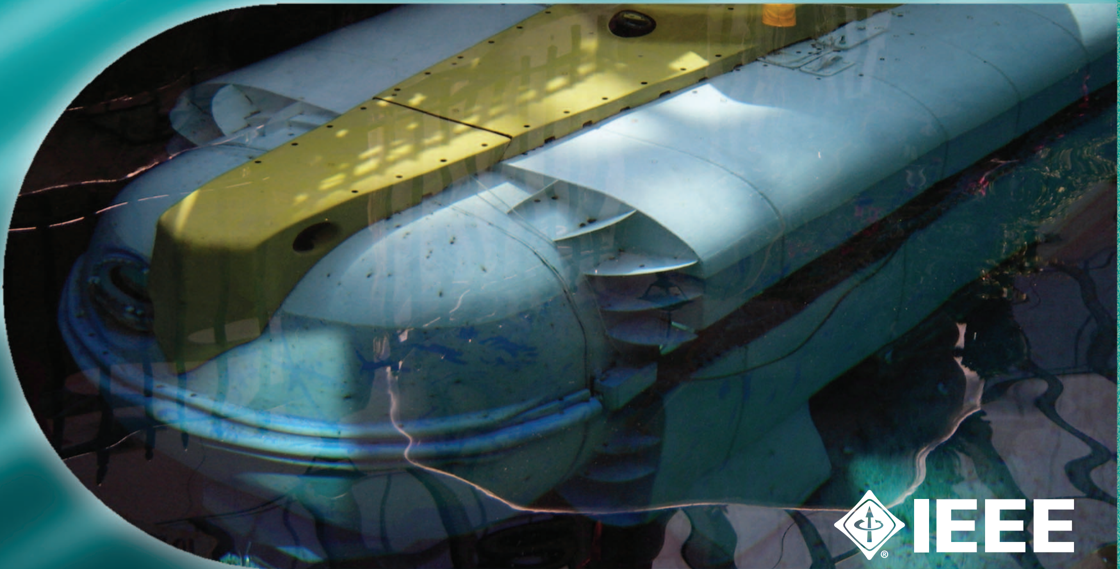
IEEE Robotics & Automation MAGAZINE

Vol. 15, No. 1 March 2008
ISSN 1070-9932

Mobile Multirobot Systems

State-of-the-Art Research and Development



IEEE
Robotics & Automation
MAGAZINE

Vol. 15, No. 1 March 2008
ISSN 1070-9932

FEATURES

DEPARTMENTS

Digital Object Identifier 10.1109/MRA.2008.917791

The articles in this issue present state-of-the-art work in mobile multirobot systems. The work presented ranges in scope from the coordination of motion and the assignment of task to the design considerations that are critical to development and evaluation. Also included are features on Haptics, the WTEC International Study, and consumer robotic products.

Background image:
©Dreamstime, iStockphoto

IEEE Robotics & Automation Magazine (ISSN 1070-9932) (IRAMEB) is published quarterly by the Institute of Electrical and Electronics Engineers, Inc. Headquarters: 3 Park Avenue, 17th Floor, New York, NY 10016-5997 USA, Telephone: +1 212 419 7900. Responsibility for the content rests upon the authors and not upon the IEEE, the Society or its members. IEEE Service Center (for orders, subscriptions, address changes): 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331 USA. Telephone: +1 732 981 0060. Individual copies: IEEE members $20.00 (first copy only), nonmembers $86.00 per copy. Subscription rates: Annual subscription rates included in IEEE Robotics and Automation Society member dues. Subscription rates available on request. Copyright and reprint permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. Copyright law for the private use of patrons (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA; (2) pre-1978 articles without a fee. For other copying, reprint, or republication permission, write Copyrights and Permissions Department, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Periodicals postage paid at New York and additional mailing offices. Postmaster: Send address changes to IEEE Robotics and Automation Magazine, IEEE, 445 Hoes Lane, Piscataway, NJ 08854 USA. Canadian GST #125634188

PRINTED IN THE U.S.A.

Let's Have an Exciting New Year!

I hope you have all passed a great change of the year! I am really curious to see what 2008 will bring for us.

On the robotics front, publication of our brand new *IEEE Transactions on Haptics* (cosponsored by the Computer and Consumer Electronics Societies) will begin soon. It will be headed by our well-esteemed member Ed Colgate.

On the magazine front, there are a couple of new features that I would like to introduce. First, we will introduce a "Position" column. In this column, scientists will be invited to make a personal statement or express a personal view on issues related to robotics and automation. Then, it will be possible for all of you to discuss this statement using the wiki at http://wiki.ieee-ras.org. After some time, Paolo Fiorini will collect the major points, and we will publish them in the magazine. I hope all of you find this new discussion arena useful and interesting.

In addition, we will discontinue the "EURON" column. It will be transformed into the "Regional" column, which will provide leaders of the robotics and automation community from all parts of the world an opportunity to contribute and give a glimpse of regional activities, in particular, the interaction between government agencies, industry, and academic researchers in developing and disseminating the robotics and automation technology.

Last but not least, we have already mentioned in the past that there was a possibility of having multimedia files such as video and software attached to articles on *Xplore*. This is now active and you can find instructions on how to do that in our *IEEE Robotics & Automation Magazine* (RAM) Web page. Thus, rather than simply providing a URL for people to access the video of the experiments described in your article, the actual video file will be posted on *Xplore* and will be downloadable long after the Web site has been shut down. This will greatly enhance the archival value of *RAM*.

Speaking of URLs, articles that include Web sites in the reference list must include, in addition to the URL, the date and the author's or company's name and contact information, so that interested readers may have some recourse if the URL becomes invalid.

This special issue is guest edited by Christopher A. Kitts and Magnus Egerstedt, and it is dedicated to multirobot systems. Furthermore, you will find the second and last part of the Haptics tutorial contributed by Vincent Hayward and Karon E. MacLean.

I hope you enjoy the issue!

Stefano Stramigioli
S.Stramigioli@ieee.org

Digital Object Identifier 10.1109/M-RA.2008.915413

IEEE Robotics & Automation Magazine

A Publication of the
IEEE ROBOTICS AND AUTOMATION SOCIETY
Volume 15, No. 1 March 2008 ISSN 1070-9932

Keep the Gradient

I shall begin by thanking all the members of the IEEE Robotics and Automation Society (RAS) for giving me the opportunity to serve as President for 2008 and 2009. It is a great honor for me, and I will do my best to meet your expectations and to uphold the high standards of excellence set by my 15 predecessors, including the five past presidents of the former Council. I am looking forward to working closely with the distinguished colleagues who run our Society and who have become good friends over the years since I first attended our flagship conference, the IEEE International Conference on Robotics and Automation, in 1986. During these 22 years, I have had the privilege to observe the growth of our Society from inside and offer my services in several capacities.

We are fortunate that the excellent past leadership in our Society has achieved a number of important successes, such as

- top-ranked publications in robotics
- new leading publication in automation science and engineering
- highly respected joint publications with other societies
- premier conferences and topical workshops in robotics and automation
- new exciting educational programs in robotics
- increased interest toward industrial activities
- strong financial reserves.

This has happened thanks to a strong team of dedicated volunteers who are committed to promote science and technology and, above all, work hard to advance our Society.

On the other hand, robotics and automation is becoming more and more pervasive in the big arena of systems science. Today, new communities of users and developers are forming with little connection to the core of robotics and automation research. A strategic goal for our Society is one of outreach and openness to these communities. It is indeed at the intersection of different disciplines that the most striking advances happen. Further progress and the expected growth of our field will depend largely on our abilities to favor such interaction and encourage cooperation.

As we undertake this endeavor, we must carry on our mission to promote and disseminate robotics and automation science through the highest possible quality in conferences and publications. This mission benefits greatly from the enormous potential of both technical and industrial activities. Our success will ultimately depend on our capacities to properly invest our finances, as well as to encourage active participation of our membership, especially the students who represent our bright future. Sharing a common vision, seeking continuous feedback, ensuring transparency, promoting new challenges, and awarding excellence are among the ingredients in this course.

The following picture, taken at the IEEE/RSJ International Conference on Intelligent Robots and Systems in San Diego last fall, features the ten officers forming our Executive Committee plus our parliamentarian. Four officers (Ken Goldberg, Ian Walker, Xiaoping Yun, and

Digital Object Identifier 10.1109/M-RA.2008.915555

From left: Xiaoping Yun (treasurer), Alex Zelinsky (vice-president for Industrial Activities), Robin Murphy (parliamentarian), Alícia Casals (vice-president for Member Activities), John Hollerbach (vice-president for Conference Activities), Kazuhiro Kosuge (president-elect), Ken Goldberg (vice-president for Technical Activities), Bruno Siciliano (president), Peter Luh (vice-president for Publications Activities), Frank Park (secretary), and Ian Walker (vice-president for Financial Activities).

Robin Murphy) are continuing to serve after the previous term under Dick Volz, and I should take this opportunity to thank the outgoing officers (Antonio Bicchi, Bill Hamel, Sukhan Lee, T.-J. Tarn, and Shigeki Sugano) for having offered their experience in the transition to the new officers (Alícia Casals, John Hollerbach, Peter Luh, Alex Zelinsky, and Frank Park). Our newly elected President-Elect Kazuhiro Kosuge will chair the Long-Range Planning Committee and cooperate with us.

Here are the challenges I have collected from our six vice-presidents, which our team is committed to pursue for the next two years.

Conference Activities Board (CAB)

Vice-President John Hollerbach

The number of conferences that RAS sponsors and cosponsors is large and growing, while the paperwork for running any conference is increasing. Our goals are

- to manage the portfolio of RAS conferences as effectively as possible (conference approval, conference openings, conference closings, and technically sponsored conferences)
- to monitor the quality of these conferences through the Steering Committee for Technical Programs to make sure they meet RAS standards
- to consolidate the Conference Editorial Board (the committed service of its Editor-in-Chief Seth Hutchinson is here acknowledged)
- to identify a unified and efficient conference software system of choice for RAS
- to utilize the RAS Web site to post and track information about conferences, to make their management easier
- to introduce a quick-start Web page to assist conference organizers in the required procedures and paperwork
- to expand the robotics and automation community that RAS serves by strategic linkages to important non-RAS conferences.

Financial Activities Board (FAB)

Vice-President Ian Walker

From the financial point of view, the key goal is to develop more accurate and timely estimates of the Society's financial state, so that we can more effectively utilize our resources. The main challenge here is that for many of the major contributors (positively and negatively) to the Society budget (e.g., final conference surpluses) we only have estimates at the time we need to make funding decisions. This uncertainty is amplified at the present, as this is a time of significant change within the IEEE, in terms of the way the IEEE calculates charges and payments to Societies. We are working in two main directions:

- to better monitor and predict the financial aspects of our Society's operations
- to predict the changes to our Society's finances resulting from the IEEE's proposed internal financial changes.

Industrial Activities Board (IAB)

Vice-President Alex Zelinsky

The outreach of our Society to the industry is a key challenge for robotics and automation growth and the impact on the new emerging markets. Our specific goals are:

- to further develop and extend the network of representatives in various countries around the world to track industrial activities, particularly new robotics product and service innovation
- to promote innovation by growing the IEEE–International Federation of Robotics Joint Forum on Innovation and Entrepreneurship in Robotics and Automation into a substantive RAS activity
- to continue to position RAS as a key player for the development of standards in robotics and automation by working with the industry and the research community; the activities of RAS will be coordinated with that of other standardization defining organizations, such as the International Standards Organization, the International Electrotechnical Commission, and the IEEE Standards Board
- to continue the technology road-mapping process for robotics and automation with the purpose of representing the industry's and community's views about the future vision for the industry and new

products or services by providing a broader view of the future possibilities and implications of robotics technologies to help guide business and government decision makers

- to encourage new IEEE Members from the industrial community to join RAS and participate in our Society activities
- to ensure that all Industrial Activities Board information is readily available and up to date for RAS Members on the Society Web site.

Member Activities Board (MAB)

Vice-President Alícia Casals

The strength of our Society relies on the quantity and involvement of its members as well as on the enthusiasm and steering power of its leading team. Our mission is to ensure that RAS can approach its current and future members, helping them to be aware of the benefits that they can obtain from our Society and the activities they can join or promote. We plan to achieve the following goals through our Standing Committees:

- to promote RAS through its chapters (so far mainly the chapter chairs have been involved) and encourage them to reach its members and promote RAS activities at the national level
- to increase educational activities involving students, such as special sessions in conferences and publications, by facilitating participation in RAS activities through special fees, as well as promoting social activities and creating the seeds for further cooperation
- to identify potential problems that impede participation of members from emerging countries in RAS activities and look for solutions related to economical or legal aspects, while taking care to provide equal opportunities for everybody
- to serve members by giving information of their possible involvement, helping on funding, mentoring young members, and looking for relationships with other societies or specialties.

Publications Activities Board (PAB)

Vice-President Peter Luh

The intellectual properties and income of RAS rest heavily on publications. The publication activities, however, are facing drastic changes in recent years because of online publications with multimedia capabilities and the concomitant push for open access and decrease in print subscriptions. In the mean time, many emerging areas stipulate new publications. We are, thus, in an exciting and challenging era for publications. Currently, RAS publishes *IEEE Transactions on Robotics (TRO)*, *IEEE Transactions on Automation Science and Engineering (TASE)*, and *IEEE Robotics and Automation Magazine (RAM)*, and the committed service of their Editors-in-Chief Alessandro De Luca (*TRO*), Nukala Viswanadham (*TASE*), and Stefano Stramigioli (*RAM*) is here acknowledged. We are also cosponsoring the following:

- *IEEE/ASME Transactions on Mechatronics*
- *IEEE/ASME Journal on Microelectromechanical Systems*
- *IEEE Transactions on Nanobioscience*
- *IEEE Transactions on Nanotechnology*
- *IEEE Sensors Journal*
- *IEEE Systems Journal*.

All these journals are among the top journals in their respective fields of interest. We are also cosponsoring the new *IEEE Transactions on Haptics*, which is expected in the second half of 2008. Our goals for the publications are:

- to work with individual editorial boards to continue improving the quality of publications as measured by the impact factors and the number of *IEEE Xplore* downloads while decreasing the time periods from submission to first decision and to final publication through a continuing improvement process and further exploration of advanced multimedia capabilities
- to work with the Financial Activities Board to analyze the financial conditions of publications and find opportunities to improve the conditions while keeping abreast of the open access issue
- to work with other boards, the general RAS membership, and other societies to encourage visionary papers and special issues on emerging areas, survey or review papers on maturing topics, IEEE Press books, possibly new journals after careful market analysis, and general submissions (including manuscripts overhauled from conference papers) for high-quality services to members and the profession.

Technical Activities Board (TAB)

Vice-President Ken Goldberg

Technical activities are truly the amplifiers of ongoing top research and future directions in our Society and beyond. Some specific goals we wish to achieve are as follows:

- to develop stronger links between the technical committees (TCs) and RAS conferences and publications by encouraging TCs to post lists of related papers and summaries of milestones and having TCs play an active role in coordinating sessions and special events at conferences
- to implement a triennial review process for our 23 TCs to facilitate creation of new TCs and to initiate a regular "TC Spotlight" feature in *RAM*
- to implement technical communities, a new category for large, highly active TCs that have at least 150 members
- to help in the Conference on Automation Science and Engineering, enabling it to grow to a major RAS conference and to encourage new research in automation, particularly in the areas of lab automation, life sciences, and security

- to increase the size of the Distinguished Lecturer (DL) program from 24 to 30
- to develop online databases to maintain information on TCs, DLs, and mailing lists for all TCs
- to involve more young researchers in the TC and DL programs, so that all TCs should have at least one cochair under age 35.

Needless to say, the list of challenges discussed is rather ambitious, and the financial implications of any new initiative shall be weighed with respect to the service provided to our members. Inspired by a truly collegiate spirit, I will cooperate with the Executive Committee, the Administrative Committee, as well as the various boards and committees to tune the ideas, enrich the list, establish new goals, and set up the most appropriate and cost-effective procedures to achieve them.

I trust that a good atmosphere is a success key in the management of a team of volunteers, who should be vigorously and enthusiastically motivated to invest their valuable time in the running of RAS. I consider interpersonal contacts to be of extreme importance, and I will promote that sense of camaraderie, which ultimately makes our Society a very pleasant environment to be active in.

The final word is for our outgoing president and good friend Dick Volz. Dick is an incredible source of knowledge within the RAS and the IEEE. His loyalty and generosity have probably no equal. I have carefully watched him during these two years I have been serving as president-elect, and I hope I have learned how to be a good president. I trust he will continue to be a precious advisor to our team of Society leaders.

I am humbled and delighted to be offered this dream-come-true prestigious opportunity. As I used to say to my pals . . . keep the gradient ;-)

Bruno Siciliano
siciliano@ieee.org
RAS President 2008–2009

Design, Control, and Applications of Real-World Multirobot Systems

Christopher A. Kitts and Magnus Egerstedt

The field of multiagent robotics has recently reached a level of maturity in that systems are beginning to transition from proof-of-concept laboratory environments to deployed real-world systems. When we started planning for this special issue of *IEEE Robotics & Automation Magazine* in the early spring of 2007, we hoped to capture this exciting development trend. With a lineup of six strong articles, spanning the area from decentralized control, diagnosis, and decision making to experimental platforms, we feel that we have come very close to our original ambitions. We hope this sentiment is shared by the readers also.

The recent flurry of activities in the general area of multiagent robotics is to a large degree performance driven since multirobot systems offer a number of advantages and additional capabilities over their single-robot counterparts, including redundancy, increased spatial coverage and throughput, flexible reconfigurability, spatially diverse functionality, and the fusing of physically distributed sensors and actuators. Applications in which these capabilities constitute enabling technologies range from remote and in situ sensing to the physical manipulation of objects, and the domains for such applications include land, sea, air, and space.

Despite remarkable research developments in the area, numerous technical challenges remain that must be overcome to field cost-effective multirobot systems. These challenges include interrobot communications, relative position sensing and actuation, control paradigms appropriate to real-time multirobot systems, the fusion of distributed sensors or actuators, man-machine interfaces allowing efficient human direction or supervision of these systems, effective reconfiguration of the system's functionality, and design approaches supporting the economical production of such systems.

The six articles in this special issue present state-of-the-art work in mobile multirobot systems with an emphasis on techniques that have matured to the point of being evaluated through experimentation. The work presented ranges in scope from the coordination of motion, to the assignment of tasks, to the design considerations that are critical to development and evaluation.

Antonelli et al. describe the application of null-space-based behavioral control, a new type of behavioral control, to the control of a group of mobile rovers that are capable of entrapping or escorting a moving target. Results from both simulations and experiments using a test bed of small tabletop rovers demonstrate the effectiveness of this control approach for constituting and maintaining an escort formation. The method is also shown to be robust to failures of individual robots, with the remaining robots dynamically restructuring themselves to achieve proper coverage of the target.

Kress-Gazit et al. present the application of hybrid feedback control to a group of automobiles that must safely navigate and park in a dynamic urban environment. These controllers combine local feedback control policies with discrete automata to satisfy high-level behavioral specifications without explicitly planning the motion of each vehicle. Through simulation, these controllers are demonstrated and show the complexity of provably correct behavior that is automatically achieved.

Bethke et al. describe the use of a task-level controller for multiple unmanned aerial vehicles that integrates vehicle health and status information into a real-time mission planning system. Using simulation and an indoor multihelicopter test bed, their experiments demonstrate significant performance improvement for response time and other metrics when such a feedback loop provides information such as payload status and fuel state to the task assignment algorithm.

Sariel et al. evaluate a distributed auction-based system for allocating tasks among a cooperative fleet of autonomous underwater vehicles for a naval mine countermeasure operation. This system, which has previously been experimentally demonstrated on a rover test bed, is evaluated in a high fidelity navy simulator capable of evaluating performance as a function of failures, limited/delayed/unreliable communications, and other real-world conditions. Through the novel integration of task scheduling and execution, their approach maintains high solution quality given the dramatic resource limitations inherent in underwater missions.

Michael et al. describe critical considerations in the design of experimental test beds for the verification and validation of large-scale multirobot control systems. In the development of their own test bed, they have focused on providing an inexpensive, flexible, scalable, and easy-to-use system to support the modeling, design, benchmarking, and validation of a wide variety of multirobot applications and control algorithms.

Bicchi et al. report on their work in developing a multiagent functional architecture for decentralized traffic management. Their platform provides a general suite of mobility and communication services that accommodates a wide variety of heterogeneous systems and that meets the critical requirements of safety, scalability, security, and reconfigurability. Their initial results in applying this architecture to a simple two or three vehicle system verify capabilities to perform accurate localization, execute collision-free motion, and manage secure communications.

As a final remark, it should be noted that this special issue (or any special issue for that matter) only represents a particular snapshot of the field, and there are undoubtedly areas and results that are not included in this issue. Although we made every effort to include most aspects of the maturing multirobot field, we cannot claim that the coverage is complete.

Digital Object Identifier 10.1109/M-RA.2007.914989

Robotics Software: The Future Should Be Open

By Herman Bruyninckx

This column introduces a number of problem claims about the pitiful state of practice in software for robotics (and for all kinds of engineering domains in general). It also presents solution claims, whose realization can lead to a long-term, macroeconomically optimal solution, both for the industry and academia.

Key problems in robotics software in the industrial and the academic practice are a chronic lack of standardization, interoperability and reuse of software libraries, both proprietary and open source. For example, we still have not standardized the Kalman filter or particle filter that everyone can and wants to use, and the same holds for many other mature robotics software components such as kinematics and dynamics, control laws, or planning algorithms. As a result, thousands of (Ph.D.) person months are lost worldwide every year in reimplementing these things for the zillionth time, without any new contribution to software reuse.

This pitiful state of the practice is not unique to robotics, and only a few engineering domains do it right: numerical linear algebra (starting many decades ago already); the World Wide Web [with (X)HTML, cascading style sheets (CSS), scalable vector graphics (SVG), and other W3C standards as the fundamental enablers]; the Java middleware ecosystem [XML processing, open services gateway initiative (OSGi, now obsolete), Eclipse, mobile phone frameworks, etc.]; and tools around the Object Management Group (OMG) standards of UML, SysML, and model-driven architecture. These examples are not tied to specific applications (this is not a coincidence but a very wise design decision about modularity and decoupling!), and they all have healthy commercial and open-source offerings, with real and rapid innovation taking place in both software development models.

Every section in this article focuses on one of the fundamental issues that has led to the retarded state of software in robotics and suggests a concrete solution. Most neighboring scientific and technologic domains (computer vision, systems and control, cognitive science, artificial intelligence, etc.) suffer from exactly the same problems, such that cooperation with those domains can lead to faster implementation of the presented solutions.

Academic Seriousness about Software

Problem Claim

The academic robotics community fails to produce a healthy software ecosystem in robotics because academics cannot get citation index or other credits for software. Hence, cooperation on the development of excellent software or appropriate standardization is not on the radar of the majority of (senior) researchers in robotics. The lack of code reuse is mainly caused by the lack of good standards and good implementations of such standards, both again caused by the lack of academic stimuli to spend time on solving these real problems. In addition, most professors in robotics have no hands-on experience with software engineering, let alone software coding, and hence they have no appropriate appreciation for the scientific and practical challenges behind the creation, distribution, and maintenance of good robotics standards and software.

Digital Object Identifier 10.1109/M-RA.2008.915411

If you would like to discuss this issue with your peers, please visit http://wiki.ieee-ras.org.

Solution: Robotics Software Journal

Our community needs to create a peer-reviewed journal on software (preferably in cooperation with the neighboring software-intensive communities). Topics of this journal are suggestions for application programming interfaces (APIs), open-source reference implementations of such APIs, best programming practices, discussions about software patterns (i.e., specific design and implementation approaches that have proven to work in various real-world systems), etc.

The most important editorial policy of the journal should be that contributions are evaluated on how they improve existing software APIs or implementations and much less on what new, innovative software is presented. Indeed, our current publication culture is driven by (often rather vacuous) claims about novelty and innovation, i.e., one tries to differentiate oneself as much as possible from existing work. However, when software is concerned, such a policy has proven to lead to fragmentation and lack of interoperability (cf. the example of the commercial Unix vendors).

Robotics Ontologies

Problem Claim

Robotics is a science of integration rather than a fundamental science, and the integration becomes ever more complex. Software support for this integration is hindered by a tremendous lack of precise semantics, in computer-readable form, of all objects used in robotics problems.

Solution: Semantic Web for Robotics

This solution is obvious since only computer-understandable representations of the meaning of all objects involved in the software can lead to the automatic support for the integration of libraries and components from different sources. One should be aware that several complementary ontologies are needed: one for objects (i.e., what are the properties and the behavior of the objects in robotics systems?), one for systems (i.e., how do the properties of systems of components influence the choice of objects with which those systems can be built?), one for profiles (i.e., what is the expertise of a given researcher? Or, what are the topics a given journal is interested in?), and possibly still some other ontologies too.

A Wikipedia approach is most appropriate for the (incremental, peer reviewed, community driven) creation of such ontologies. Technically, nothing more is needed than the standardization of computer-readable semantic tags. Again, this effort should be shared with the neighboring scientific domains.

Funding Agency Policies

Problem Claim

Funding agencies worldwide often pride themselves to fund only fundamental scientific developments, but they forget to provide money to solve the aforementioned software and ontology problems. This leads to increasingly diminishing returns in fundamental results: the lack of code reuse brakes the rate of fundamental progress by preventing researchers from standing on the (software) shoulders of giants.

Solution: Fund Middleware Projects

Only a slight change in policy is needed: the funding agencies should also provide funding for (some, not many!) standards and ontology creating projects (and for open-source software that supports them), and they should give significant incentives to other research projects to apply the results.

Lack of Interest in MDE

Problem Claim

The implicit assumption behind model-driven engineering (MDE) is that system designers will just have to provide specifications of what they want their systems to achieve, and (formally verified) code will be generated automagically. This assumption is too simplistic: the influence of all forces, as defined in the context of software patterns, working on the software design is too high for formally verifiable and one-size fits-all solutions to be generated automatically. Nevertheless, the robotics community is seriously neglecting the progress in MDE, hence throwing away lots of opportunities to let software engineering and practice mature in the domain of robotics.

Solution: Create SysML Profiles for Robotics

The robotics literature contains hundreds of articles about architectures, all of which use graphical models with boxes and arrows, but the meaning (semantics!) of these symbols has never been standardized, and the practical constraints on real-world implementations are almost never made explicit. Nevertheless, the recent standardization of SysML (and of its real-time and embedded specialization MARTE) provides tremendous opportunities to start with reusable and semantically well-defined designs of complex software systems.

No other competing framework of software design standards exists, and the creation of SysML supporting tools (both in commercial and open-source form) is booming. So, the robotics community should not hesitate and should begin to use these system modeling standards, again stimulated by editorial policies of journals and conferences, and by project requirements from funding agencies.

Asynchronous Programming

Problem Claim

Robot systems are increasingly multiagent, and it is very difficult to carry out the asynchronous data exchange and activity synchronization between agents.

The problem is to find an appropriate trade-off between efficiency and robustness: asynchronous communication and synchronization will go wrong and can never be avoided by design. Robotics engineers are most often not even aware of the problems caused by asynchronous software activities.

Solution: Integrate IPC in Engineering Curriculum

Interprocess communication (IPC) programming is still not integrated in academic or industrial engineering curricula. The robotics community should identify the (all in all rather limited number of) IPC use cases that are relevant in robotics multiagent systems, and create a Wikibook about the subject, with worked-out solutions. Again, only the wiki approach to content creation is appropriate in this case since a significant amount of fine-tuning and peer review will be needed before the content matures.

Monolithic Commercial Software

Problem Claim

Commercial software products in robotics are invariably large monolithic systems in which the users cannot replace components by alternatives from other vendors or by their own software. This reduces the flexibility and appropriateness of the commercial software to unworkably low levels.

Robotics software is typically provided only by manufacturers of robotics hardware, with some framework exceptions such as Real-Time Innovations and Microsoft Robotics Studio. None of these software platforms supports the flexibility required to build optimal or innovative robot control software systems for all applications.

Solution: Apply FOS Business Models

There is only one technological approach toward making robust complex systems: to integrate small and stable subsystems via semantically well-defined and standardized interfaces.

This approach is followed in all technological domains except in software; for example, the most complex buildings or telecommunication systems can be designed and constructed with standardized components that can be purchased from several vendors competing in a free market. Such a free market with competitors offering fully replacable building blocks does not exist in software in general and certainly not in robotics.

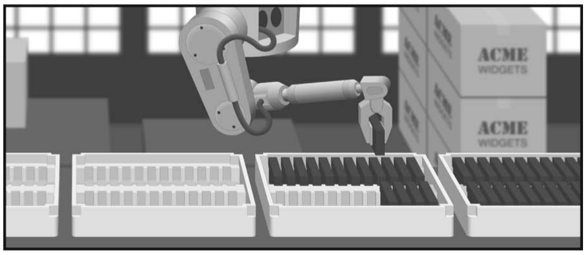
As is the case with general software, practice shows that only free software and open-source software (FOS) projects really work toward a free IT market. Economic theories have since long described the negative effects of increasing returns in noninteroperable technological networks and the necessity to have natural, vendor-neutral monopolies for all large-scale basic infrastructure. Nevertheless, software customers are still not demanding their commercial providers to obey these economic realities, with expensive, monolithic, and underachieving software as a result.

Both customers and providers should begin to understand how to make money on top of FOS infrastructure and how this can lead to optimal performance, robustness, and cost. Roughly speaking, the business model for FOS is to capitalize on software services (including the service to write new software!) and not on vendor lock-in, as is the case now. One inspiring example is what is happening in the mobile telephony market, in which more and more vendors use an increasing amount of Linux and Java FOS frameworks.

Concluding Claim: Open Content, Standards, and Software

There is only one solution to maximize the long-term, macroeconomic benefits for the robotics industry and for academic robotics research: the closely integrated development of open content, open standards, and open source. However, the strategy should be to adopt them in this particular order and not the other way around: also for all existing open-source projects in robotics, lock-in is a significant problem because the interoperability of these projects is close to zero because of the current lack of ontologies (open content) and APIs (open standards).

Any vendor or project lock-in we are suffering from is the result of our own voluntary choices. We, robotics scientists, are the essential components in all editorial boards of journals and conferences, and we are in the think tanks and the review boards of funding agencies. Hence, we have the future of software development in robotics in our own hands. Let's take that responsibility seriously and open up that future in all possible ways.



Kosuge Elected RAS President-Elect

At their 3 November meeting in San Diego, California, the IEEE Robotics and Automation Society (RAS) Administrative Committee (AdCom) elected Kazuhiro Kosuge of Tohoku University, Sendai, Japan, as president-elect. He will serve as president-elect from 2008–2009, and, in 2010, he will succeed Siciliano as RAS president.

Kosuge has served the Society in many capacities, including several terms on the RAS AdCom, vice-president for membership, conference board meetings chair, program cochair for the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 1998, the IEEE International Conference on Robotics and Automation (ICRA) 2007, and general chair for IROS 2004. He will be general chair for ICRA 2009.

Kosuge joins a long line of dedicated and distinguished leaders, some of whom are shown in the photo taken at IROS 2007 in San Diego, California.

RAS presidents. From left: T.-J. Tarn (1992–1993), Art Sanderson (1989–1990), George Bekey (1996–1997), Toshio Fukuda (1998–1999), Dick Volz (2006–2007), Bruno Siciliano (2008–2009), and Kazuhiro Kosuge (2010–2011).

Richard Volz Elected IEEE Division X Director-Elect

RAS Past President Richard Volz has been elected director-elect for Division X by the members of the societies in the division. He will serve as director-elect in 2009, and, in 2010, he will succeed William A. Gruver as Division X director.

The director of Division X, along with the directors of the other nine IEEE Technical Divisions, the directors of the ten IEEE geographic regions, the chairs of the major IEEE boards, and the IEEE officers, constitute the Board of Directors, the governing body of the IEEE. The directors also work to facilitate communication and cooperation among the societies and regions in their Divisions.

In addition to the RAS, Division X includes the IEEE Computational Intelligence Society; the IEEE Control Systems Society; the IEEE Engineering in Medicine and Biology Society; the IEEE Lasers and Electro-Optics Society; the IEEE Systems, Man, and Cybernetics Society; the IEEE Sensors Council; and the IEEE Systems Council.

Volz has been an active Member of the IEEE for more than 40 years and has been active in RAS since the establishment of *IEEE Journal on Robotics and Automation*, the predecessor to the current *IEEE Transactions on Robotics Automation* and *IEEE Transactions on Automation Science and Engineering* (*T-ASE*). During his term as RAS President, he also served as a Member of the IEEE Technical Activities Board, Finance and Conference Committees, and as a Member of the IEEE Publication Services and Products Board Strategic Planning Committee and the IEEE Careers Committee.

Digital Object Identifier 10.1109/M-RA.2007.914984

RAS Members Elected to the IEEE 2008 Fellows Class

We are proud to announce that 14 RAS Members were recently elected IEEE Fellows. Only 0.1% of active IEEE Members are elected Fellow Grade each year, and each nominee's Fellow nomination form, CV, and letters of recommendation are carefully evaluated.

The following new Fellows' nominations were reviewed by the RAS.

- Max Meng, Chinese University of Hong Kong, *for contributions to medical robotics*
- Zexiang Li, Hong Kong University of Science and Technology, *for contributions to robotic manipulation, nonholonomic motion planning, and workpiece utilization*
- Tianyou Chai, Northeastern University, China, *for contributions to adapt intelligent decoupling control and integrated automation of complex industrial processes*
- Fan-Tien Cheng, National Cheng Kung University, China, *for contributions to semiconductor manufacturing automation and force optimization in multiple-chain robotic mechanisms*
- Narahari Yadati, Indian Institute of Science, *for contributions to the design of manufacturing systems, supply chain networks, and electronic markets*
- Alexander Zelinsky, Commonwealth Scientific Industrial Research Organization, *for contributions to vision-based robotics*
- Peter Corke, Commonwealth Scientific Industrial Research Organization, *for contributions to visual-based robot control and its applications to field robotics*
- Roy Featherstone, Australian National University, *for contributions to multibody dynamics*
- Steven Holland, General Motors Research and Development, *for leadership in the industrial application of robotic technology*

2008 IEEE Fellows


Max Meng


Zexiang Li


Tianyou Chai


Fan-Tien Cheng


Narahari Yadati


Alexander Zelinsky


Peter Corke


Roy Featherstone


Steven Holland


Vincent Hayward


Roland Siegwart


Dragan Nesic


Karen Panetta


Ju-Jang Lee

- Vincent Hayward, McGill University, Quebec, Canada, *for contributions to robot manipulator programming and the development of haptics interface technology*
- Roland Siegwart, Swiss Federal Institute of Technology (ETH), Zurich, *for contributions to mobile, networked, and microscale robots.*

The following RAS IEEE Fellows were evaluated by other societies.

- Dragan Nesic (CS), University of Melbourne, Australia, *for contributions to the analysis and control of networked nonlinear sampled-data systems*
- Karen Panetta (EDUC), Tufts University, Massachusetts, *for leadership in engineering education and curriculum development to attract, retain, and advance women in engineering*
- Ju-Jang Lee (IE), Korea Advanced Institute of Science and Technology, *for contributions to intelligent robust control and robotics.*

RAS Awards

Spansion, Inc., Sponsors IEEE CASE Best Paper Awards

Spansion, Inc., the Silicon Valley company, which is the world's largest NOR Flash memory provider has offered to sponsor the annual IEEE Conference on Automation Science and Engineering (CASE) Best Paper for at least the next five years.

The first awards were presented at the 2007 IEEE CASE Awards Banquet in Scottsdale, Arizona, in September 2007. The cowinners of the Best Paper Award were:

- Wenhui Wang, Xinyu Liu, and Yu Sun, for "Autonomous Zebrafish Embryo Injection Using a Microrobotic System"
- Timothy Molter, Sarah McQuaide, Meng Zhang, Mark Holl, Lloyd Burgess, Mary Lidstrom, and Deirdre Meldrum for "Algorithm Advancements for the Measurement of Single Cell Oxygen Consumption Rates."

Cowinners of the Best Student Paper Award were:

- T.H. Tran, Ngai Ming Kwok, Steven Scheding, and Q.P. Ha for "Dynamic Modelling of Wheel-Terrain Interaction of a UGV"
- Ehsan Saeedi, Samuel Kim, James Etzkorn, Deirdre Meldrum, and Babak Parviz for "Automation and Yield of Microscale Self-Assembly Processes."

2006 T-ASE Googol Best New Application Paper Award

Two groups of authors were cowinners of the 2006 *T-ASE* Googol Best New Application Paper Award offered for the Best New Application Paper published in *T-ASE*. This award was also presented at CASE 2007.

- Jae Wan Kwon, Sanat Kamal-Bahl, and Eun Sok Kim, for "In Situ DNA Synthesis on Glass Substrate for Microarray Fabrication Using Self-Focusing Acoustic Transducer," in the April 2006 issue, pp. 152–158.
- Lixin Dong, Nelson Bradley, Fukuda Toshio, and Arai Fumihito, for "Towards Nanotube Linear Servomotors," in the July 2006 issue, pp. 228–235.

KUKA Roboter to Sponsor Best Service Robotics ICRA Paper Award

The KUKA Roboter, a leading robotics company based in Germany, has agreed to sponsor a new paper award in the area of service robotics at ICRA, beginning in 2008.

RAS Members Elect Six New AdCom Members

A record of more than 1,000 RAS Members voted to elect six new Members of the Society's AdCom. This year, for the first time, Members were able to vote via the Internet, and most Members chose the electronic medium, although they still had the option to mail or fax paper ballots. The new AdCom Members, who will serve a three-year term, are as follows:

- Alessandro De Luca, *Università di Roma "La Sapienza" (Italy)*
- Peter Corke, *CSIRO (Australia)*
- Lynne Parker, *University of Tennessee-Knoxville (USA)*
- Stefano Stramigioli, *University of Twente (The Netherlands)*
- Shigeki Sugano, *Waseda University (Japan)*
- Satoshi Tadokoro, *Tohoku University (Japan)*

2007 *IEEE Robotics and Automation Magazine* Reviewers

Thanks to the following individuals who served as reviewers for the *IEEE Robotics & Automation Magazine* in 2007. Without their dedicated work, we could not have published this magazine.

Beshahwired Ayalew
Bryan Adams
Luigi Biagiotti
John Billingsley
Bradley Bishop
Clifford Bonaventura
Johann Borenstein
Mike Bosse
Thomas Braunl
Alberto Broggi
Michael Bruenig
Carlos Canudas-de Wit
Stefano Caselli
Damien Chablat
Enya Cheng
Gordon Cheng
Greg Chirikjian
Oscar Jr. Chuy
Anders Lyhne Christensen
Christopher Clark
Peter Corke
Mark Cutkosky
Dimos Dimarogonas
Robert Dougherty
Matthew Dunbabin
Michael Drew
Tolga Eren
Riccardo Falconi
John Feddema
Trevor Fitzgibbons
Masahiro Fujita
Tove Gustavi
Jason Gu
Musad Haque
Kensuke Harada
Geir Hovland
Xiaoming Hu
Mas Ignacio
Jared Jackson
Kevin Jones
Shuuji Kajita
Peter Kazanzides
Chetan Kapoor
Charlie Kemp
Jongrae Kim
Derek Kingston
Eric Klavins
Haldun Komsuoglu
Kiju Lee
Mark Lee
Robert Lee
Glen Lightsey
Stephen Lindemann
Frank Lingelbach
Hod Lipson
Dikai Liu
James McLurkin
John-Michael McNew
Manuel Mazo
Tejas Mehta
Hideaki Minakata
Eric Monacelli
Matt Moses
Nima Moshtagh
Satoshi Murata
Mansard Nicolas
Koichi Nishiwaki
Jeff Ota
Christian Ott
Lucia Pallottino
Apostolos Pantazis
Michael Piovoso
Matt Powers
Robin Qiu
Mike Rasay
Anders Robertsson
Daniela Rus
David Russell
Ketan Savla
Kerstin Severinson-Eklundh
Jimmy Sastra
Ketan Savla
Cristian Secchi
Brian Schucker
Roland Siegwart
Sanjiv Singh
Brian Smith
Kyle Stanhouse
Salah Sukkarieh
Michael Swartwout
Mahmoud Tarokh
Philip Voglewede
Eric Westervelt
Stefan Williams
Gordon Wyeth
Tao Yang
Kazuhito Yokoi

RAM Multimedia Files on *Xplore*

The IEEE has made it possible to post videos and other multimedia files to accompany journal and magazine articles on *Xplore*. Authors of accepted articles are encouraged to post accompanying videos. Instructions for the same are posted on the Web site at http://www.ieee-ras.org.

The following articles from December 2007 now have videos posted:

- *Morphology Control in a Multirobot System* by Anders Lyhne Christensen, Rehan O'Grady, and Marco Dorigo
- *Robotic Self-Replication* by Kiju Lee and Gregory S.Chirikjian
- *Toward a Scalable Modular Robotic System* by Satoshi Murata, Kiyoharu Kakomura, and Haruhisa Kurokawa

The purpose of the award, which will include a US$1,000 honorarium, is to promote advancement between robotics science research and industry research and development in the area of service robotics applications (both professional and domestic). The sponsorship by KUKA will be initially for five years (2008–2012).

"We are grateful to KUKA Roboter for their generous support of ICRA," said Dick Volz, RAS past president, who with Rainer Bischoff, KUKA Coordinator of Cooperative Research Projects, and KUKA executives worked to establish the award.

AdCom Approves George Saridis Leadership Award

At their 3 November meeting, the IEEE RAS AdCom voted to endow the George Saridis Leadership Award. The award is named in honor of the late Prof. George Saridis, founding president of the Robotics and Automation Council, which later became the RAS.

This award will recognize the outstanding contributions of an individual for exceptional leadership, innovation, and dedication that benefit the robotics and automation community. Up to two awards will be given each year, and no award will be given if no qualified candidate is identified.

The Saridis Award and the KUKA and Spansion, Inc. Awards have been submitted to the IEEE Technical Activities Awards and Recognition Committee for formal approval.

Invention and Entrepreneurship in Robotics and Automation

The RAS AdCom voted to approve a revised memorandum of agreement with the International Federation of Robotics (IFR) for cosponsorship of the Invention and Entrepreneurship in Robotics and Automation Award.

2009 AdCom Nominations and Petitions

The RAS AdCom Nominations Committee is accepting nominations, including self-nominations, for the 2008 Administrative Committee elections. Candidates may petition to be on the ballot by submitting a petition with signatures of 2% of RAS voting members (121 Graduate Student and higher grade Members in 2008). Nominations and petitions should be e-mailed or faxed to Rosalyn Snyder (E-mail: r.g.snyder@ieee.org, Fax: +1 919 882 9734).

RAS and the IFR agree to jointly sponsor the Invention and Entrepreneurship Award. The purpose of this award is to highlight and honor the achievements of the inventors with value creating ideas and entrepreneurs who propel those ideas into world-class products. At the same time, the joint disposition of the award underlines the determination of both organizations to promote stronger collaboration between robotics science and robotics industry. Up to one award will be given annually to the individual(s) whose entrepreneurial efforts have taken an earlier conceptual innovation and evolved it into a commercialized product.

The award will include an honorarium of US$2,000, jointly provided by the IFR and RAS, which will be shared by all winners. The next award is scheduled to be presented at the 2008 Innovation and Entrepreneurship Workshop in Munich in June 2008. Contact RAS Vice-President for Industrial Activities Alex Zelinsky (e-mail: alex.zelinsky@csiro.au) for information.

For a complete description of the award and the nominations procedures, please see http://www.ieee-ras.org/member/awardsRAS or the announcement in this issue.

2007 IROS Paper Awards

Congratulations to the winners of the IROS 2007 paper awards.

- Harashima Award for Innovative Technologies: Mark Spong
- RSJ/SICE Best Paper Award: "Design, Fabrication, and Analysis of a 3-DOF, 3-cm Flapping-Wing MAV," Robert Wood
- ICROS Best Application Paper Award: "Robust Stereo Tracking for Space Applications," Fabien Dionnet and Eric Marchand
- SARCOS Best Student Paper Award: "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models," Jonathan Ko, Daniel Klein, Dieter Fox, and Dirk Haehnel
- Boston Dynamics Best Video Award: "The First Flight of an Insect-Sized Robotic Fly," Robert Wood
- Hewlett-Packard Most Innovative Paper Award: "Chemical Robot Design of Self-Walking Gel," Shingo Maeda, Yusuke Hara, Ryo Yoshida, and Shuji Hashimoto

The Technical Committee on Haptics

By Hong Z. Tan

The word *haptics* refers to sensing and manipulation through the sense of touch. (In the interest of space and readability, I have taken the liberty of using definitions that have been developed by many haptics researchers. Proper citations, including a more personal account of the first year of the Technical Committee on Haptics, can be found at www.worldhaptics.org under "archives.") The term *cutaneous* or *tactile sense* refers to the awareness of stimulation of the outer surface of the body mediated by mechanoreceptors in the skin. The term *kinesthesis* or *proprioception* denotes the awareness of joint-angle positions and muscle tensions mediated by sensory organs embedded in the muscles and joints. Modern haptics is concerned with the science, technology, and applications associated with the information acquisition and object manipulation through touch, including all aspects of manual exploration and manipulation by humans, machines, and the interactions between the two, performed in real, virtual, teleoperated, or networked environments. The technical scope of the Technical Committee on Haptics (TCH) embraces all aspects related to haptic interactions, from basic science to technological developments to applications.

Earlier, haptics research focused on sensory substitution that conveyed imagery or speech information to individuals with visual or auditory impairments via their sense of touch. Typical devices used solenoid and piezoelectric actuators and electrical stimulators. With the advent of force-feedback technology, there were renewed interests in using haptic interfaces in teleoperator systems and virtual environments. Research on robotic hands and manual grasping further underscored the need for spatially distributed force sensing and display. Although the technologies for vibrotactile stimulators and point-based force-feedback devices are relatively mature and available commercially, finger-tip haptics, the development of devices consisting of tightly packed pin arrays and those conveying surface curvature, contact friction, and slip, is now a topic of hot pursuit in many research laboratories. In the recent years, haptics has permeated our daily lives by showing up in consumer products such as personal digital assistants, game consoles, cell phones, and touch screens.

The TCH was established in October 2006 under the IEEE Robotics and Automation Society (RAS) and is cosponsored by the IEEE Computer Society. The mission of the TCH is to integrate the diverse interests of the highly interdisciplinary research community and to improve communication among the different research areas. Haptics research by its nature is highly multidisciplinary and interdisciplinary and covers many fields such as robotics, control, neuroscience, psychology, rendering, algorithms, interaction design, multimodal, and multisensory research, to name just a few. Major breakthroughs can be anticipated through the integration and crossfertilization of different disciplines. The TCH serves the haptics community by coordinating the scheduling of major haptics conferences, facilitating special conference sessions, workshops, and tutorials, organizing special journal issues on haptics, and contributing toward a journal on haptics.

The TCH is our latest attempt at building a home for the international haptics research community. Following several workshops and conference sessions in the early 1990s, Ed Colgate and Dov Adelstein started the first Haptics Symposium in 1992. In 1996, Ken Salisbury and Mandayam Srinivasan organized the first PHANToM User's Group Workshop following the commercialization of the PHANToM series of force-feedback devices, which have since become the PUMA for haptics research. The EuroHaptics Conference was founded by Alan Wing and Matthias Harders, and its first meeting was held in 2001. In March 2005, Massimo Bergamasco and Antonio Bicchi successfully hosted the first World Haptics Conference (WHC) in Pisa, Italy, which brought together almost 400 haptics researchers from all over the world. To leverage the momentum generated by the first WHC, Antonio Bicchi proposed the idea of launching a TCH in October 2005. Discussions with top haptics researchers ensued over e-mail, and the pros and cons of being associated with the IEEE were debated. A common theme at these discussions was the need for TCH to be inclusive to reflect the diversity of haptics researchers. As far as I am aware, the TCH is the first to have joint sponsorship from two IEEE societies.

Among our many accomplishments in the first year, we received the RAS Most Active Technical Committee of the Year Award in 2007 and have successfully launched the *IEEE Transactions on Haptics* for 2008. Needless to say, the success of TCH has been a group effort. I have been ably assisted by my cochairs, members of the executive committee, and our senior advisors. Of the RAS leadership, Dick Volz has been a great president, who has often made the impossible happen. Ken Goldberg and Stephanie White (then vice president of Technical Activities, Computer Society) have guided us with great openness. We are also thrilled to be working with the new RAS President Bruno Siciliano, who has been very supportive of TCH.

To learn more about TCH and to join us as a member, please visit our Web site at www.worldhaptics.org. Information about all RAS technical committees is at http://tab.ieee-ras.org/.

Digital Object Identifier 10.1109/M-RA.2008.916107

ROV Competition Helps Students Develop Technical Skills and Build Marine Technical Workforce

By Jill Zande and Caroline Brown

Although we owe so much of our life to the marine environment, we devote too few efforts to the needs of that ecosystem. To engage more students in marine studies, the MATE center has established a series of competition in underwater robotics. With the following column by the center director, we hope to motivate RAS members to look at oceans too for their research and teaching activities.

Paolo Fiorini,
RAS Education Committee Cochair

On a global basis, the economic importance of ocean industries cannot be overstated. Between three and five percent of Europe's gross domestic product (GDP) is estimated to be generated by marine-based industries and services (not including the value of raw materials, such as oil, gas, or fish), and Europe's maritime regions account for more than 40% of its GDP [1]. Ocean industries contribute nearly Can$23 million [2] and more than US$117 billion [3] to their respective GDPs.

Marine-related industries require highly skilled technical professionals to continue to grow their contribution to the global economy. Recruiting well-trained and competent professionals is critical to their survival.

Yet, despite the critical need for a well-prepared marine workforce, reliable information about marine technology careers has not been widely available to students and educators, which results in an historical shortage of well-trained and educated technical workers. To help address the increasing need for an appropriately trained and educated marine technology workforce, the Marine Advanced Technology Education (MATE) Center was established in 1997 with funding from the National Science Foundation (NSF).

Improving Marine Technology Education

Headquartered at Monterey Peninsula College in Monterey, California, MATE's mission is to improve marine technical education and increase the number of highly skilled technical professionals who enter ocean-related occupations.

MATE has a unique approach to helping students develop the skills needed to enter the marine technology workforce (Figure 1). The MATE model first focuses on conducting marine workforce studies that outline the needs of marine technology employers. These needs are taken into account when developing occupational definitions and guidelines for occupational knowledge and skills for marine technology careers, curricula, courses, and educational and career-management programs at community colleges and other educational institutions and faculty development programs that help educators

Digital Object Identifier 10.1109/M-RA.2007.914923

Outcomes	Products
1) Needs Identification	List of Critical Workforce Needs from Industry Queries
2) Occupational Definitions	↓ Industry- and Government-recognized Occupational Categories
3) Occupational Knowledge and Skill Guidelines	↓ Industry-Identified Knowledge and Skills for Specific Occupations
4) Skill Cluster Competencies	↓ Knowledge and Skill Grouped by Subject Area
5) Instructional Materials and Services	↓ Competency-Based Assessments, Modules, Courses, Faculty Development Workshops, and Internships
6) Educational Programs	↓ Degree and Certificate Programs Based on Instructional Materials
7) Career Management Programs	↓ Job Placement Programs, Professional Development Courses

Figure 1.

incorporate marine-related technology information and activities into their classrooms.

Another critical component of the MATE model is a student robotics competition that focuses on underwater robots or remotely operated vehicles (ROVs). The International Student ROV Competition is the first student robotics competition to focus exclusively on ROVs. The competition is held in partnership with the Marine Technology Society (MTS) ROV Committee and supported by the NSF, the National Oceanic and Atmospheric Administration, the MTS ROV Committee, and other ocean and technology-related organizations, including IEEE's Oceanic Engineering Society.

Competition Simulates Workplace Environment

Marine industries that rely on ROVs include scientific research, offshore oil and gas exploration, telecommunications engineering and construction, underwater archaeology, underwater construction and structural inspections, and port and harbor security. MATE's ROV Competition presents middle school, high school, community college, and university students with the same types of challenges that scientists and engineers face when working in these environments.

Using missions that simulate a high-performance workplace environment, student teams from all over the world compete with ROVs that they design and build. Examples of competition missions include installing, recovering, repairing, and maintaining simulated electronics instruments and maneuvering through an obstacle course designed to simulate an oil pipeline.

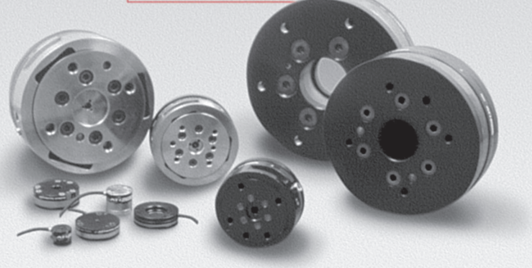
In addition to the underwater missions, teams must make oral and written engineering presentations to a panel of judges who represent various aspects of the marine industry or who represent marine industry, research, government, and the military. Each team is evaluated on the design, construction, and performance of its ROV, its ability to communicate what it learned, and how it can put its knowledge to use in designing and building its ROV.

Students Apply STEM Skills and Connect with Industry Professionals

The competition encourages students to apply science, technology, engineering, mathematics (STEM) skills and teaches teamwork and critical thinking skills. In addition, it helps students become aware of careers where they can apply these skills, a critical step in addressing the shortage of qualified engineers and technical professionals.

An important aspect of the ROV Competition is that it facilitates connections between students, educators, and employers—connections that both academia and industry often claim are lacking. MATE enlists employers—industries, research institutions, government agencies, professional societies, and corporate and private foundations—to become involved in the event by helping to support the competing schools and colleges by contributing funds, equipment, supplies, and technical expertise.

Since it was first held in 2002, the ROV Competition has grown to include nearly 100 supporting organizations. Competition supporters have contributed building materials, supplies, and equipment as well as travel and room and board for participating teams. In total, industry professionals volunteer their time as mentors, technical advisors, and competition judges. Most of the student teams are provided with some form of direct contact—such as mentoring, funding, equipment donations, or visits to



facilities—with industry professionals during the design and building process.

Conclusions

Each year, the competition is held in a new location and focuses on a different theme that exposes students to the many aspects of the marine technology sector. In 2008, the ROV Competition will be held in partnership with Ridge 2000, an interdisciplinary research program sponsored by the NSF and designed to study the biology, chemistry, geology, and geophysics of Earth's ocean ridge systems. Hosted at the Scripps Institution of Oceanography and the University of California, San Diego, the competition will be held on 26–28 June 2008.

The competition theme focuses on hydrothermal vents, seafloor hot springs that discharge continuous streams of hot fluids into the surrounding cold ocean water, and the technologies used to study these deep-sea environments. Before the June event, student teams will participate in one of 17 regional events held in pools around the world. (IEEE in Hawaii has been very supportive of the Hawaii Underwater Robot Competition.) The top winners from the regional competitions will earn a spot in the international competition.

The International Student ROV Competition is an integral part of a model that empowers students to make informed decisions about marine technology careers and the key skills and experiences that will enable them to seek gainful employment in ocean-related fields. By designing and building ROVs, participating in real-world competition scenarios, and coming into direct contact with industry professionals, students apply STEM skills in a fun and exciting manner and increase their knowledge of marine-related technical careers.

More information about MATE's ROV Competition can be found at http://www.marinetech.org/rov_competition/.

References

[1] European Commission Maritime Affairs Maritime Policy Task Force. (2006, July 6). Towards a future maritime policy for the union: A European vision for the oceans and seas, p. 3. [Online]. Available: http://ec.europa.eu/maritimeaffairs/pdf/com_2006_0275_en_part2.pdf

[2] Fisheries and Oceans Canada Statistical Services. (2005, Dec. 9). Canada's ocean industries: Contribution to the economy 1988–2000. [Online]. Available: http://www.dfo-mpo.gc.ca/communic/Statistics/oceans/economy/contribution/table3_25a_e.htm

[3] U.S. Commission on Ocean Policy. (2004, Sept. 20). An ocean blueprint for the 21st century, p. 3. [Online]. Available: http://www.oceancommission.gov/documents/full_color_rpt/01_chapter1.pdf

Jill Zande is the associate director, coprincipal investigator, and ROV Competition coordinator for the MATE Center. She is a member of the MTS Education Task Force, the editorial board of the *MTS Journal*, and the MTS ROV and Education Committees, and she is chair of the MTS Monterey Bay section. She earned a master's degree in oceanography and coastal studies at Louisiana State University, where she researched hydrocarbon seep communities in the Gulf of Mexico and seagrass communities in coastal environments.

Caroline Brown is a science writer for the MATE Center. She earned her master's degree in journalism from the University of North Carolina, where she studied the use of mass communication in immigrant communities.

Address for Correspondence: Jill Zande and Caroline Brown, The MATE Center, Monterey Peninsula College, 980 Fremont St., Monterey, CA 93940 USA. Phone +1 831 645 1393. E-mail: jzande@marinetech.org, caroline@carolinebrown.com.

ICRA'08 Comics Contest

Organizers: Aude Billard, Jorge Cham

The ICRA'08 IEEE-RAS robot comics contest is an initiative sponsored by the Robotics and Automation Society (RAS).

- Folks are invited to compete either as teams or on an individual basis. The jury will be composed of five experts in robotics and comics design.
- RAS will award a prize of US$500.00 to the winning team. The best comics will be published in the IEEE-RAS magazine, with a photograph of the winning team.
- All comics will be displayed during the awards ceremony, see the ICRA'08 program.

Who can compete:

Any registered participant at ICRA'08. Register for ICRA'08 at http://www.icra2008.org.

How to compete:

To enter the competition, you can either submit your comics in pdf or jpg format by registering at **http://lasa.epfl.ch/icra08/comics.html**, or you can have it scanned at the desk on the first day of ICRA'08.

The competition will close at 6:00 p.m. on the first day of the conference, 20 May 2008. All entries must be accompanied by a statement that the work is original and all rights belong to the entrant.

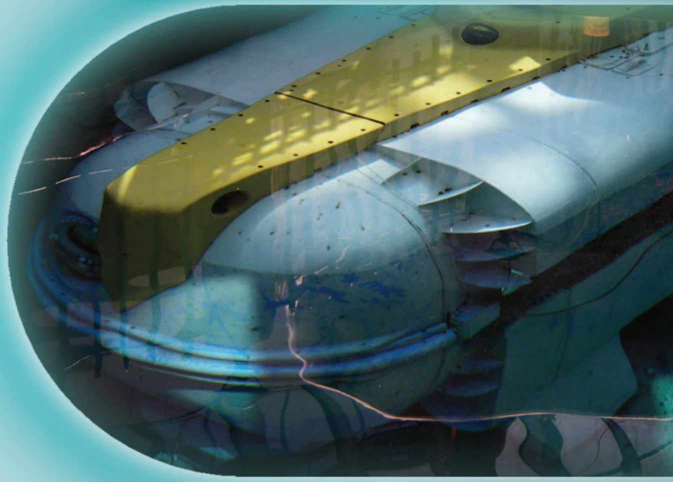
The Entrapment/ Escorting Mission

An Experimental Study Using a Multirobot System

BY GIANLUCA ANTONELLI, FILIPPO ARRICHIELLO,
AND STEFANO CHIAVERINI

© DREAMSTIME

© DREAMSTIME



© ISTOCKPHOTO

Mobile Multirobot Systems

In recent years, multirobot systems have been the object of widespread research interest in the scientific community, given their application in different fields of robotics such as service, military, or educational robotics. The interest in using multirobot systems is due to their unique characteristics such as increasing the redundancy and the flexibility of mission execution, making the system tolerant to possible robot faults, accomplishing missions not physically executable by a single robot, or achieving the same mission of a single robot while reducing the execution time and increasing the performance. Moreover, the flexibility of multirobot systems is increased by the realization of systems with different typologies of autonomous vehicles such as wheeled mobile robots [15], [10], autonomous underwater vehicles [13], unmanned aerial vehicles [22], [6], and marine surface vessels [8], [16]. The research in multirobot systems has matured to the point where systems with hundreds of robots [18], [14] or teams of heterogeneous robots [9], [11] are being proposed.

Consistent research is devoted to applications such as exploration and mapping of unknown environments, pushing large objects, or studying biological systems, but few studies explicitly address the entrapment/escorting or catching problem. The entrapment/escorting mission consists in surrounding a moving target by reducing its escape windows (or, similarly, protecting a target by reducing the intrusion paths for an external agent) and can have different applications such as robotic surveillance security systems, military robotics, or entertainment robotics. In [17], a set of fuzzy rules are proposed to surround and entrap an escaping target, and these rules are experimentally validated on a three-robot system. In [23], an approach is presented to track and acquire a target and is experimentally validated by the use of two mobile robots.

From a control point of view, multirobot systems pose broadly different problems, such as motion planning and coordination, behavior emergence in unknown environments or unpredictable situations, information sharing, and the choice of sensor equipment. Among the possible control techniques, most control strategies for mobile robots resort to biologically inspired concepts, i.e., using elementary control rules of various animals (e.g., ants, bees, birds, and fishes) and trying to reproduce their group behavior (e.g., foraging, flocking, homing, and dispersing [19]) in cooperative robotic systems. Behavior-based approaches give the system the autonomy to operate in unpredicted situations using sensors to obtain information about the environment; thus, they are useful in guiding a multirobot system in an unknown or dynamically changing environment.

In this article, the entrapment/escorting mission is handled by resorting to the kinematic control presented in [4] and [5]. The proposed approach is based on a new kind of behavioral control, the null space-based behavioral (NSB) control [13]. This method differs from other existing behavioral coordination methods in the way that the outputs of the single elementary behaviors are merged to compose the final behavior. The NSB has been extensively tested in formation control missions [2], while in this article, its application to the entrapment/escorting mission is discussed. In particular, the control strategy has been validated both

Digital Object Identifier 10.1109/M-RA.2007.914932

in simulation and in several experimental case studies, where a team of six Khepera II mobile robots has to entrap a moving target represented by a tennis ball randomly pushed by hand. The simulative and experimental results show the effectiveness of the approach. Moreover, the control approach has been made robust such that in spite of the loss of a vehicle, in case of failure of one or more vehicles, the system autonomously reconfigures itself to correctly achieve the mission. Accordingly, in the experimental case studies shown, an intentional failure of one of the robots is imposed so as to show the structural robustness and the dynamic scalability property of the proposed technique with respect to the eventual loss of vehicles.

The NSB Control for Multirobot Systems

In a general robot mission, the accomplishment of several tasks at the same time is of interest. For instance, in a formation control mission, it is required that the vehicles maintain a given relative position while avoiding obstacles. A possible technique to handle the composition of the tasks has been proposed in [7], which consists in assigning a relative priority to single task functions by resorting to the task priority inverse kinematics introduced in [20] for ground-fixed redundant manipulators. Nevertheless, as discussed in [12], in the case of conflicting tasks, it is necessary to devise singularity robust algorithms that ensure proper functioning of the inverse velocity mapping.

Based on these works, this approach to the composition of the tasks has been developed in [4] in the framework of the singularity robust task priority inverse kinematics [12].

By defining the task variable to be controlled as $\boldsymbol{\sigma} \in \mathbb{R}^m$ and the system configuration as $\boldsymbol{p} \in \mathbb{R}^l$,

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}), \tag{1}$$

with the corresponding differential relationship

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}} \boldsymbol{v} = \boldsymbol{J}(\boldsymbol{p})\boldsymbol{v}, \tag{2}$$

where $\boldsymbol{J} \in \mathbb{R}^{m \times l}$ is the configuration-dependent task Jacobian matrix, and $\boldsymbol{v} \in \mathbb{R}^l$ is the system velocity.

An effective way of generating motion references $\boldsymbol{p}_{\mathrm{d}}(t)$ for the vehicles starting from the desired values $\boldsymbol{\sigma}_{\mathrm{d}}(t)$ of the task function is to act at the differential level by inverting the (locally linear) mapping [2]. In fact, this problem has been widely studied in robotics (see, e.g., [24] for a tutorial). A typical requirement is to pursue a minimum-norm velocity, leading to a closed-loop inverse kinematics (CLIK) least-square solution:

$$\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{J}^{\dagger}(\dot{\boldsymbol{\sigma}}_{\mathrm{d}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1}(\dot{\boldsymbol{\sigma}}_{\mathrm{d}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}), \tag{3}$$

where $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains, and $\tilde{\boldsymbol{\sigma}}$ is the task error defined as $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{\mathrm{d}} - \boldsymbol{\sigma}$.

The NSB control intrinsically requires a differentiable analytic expression of the tasks defined, so that it is possible to compute the required Jacobians. In detail, based on the analogy of (3), the single task velocity is computed as

$$\boldsymbol{v}_i = \boldsymbol{J}_i^{\dagger}(\dot{\boldsymbol{\sigma}}_{i,\mathrm{d}} + \boldsymbol{\Lambda}_i\tilde{\boldsymbol{\sigma}}_i), \tag{4}$$

where the subscript i denotes the ith task quantities. If the subscript i also denotes the degree of priority of the task with, e.g., task 1 being the highest-priority one, according to [12], the closed-loop solution (3) is modified into

$$\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1\right)\left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^{\dagger}\boldsymbol{J}_2\right)\boldsymbol{v}_3\right]. \tag{5}$$

The NSB control always fulfills the highest-priority task at nonsingular configurations. Remarkably, (5) has an agreeable geometrical interpretation. Each task velocity is computed as if it were acting alone. Then, before adding its contribution to the overall vehicle velocity, a lower-priority task is projected onto the null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it.

The Escorting Mission

The mission of escorting a target can be seen as the requirement of surrounding a target whose movement is not known a priori but can be measured in real time. To achieve the mission, the multirobot system has to entrap the target and reduce its possible escape windows by properly distributing the team members around it. Thus, with reference to the planar case, the escorting mission can be satisfied by placing the n vehicles of the team at the vertices of a regular polygon of order n centered in the target and whose sides define a sort of intrusion/escape window (see Figure 1).

Following the NSB approach, the escorting mission is decomposed into elementary subproblems to be individually described and solved, which are as follows:

1) command the robots' centroid to be coincident with the target
2) move the robots on a given circumference around the centroid
3) properly distribute the robots along the circumference
4) avoid collisions among the robots themselves and with obstacles.

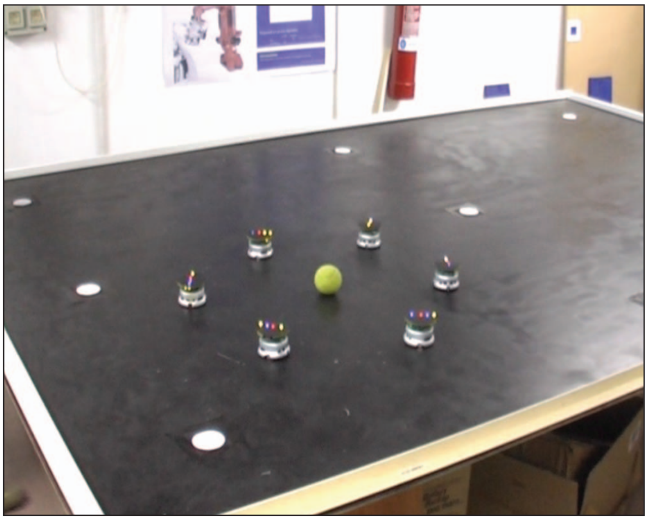
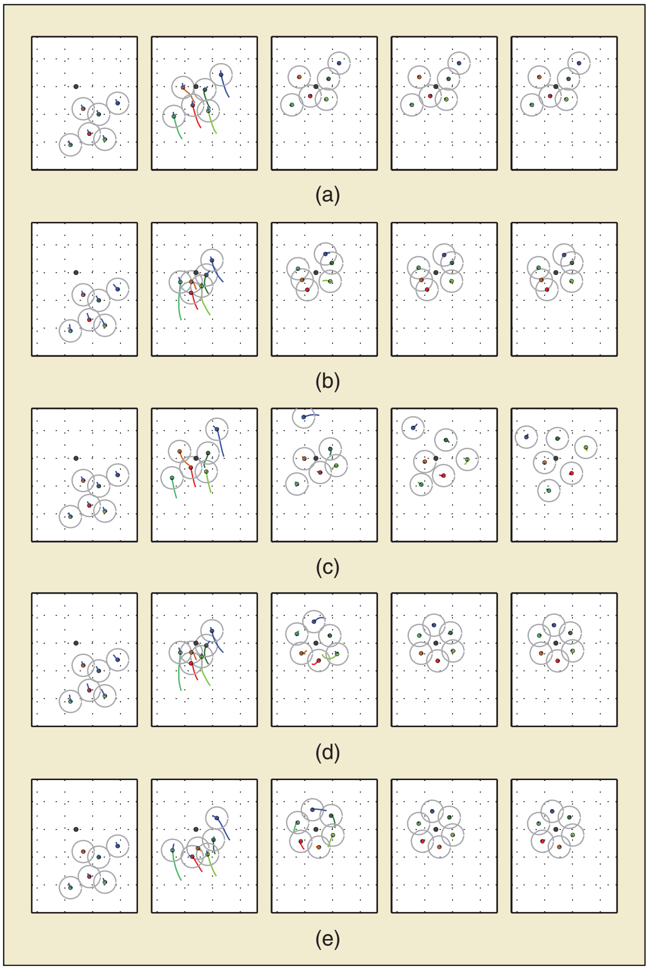


Figure 1. *The entrapment/escorting mission.*

Table 1. Selective activation, relative priority and CLIK gains of the behaviors in the five cases considered.

	Priority					
Task	**A**	**B**	**C**	**D**	**E**	**CLIK Gains**
Centroid on the target	2	2	2	2	3	$\Lambda = 2.0$
Distribution on a circumference	–	3	–	3	2	$\Lambda = 0.5$
Polygon with equal edges	–	–	3	4	4	$\Lambda = 3.0$
Obstacle avoidance	1	1	1	1	1	$\Lambda = 1.0$


(a)

(b)

(c)

(d)

(e)

Figure 2. Simulations of the entrapping mission with partial or full activation of the elementary behaviors: (a) obstacle + centroid; (b) obstacle + centroid + circular; (c) obstacle + centroid + polygon; (d) obstacle + centroid + circular + polygon; and (e) obstacle + circular + centroid + polygon.

For each behavior, a suitable task function is properly designed. Without entering the mathematical details, which can be found in [5], the task function definitions are reported below.

1) For the centroid position, the two-dimensional task function $\boldsymbol{\sigma}_c$ is simply given by

$$\boldsymbol{\sigma}_c = \boldsymbol{f}_c(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p}_i = \bar{\boldsymbol{p}}. \qquad (6)$$

2) The n-dimensional task function

$$\boldsymbol{\sigma}_s = \begin{bmatrix} \vdots \\ \frac{1}{2}(\boldsymbol{p}_i - \boldsymbol{c})^T(\boldsymbol{p}_i - \boldsymbol{c}) \\ \vdots \end{bmatrix} \qquad (7)$$

can be used to keep each robot of the team at a given distance r from a point $\boldsymbol{c} \in \mathbb{R}^2$ by setting

$$\boldsymbol{\sigma}_{s,d} = \begin{bmatrix} \vdots \\ r^2/2 \\ \vdots \end{bmatrix}. \qquad (8)$$

3) Properly distributing the robots along a given circumference is equivalent to making equal the relative distance between successive robots along this circumference. The latter task can be achieved by properly assigning the perimeter of the polygon inscribed in the circumference [5]. In fact, a regular polygon has the maximum perimeter among all the polygons of the same order inscribed on a given circumference. In this article, instead, the same configuration is pursued by requiring that the robots place themselves at the vertices of a polygon with sides of the same length. This is achieved by simply imposing the same distance between adjacent vehicles. It is worth noting that the task function definition used in this article has been shown to be more efficient than that proposed in [5] in the experimental runs.
4) The obstacle avoidance task function is defined individually for each vehicle, i.e., it is not an aggregate task function. In fact, each vehicle needs to avoid both environmental obstacles and the other vehicles. With reference to the generic vehicle of the team, in the presence of a punctual obstacle in the advancing direction, the task function has to elaborate a driving velocity, aligned to the vehicle-obstacle direction, that keeps the vehicle at a safe distance d from the obstacle. Therefore,

$$\sigma_o = \|\boldsymbol{p} - \boldsymbol{p}_o\|$$
$$\sigma_{o,d} = d,$$

where $\boldsymbol{p}_o$ is the obstacle position.

According to (4), each elementary behavior outputs a velocity reference command to each robot of the team. To obtain the actual motion reference commands to the robots, the outputs that accomplish the single behaviors are merged by (5) on the basis of the active behaviors and on their priority orders.

Simulations

Extensive simulations have been performed with a selective activation of the behaviors and with different priority orders to better emphasize the meaning of each behavior and the importance of the priority orders. The simulations concern a team of nonholonomic robots that, starting from the same initial configuration, are commanded to accomplish different missions, depending on the active behaviors. It is worth noting that the simulation software uses the same control code realized to perform the experiments. Of course, in the simulations, instead of reading data from the camera vision system and sending data to the robots' actuators, the control code exchanges data with a kinematic simulator and a graphical interface. Besides simplifying the debugging of the control code, the simulator also allows the analysis of the behavior of the robots in ideal conditions that set the target performance to be pursued in the experiments. In particular, the absence of stochastic phenomena (e.g., measurement noise, variable delivery time, or loss of data in the radio communication) allows a repeatable comparison of different missions executed by starting from the same initial conditions.

The performed simulations concern five different situations denoted from A to E. Table 1 reports the active behaviors and their relative priority order for each case considered, and the CLIK gains are also given. For instance, in situation B, the highest-priority task is obstacle avoidance, the second-priority task is to keep the centroid of the team on the target, and the third-priority task is to distribute the robots on a circumference centered in the target. The task of placing the robots at the vertices of a regular polygon is not active. Remarkably, obstacle avoidance is always active and chosen as the primary task in all the missions to ensure safe execution of the mission.

Figure 2 reports several steps of the simulation for all the cases considered. In particular, Figure 2(a) shows the steps of mission A in which the robots have to keep their centroid on the target while avoiding collisions with it and among themselves. In this mission, the only control objective is the centroid. The shape of the robots thus remains uncontrolled. However, note that the final shape is not much different from the initial one. This can be explained by recalling that, among all the possible solutions for a single task, the NSB approach chooses at each step the one with the minimum velocity norm. As a consequence, the robots do achieve the mission, minimizing the motion in the null space of the centroid task function.

Figure 2(b) shows a mission (case B) in which the robots have to keep their centroid on the target and arrange themselves on a circumference of fixed radius. It is worth noting that the distribution along the circumference is uncontrolled, and thus the robots do not reach a regular polygonal shape. The addition of a behavior that places the robots at the vertices of a polygon with equal edges permits the accomplishment of the mission of entrapping the target. Figure 2(c) then shows the mission (case C) related to this elementary behavior, in which all the distances between adjacent vehicles surrounding the target are equal.

Cases D and E differ only in the order of priority of the active behaviors. The obtained simulation results are reported in Figure 2(d) and (e), which illustrate how the entrapment/escorting mission can be globally achieved by the use of the four proposed-task functions. Nevertheless, leaving out the obstacle avoidance behavior—the chosen elementary behaviors are not conflicting—all the tasks can be simultaneously solved at the end. Figure 2(d) and (e) shows that, at the last step, the target is surrounded by the vehicles that regularly distribute themselves around it. However, the different order of the priority of the tasks in the two cases changes the transient of the respective simulations.

Experiments

In the following section, the experimental setup and the results of the execution of several escorting missions with intentionally caused faults are reported.

Experimental Setup

The multirobot setup available at Laboratorio di Automazione Industriale of the Università degli Studi di Cassino, Italy, is composed of several Khepera II mobile robots manufactured by K-Team [1]. These are differentially driven mobile robots (with unicycle-like kinematics) with an approximate diameter of 8 cm. Each can communicate through a Bluetooth module with a remote Linux-based PC where the NSB has been implemented. To allow the needed absolute position measurements, we have developed a vision-based system using two CCD cameras, two

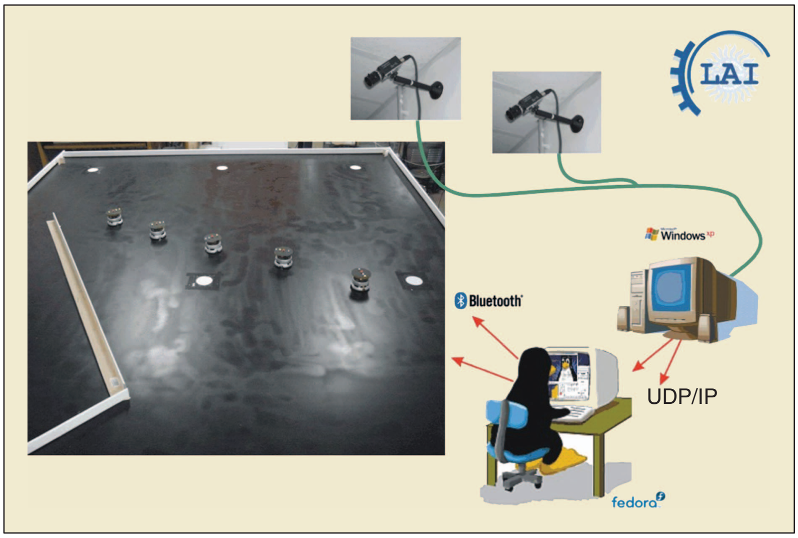


Figure 3. *Sketch of the multirobot setup available at Laboratorio di Automazione Industriale of the Università degli Studi di Cassino, Italy.*

Table 2. Order of priority and CLIK gains for the behaviors in the two experiments.

Task	Gain	Priority First Experiment	 Second Experiment
Obstacle avoidance	$\Lambda = 1.0$	1	1
Distribution on a circumference	$\Lambda = 0.5$	2	3
Centroid on the target	$\Lambda = 2.0$	3	2
Polygon with equal edges	$\Lambda = 3.0$	4	4

Matrox Meteor-II frame grabbers, and self-developed C++ image-processing functions. The acquired images are 1,024 × 768 RGB bitmaps. The measurement error has an upper bound of ≈0.5 cm and ≈1°. The remote PC, which implements the NSB control, receives the position measurements from the vision system at a sampling time of 100 ms. The NSB outputs the desired linear velocities for each robot, and, therefore, a heading controller is needed to obtain the wheels' desired velocities. We have developed a heading controller derived from the one reported in [21]. The remote PC sends (through the Bluetooth module) the wheels' desired velocities with a sampling time of $T = 80$ ms to each vehicle. The wheels' controller (onboard each robot) is a proportional integral derivative control loop developed by the manufacturer. A saturation of 40 cm/s and 100°/s has been introduced for the linear and angular velocities, respectively. Moreover, the encoders' resolution is such that a quantization of ≈0.8 cm/s and ≈9°/s is experienced. A sketch of the setup is shown in Figure 3.

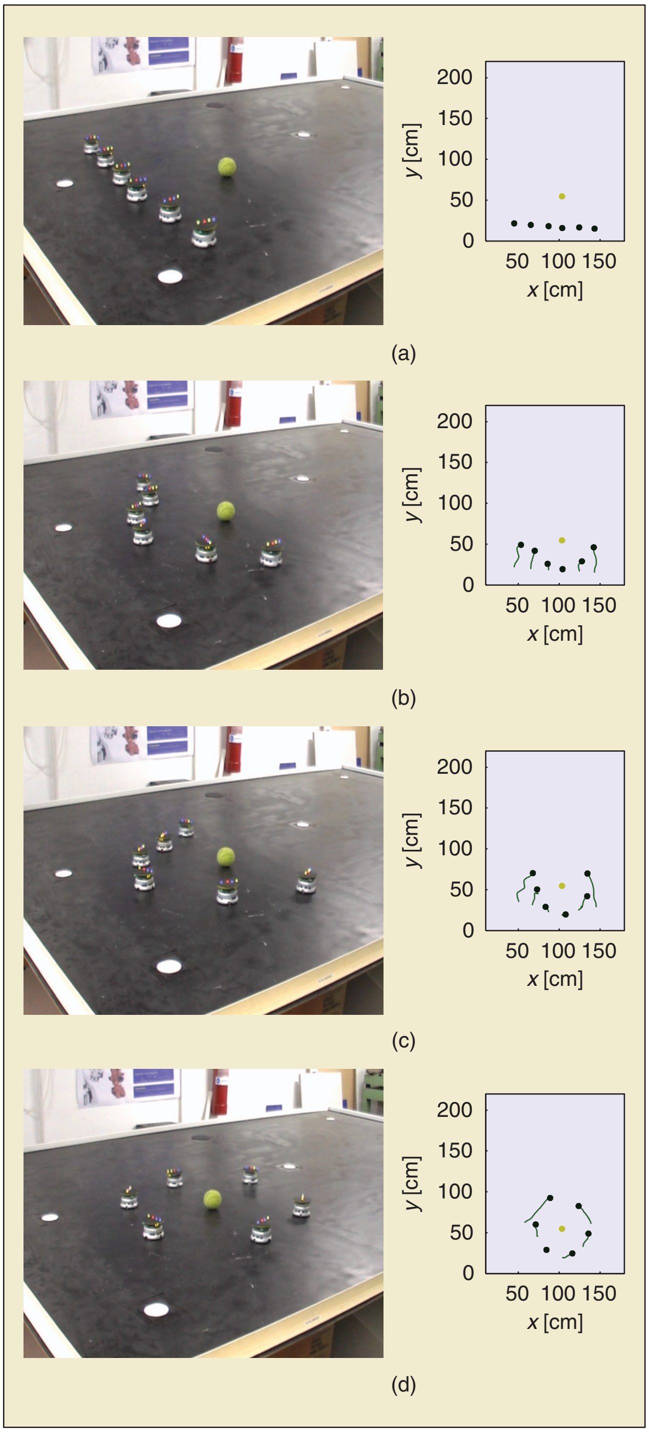


Figure 4. *First set of snapshots of the first escorting experiment: from $t = 0$ s to $t \approx 5$ s.*

Experimental Results

As a challenging case study, we report the experimental results of two different executions of a mission where a tennis ball is the target to be entrapped by a team of six Khepera II mobile robots. In particular, the vehicles should guarantee an escaping window of 40 cm while the safety distance imposed on the vehicles is 20 cm. The desired radius of the surrounding circumference is modified to guarantee the desired escaping window according to the number of robots, i.e., it is modified during the experiments to take into account the loss of one or more vehicles. To underline the effects of the task priority, the two executions differ only in the priority orders, while the topology of the mission and the task gains are exactly the same. The video images of the experiment are presented in two synchronized frames: the one on the left shows the videos acquired by a hand video camera, and the one on the right reports animations obtained using experimental data [25]. These animations are achieved through a self-developed C-based program that uses the OpenGL graphics library under the Linux environment.

For the first mission (the relative video is named `RAM_CIRCULAR.mpg`), we report a 30-s long section of the escorting mission. Initially, the ball is still, and the six robots have to surround it. Then, at $t \approx 6$ s one robot is moved away from the arena to simulate a failure, then it is put back in the arena at $t \approx 9$ s. Moreover, at $t \approx 16$ s, the target is pushed to impose a reconfiguration to the robots. The order of priority of the four tasks and the corresponding CLIK gains are summarized in Table 2. The video shows that the robots' positions in the circumference are not fixed a priori. After the failure of the robot, in fact, it is put back in the arena in a random position, and the platoon automatically reconfigures to include the recently added robot.

In Figure 4, the first five seconds are reported. The target is still, and the vehicles are required to surround it. It can be observed that the obstacle avoidance task is always the primary task, and the vehicles avoid hitting each other during the movement. Moreover, no predefined position is assigned around the target. A hexagon-like configuration is the natural structure of the six-robot formation since the regular polygon guarantees the minimum distance between adjacent points on a given circumference.

A fault is caused at $t \approx 6$ s by moving away a robot by hand and further obscuring it to the camera. The algorithm recognizes the absence of a robot as a major fault, i.e., the vehicle is lost, and the remaining robots have to complete the mission, ignoring the damaged robot and considering it as an obstacle. After the reconfiguration is successfully achieved, the robot is put again in the arena at $t \approx 9$ s. In Figure 5, the second group of snapshots are given, from $t \approx 5$ s to $t \approx 14$ s. Figure 5(a) and (b) shows the moment in which one of the robot is moved away. In Figure 5(b) and (c), the remaining vehicles are no longer minimizing the escape space of the target and need to reconfigure to achieve the lowest-priority task. From the geometrical point of view, it can be observed that this is achieved by positioning the vehicles from the vertices of a six-side regular polygon to those of a five-side regular one [Figure 5(b)] and modifying the desired radius accordingly. Moreover, when the vehicle is put back in the arena [Figure 5(c)] the formation is again rearranged into a hexagon. Note that, since the position of the robots in the formation is not specified, after recovering from the fault, the vehicle takes a different position from the one it had before the fault [Figure 5(d)].

The target is pushed twice to demonstrate that the algorithm is working in real time, and the vehicles reconfigure such that the escort mission is still accomplished. This can be seen

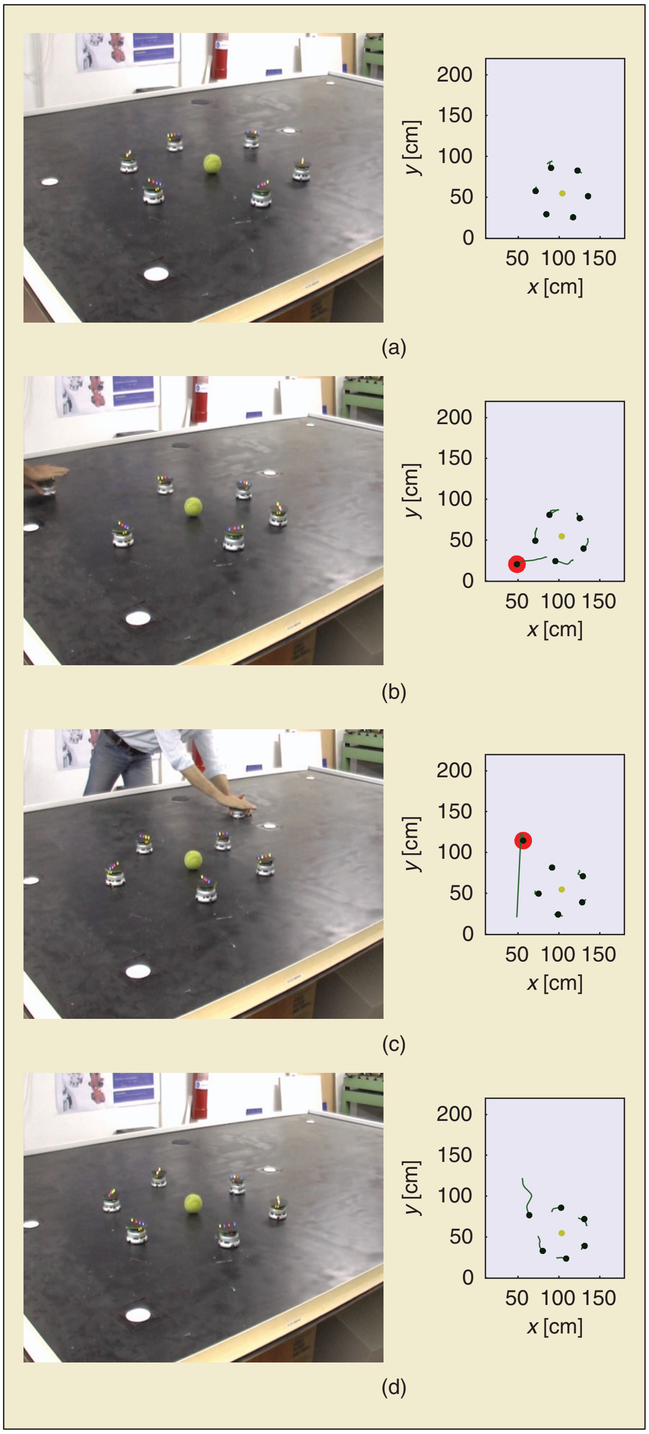


Figure 5. *The second set of snapshots of the escorting experiment: from t ≈ 6 s to t ≈ 14 s. A fault is caused by (b) moving away a robot by hand and (c) further obscuring it to the camera. (d) After the reconfiguration is successfully achieved, the robot is put again in the arena.*

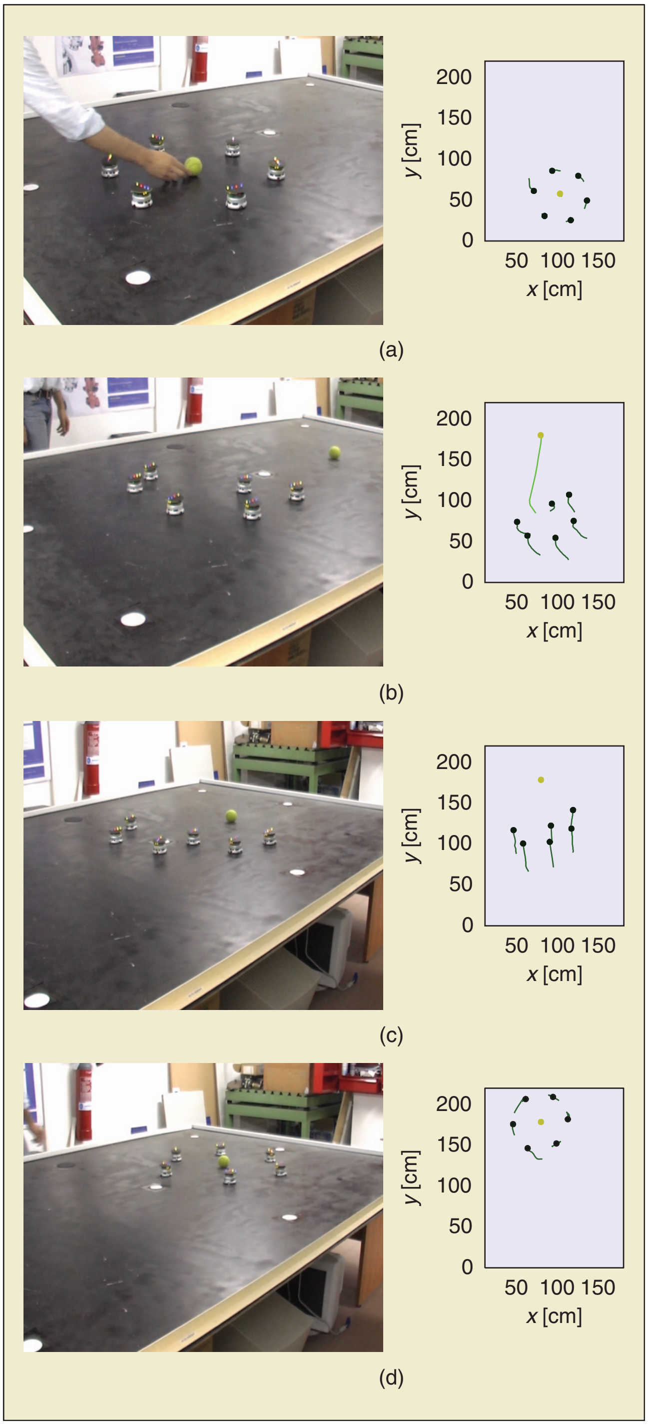


Figure 6. *Third set of snapshots of the escorting experiment: from t ≈ 14 s to t ≈ 29 s.*

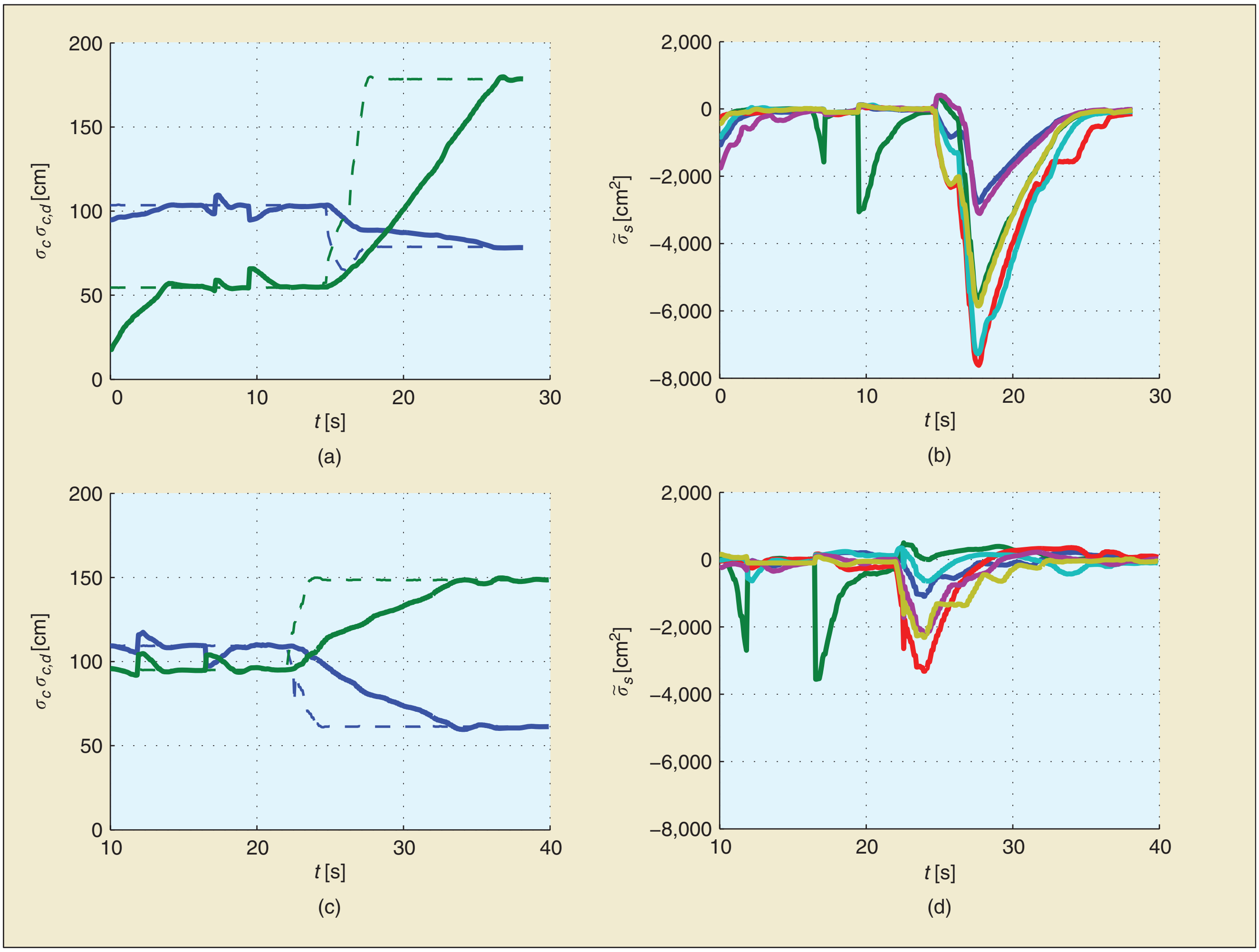


Figure 7. *Errors of the (a)–(c) centroid and circular (c)–(d) task functions relative to the two experiments shown with different tasks' priority order. (a)–(b) obstacle + circular + centroid + polygon and (c)–(d) obstacle + centroid + circular + polygon.*

in Figure 6, where the last ≈15 s of the mission is reported. Moreover, it can also be observed that, after the motion of the target, the vehicles reconfigure themselves with a different position relative to the first steady-state condition.

A second experiment was done that differs from the first one only in the priority orders of the tasks. These are reported in Table 2 together with the corresponding CLIK gains. The complete experimental results are not reported here, but see [25] for a video of the complete experiment (named `RAM_CENTROID.mpg`).

Finally, the time history of the centroid task function (solid line) against its desired value [i.e., the ball position (dashed line)] and the time history of the errors of the circular task function for both the experiments are reported in Figure 7. The errors are first convergent to zero. Then, several transients caused by the abrupt fault, the abrupt vehicle recovery, and the target movement can be observed. The behavior of the team in the two experiments is quite similar. However, it is worth noting from Figure 7(a) and (b) that the circular task function has a more regular shape when it has higher priority (in the first experiment).

Conclusions

The problem of escorting a moving target with a team of mobile robots was solved in this article by resorting to a formation control algorithm that can be cast in the framework of the NSB control approach. The overall mission, therefore, is decomposed into properly defined elementary tasks that are hierarchically arranged, so that the higher-priority tasks are not influenced by the lower-priority ones. The validity of the proposed approach has been proved by both simulation case studies and experimental results with a team of six Khepera II mobile robots. Stability analysis concerning effective conditions needed to verify that the behaviors of specific missions are properly defined and merged is under investigation. Future improvements might regard decentralization of the algorithm, consideration of the vehicles' nonholonomicity in the definition of the task functions, and the introduction of a piecewise-constant constraint for the linear velocity to allow application of the method to teams of cruise vehicles (e.g., a fleet of vessels or a flight of planes).

Keywords

Multirobot, coordination control, behavioral approach.

References

[1] K-Team. [Online]. Available: http://www.k-team.com/

[2] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini, "Experiences of formation control of multi-robot systems with the Null-Space-

based Behavioral control," in *Proc. 2007 IEEE Int. Conf. Robotics and Automation*, Rome, 2007, pp. 1068–1073.

[3] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The Null-Space-based Behavioral control for autonomous robotic systems," *J. Intell. Service Robot.*, vol. 1, no. 1, pp. 27–39, Jan. 2008.

[4] G. Antonelli and S. Chiaverini, "Kinematic control of a platoon of autonomous vehicles," in *Proc. 2003 IEEE Int. Conf. Robotics and Automation*, Taipei, 2003, pp. 1464–1469.

[5] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, pp. 1285–1292, Dec. 2006.

[6] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proc. IEEE*, vol. 94, no. 7, pp. 1306–1324, July 2006.

[7] B. Bishop and D. Stilwell, "On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles," in *Proc. 2001 IEEE Int. Conf. Control Applications*, México, 2001, pp. 823–828.

[8] E. Borhaug, A. Pavlov, R. Ghabcheloo, K. Pettersen, A. Pascoal, and C. Silvestre, "Formation control of underactuated marine vehicles with communication constraints," in *Proc. 7th IFAC Conf. Manoeuvring and Control of Marine Craft*, Lisbon, 2006.

[9] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE J. Robot.*, vol. 21, no. 3, pp. 376–386, June 2005.

[10] S. Carpin and L. Parker, "Cooperative motion coordination amidst dynamic obstacles," *Distrib. Autonom. Robot. Syst.*, vol. 5, pp. 145–154, 2002.

[11] L. Chaimowicz, B. Grocholsky, J. Keller, V. Kumar, and C. Taylor, "Experiments in multirobot air-ground coordination," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, p. 4053.

[12] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 13, no. 3, pp. 398–410, 1997.

[13] E. Fiorelli, N. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. Fratantoni, "Multi-AUV control and adaptive sampling in monterey bay," in *Proc. IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations*, Sebasco, ME, June 2004, pp. 134–147.

[14] A. Howard, L. Parker, and G. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *Int. J. Robot. Res.*, vol. 25, nos. 5–6, p. 431, 2006.

[15] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. Ganino, M. Garrett, S. Joshi, and P. Schenker, "CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration," *IEEE Trans. Syst. Man Cybern. A*, vol. 33, no. 5, pp. 550–559, 2003.

[16] I. Ihle, R. Skjetne, and T. Fossen, "Nonlinear formation control of marine craft with experimental results," in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, The Bahamas, 2004, vol. 1, pp. 680–685.

[17] T. Kamano, T. Yasuno, T. Suzuki, Y. Hasegawa, H. Harada, and Y. Kataoka, "Design and implementation of fuzzy cooperative catching controller for multiple mobile robots," in *Proc. 26th Ann. Conf. of the IEEE Industrial Electronics Society*, 2000, pp. 1749–1754.

[18] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams," *Experimental Robotics IX.* (Springer Tracts in Advanced Robotics), New York: Springer Verlag, 2005.

[19] M. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robot. Autonom. Syst.*, vol. 16, no. 2, pp. 321–331, 1995.

[20] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[21] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: design, implementation, and experimental validation," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 6, pp. 835–852, 2002.

[22] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, no. 1, p. 3, 2002.

[23] P. Pirjanian and M. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, 2000, vol. 3, pp. 2696–2702.

[24] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 3, pp. 201–212, 1990.

[25] Robotics Research group at the Industrial Automation Laboratory, University of Cassino, Cassino, Italy. [Online]. Available: http://webuser.unicas.it/lai/robotica/video/

Gianluca Antonelli received the laurea degree in electronic engineering and the research doctorate degree in electronic engineering and computer science from the University of Naples, Italy, in 1995 and 2000, respectively. He is currently an associate professor at the University of Cassino, Italy. He is a Senior Member of the IEEE since June 2006. Since September 2005, he has been the associate editor of *IEEE Transactions on Robotics*. He is also the editor of the *Journal of Intelligent Service Robotics* since November 2007. His research interests include simulation and control of underwater robotic systems, force/motion control of robot manipulators, multirobot systems, and identification. He has published 22 international journal papers and about 60 conference papers. He is the author of *Underwater Robots* (Springer Verlag; first edition, 2003; second edition, 2006).

Filippo Arrichiello received the laurea degree in mechanical engineering from the University of Naples, Italy, and the research doctorate degree in electrical and information engineering from the University of Cassino, Italy, in 2003 and 2007 respectively. He is currently a research assistant at the University of Cassino, Italy. From March to September 2005, he attended the Marie Curie Program at the Norwegian University of Science and Technology in Trondheim, Norway. His research activity mainly concerns industrial and mobile robotics with specific interest in multirobot system applications. He has published 14 journal and conference papers.

Stefano Chiaverini received the laurea and the research doctorate degrees in electronics engineering from the University of Naples, Italy, in 1986 and 1990, respectively. He is currently a professor of automatic control at the engineering faculty of the University of Cassino, Italy, where he is vice head of the Department of Automation, Electromagnetics, Information Engineering, and Industrial Mathematics. His research interests include manipulator inverse kinematics techniques, redundant manipulator control, cooperative robot systems, force/position control of manipulators, underwater robotic systems, and mobile robotic systems. He has published more than 150 international journal and conference papers, and he is the coeditor of *Complex Robotic Systems* (Springer Verlag, 1998) and is the coauthor of *Fondamenti di Sistemi Dinamici* (McGraw Hill, 2003). From January 2000 to February 2004, he has been an associate editor of *IEEE Transactions on Robotics and Automation* and, since August 2004, the editor of *IEEE/ASME Transactions on Mechatronics*. He is a Senior Member of the IEEE and a member of the IEEE Robotics and Automation Society conference editorial board.

Address for Correspondence: Filippo Arrichiello, Università degli Studi di Cassino, Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Via G. Di Biasio 43, 03043, Cassino (FR), Italy. E-mail: f.arrichiello@unicas.it.

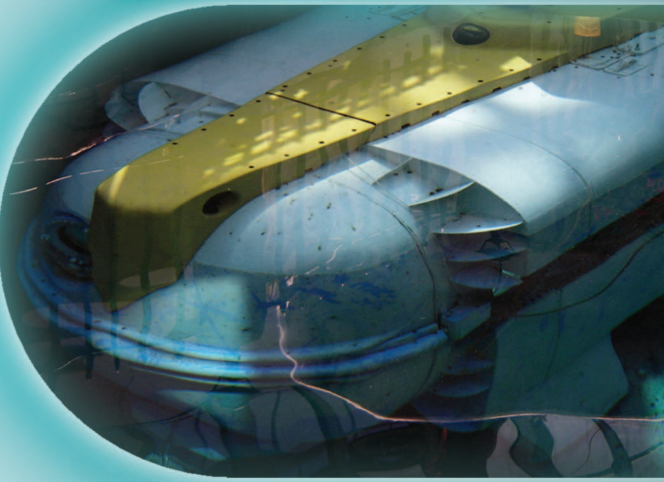
Courteous Cars

Decentralized Multiagent Traffic Coordination

BY HADAS KRESS-GAZIT, DAVID C. CONNER,
HOWIE CHOSET, ALFRED A. RIZZI,
AND GEORGE J. PAPPAS


© DREAMSTIME


© DREAMSTIME


© ISTOCKPHOTO

Mobile Multirobot Systems

A major goal in robotics is to develop machines that perform useful tasks with minimal supervision. Instead of requiring each small detail to be specified, we would like to describe the task at a high level and have the system autonomously execute in a manner that satisfies that desired task. While the single-robot case is difficult enough, moving to a multirobot behavior adds another layer of challenges. Having every robot achieve its specific goals while contributing to a global coordinated task requires each robot to react to information about other robots, for example, to avoid collisions. Furthermore, each robot must incorporate new information into its decision framework to react to environmental changes induced by other robots since this knowledge may effect its behavior.

This article uses the approach presented in [1], in which low-level continuous feedback control policies are combined with a formally correct discrete automaton, thus satisfying a specified high-level behavior for any initial state in the domain of the low-level policies. This allows the approach to be applied to systems that react to changing dynamic environments and that may have complex nonlinear constraints, such as nonholonomic constraints, input bounds, and obstacles or body shape. Furthermore, given a collection of local feedback control policies, the approach is fully automatic and correct by construction.

Multirobot high-level behavior is captured naturally in a decentralized manner in this approach. By allowing each robot's automaton to depend on information gathered locally from other robots and the environment, each robot can react during the execution to the other robots' behaviors. The approach [1] also supports creating a single centralized controller for the group of robots. However, such a controller would encode global knowledge of all robots' state and therefore will not scale well. Furthermore, agent synchronization issues might emerge. By choosing the decentralized approach, the controller remains tractable and the agent's behavior only depends on local events. Although the decentralized approach has some limitations too, it seems more suited for multirobot behaviors.

The approach combines the strengths of control theoretic and computer science approaches. Control theoretic approaches offer provable guarantees over local domains; unfortunately, the control design requires a low-level specification of the task. In the presence of obstacles, designing a global control policy becomes unreasonably difficult. In contrast, discrete planning advances from computer science offer the ability to specify more general behaviors and generate verifiable solutions at the discrete level but lack the continuous guarantees and robustness offered by feedback.

By using a collection of local feedback control policies that offer continuous guarantees and composing them in a formal manner using discrete automata, the approach automatically creates a hybrid feedback control policy that satisfies a given high-level specification without ever planning a specific configuration space path. To be more specific, given the robot's workspace, its limitations, its sensors (i.e., the local information it can get from the environment and the other robots), and the high-level specifications it should satisfy, the

Digital Object Identifier 10.1109/M-RA.2007.914921

approach first populates the configuration space with local continuous feedback control policies. These policies drive the robot in paths that are guaranteed to stay in the appropriate lane while avoiding collisions with static obstacles. Furthermore, these policies induce a discrete graph, i.e., if policy Φ_A drives the robot to the domain of policy Φ_B, there is a discrete transition from policy Φ_A to Φ_B. Using this discrete graph, the approach automatically synthesizes a discrete automaton that satisfies the high-level specifications.

These high-level specifications are given in a subset of linear temporal logic (LTL). Loosely speaking, temporal logic extends propositional logic (AND, OR, NOT) by adding temporal connectives (ALWAYS, EVENTUALLY, . . .), thus enabling one to reason about propositions that can change truth value with time. The specifications that are considered in this article usually depend on the local input from the environment and from the other robots that are part of the environment from one robot's perspective. Finally, the system continuously executes the automaton based on the state of the environment and the vehicle by activating the continuous policies. Given proper sensor function, this execution guarantees that the robot will satisfy its intended behavior using a decentralized approach.

As a demonstration of the general approach, this article presents a familiar example: conventional Ackermann-steered vehicles operating in an urban environment. Figure 1 shows the environment and a simulation snapshot with eight currently active vehicles. The vehicles in this simulation execute one of two automata. The first automaton satisfies the high-level specification "drive around until you find a free parking space and then park." The second automaton satisfies the specification "Leave the block, obeying traffic rules, through Exit$_i$," where i is given as input. This article discusses the design and deployment of the local feedback policies, the automatic generation of automata that satisfy high-level specifications, and the continuous execution.

The approach to composing low-level policies is based on our earlier work using sequential composition [2], [3]. Sequential composition depends on well-defined policy domains and well-defined goal sets to enable tests that the goal set of one policy is contained in the domain of another. For idealized (point) systems, several techniques are available for generating suitable policies [4]–[8]. Our recent work extends these ideas to a more complex system model with Ackermann steering, input bounds, and the shape of the vehicle [1].

Building on the sequential composition idea [2], a recent work has shown how to compose local controllers in ways that satisfy temporal specifications given in temporal logic [9] rather than final goals. In [10]–[12], powerful model checking tools were used to find the sequence in which the controllers must be activated for the system to satisfy a high-level temporal behavior. Although these approaches can capture many interesting behaviors, their fundamental disadvantage is that they are open-loop solutions. They find sequences of policies to be invoked rather than an automaton and therefore cannot satisfy reactive behaviors that depend on the local state of the environment, as determined at run time, or handle uncertain initial conditions.

This work builds on the approach taken in [13], which is based on an automaton synthesis algorithm introduced in [14]. By creating automata rather than specifying sequences of policies, the robot can satisfy behaviors that depend on local information gathered during run time.

Local Continuous Feedback Control Policies

Local continuous feedback control policies form the foundation of the control framework; the policies are designed to provide guaranteed performance over a limited domain. Using continuous feedback provides robustness to noise, modeling uncertainty, and disturbances. This section presents the system model used in the control design, the formulation of the local policies, and the method of deployment.

System Modeling

Although this approach can be applied to different robot models, this article focuses on the control of a rear-wheel drive car-like vehicle with Ackermann steering. The vehicle, which is shown schematically in Figure 2, is sized based on a standard minivan.

The vehicle pose, g, is represented as $g = \{x, y, \theta\}$, where (x, y) is the location of the midpoint of the rear axle with respect to a global coordinate frame and θ is the orientation of the body with respect to the global x-axis. The angle of the steering wheel is $\phi \in \mathbf{I} = (-\phi_{\max}, \phi_{\max})$, a bounded interval.

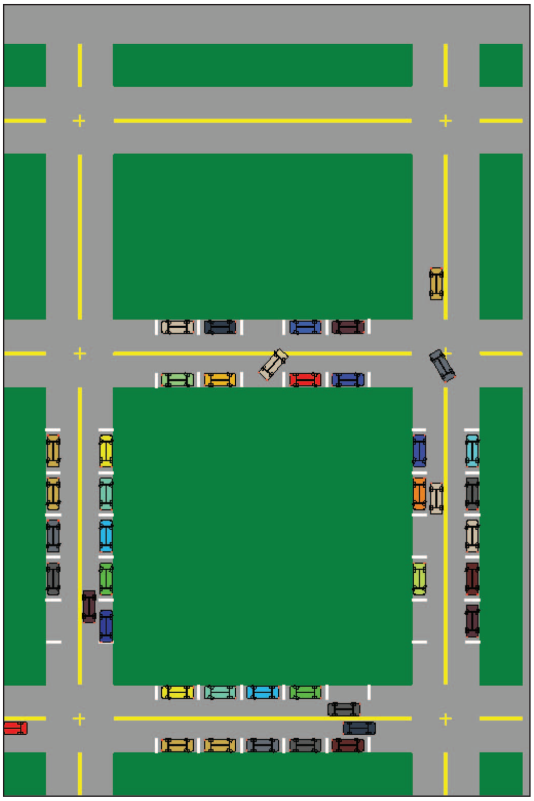


Figure 1. *The environment has 40 parking spaces arranged around the middle city block. For any vehicle, the high-level specification encodes either "drive around until you find a free parking space and then park" or "leave your parking space and exit the block."*

The nonholonomic constraints inherent in the rolling contacts uniquely specify the equations of motion via a nonlinear relationship between the input velocities and the body pose velocity. Let the system inputs be $u = \{v, \omega\} \in \mathcal{U}$, where $\mathcal{U}$ is a bounded subset of $\mathbb{R}^2$, v is the forward velocity, and ω is the rate of steering. The complete equations of motion are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{1}{L}\tan\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} A(g, \phi) \\ 0 \quad 1 \end{bmatrix} u. \tag{1}$$

More compactly, the body pose velocity is $\dot{g} = A(g, \phi)u$, where $A(g, \phi)$ encodes the nonholonomic constraints.

In addition to nonholonomic constraints, the system evolution is subject to configuration constraints. The body pose is restricted by the obstacles in the environment. The pose is further constrained by local conventions of the road, such as driving in the right lane. There is an absolute mechanical limitation of $\pm\phi_{\max}$. For safety and performance reasons, we allow further steering angle constraints at higher speeds. The system inputs are constrained based on speed limits in the environment and system capabilities.

Local Policy Development

The hybrid control framework uses local feedback control policies to guarantee behavior over a local domain. These local policies are then composed in a manner that allows reasoning on a discrete graph to determine the appropriate policy ordering that induces the desired global behavior. For the policies to be composable in the hybrid control framework, the individual policies must satisfy several requirements: 1) domains lie completely in the free configuration space of the system, 2) under influence of a given policy, the system trajectory must not depart the domain except via a specified goal set, 3) the system must reach the designated goal set in finite time, and 4) the policies must have efficient tests for domain inclusion given a known configuration [3], i.e., it is easy to check whether the vehicle is in the domain of a certain policy. This article focuses on one design approach that satisfies these properties.

The navigation tasks are defined by vehicle poses that must be reached or avoided; therefore, this article defines cells in the vehicle pose space. Each cell has a designated region of pose space that serves as the goal set. Over each cell, we define a scalar field that specifies the desired steering angle, ϕ_{des}, such that steering as specified induces motion that leads to the goal set. Taking the steering angle derivative with respect to body pose gives a reference steering vector field over the cell. This leads to a relatively simple constrained optimization problem over the bounded input space. The resulting policies are able to satisfy the four requirements given earlier.

The approach to defining the cell boundary and desired steering angle is based on a variable structure control approach [15]. The cells are parameterized by a local path segment in the workspace plane [Figure 3(a)]. The workspace path is lifted to a curve in body pose space by considering the path tangent vector orientation as the desired heading. One end of the path serves as the center of the goal set. This work uses line segments and circular arcs for the path segments. Other path shapes are possible at a cost of more complex derivative calculations [16].

To perform the control calculations, the body pose is transformed to a local coordinate frame assigned to the closest point on the path to current pose. The policy defines a boundary in the local frames along the path. Figure 3(b) shows the cell boundary defined by the local frame boundaries along the path; the interior of this tube defines the cell. The size of the tube can be specified subject to constraints induced by the path radius of curvature and the vehicle steering bounds. The cell can be tested for collision with an obstacle using the technique outlined in [3].

We define a surface in the local frame to serve as a sliding surface for purposes of defining a desired steering angle [15]. To generate a continuous steering command, the sliding surface is defined as a continuous function with a continuous bounded derivative; a blending zone is defined around the sliding surface. Outside the blending zone, the desired steering is set to a steering limit, ϕ_{lim}, where $|\phi_{\text{lim}}| \le \phi_{\max}$. The sign of ϕ_{lim} depends on the current direction of travel (forward or reverse) and whether the current body pose in local

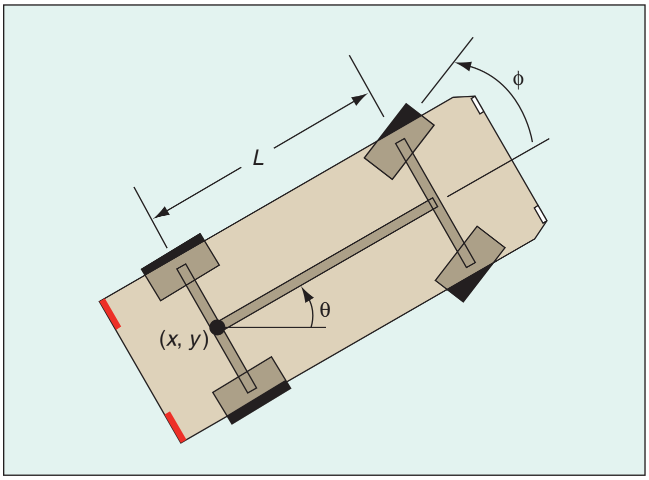
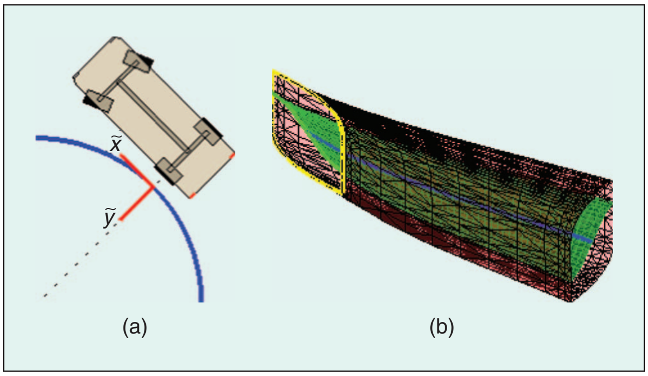


Figure 2. *Car-like system with Ackermann steering. The inputs are forward velocity and steering angle velocity.*



(a) (b)

Figure 3. *Control policy based on [15]: (a) workspace path with local frame defined and (b) the cell boundary forms a tube around the path in pose space. The sliding surface is shown in the cell interior.*

coordinates is above or below the sliding surface. (For policies that move the system in reverse, the positive or negative signs are swapped.) Inside the blending zone, let

$$\phi_{\text{des}} = \eta\phi_{\text{lim}} + (1-\eta)\phi_{\text{ref}}, \tag{2}$$

where $\eta \in [0, 1]$ is a continuous blending function based on distance from the sliding surface and ϕ_{ref} is the steering command that would cause the system to follow the sliding surface. Thus, (2) defines a mapping from the body pose space to the desired steering angle for any point in the cell. The sliding surface is designed such that steering according to ϕ_{des} will cause the system to move toward the sliding surface and then along the sliding surface toward the specified curve in the desired direction of travel. At the boundary of the cell, the desired steering must generate a velocity that is inward pointing, which constrains the size and shape of a valid cell.

For a closed-loop policy design, the system must steer fast enough so that the steering angle converges to the desired steering angle faster than the desired steering angle is changing. This induces an additional constraint on the input (velocity and rate of steering) space. Given this constraint, a simple constrained optimization is used to find a valid input. Each policy is verified to ensure that a valid input exists over its entire domain during specification.

The vehicle closed-loop dynamics over the cell induce a family of integral curves that converge to the curve specifying the policy. To guarantee that an integral curve never exits the cell during execution, we impose one additional constraint. Define the steering margin, ϕ_{margin}, as the magnitude of the angle between the desired steering along the cell boundary and the steering angle that would allow the system to depart the cell. During deployment, the policies must be specified with a positive steering margin. To use the control policy, we require that $|\phi_{\text{des}} - \phi| < \phi_{\text{margin}}$. Initially, if $|\phi_{\text{des}} - \phi| \geq \phi_{\text{margin}}$, the system halts and steers toward the desired steering angle until $|\phi_{\text{des}} - \phi| \leq \phi_{\text{margin}}$. Invoking the policies this way guarantees that the system never departs the cell, except via the designated goal set; i.e., the policy is conditionally positive invariant [3]. As the vehicle never stops once the steering policy becomes active, the system reaches the designated goal in finite time.

Local Policy Deployment

To set up the basic scenario, we define the urban parking environment, shown in Figure 1, based on a green practices guideline for narrower streets [18]. The regularity of the environment allows an automated approach to policy deployment.

First, we specify a cache of local policies using the generic policy described earlier. The cache uses a total of 16 policies: one policy for normal traffic flow, four policies associated with left and right turns at the intersections, six policies associated with parking, and five associated with leaving a parking space. Ten of the policies move the vehicle forward, and six move the vehicle in reverse. Each policy in the cache is defined relative to a common reference point. At this point, the specification of the free parameters for each policy in the cache is a trial-and-error process that requires knowledge of the environment, the desired behaviors, and some engineering intuition. During specification of the policies, we verify that the convergence and invariance properties are satisfied and that the policies are free of obstacle collision based on the road layout.

Policies from the cache are then instantiated at grid points defined throughout the roadways. This is done offline based on knowledge of the local roadways. The instantiation process selects a subset of the policies in the cache based on the grid point location. Given the cache and specified grid points, the instantiation process is automated. Normally, the test for obstacle collision would be conducted as the policies are instantiated, but the regularity of the roadway renders this unnecessary. For intersections, the four turning policies are deployed for each travel direction along with the basic traffic flow policy. For the straight traffic lanes, the grid points lie in the middle of the traffic lanes aligned with the front of the parking space markers; the orientation is defined by the traffic flow. The basic traffic flow policy is always deployed at these grid points.

If a potential parking space is adjacent to the grid point, a special parking policy is instantiated. Although considered a single policy by the automaton synthesis, each parking policy is actually composed of several policies from the cache. The parking component policies are only instantiated when the parking behavior is invoked for the first time by the global parking automaton (see “Automation Synthesis” section). Figure 4 shows an example parking maneuver induced by the composition of the local feedback control policies. The same applies for special leaving policies that are a composition of several policies causing the vehicle to leave a parking space. For the region defined in Figure 1, there are initially a total of 306 policies, including 40 parking policies associated with the 40 possible parking spaces. Five policies are instantiated for each parking behavior invoked, and five policies instantiated for leaving a parking space. These are added on an as-needed basis; the appropriate nodes are appended to the automaton.

As part of the instantiation process, we test for goal set inclusion pairwise between policies. The policies in the cache are specially defined so that policies instantiated at neighboring grid points prepare one another appropriately. If the goal set of one policy is contained in the domain of a second, the first is said to prepare the second [2]. This pairwise test defines the

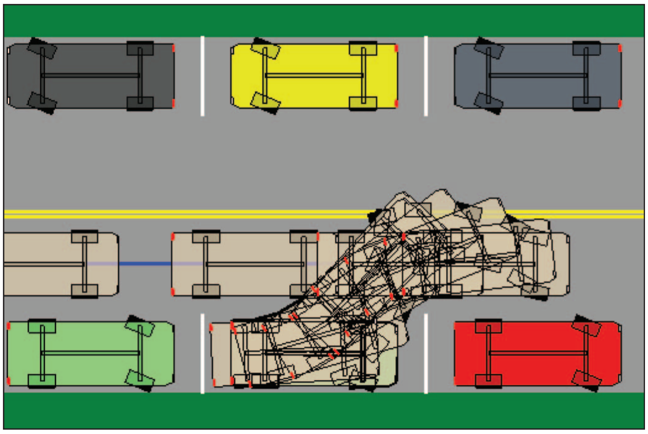


Figure 4. *Parking behavior induced by the composition of local policies. The feedback control policies guarantee the safety of the maneuver.*

prepares graph, which encodes the discrete transition relation between policies. This graph forms the foundation of the automaton synthesis approach described in the next section. The policy specification, instantiation, and prepares testing is done offline prior to the system generating the automaton.

Automaton Synthesis

This section describes the method used to create the automata that governs the local policies' switching strategy. These automata are guaranteed to produce paths that satisfy a given specification in different dynamic environments, if such paths exist.

Synthesis Algorithm

We are given a set of binary inputs (e.g., a binary input that is true when the closest parking spot is empty and false otherwise, a local hazard detected), a set of binary outputs (e.g., whether or not to activate policy Φ_i, signal left (right) turn, parking here, leaving adjacent spot), and a desired relationship between inputs and outputs (e.g., "if you sense an empty parking space, invoke a parking policy"). The realization or synthesis problem consists of constructing a system that controls the outputs such that all of its behaviors satisfy the given relationship or determine that such a system does not exist.

The relationship is given as an LTL with a specific structure [9], and the system is built using the algorithm introduced in [14]. There, the synthesis process is viewed as a game played between the system, i.e., the robot, which controls the outputs, and the environment, which controls the inputs. The two players have initial conditions and a transition relation defining the moves they can make. The winning condition for the game is given as a Generalized Reactivity (1) (a fragment of LTL) formula σ. The way the game is played is that at each step, first the environment makes a transition according to its transition relation, and then the system makes its own transition (constraints on the system transitions include obeying the prepares graph). If the system can satisfy σ no matter what the environment does, we say that the system is winning and we can extract an automaton. However, if the environment can falsify σ, we say that the environment is winning and the desired behavior is unrealizable, which means that there is no automaton that can satisfy the requirements.

The synthesis algorithm [14] takes the initial conditions, transition relations, and winning condition, and then checks whether the specification is realizable. If it is, the algorithm extracts a possible, but not necessarily unique, automaton that implements a strategy that the system should follow to satisfy the desired behavior.

Writing Logic Formulas

Informally, LTL formulas are built using a set of boolean propositions, the regular boolean connectives not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\Rightarrow$), if and only if ($\Leftrightarrow$), and temporal connectives. The temporal connectives include next ($\bigcirc$), always ($\Box$) and eventually ($\Diamond$). These formulas are interpreted over infinite sequences of truth assignments to the propositions. For example, the formula $\bigcirc(p)$ is true if in the next position p is true. The formula $\Box(q)$ is true if q is true in every position in the sequence. The formula $\Box\Diamond(r)$ is true if always eventually r is true, i.e., if r is true infinitely often.

The input to the algorithm is an LTL formula

$$\varphi = (\varphi_e \Rightarrow \varphi_s).$$

φ_e is an assumption about the inputs and thus about the behavior of the environment, and φ_s represents the desired behavior of the system. More specifically,

$$\varphi_e = \varphi_i^e \wedge \varphi_t^e \wedge \varphi_g^e; \quad \varphi_s = \varphi_i^s \wedge \varphi_t^s \wedge \varphi_g^s.$$

φ_i^e and φ_i^s describe the initial condition of the environment and the system. φ_t^e represents the assumptions on the environment by constraining the next possible input values based on the current input and output values. φ_t^s constrains the moves the system can make, and φ_g^e and φ_g^s represent the assumed goals of the environment and the desired goals of the system, respectively. For a detailed description of these formulas, see [13].

Translating this formula to a game, the initial condition is $\varphi_i^e \wedge \varphi_i^s$, the transition relations for the players are φ_t^e and φ_t^s, and the winning condition is $\sigma = (\varphi_g^e \Rightarrow \varphi_g^s)$. Note that there are two ways for the system to win. It wins if either φ_g^s is satisfied, i.e., the system reaches its goals, or φ_g^e is falsified. The latter case implies that if the environment does not satisfy its goals (either a faulty environment or the system interfered), then a correct behavior of the system is no longer guaranteed. Furthermore, if during an execution of the automaton, the environment violates its own transition relation, the automaton is no longer valid.

In the following sections, we explain in detail how to encode the specifications. "Adhering to Traffic Laws" section first describes an LTL formula that encodes appropriate behavior in traffic, i.e., the reaction to hazardous conditions and the activation of the turn signals. This LTL formula captures the multirobot aspect of the behavior. "Parking" and "Leaving" sections then add the more specialized behavior for the parking and leaving tasks, respectively.

Adhering to Traffic Laws

Socially acceptable driving behavior includes stopping at stop lights, driving in the designated lane, keeping a safe distance from vehicles ahead, and using the left and right turn signals. To encode such behavior, we define one input, hazard, which becomes true whenever the car must stop. Such an input may be the result of a proximity sensor in the case of keeping a safe distance from another vehicle or of a vision system recognizing a red light at the intersection or another vehicle signaling that it is about to make a turn. The hazard is also used to cause a vehicle intending to park to wait on a vehicle that is ready to leave an occupied parking space. Although in some cases, the more natural reaction to such conditions is to slow down rather than stop, here we take the more conservative approach for simplicity. The local feedback control policies serve as outputs. Additional output propositions are signalL and signalR, which indicate whether the left (right) turn signal should be activated, and the proposition "stop," which indicates whether the vehicle should stop. These outputs are detectable by other robots. The formula encoding this behavior is given in the following list.

1) *Assumptions on the environment:* Initially, there is no need to stop; therefore, $\varphi_i^e = \neg\text{hazard}$. We do not impose any further restrictions on the behavior of the hazard input; thus, it can become true or false at any time. To keep the structure of the formula, we encode both φ_t^e and φ_g^e as the trivial formula true

$$\varphi_t^e \wedge \varphi_g^e = \square\,(\text{TRUE}) \wedge \square\,\Diamond(\text{TRUE}),$$

which means that these formulas are always satisfied.

2) *Constraints on the behavior of the vehicle (system):* Initially, the vehicle must be in the domain of an initial policy, no turn signal is on (we assume the vehicle starts by driving straight

$$\varphi_i^s = \vee_{i\in\text{InitialPolicy}}\Phi_i \wedge \neg\,\text{SignalL} \wedge \neg\,\text{SignalR} \wedge \neg\,\text{stop},$$

which will be changed in the "Leaving" section), and the vehicle is not required to stop. The vehicle can only transition from one policy to the next based on the prepares graph from the "Local Policy Deployment" section (first line of φ_t^s below). It must turn the left turn signal only if it is turning left and the same for the right turn signal (second and third line). It must stop if and only if the hazard signal is true (last line).

$$\varphi_t^s = \begin{cases} \wedge_i\square\,(\Phi_i \Rightarrow (\bigcirc\,\Phi_i\ \vee_{j\in\text{SuccessorsOfPolicy}_i}\bigcirc\,\Phi_j)) \\ \wedge\square\,((\vee_{j\in\text{LeftTurnPolicies}}\bigcirc\,\Phi_j) \Leftrightarrow \bigcirc\,\text{signalL}) \\ \wedge\square\,((\vee_{j\in\text{RightTurnPolicies}}\bigcirc\,\Phi_j) \Leftrightarrow \bigcirc\,\text{signalR}) \\ \wedge\square\,(\bigcirc\,\text{hazard} \Leftrightarrow \bigcirc\,\text{stop}). \end{cases}$$

Finally, since we are only concerned with obeying traffic laws and we do not require the vehicle to go anywhere, we simply write $\varphi_g^s = \square\,\Diamond(\text{TRUE})$.

Parking

In this scenario, a vehicle is searching for an empty parking space and parks once it finds one. Starting from the formula in the "Adhering to Traffic Laws" section, we define another input, park, which becomes true when an empty parking space is found.

1) *Assumptions on the environment:* We add these subformulas to φ_e of the "Adhering to Traffic Laws" section: Initially there is no parking near the vehicle; therefore, we add $\neg\,\text{park}$ to φ_i^e. We can only determine whether there is a free parking space if we are in a policy next to it, i.e., park cannot become true if the vehicle is not next to a parking space or in one (first subformula). Also, for implementation reasons, we assume that the input park remains true after parking (second subformula). These subformulas are added to φ_t^e

$$\begin{cases} \square\,([(\neg(\vee_{i\in\text{ParkPolicy}}\Phi_i)) \wedge (\neg(\vee_{j\in\text{PreparesParkPolicy}}\Phi_j))] \\ \quad\Rightarrow \neg\bigcirc\,\text{park}) \\ \wedge \\ \square\,((\text{park} \wedge (\vee_{i\in\text{ParkPolicy}}\Phi_i)) \Rightarrow \bigcirc\,\text{park}). \end{cases}$$

We have no assumptions on the infinite behavior of the environment (we do not assume that there is an empty parking spot); therefore, the goal component remains set to true.

2) *Constraints on the behavior of the vehicle (system):* Here, we add the parking requirement to φ_t^s, which state that the vehicle cannot park if there is no parking space available, indicated by the park input (first line). If there is an empty parking space, it must park (second line).

$$\begin{cases} \wedge_{i\in\text{ParkPolicy}}\square\,(\neg\bigcirc\,\text{park} \Rightarrow \neg\bigcirc\,\Phi_i) \\ \wedge\square\,(\bigcirc\,\text{park} \Rightarrow (\vee_{i\in\text{ParkPolicy}}\bigcirc\,\Phi_i)). \end{cases}$$

Finally, we replace φ_g^s by adding a list of policies the vehicle must visit infinitely often if it has not parked yet. These policies define the area in which the vehicle will look for an available parking space.

$$\varphi_g^s = \wedge_{i\in\text{VisitPolicy}}\,\square\,\Diamond(\Phi_i \vee \text{park} \vee \text{stop}).$$

Note that the goal condition is true if either the vehicle visits these policies infinitely often (when there is no parking space available) or it has parked or it has stopped (because of an accident ahead of it or a broken stop light).

Leaving

In this scenario, a vehicle is leaving its parking space and exiting the block via some specified exit. As before, starting from the formula in the "Adhering to Traffic Laws" section, we define as additional inputs Exit Φ_i for $i \in$ ExitPolicies. These are inputs that are constant and define which exit the vehicle should use (the proposition that is true), thus two vehicle leaving may use the same generated automaton with different inputs.

1) *Assumptions on the environment:* We add these subformulas to φ_e. Initially only one Exit Φ_i is true. This is added to φ_i^e

$$\vee_{i\in\text{ExitPolicies}}\,(\text{Exit}\,\Phi_i \wedge_{j\in\text{ExitPolicies},\,j\neq i} \neg\,\text{Exit}\,\Phi_j).$$

We require the input to be constant, which means that they cannot change. Therefore, we add to φ_t^e

$$\vee_{i\in\text{ExitPolicies}}\,(\text{Exit}\,\Phi_i \Leftrightarrow \bigcirc\text{Exit}\,\Phi_i).$$

We have no assumptions on the infinite behavior of the environment; therefore, the goal component remains set to true.

2) *Constraints on the behavior of the vehicle (system):* Initially, the car is leaving a parking space, hence it must turn on the left turn signal. We modify φ_i^s to be
$\varphi_i^s = \vee_{i\in\text{InitialPolicy}}\Phi_i \wedge \text{SignalL} \wedge \neg\text{SignalR} \wedge \neg\text{stop}$.

We do not add any further subformulas to φ_t^s of the "Adhering to Traffic Laws" section. As for φ_g^s, we replace it with the requirement that the vehicle must go to the designated exit policy if it has not stopped.

$$\varphi_g^s = \wedge_{i\in\text{ExitPolicies}}\square\,\Diamond((\Phi_i \Leftrightarrow \text{Exit}\Phi_i) \vee \text{stop}).$$

Continuous Execution of Discrete Automata

The synthesis algorithm generates an automaton that governs the execution of the local policies; however, the continuous evolution of the system induced by the local policies governs the state transitions within the automaton. In this section, we discuss the implementation of the policy switching strategy.

Execution

A continuous execution of the synthesized automaton begins in an initial state q_0 that is determined by linearly searching the automaton for a valid state according to the initial body pose of the vehicle. From state q_i at each time step, the values of the binary inputs are evaluated. (We assume the time step is short compared with the time constant of the closed-loop dynamics.) On the basis of these inputs, all possible successor states are determined. If the vehicle is in the domain of policy Φ_l, which is active in a successor state q_j , the transition is made. Otherwise, if the vehicle is still in the domain of Φ_k, which is active in state q_i, the execution remains in this state. The only case in which the vehicle is not in the domain of Φ_k, or in any successor Φ_l, is if the environment behaved badly. It either violated its assumptions, thus rendering the automaton invalid, or it caused the vehicle to violate the prepares graph (e.g., a truck running into the vehicle). In the event that a valid transition does not exist, the automaton executive can raise an error flag, thereby halting the vehicle and requesting a new plan. This continuous execution is equivalent to the discrete execution of the automaton [10], [12].

Guarantees of Correctness

We have several guarantees of correctness for our system, starting from the high-level specifications and going down to the low-level controls. First, given the high-level specification encoded as an LTL formula, the synthesis algorithm reports whether the specification is realizable or not. If an inconsistent specification is given, such as, "always keep moving and if you see a stop light stop," the algorithm will return that there is no such system. Furthermore, if a specification requires an infeasible move in the prepares graph, such as "always avoid the left north or south road and eventually loop around all the parking spaces," the algorithm will report that such a system does not exist.

Second, given a realizable specification, the algorithm is guaranteed to produce an automaton such that all its executions satisfy the desired behavior if the environment behaves as assumed. The construction of the automaton is done using φ_t^e, which encodes admissible environment behaviors; if the environment violates these assumptions, the automaton is no longer correct. The automaton state transitions are guaranteed to obey the prepares graph by the low-level control policy deployment unless subject to a catastrophic disturbance (e.g., an out of control truck). Modulo a disconnect between φ_t^e and the environment, or a catastrophic disturbance to the continuous dynamics, our approach leads to a correct continuous execution of the automaton that satisfies the original high-level desired behavior.

Sensors, or more specifically, the binary inputs used by the automaton, are of great importance in this framework. First, as mentioned earlier, they must satisfy the assumptions made about them in the LTL formula; otherwise, the automaton will not be correct. Second, even if they do satisfy these assumptions, they may still cause correct yet unintended behavior. For example, if the proximity sensor set the hazard input to true whenever another vehicle was in a certain radius, even if that vehicle was behind in a forward driving lane, both vehicles may get deadlocked, i.e., both would stop forever. Although this behavior satisfies the original specification, it does not follow the spirit of finding a parking space. (This is a classical problem in concurrent systems. There, fairness assumptions are imposed on the inputs to ensure that the system will not deadlock.) On the other hand, both cars stopping might be a desired behavior when an accident occurred; therefore, we would not want to forbid it in the specifications. Such unintended behavior would not be present in a centralized approach where the controller has full knowledge and not just local information as is the case here. However, with careful design of the inputs, such behaviors can be avoided.

Results

The approach is verified in a simulation executed using MATLAB. First, the workspace is laid out, and a cache of policies is specified. Second, the policies are automatically instantiated in the configuration space of the vehicle, and the prepares graph is defined. Next, the LTL formulas are written. Each LTL formula is then given to the automatic synthesis algorithm implemented by Piterman et al. [14] on top of the temporal logic verifier system [17]. At this point, the resulting automaton is used to govern the execution of the local policies, based on the local behavior of the environment. The vehicles are able to react in real time to disturbances via the local continuous feedback and environmental changes sensed locally due to the automaton.

In such an execution, we must simulate the sensors that govern the behavior of the park and hazard inputs. The park input is set to true whenever there is a free parking space near by. The hazard input that enables the traffic law abiding behavior and thus the multirobot task should be set to true whenever the car must stop. Here, we simulate a proximity sensor with added logic that sets hazard to true whenever the car is too close to a car ahead of it (keeping safe distance), whenever a car ahead is backing up to park (being polite), whenever the car is leaving a parking space and another car passes by, and whenever another car is leaving a parking space which the car will park in next. We also simulate a vision system that detects whether the stoplight is red.

In the following example, the workspace is the one shown in Figure 1, with the 306 policies instantiated as described in the "Local Policy Deployment" section. In the parking LTL formula, the visit policies correspond to the eight lanes around the parking spaces (four going clockwise and four going counter clockwise), and the initial policies correspond to the ten entry points to the workspace. Likewise, in the leaving automaton, the 40 parking spaces are the possible initial policies, and the ten exit points are the possible goals. Initially, 35 of the 40 parking spaces were randomly specified as occupied.

In this simulation, eight cars enter the block at different times and from different entry points, looking for a parking space. During the execution, an additional three cars leave their parking spaces and exit the workspace. Figure 5 shows a general snapshot of the simulation. At this point in time, seven cars are moving in the workspace. Cars that are marked with red ellipses are the cars whose hazard input is true; therefore, they have stopped. All stopped cars in this figure are obeying stoplights.

Figure 6 shows several close-up looks at different traffic behaviors encountered during the simulation. In Figure 6(a), the blue car that is leaving the parking space has stopped, indicated by a red ellipse, to let the brown car drive by. This hazard was invoked based on a proximity sensor. In Figure 6(b), red car is parking while the blue car waits for it to finish before passing. In Figure 6(c), the orange car is stopping to allow the gray car to complete a left turn. The white car on the left is leaving the parking space that later will be occupied by the brown car. Figure 6(d) shows two cars stopping before a stoplight. While the white car stopped based on the stoplight, the black car behind stopped based on the proximity to the car ahead of it. Figure 6(e) and (f) is the two snapshots of two cars parking simultaneously in opposite lanes. The car that started the parking maneuver later (bottom lane) pauses to allow the other car to park safely.

The video of this simulation can be viewed at [19].

Conclusions and Future Work

In this article, we have demonstrated, through the parking and leaving example, how high-level specifications containing multiple temporally dependent goals can be given to a team of realistic robots, which in turn automatically satisfy them. By switching between low-level feedback control policies and moving in a well-behaved environment, the correctness of each robot's behavior is guaranteed by the automaton. The system satisfies the high-level specification without needing to plan the low-level motions in configuration space.

Furthermore, given a collection of local feedback control policies, the approach is fully automatic and correct by construction.

Sensor inputs play a crucial role in this framework, as explained in the "Continuous Execution of Discrete Automata" section. A hazard input becoming true at the wrong time may lead to deadlock. Deciding when and how long to stop is a hard problem even for humans, as sometimes demonstrated at four-way stops, let alone robots. Therefore, in the future, we wish to explore how such inputs should be designed, implemented, and verified.

We plan to extend this work in several other directions. At the low level, we wish to consider more detailed dynamics. At the high level, we intend to address more complex robot coordination and tasks. Our research also focuses on accessible specification languages such as some form of natural language. Furthermore, we plan to run several experiments with real systems that demonstrate the work described in this article.

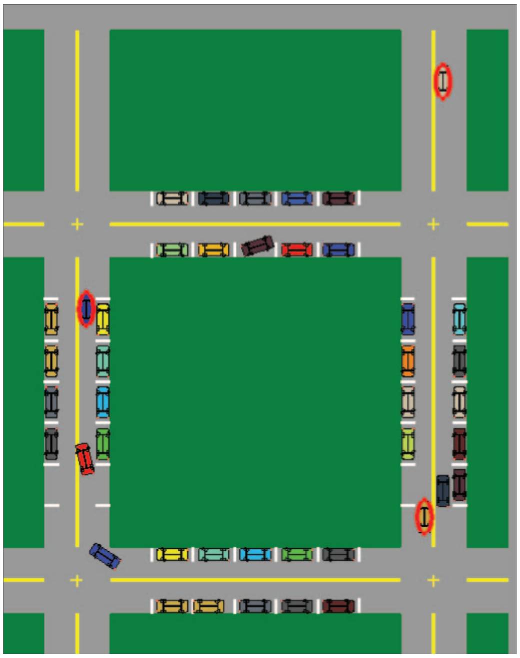


Figure 5. *A snapshot of the simulation. Cars surrounded by red ellipses are cars that are stopping because of the hazard input, in this case based on a stoplight.*

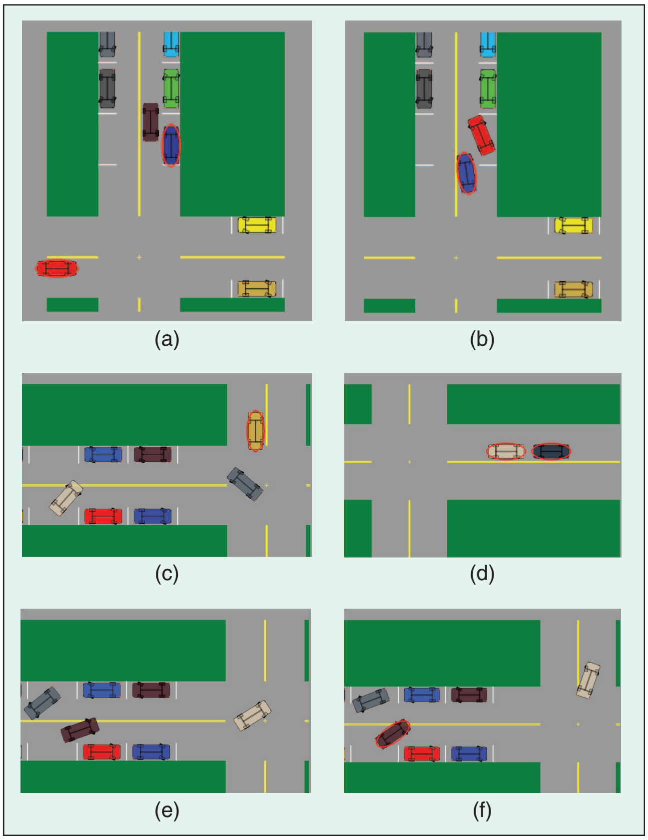


Figure 6. *Close-up looks at different behaviors seen throughout the simulation. (a) Blue car leaving. (b) Red car parking. (c) Yielding to turn in progress. (d) Two cars at stoplight. (e) Two cars parking. (f) Two cars parking.*

Acknowledgment

This work was partially supported by Army Research Office MURI DAAD 19-02-01-0383.

Keywords

Multirobot, hybrid control, motion planning.

References

[1] D. C. Conner, H. Kress-Gazit, H. Choset, A. A. Rizzi, and G. J. Pappas, "Valet parking without a valet," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, CA, Oct. 2007, pp. 572–577.

[2] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 534–555, 1999.

[3] D. C. Conner, H. Choset, and A. A. Rizzi, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies," in *Proc. Robotics: Science and Systems II*, Philadelphia, PA, 2006, pp. 57–64.

[4] A. A. Rizzi, "Hybrid control as a method for robot motion programming," in *IEEE Int. Conf. Robotics and Automation*, May 1998, vol. 1, pp. 832–837.

[5] D. C. Conner, A. A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, Oct. 2003, pp. 3546–3551.

[6] L. Yang and S. M. Lavalle, "The sampling-based neighborhood graph: An approach to computing and executing feedback motion strategies," *IEEE Trans. Robot. Automat.*, vol. 20, pp. 419–432, June 2004.

[7] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot planning and control in polygonal environments," *IEEE Trans. Robot.*, vol. 21, pp. 864–874, Oct. 2005.

[8] S. R. Lindemann, I. I. Hussein, and S. M. LaValle, "Realtime feedback control for nonholonomic mobile robots with obstacles," in *IEEE Conf. Decision and Control*, San Diego, CA, 2006, pp. 2406–2411.

[9] E. A. Emerson, "Temporal and modal logic," *Handbook of Theoretical Computer Science (vol. B): Formal Models and Semantics*. Cambridge, MA: MIT Press, 1990, pp. 995–1072.

[10] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for mobile robots," in *IEEE Int. Conf. Robotics and Automation*, 2005, pp. 2020–2025.

[11] G. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: A temporal logic approach," in *IEEE Conf. Decision and Control*, Seville, Spain, 2005, pp. 4885–4890.

[12] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from LTL specifications," in *9th Int. Workshop on Hybrid Systems: Computation and Control*, Santa Barbara, CA, 2006.

[13] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Where's waldo? Sensor-based temporal logic motion planning" in *IEEE Int. Conf. Robotics and Automation*, 2007, pp. 3116–3121.

[14] N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of Reactive(1) Designs," in *VMCAI*. Springer-Verlag, New York, Jan. 2006, pp. 364–380.

[15] A. Balluchi, A. Bicchi, A. Balestrino, and G. Casalino, "Path tracking control for Dubin's car," in *IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, 1996, pp. 3123–3128.

[16] W. L. Nelson, "Continuous curvature paths for autonomous vehicles," in *IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 1989, vol. 3, pp. 1260–1264.

[17] A. Pnueli and E. Shahar. (1996, May 5). The TLV system and its applications. [Online]. Available: http://www.cs.nyu.edu/acsys/tlv/

[18] Green home building guidelines. http://www.nahbrc.org/greenguidelines/userguide_site_innovative.html. Accessed Jan. 2008.

[19] Multiple cars in traffic. http://www.grasp.upenn.edu/~hadaskg/MultipleCarsInTraffic.avi

Hadas Kress-Gazit graduated with B.Sc. degree in electrical engineering from the Technion in 2002. During her undergraduate studies, she worked as a hardware verification engineer for IBM. After graduating and prior to entering graduate school, she worked as an engineer for RAFAEL. In 2005, she received the M.S. degree in electrical engineering from the University of Pennsylvania, where she is currently pursuing a Ph.D. Her research focuses on generating robot controllers that satisfy high-level tasks using tools from the formal methods, hybrid systems, and computational linguistics communities. She is a Student Member of the IEEE.

David C. Conner received the Ph.D. degree in robotics from Carnegie Mellon University in 2007. He received an M.S. degree in robotics in 2004. His research interests include applications of differential geometry, hybrid systems, and control theory for mobile robot navigation and control. He graduated with B.S. and M.S. degrees in mechanical engineering from Virginia Tech (VPI & SU) in 1991 and 2000, respectively. He is currently a research scientist with TORC Technologies in Blacksburg, Virginia. He is a Student Member of the IEEE.

Howie Choset received the Ph.D. degree in mechanical engineering from the California Institute of Technology in 1996. He is currently an associate professor at Robotics Institute at Carnegie Mellon University, where he conducts research in motion planning and design of serpentine mechanisms, coverage path planning for demining and painting, mobile robot sensor-based exploration of unknown spaces, and education with robotics. He is a Member of the IEEE.

Alfred A. Rizzi received the Sc.B. from MIT in 1986 and the M.S. and Ph.D. degrees from Yale University in 1990 and 1994, respectively, all in electrical engineering. Prior to entering graduate school, he served as a design engineer with the Northrop Corporation. He is currently a lead robotics scientist at Boston Dynamics. His research focus includes a combination of topics related to dynamically capable robotic systems and industrial and commercial application of such systems. He is a Member of the IEEE and a member of the editorial board for the *International Journal of Robotics Research*.

George J. Pappas received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1998. He is currently a professor in the Department of Electrical and Systems Engineering and the Deputy Dean of the School of Engineering and Applied Science. His research focuses on the areas of hybrid and embedded systems, hierarchical control systems, distributed control systems, nonlinear control systems, and geometric control theory, with applications to robotics, unmanned aerial vehicles, and biomolecular networks. He is a Senior Member of the IEEE.

Address for Correspondence: Hadas Kress-Gazit, GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 USA. E-mail: hadaskg@grasp.upenn.edu.

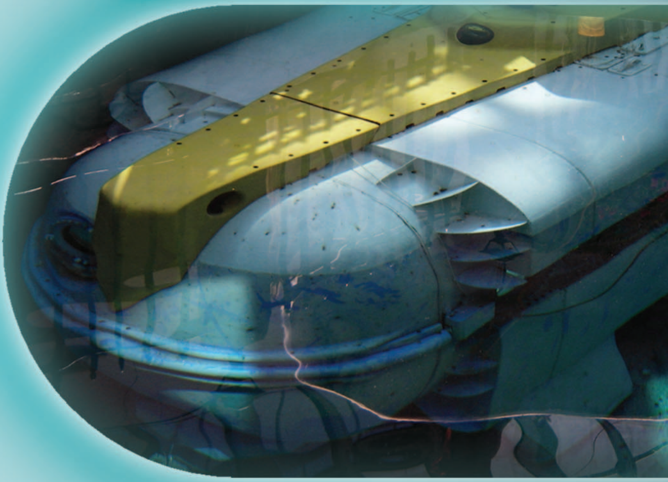
UAV Task Assignment

An Experimental Demonstration with Integrated Health Monitoring

**BY BRETT BETHKE, MARIO VALENTI,
AND JONATHAN P. HOW**

© DREAMSTIME

© DREAMSTIME



© ISTOCKPHOTO

Mobile Multirobot Systems

Unmanned aerial vehicles (UAVs) are becoming vital warfare and homeland security platforms because they have the potential to significantly reduce cost and risk to human life while amplifying warfighter and first-responder capabilities. To date, these vehicles have been operated in real missions with some success, but there remain challenging barriers to achieving the future vision of multiple UAVs operating cooperatively with other manned and unmanned vehicles in national airspace and beyond [1]. Among these is the problem of developing efficient and effective algorithms for simultaneously controlling and coordinating the actions of multiple autonomous vehicles in a dynamic environment. A particular concern is that off-nominal conditions or degraded components could reduce the capabilities of these UAVs to accomplish the mission objectives.

This article builds on the very active area of planning and control for autonomous multiagent systems (see [2] and [3] and the references therein). In principle, some of the issues raised in this problem are similar to questions arising in manufacturing systems [4], [5] and air transportation [6]–[8]. In addition, similar problems have been investigated under the Defense Advanced Research Projects Agency sponsored mixed initiative control of teams of autonomous agents [9]–[11]. While these efforts have made significant progress in understanding how to handle some of the complexity inherent in multiagent problems, the research in this article considers issues related to how vehicle health (e.g., fuel management and vehicle failures) affects the real-time mission planning (e.g., the task assignment). This work represents a step toward enabling robust decision making for distributed autonomous UAVs by improving the team's operational reliability and capabilities through better system self-awareness and adaptive mission planning.

The proposed methods for solving the overall multiagent problem typically involve formulating several smaller subproblems, each of which is simpler and, therefore, easier to solve [12]. One such solution architecture is shown in Figure 1, in which a number of components are combined to achieve the overall goals of the mission. The mission planning component is the highest level in the system. It keeps track of the mission objectives and generates tasks, which are discrete actions whose completion will aid the overall accomplishment of the mission. Examples of tasks include searching for, identifying, or tracking an object of interest. The mission planner provides the list of tasks to the task assignment component, which decides which of the available vehicles should perform each task based on the information about the tasks and the capabilities of the vehicles. Once the assignments have been made, they are sent to the trajectory designer, which plans feasible trajectories for each vehicle. The output of the trajectory designer is a sequence of waypoints for each vehicle to follow. These waypoints are sent to the vehicle controllers, which compute the actual controls needed to follow the waypoint plans.

Digital Object Identifier 10.1109/M-RA.2007.914931

Knowledge of the fuel state of the vehicle is important to be able to estimate the remaining useful flight time of the vehicle.

Inherent in each of the components in the architecture is a set of interconnected models used to predict future system behavior. For example, the controller contains a model of the control input dynamics of the vehicle, while the task assignment component contains a model of the performance each vehicle can be expected to produce if assigned to a given task. In the most general sense, system actions are selected by searching for actions that lead to desirable, predicted outcomes as given by the system models. Clearly, the performance of the system, therefore, depends heavily on the accuracy of these models.

One strategy for improving the accuracy of the models is to include additional feedback loops that provide information that can be used to adjust the models in real time. The amount, type, and quality of feedback information that each component receives plays a large role in how effectively the system can deal with dynamically changing factors in the environment, mission objectives, and state of the vehicles. Intuitively, feedback is necessary wherever there is uncertainty in the system, so that the initial plan of action made by each of the components of the planner can be modified when changes occur. Uncertainty may be present at all levels of the planning architecture as a result of incomplete knowledge of many factors, such as actuator performance at the control level, dynamic constraints at the trajectory design level, sensor health at the task assignment level, and long-term maintenance needs at the mission management level.

This article focuses on the health management problem at the task assignment level, developing a feedback mechanism for the performance model used by the task assignment algorithm. The assignment problem has been studied extensively [13]–[15]. However, most of the work done to date has used only a static vehicle performance model, making it difficult for these approaches to adapt to unexpected changes, such as sensor failures, during the course of the mission. The goal of this article is to develop a feedback loop that uses health state information to update the performance model in real time.

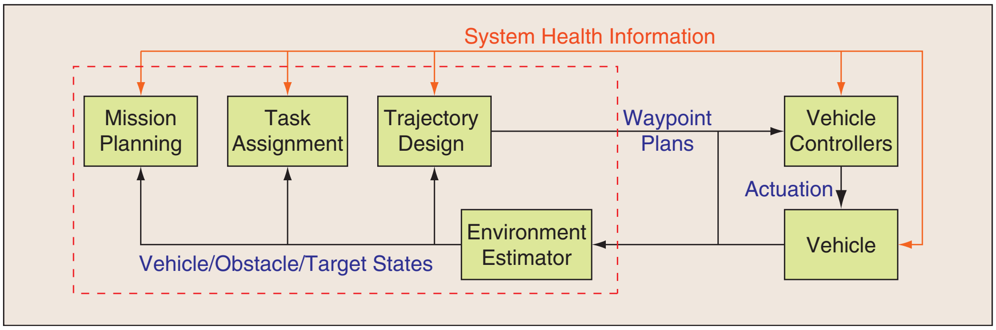


Figure 1. *Overall autonomous mission system architecture.*

By updating the performance model of an already existing algorithm, previous work on the task assignment problem can be leveraged and extended without requiring the modification of the existing algorithm. Its performance can be improved only by improving the quality of information available to make assignments.

Selection of Performance Model

The selection of the performance model incorporating health state information about the vehicle is clearly an important aspect of the feedback design. The particular details of the model depend on the mission problem in question and the vehicle hardware being used. However, there are a number of classes of general features that may be appropriate to be included in a performance model.

Vehicle Translational Dynamics

At the level of the task assignment problem, the vehicle dynamics are usually abstracted as being first order with a maximum speed $v_{\max}$. This abstraction allows the task assignment algorithm to capture important aspects of the vehicles' performance (in particular, how long they can be expected to take to reach a particular task), while being sufficiently simple to allow computational tractability. Recall that the trajectory planning and control levels below the task assignment level are responsible for carrying out those lower-level functions, allowing this simplification to be made. Note also that this is the model used in most of the previous work on task assignment.

Propulsion System State

The vehicle propulsion system may be abstracted as an entity that enables the vehicle to move at the maximum speed $v_{\max}$. Health feedback about the propulsion system may dynamically modify $v_{\max}$ to reflect the state of the propulsion system. For example, knowledge of a failing motor may cause $v_{\max}$ to decrease from its nominal value.

Fuel State

Knowledge of the fuel state of the vehicle is important to be able to estimate the remaining useful flight time of the vehicle. The performance model should include an estimator that performs the remaining flight time calculation based on the remaining fuel, average fuel consumption rates, and perhaps other environmental factors. Use of this information allows the task assignment algorithm to safely make assignments while ensuring that vehicles can return to the base before running out of fuel.

Sensor States

The current performance level of any sensing system onboard the vehicle should be included in the model if they are required to carry out tasks. For example, if an onboard camera is to be used for a surveillance task, the state of the camera (e.g., quality of the video signal) should be accounted for in the model.

Communication System State

Communication with other vehicles is often a requirement to enable vehicles to coordinate their actions with each other or relay messages to a distant ground station. Accounting for a vehicle's current estimated transmission and reception distances may allow the tasking system to avoid sending a vehicle to a location where it will be out of the communication range.

Modification of RHTA to Include Health Feedback: An Example

For the purposes of illustration, an example of incorporating a simple health feedback loop in the receding horizon task assignment (RHTA) algorithm is presented here. Briefly, the RHTA algorithm works as follows (for more details, see Alighanbari, 2004; Algorithm 2.3.1 [16]). Given the set of tasks W, distances between tasks $d(i, j)$, and vehicles V, RHTA enumerates all possible task sequences of specified length n_c. These sequences are called petals. The value of each petal is estimated as

$$S_{vp} = \sum \lambda^{T_{ip}} s_{wd},$$

where T_{ip} is the time at which task i is completed in petal p, s_{wd} is the task value, and λ is a time discount factor. Given the values of all the petals S_{vp}, RHTA solves the following optimization problem to select the optimal petal for each UAV:

$$\begin{aligned} \max\ J &= \sum_{v=1}^{N_v} \sum_{p=1}^{N_{vp}} S_{vp} x_{vp} \\ \text{subject to} \quad & \sum_{v=1}^{N_v} \sum_{p=1}^{N_{vp}} A_{vpi} x_{vp} \leq 1, \quad x_{vp} \in \{0, 1\} \\ & \sum_{p=1}^{N_{vp}} x_{vp} = 1, \quad \forall\, v \in \{1, \ldots, N_v\}. \end{aligned}$$

Here, x_{vp} is a binary variable that is equal to 1 if the pth petal is selected and 0 if not, and A_{vpi} equals 1 if task i is visited by vehicle v in petal p and 0 otherwise.

In the example, health state information is represented by adding a fuel state to the vehicle model. In this case, the fuel model is straightforward.

- The vehicle's fuel level f_i decreases at a constant rate k_{fuel} anytime the vehicle is flying.
- If f_i reaches zero before the vehicle refuels, the vehicle crashes and is lost.
- In addition, the occurrence of failures is modeled as a Poisson process with time intensity ρ_f; when a failure occurs, the rate of fuel burn increases to $k_{\text{fuel,failure}} > k_{\text{fuel}}$. Thus, this failure mode increases the rate at which fuel is burned (and, thus, decreases the time a vehicle can complete tasks).

Due to the inclusion of randomly occurring failures, the fuel model is able to capture some of the uncertainty in the health state of the vehicle. If a failure occurs, the optimal task assignment may change due to the fact that the failed vehicle may no longer be able to service its assigned task. When this happens, the task assignment algorithm must be able to calculate the new optimal solution, subject to the new constraint imposed by the failure.

Unmanned aerial vehicles (UAVs) are becoming vital warfare and homeland security platforms.

To handle these types of scenarios, the RHTA algorithm was extended to include the fuel state in the vehicle model. This was accomplished by including an estimate of each vehicle's operational radius, which is defined as $r_i \equiv v_{\max}\ (f_i / k_{\text{fuel}})$. The quantity r_i represents the maximum distance a vehicle can fly given its current fuel state, before running out of fuel. This information can be used to effectively prune the list of petals that RHTA considers to ensure that the vehicle can always safely return to the base before its fuel is exhausted. Specifically, the following constraint was added to the RHTA optimization problem:

$$L_i + d(w_{n_c}, x_{\text{base}}) \leq r_i.$$

Here, $d(w_{n_c}, x_{\text{base}})$ represents the normal Euclidean distance between the last waypoint in the petal and the base, and

$$L_i = d(v, w_1) + \sum_{j=2}^{n_c} d(w_{j-1}, w_j)$$

is the total length of the petal. The constraint effective rejects a petal if the length of the petal plus the distance from the terminal waypoint w_{n_c} to base is greater than the current operational radius of the vehicle. This ensures that the vehicle visits only waypoints that allow it to return safely to the base.

With this extension, RHTA will assign a vehicle to return to the base when every possible permutation of waypoints is rejected by the pruning criterion. Thus, this method provides a simple rule that determines when a vehicle should return to the base for refueling since it cannot safely service any of the remaining tasks. Note that this method can create some problems if the above rule is followed too strictly since too many vehicles may be sent back to the base unnecessarily (i.e., when they still have large operational radii) if there are few or no active tasks. This problem can be solved by inserting artificial loiter tasks $(w_{\text{loiter}}, p_{\text{loiter}})$ into W. These tasks are treated in the same way as real tasks by the RHTA algorithm, but their purpose is to force the vehicles to remain in advantageous areas.

Simulation Results

A multivehicle mission simulation was developed to test the task assignment algorithms. This simulation includes a base location and a number of vehicles (20 were simulated in the following tests), as well as a mechanism to randomly generate tasks and vehicle failures. The simulation runs RHTA to repeatedly assign tasks to vehicles and simulate the resulting system response.

The mission was to be carried out over a period of time longer than the flight endurance of the UAVs being used.

There are two metrics of performance calculated in the simulation: the average time it took to service each task (response time) and how many vehicles were lost during the mission (vehicle loss occurs when a vehicle runs out of fuel before returning to the base).

Simulation results are shown in Figure 2. The first test used RHTA in its original form. Since unmodified RHTA does not account for vehicle failures, it will command a failed vehicle to continue toward its original target despite the risk that it may run out of fuel and crash before returning to the base. The performance of unmodified RHTA results in an average service time of 21.3 s, and a vehicle loss rate of 25%.

The second test used the modified form of RHTA, which proactively recalls failed vehicles to the base while quickly reassigning a new, healthy vehicle to the task, using the idea of the operational radius discussed previously.

The results in Figure 2 clearly show that the modified RHTA provides a faster average response time due to its proactive reassignment behavior. The improvement in response time is about 18%, which is significant considering that the speed of the vehicles has not been changed, only the way they are assigned. In addition, the vehicle loss rate is significantly reduced (by 20%) because failed vehicles are automatically returned to the base instead of continuing toward their assigned tasks.

Flight Results

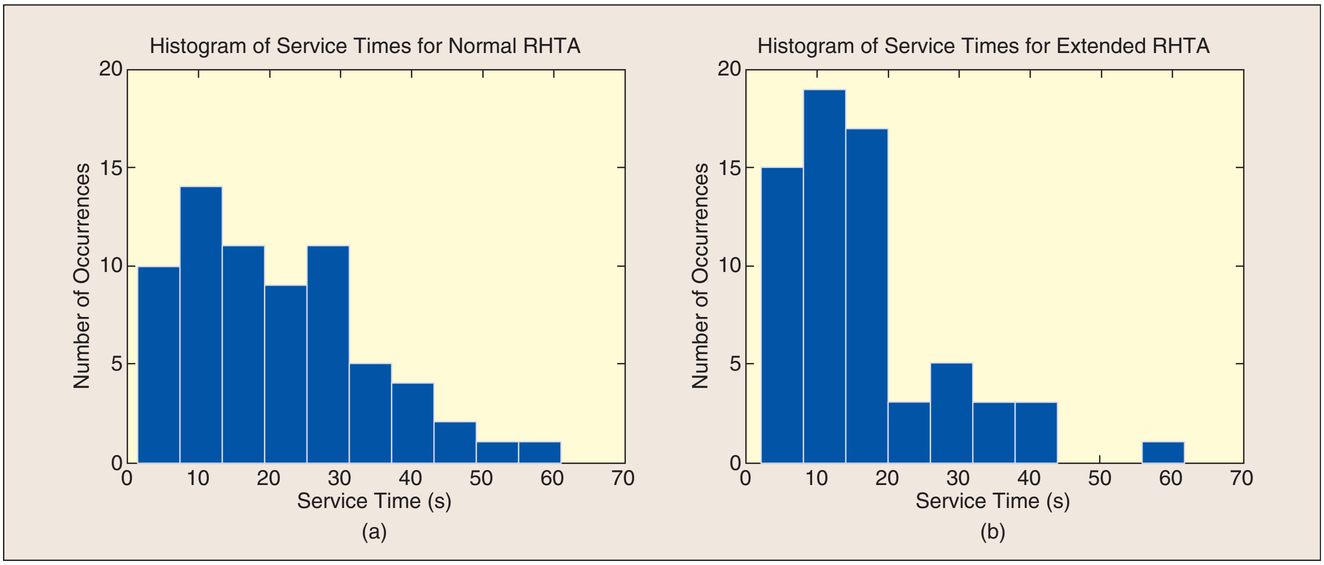
A set of experiments incorporating all aspects of the work presented thus far was conducted to demonstrate a complete, fully autonomous, persistent search and track mission on MIT's RAVEN (Real-time indoor Autonomous Vehicle test Environment) platform [17]. In these experiments, the UAVs used were Draganfly V Ti Pro R/C helicopters (see Figure 3). The mission goals were to search for, detect, estimate, and track an unknown number of ground vehicles in a predefined search region. The mission was to be carried out over a period of time longer than the flight endurance of the UAVs being used (around 5–10 min, depending on the charge of the battery), necessitating the coordination of multiple UAVs coming in and out of the flight area as required to maintain coverage. Finally, active health monitoring was required to detect and adapt to potential vehicle camera failures during the test.

To carry out the mission, a cooperative vision-based target estimation and tracking system [18], [19] was combined with the modified RHTA algorithm. Furthermore, the RHTA tasking system was interfaced to an autonomous mission system [12] that employed battery monitors to estimate the time of flight remaining for each UAV in the search area and handled requests by the tasking system to activate vehicles for use in the search or tracking activities.

The experiment setup is shown in Figure 3. Three UAVs are initially stationed at their base location at the far north end of the flight area, while two ground vehicles are positioned at random locations in the southern region. For these experiments, one of the vehicles was positioned on top of a box, while the other was located on the ground and was free to move.

The progression of the mission is according to the following sequences.

1) At the beginning of the test, the tasking system requests a single UAV from the mission system.
2) Once the requested UAV is airborne, the tasking system commands this UAV to begin an area search. During this initial detection phase, the UAV keeps track of how many distinct targets it has detected so far and stores them in a target list. The detection phase lasts for 2 min.
3) After the detection phase ends, the tasking system requests another UAV from the mission system.



(a)
(b)

Figure 2. *Simulation results: (a) Normal RHTA: median service time, 18.8 s; average service time, 21.3 s; vehicles lost, 5 of 20 (25.0%). (b) Extended RHTA: median service time, 14.0 s; average service time, 17.4 s; vehicles lost, 1 of 20 (5%).*

4) Once the second UAV is airborne, the system enters the tracking phase. The tasking system commands the second UAV into the search area so that there are now two UAVs in the area. Together, these two UAVs sequentially visit each location in the target list found during the detection phase. The UAVs spend 1 min at each location before moving on to the next. If there is a target at the given location when the UAVs arrive, they begin tracking the target. Additionally, although the tracking logic is designed to prevent collisions between the vehicles, a potential function-based method is used to ensure an additional level of safety. If a UAV comes too close to another UAV or an obstacle in the environment, it is repelled away by seeking to move to an area of lower potential.
5) At any point in the mission, the tasking or mission systems may determine that a particular UAV needs to return to the base. The reason for this may be either that the UAV is getting low on the remaining battery lifetime or that the UAV's camera has failed or is performing poorly. In either case, when a return-to-base condition is detected, the tasking system sends a sequence of waypoints to the UAV to command it back to the base. Once at the base location, the mission system lands the UAV and schedules any necessary refuelling or maintenance. At the same time, another UAV is launched and sent to the search area. In this manner, the mission is able to continue as UAVs cycle in and out.
6) The mission continues until a preset mission time expires or the human operator stops the mission. For these experiments, the mission time was 11 min.

In the detection phase, a single vehicle explored the search area and detected the presence of two ground vehicles. Figure 4 shows an early segment of the tracking phase after the second UAV had entered the search area. In this phase, the two UAVs estimated and tracked the position of the eastern ground vehicle using the vision tracking system [18], [19].

Figure 5 shows the time history of the mission for all the three UAVs used in the experiment. At t = 0, UAV 1 is taking off and surveying the area. It then requests a second vehicle for support at t = 182 s, and UAV 2 takes off and begins assisting in tracking targets. At t = 304 s, UAV 1 receives a low battery warning and returns to base, while UAV 3 takes off to replace UAV 1. At t = 433 s, UAV 3 experiences a simulated camera failure. The system detects the failure and sends UAV 3 back to base while commanding UAV 1 to take off again. The mission ends at t = 650 s. At several points during the mission, UAVs were successfully changed out because of low-battery states. In addition, a simulated camera failure during the tracking phase of the mission resulted in the failed vehicle returning to the base and a replacement vehicle being sent out. Due to these correct system responses, the goals of the

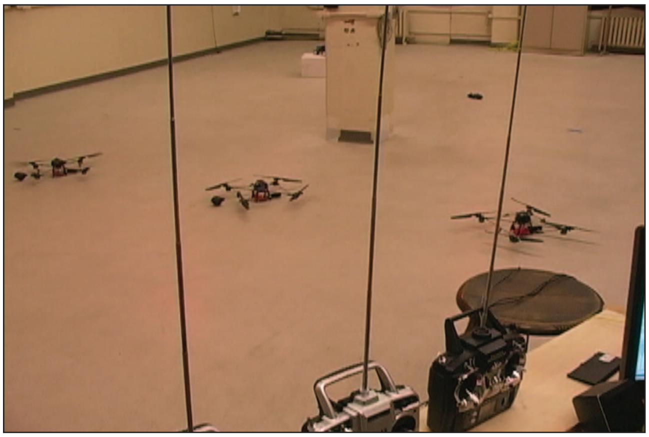


Figure 3. *Persistent search and track mission setup.*

The RHTA algorithm was extended to include the fuel state in the vehicle model.

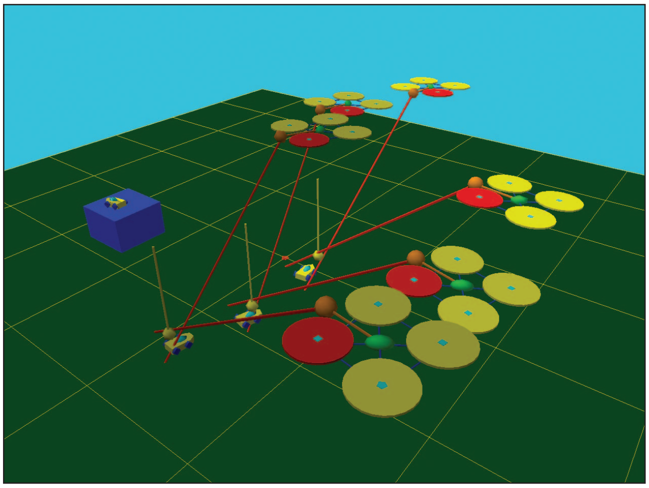


Figure 4. *Time-lapse image of one phase of the persistent mission showing cooperative tracking of a moving ground vehicle using two UAVs.*

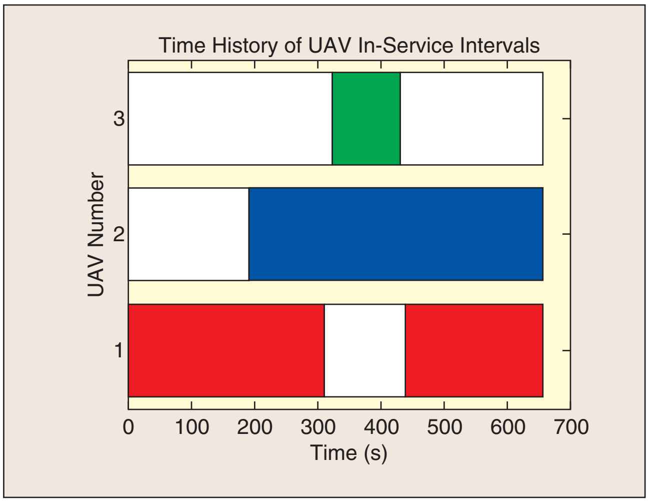


Figure 5. *Time history of the persistent surveillance mission. Colored blocks indicate times when that UAV was actively flying in support of accomplishing the mission.*

overall mission were able to be accomplished continuously over the course of the mission.

Conclusions

The health-aware task assignment algorithm developed in this article was demonstrated to be effective both in simulation and actual flight experiments. These initial results are very promising; however, more can be done in the health management problem in terms of accounting for other types of health states (sensor performance and control actuator failure modes). Furthermore, an important concept in the health management problem is to provide a robust performance in the face of uncertainty. Future work will focus on embedding more sophisticated stochastic models of numerous health states (including fuel usage and sensor performance) into the problem formulation and devising techniques to maximize performance while being robust to the uncertainty inherent in the problem.

Acknowledgments

We would like to thank Jim McGrew, Brandon Luders, Josh Redding, and Spencer Ahrens at the Massachusetts Institute of Technology (MIT), for their assistance in this research. Brett Bethke is sponsored by the Hertz Foundation and the American Society for Engineering Education. The research has been supported also by the Boeing Company and AFOSR grant FA9550-04-1-0458.

Keywords

UAVs, task assignment, health management, autonomous systems.

References

[1] Office of the Secretary of Defense. (2005). Unmanned Aircraft Systems Roadmap. [Online]. Available: http://www.acq.osd.mil/usd/Roadmap Final2.pdf

[2] H. Paruanak, S. Brueckner, and J. Odell, "Swarming coordination of multiple UAV's for collaborative sensing," in *Proc. 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conf.*, San Diego, CA, Sept. 2003, p. 6525.

[3] P. Gaudiano, B. Shargel, E. Bonabeau, and B. Clough, "Control of UAV SWARMS: What the bugs can teach us," in *Proc. 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conf.*, San Diego, CA, Sept. 2003, p. 6624.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2000.

[5] J. C. Hartman and J. Ban, "The series-parallel replacement problem," *Robot. Comput. Integr. Manuf.*, vol. 18, nos. 3–4, pp. 215–221, 2002.

[6] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operat. Res.*, vol. 46, no. 3, pp. 406–422, 1998.

[7] K. Andersson, W. Hall, S. Atkins, and E. Feron, "Optimization-based analysis of collaborative airport arrival planning," *Transport. Sci.*, vol. 37, no. 4, pp. 422–433, 2003.

[8] J. Rosenberger, E. Johnson, and G. Nemhauser, "Rerouting aircraft for airline recovery," *Transport. Sci.*, vol. 37, no. 4, pp. 408–421, 2003.

[9] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, vol. 3, pp. 2816–2822.

[10] C. Johnson, "Inverting the control ratio: human control of large autonomous teams," presented at the Int. Conf. on Autonomous Agents and Multiagent Systems: Workshop on Humans and Multi-Agent Systems Melbourne, Australia, July 2003.

[11] J. Wohletz, D. Castanon, and M. Curry, "Closed-loop control for joint air operations," in *Proc. American Control Conf.*, Arlington, VA, June 2007, vol. 6, pp. 4699–4704.

[12] M. Valenti, B. Bethke, J. How, D. Pucci de Farias, and J. Vian, "Embedding health management into mission tasking for UAV teams," presented at the American Controls Conf., New York, NY, June 2007.

[13] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 911–922, 2002.

[14] C. Schumacher, P. R. Chandler, S. J. Rasmussen, and D. Walker, "Task allocation for wide area search munitions with variable path length," in *Proc. 2003 American Control Conf.*, Denver, CO, June 2003, vol. 4, pp. 3472–3477.

[15] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," presented at the IEEE Conf. Decision and Control, Dec. 2004.

[16] M. Alighanbari, "Task assignment algorithms for teams of UAVs in dynamic environments," M.S. thesis, Dept. Aeronaut. Astronaut., Massachusetts Institute of Technology, Cambridge, 2004.

[17] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multivehicle flight testbed for fault detection, isolation, and recovery," in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, Aug. 2006, p. 6200.

[18] B. Bethke, M. Valenti, and J. How, "Cooperative vision based estimation and tracking using multiple UAVs," in *Proc. Conf. Cooperative Control and Optimization*, Gainesville, FL, Jan. 2007, pp. 179–189.

[19] B. Bethke, "Persistent vision-based search and track using multiple UAVs," M.S. thesis, Massachusetts Institute of Technology, Cambridge, 2007.

Brett Bethke is a research assistant at the Aerospace Controls Laboratory at MIT, Cambridge. He received his bachelors degree in physics and aerospace engineering from MIT and a fellowship from the Hertz Foundation and the American Society for Engineering Education. In June 2007, he received his master's degree in aeronautics and astronautics and is currently pursuing his Ph.D. degree in the Aeronautics and Astronautics Department at MIT.

Mario Valenti received his bachelor's degree in electrical and mechanical engineering from the Temple University, Philadelphia, Pennsylvania, and his master's degree in electrical engineering and computer science from MIT, Cambridge. He graduated with a Ph.D. in electrical engineering and computer science from MIT in June 2007.

Jonathan P. How is a professor in the Department of Aeronautics and Astronautics, MIT, Cambridge, where he leads the Aerospace Controls Laboratory. He graduated from the University of Toronto with a B.Sc. in aerospace engineering and received the M.Sc. and Ph.D. degrees in aeronautics and astronautics from MIT in 1989 and 1993, respectively. He is a Senior Member of the IEEE.

Address for Correspondence: Jonathan P. How, Aerospace Controls Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, MIT Rm 33-326, Cambridge, MA 02139 USA. E-mail: jhow@mit.edu.

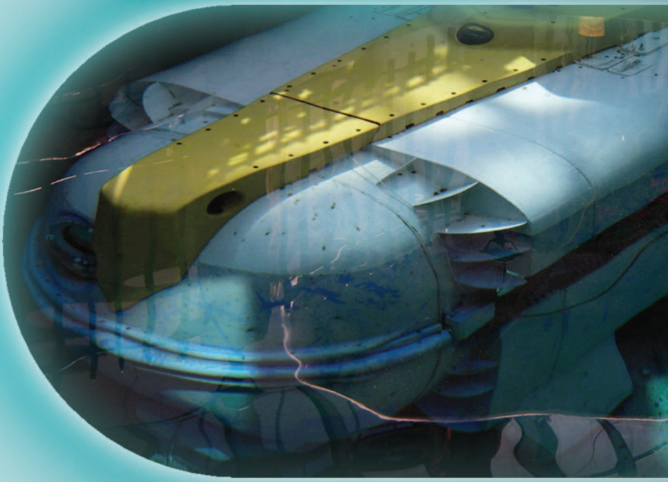
Naval Mine Countermeasure Missions

A Distributed, Incremental Multirobot Task Selection Scheme

BY SANEM SARIEL, TUCKER BALCH, AND NADIA ERDOGAN

© DREAMSTIME

© DREAMSTIME



© ISTOCKPHOTO

Mobile Multirobot Systems

Undersea operations using autonomous underwater vehicles (AUVs) provide a different and in some ways a more challenging problem than tasks for unmanned aerial vehicles and unmanned ground vehicles. In particular, in undersea operations, communication windows are restricted, and bandwidth is limited. Consequently, coordination among agents is correspondingly more difficult. In traditional approaches, a central planner initially assigns subtasks to a set of AUVs to achieve the team goal. However, those initial task assignments may become inefficient during real-time execution because of the real-world issues such as failures. Therefore, initial task allocations are usually subject to change if efficiency is a high concern. Reallocations are needed and should be performed in a distributed manner. To provide such flexibility, we propose a distributed auction-based cooperation framework, distributed and efficient multirobot-cooperation framework (DEMiR-CF) [1], which is an online dynamic task allocation (reallocation) system that aims to achieve a team goal while using resources effectively. DEMiR-CF, with integrated task scheduling and execution capabilities, can also respond to and recover from real-time contingencies such as communication failures, delays, range limitations, and robot failures. It has been implemented and tested extensively in the multirobot multitarget exploration domain [2] and in complex missions of interrelated and resource constrained tasks [3]. In this article, we report the performance of the framework against real-world difficulties encountered in multi-AUV coordination for the naval mine countermeasure (MCM) mission obtained through several experiments on the U.S. Navy's Autonomous Littoral Warfare Systems Evaluator-Monte Carlo (ALWSE-MC) simulator [4]. DEMiR-CF supports a distributed strategy for real-time task execution and is designed to use the advantages of auction-based approaches. Additional precaution routines are integrated into the framework to enhance solution quality. Other works in auction-based coordination research include M+ [5], MURDOCH [6], TraderBots [7], and the allocation scheme by Lemaire [8]. According to the review given in [9], existing auction-based systems are not fully capable of replanning task distributions, redecomposing tasks, rescheduling commitments, and replanning coordination during execution. Our approach aims at filling these gaps. We propose an integrated cooperation framework for multirobot task execution and analyze the performance of the precaution routines and solution quality maintenance schemes for single-item auctions in a multi-AUV coordination context [10]. Experiments are performed in a realistic simulation environment with real-time constraints and events such as AUV failures and limitations, and delays in communication range. Precaution routines embedded into the framework not only recover from failures but also serve to

Digital Object Identifier 10.1109/M-RA.2007.914920

Additional precaution routines are integrated into the framework to enhance solution quality.

maintain a high solution quality. Our experiments show that communication delays significantly influence the solution quality and should be analyzed in multirobot systems, especially working in harsh environments. As the experiments and scenarios demonstrate, online task handling performance of DEMiR-CF is considerably promising.

Naval MCM Missions

Naval MCMs are actions taken to counter the effectiveness of underwater mines. MCM operations include finding and seizing mine stockpiles before they are deployed, sweeping desired operational areas, identifying mined areas to be avoided, and locating and neutralizing individual mines [11]. Our research is focused on the subset of MCM operations that involve locating and mapping all individual mines in an operational area. In general, recognizing proud mines on the seafloor is not overly difficult; the difficulty arises with the abundance of nonmine objects on the seafloor that possess mine-like characteristics (e.g., geologic outcroppings, coral, manmade debris) [12]. This ample supply of false alarms has necessitated the following strategy typically employed by the navy: detect and classify mine-like objects (MLOs) with high-coverage rate sensors (e.g., sidelooking sonar), employ advanced signal processing techniques for maximal false alarm reduction, and then revisit the remaining MLOs with identification-quality assets (e.g., electrooptic sensors) to confirm them as mines or dismiss them as false alarms. A sample image in which an MLO remains is illustrated in Figure 1.

The reference mission in this research is to detect, classify, and identify underwater mines in a given operational area simulated in ALWSE-MC [4], an analysis package designed to simulate multiple autonomous vehicles performing missions in the littoral regions, including mine reconnaissance, mapping, surveillance, and clearance. This mission employs two types of vehicles: unmanned underwater vehicles (UUV), which are free-swimming AUVs and possess large-footprint sensors (e.g., side-scan sonar) for detection and classification (D/C) of mines, and seafloor crawlers equipped with short-range, identification-quality sensors (e.g., camera). The crawlers have the ability to stop at an object and take a picture with a camera.

Figure 1. Sidelooking sonar sensors may fail in correctly classifying mines because of their similarities to some nonmine objects in undersea habitat [12].

The MCM domain has important similarities to some of the well-known domains where the use of a multirobot team is usually beneficial. The search and rescue domain where different types of robots are required is one example. Searching for victims in the disaster area is similar in nature to the detection of mines. Rescue operations in which first aid is provided to victims are also similar to the classification tasks. Another interesting domain, the space exploration mission, has a high resemblance in form also. The mission can be divided into two submissions: searching for important points to reconsider and revisiting the sample points determined in the first phase to further investigate specific locations and collect scientific data with more specialized robots. Therefore, we believe that the solutions offered to carry out the MCM mission can be successfully applied to these domains also.

DEMiR-CF

The MCM mission is performed undersea and in real time. Managing the overall robot team by a central authority is not usually possible because of the limitations of the real-world environment. Therefore, each individual robot should find a way to solve the global problem from its local perspective while assuming a global approach is possible in a distributed setting.

To meet the real-world limitations, we propose a dynamic and distributed task allocation scheme, DEMiR-CF, to coordinate robots that cooperate to fulfill different parts of a mission. DEMiR-CF is designed for complex missions including interrelated tasks that require diverse (heterogeneous) capabilities and simultaneous execution [1], [13]. Dynamism is achieved through incremental selection and allocation of the targets. By means of the distributed characteristic of the proposed allocation scheme, each robot is allowed to select a candidate task for itself and, next, the robots proceed to cooperate in the process of selecting the most suitable robots for the tasks. A time-extended view is considered while selecting tasks after forming rough schedules. The framework combines the dynamic priority-based task selection scheme, distributed task allocation procedures and coalition formation schemes as cooperation components, and Plan B precaution routines, some of which are implemented by the coalition maintenance or dynamic task selection scheme. These components are integrated into a single framework to provide an overall system that finds near-optimal solutions for real-time task execution. The modules that embody the framework and information flow among them are given in Figure 2. Each robot keeps a model, which includes information on current status, of the other robots and the mission tasks. The model update module, the (system) consistency checking module, and the dynamic task selector module perform Plan B precaution routines by either updating the model maintained by the robot or activating the warning mechanisms. Model updates are initiated by either incoming information from the other robots or information perceived by the robot itself. If a system inconsistency arises, the consistency checking module is responsible for initiating warning mechanisms and informing the corresponding robots. The dynamic task selector module selects the most suitable task by considering the model of the robot. The distributed allocation scheme ensures the distributed task allocation by executing the required negotiation procedures for the selected task. The execution or coalition scheme implements synchronized task execution and coalition maintenance procedures. Task models are updated according to the selected task and the task currently in execution. A sample flow of the operations in the framework (as depicted in Figure 2) is summarized as follows:

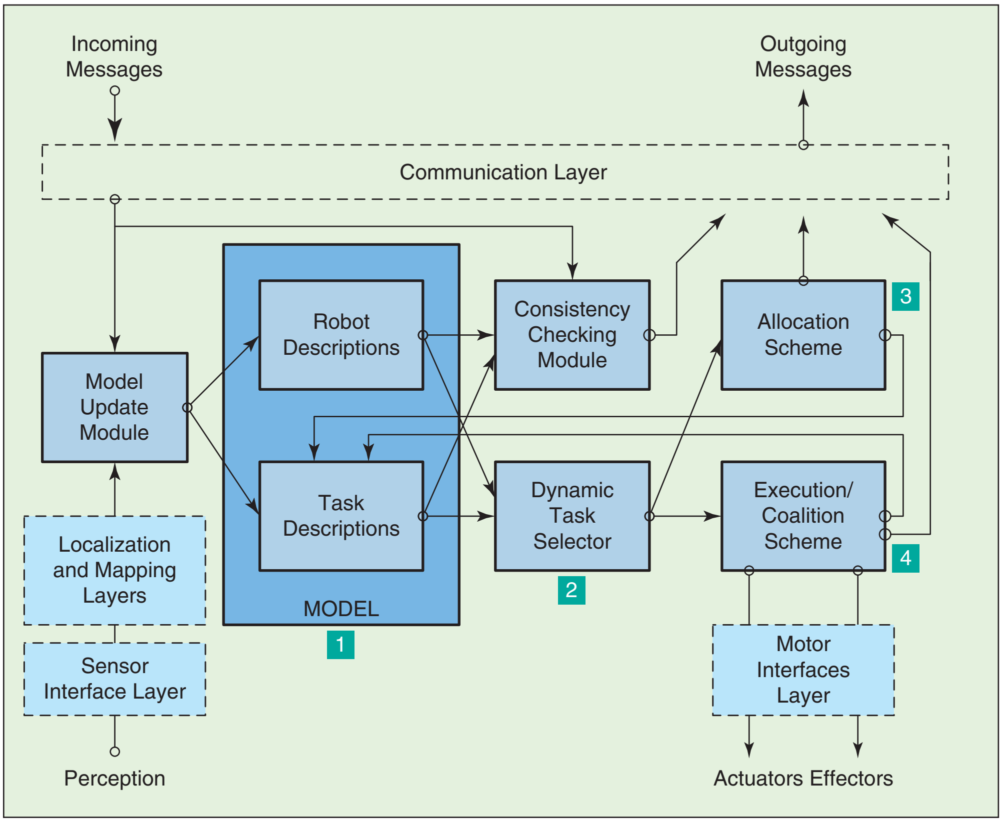


Figure 2. *DEMiR-CF modules.*

- Initially, robots are delivered the mission task definitions.
- Each robot selects the most suitable candidate task to execute through global cost consideration (dynamic task selection or switching).
- Robots offer auctions for the tasks they have selected. During auction steps, inconsistencies are cleared and conflicts are resolved.
- Task assignments are made for the announced tasks, making sure that each robot takes part in the most suitable execution when the global solution quality is considered.
- Dynamic task selection or switching proceeds simultaneously with task execution. This allows the robot to switch between tasks when executing the candidate task becomes more profitable than continuing with the current task and handling real-time contingencies at the same time. Hence, corresponding auction and selection procedures (second through fourth items) are applied continually.

DEMiR-CF is designed with the capability to deal with real-time situations. The framework can efficiently respond to these events and maintain the solution quality simultaneously with real-time task execution.

Plan B Precautions

Plan B precautions are taken in DEMiR-CF by the model update module, which updates the system model of the robot, and the system consistency checking module. The model update module uses incoming information from the other robots and its own perception data to update the world model. The system consistency checking module provides warning that initiate actions to keep the system consistent.

Recovery operations may include warning other robots about the problem or changing the model accordingly. Inconsistencies usually arise when robots are not informed about tasks that are achieved, under execution, or under auction in real-world operations. To keep system consistency, robots use explicit communication and broadcast the information as follows:

- Tasks known to be achieved in predefined time periods to prevent redundant executions. (This feature provides a bucket-brigade type of information sharing that enables information transition from one robot to another where point-to-point access is not possible, and consequently communication range limitations are resolved.)
- Newly discovered online tasks that are not yet achieved.
- Task execution messages in predefined time periods. (These messages contain the updated cost value and the estimated task achievement deadline information. Therefore, they serve as clues, meaning that the executer robot is still alive and the task is under execution.)
- Task achievement message when the task is achieved.
- Cancellation message if the task execution is canceled.
- Task invalidation message when an invalidity is detected.

Incoming messages from other robots are taken as clues for being marked as running properly. Some misleading beliefs such as setting the state of a robot as failed although it is running

In undersea operations, communication windows are restricted and bandwidth is limited.

properly may cause parallel executions. This is a desired feature from the point of view of the completion of mission. Designed precautions resolve these kinds of inconsistencies if communication resources permit in later steps. In designing the precautions, it is assumed that robots are trusted and benevolent.

Task Representation for the MCM Mission

Our general task representation is capable of describing complex tasks with interdependencies [1]. However, in this particular case study, tasks do not have interdependencies. Two types of tasks are defined for vehicles: visit waypoint (w) and identify MLO (t). The task representation includes the capabilities required for each type of task: reqcap_w contains side-scan sonar and reqcap_t contains cameras besides the standard capabilities of AUVs common in both types of vehicles. The coverage mission (M_{C}) contains predefined number of waypoints ($w_i \in M_{\mathrm{C}}$, $0 < i \leq ||M_{\mathrm{C}}||$) to be visited by all UUVs ($R_{\mathrm{UUV}} \subset R$). One way to represent a task is to directly assign it for each waypoint. However, this representation has a drawback of high communication requirements for efficient completion of the mission. Instead, we represent tasks as interest points of regions or search areas ($W_k = \cup w_i$, $\forall w_i$ is unvisited, and $W_k \subseteq M_{\mathrm{C}}$). These regions (and the corresponding centers) are determined by the robots during runtime dynamically although the waypoint locations are fixed at known coordinates. Therefore, both the allocation of the waypoints to the robots and the paths constructed to traverse these waypoints are determined online by negotiations. Negotiating the interest points (regions) instead of the individual waypoints reduces the communication overhead. Regions determined by different UUVs may vary during runtime and may sometimes overlap. However, the uncertainty related to the region determination is within an acceptable range, especially when the cost is compared with the requirements of complete knowledge sharing by representing each waypoint as a task. Before defining the regions, the relative distance values, $\mathrm{reldist}(r_j, w_i)$, are determined for each unvisited waypoint w_i using (1), where function dist returns the Euclidean distance between points. r_k locations are the latest updated locations of the robots. If there is no known active robot assumed to be running properly, $\mathrm{reldist}(r_j, w_i)$ is the value of the distance between the robot and the waypoint

$$\mathrm{reldist}(r_j, w_i) = \mathrm{dist}(r_j, w_i) - \min_{\forall k \neq j} (\mathrm{dist}(r_k, w_i)). \qquad (1)$$

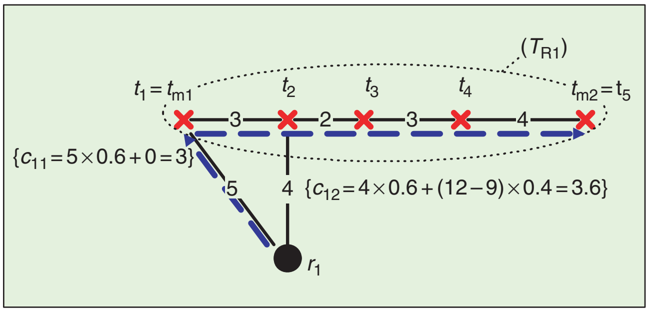


Figure 3. *Target selection strategy by the FAC heuristic function.*

Each robot defines its regions ($W_{jk}, 1 \leq k \leq ||R_{\mathrm{UUV}}||$). The number of regions equals the number of UUVs believed to be running properly. After sorting the $\mathrm{reldist}(r_j, w_i)$ values of the unvisited waypoints in descending order as an array, the array is cut into subarrays that represent the regions. Each region contains approximately an equal number of waypoints. Each robot specifies the region of highest interest as its first region. If the robots are closely located, the regions of highest interest may overlap. In this case, negotiations are needed to resolve conflicts and to assign only one robot for each region.

The identification mission (M_{I}) contains an unknown number of tasks for the MLO locations ($t_i \in M_{\mathrm{I}}, 0 < i \leq ||M_{\mathrm{I}}||$) to be visited by the crawlers. Therefore, the tasks in M_{I} are generated online during runtime.

Exploration for Detection of MLO Locations

To begin the mission, the UUVs survey the operational area following waypoints determined a priori; however, corresponding regions containing waypoints may be reassigned by the negotiations among UUVs autonomously. After determining regions, each UUV proposes an auction for the region of highest interest (interest point). After negotiations on several auctions, each UUV is assigned to the closest region (interest point). If more than one robot is almost at the same distance from the interest point, the one with the smaller id number is assigned to the region. The other UUVs continue to offer auctions for the remaining regions. Allocations of the regions may also change during run time to maintain higher solution quality. Whenever UUVs detect failures or recoveries from failures, they change their region definitions accordingly and offer new auctions. After the region assignments are completed, each robot visits waypoints in its region (W_j) in a sequence identified by an ordering of their cost values from the smallest to the largest:

$$\begin{aligned} c(r_j, w_i) &= \alpha \cdot \mathrm{dist}(r_j, w_i) \\ &\quad + (1 - \alpha) \cdot [\mathrm{dist}(w_{f1}, w_{f2}) \\ &\quad - \max(\mathrm{dist}(w_i, w_{f1}), \mathrm{dist}(w_i, w_{f2}))] \\ \{\mathrm{dist}(w_{f1}, w_{f2}) &= \max(\mathrm{dist}(w_k, w_l)), w_{l,k,j,f_1,f_2} \in W_j\}. \qquad (2) \end{aligned}$$

This heuristic function considers boundary targets, w_{f1} and w_{f2} in W_j, which are the targets with the maximum distance value. The basic idea of this function is to forward the robot to one of these boundary targets since these targets determine the diameter of the region (W_j) and both of them should be visited. If the robot initially heads toward one of the boundary targets, the diameter (the longest path) can be traveled by visiting other targets along the path. A sample illustration of this cost function is given in Figure 3. In this figure, although t_2 is closer to r_1 than t_1, with the farthest addition cost (FAC) heuristic applied, t_1's cost value is smaller than that of t_2 $(3 < 3.6)$, which results in a better route shown by the dashed arrows. The cost penalty applied to forward the robot to the boundary targets is limited to a small degree. By introducing a constant (α), this degree of direction can be adjusted. When α is assigned a value of $2/3$, this heuristic function produces close to optimal results for the multirobot multitarget allocation domain [2]. If more than one pair of boundary targets exist, the pair that has a member at the smallest distance from the UUV is selected.

An illustrative example of the generation of the search regions (areas) and the traversed path patterns by the robots are depicted in Figure 4. Since there are three robots in this figure, three search regions are determined and covered by the robots.

As UUVs detect the MLOs on their way, they broadcast these estimated target positions to all AUVs (hence, tasks for crawlers are generated online during execution). Then, MLO information can propagate to all other AUVs in the group that can possibly be reached. Periodic broadcasting of important information (coming from either owned sensors or external agents) is a way to handle communication range limitations.

Identification of MLOs

When the crawlers are informed about the MLO locations, they update their world knowledge and dynamically select the best MLO targets to visit and propose auctions. Therefore, they can switch among tasks when new tasks appear if it is more profitable. It is also possible that a crawler may inadvertently discover a mine without being informed of its position by a UUV. In this case, the crawler identifies the target, adds it to its task list as an achieved task, and broadcasts achievement information to maintain the system consistency. Crawlers determine their bid values by using the cost functions proposed for the multirobot multitarget exploration domain [2].

In the identification task, when crawlers are within an area close to an MLO location, they begin keeping time while surveying the MLO location. Whenever the time limit is reached, they set the task status as achieved and broadcast this information. If a detection event occurs during this time period, the MLO location is considered to be an actual mine; otherwise, it is determined as a false alarm after deadline. In either case, the task is marked as achieved.

Experimental Results on the MCM Mission

The performance of our framework and the precaution routines is evaluated in ALWSE-MC. Three sample scenarios in the simulator are given to illustrate the performance of our framework for the naval MCM mission. The MCM mission movies are available online at [14]. UUVs are equipped with sensors capable of detecting mines within 30 ft from the skin of a target. However, they are not able to correctly identify them. The crawlers are equipped with cameras that can both detect and identify mines within 20 ft. None of the AUVs have predefined search patterns. UUVs have internal navigation errors; therefore, their estimated location values are different from actual locations in most cases. Two AUVs can communicate each other whenever the receiver AUV is in the sender AUV's transmitter range, within its transmitter beam width, and the sender AUV is within the transmitter AUV's receiver beam width.

Naval MCMs are actions taken to counter the effectiveness of underwater mines.

All UUVs and crawlers begin execution from a deployment area. There is no a priori information about mine locations. Around 121 waypoint locations (environment size: 200 × 200) are known but are not assigned initially. UUVs begin negotiations and divide the overall mission area into three (known number of UUVs) regions. Since they are within the line of sight, they can communicate their location information. Therefore, initially defined regions are nearly the same for all UUVs. Figure 5 illustrates a successful mission scenario

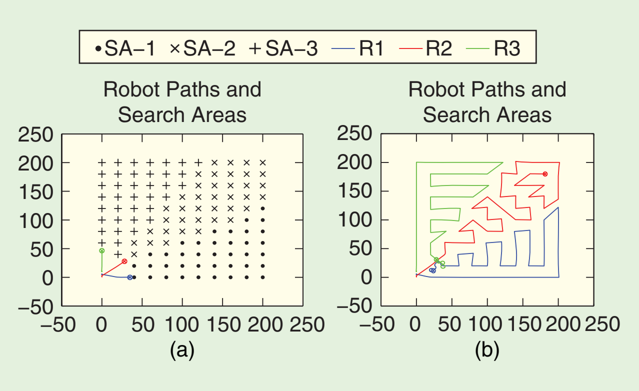


Figure 4. *(a) Mission execution begins. The overall area is divided into regions. (b) Robots patrol the area in the corresponding regions.*

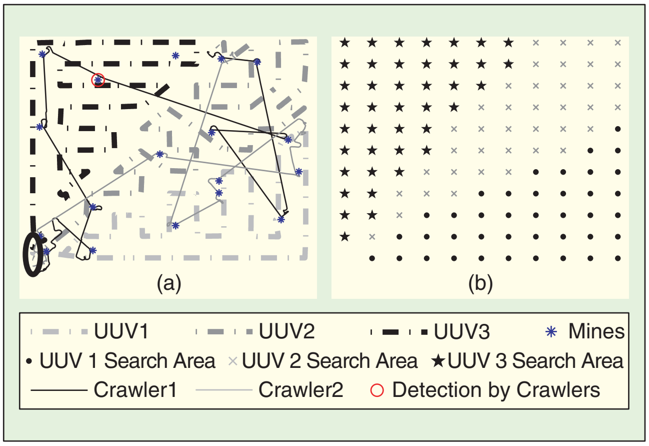


Figure 5. *Scenario 1: (a) The UUVs cover the area, and the crawlers visit the MLO locations. (b) The UUV regions are illustrated.*

DEMiR-CF is designed with the capability to deal with real-time situations.

with three UUVs and two crawlers. Allocations of waypoints after negotiations can be seen in Figure 5(b). Since there are no failures, waypoint assignments do not change during run time. However, the crawlers sometimes switch among tasks if they are not informed about tasks that are being executed, and sometimes parallel executions occur. Whenever they are in communication range, they can resolve the conflicts efficiently by means of the precaution routines. As shown in Figure 5(a), the crawlers can also detect mines without being informed (red circled in the figure). The routes of the crawlers may seem somewhat random. However, it should be noted that the tasks related to the MLO locations appear online during run time when they are discovered, and the communication range is limited.

In Scenario 2, UUV3 fails in the same setting of Scenario 1 (Figure 6, the location of the failure is indicated with a red arrow in the figure). Initial regions for all UUVs change after UUV3 fails [Figure 6(b)]. The other UUVs revise their region definitions and, after negotiations, they share the full area as indicated in the figure. The visited waypoints are not in their region coverage. Because of the uncertainties, some waypoints may remain uncovered in the schedules (indicated with the red diamond in the figure). However, this uncertainty-related problem is resolved by UUV2, and the mission is completed.

In the Scenario 3 (Figure 7), UUV3 fails and the other UUVs detect the failure and they negotiate the remaining unvisited waypoints and new schedules are determined as in Figure 7(b). While these UUVs execute their tasks, UUV4 is released from the deployment area. Detecting the arrival of a new UUV, the other UUVs change their region definitions accordingly [Figure 7(d)] and offer auctions for these areas. Initially UUV4 is not informed about the visited waypoints

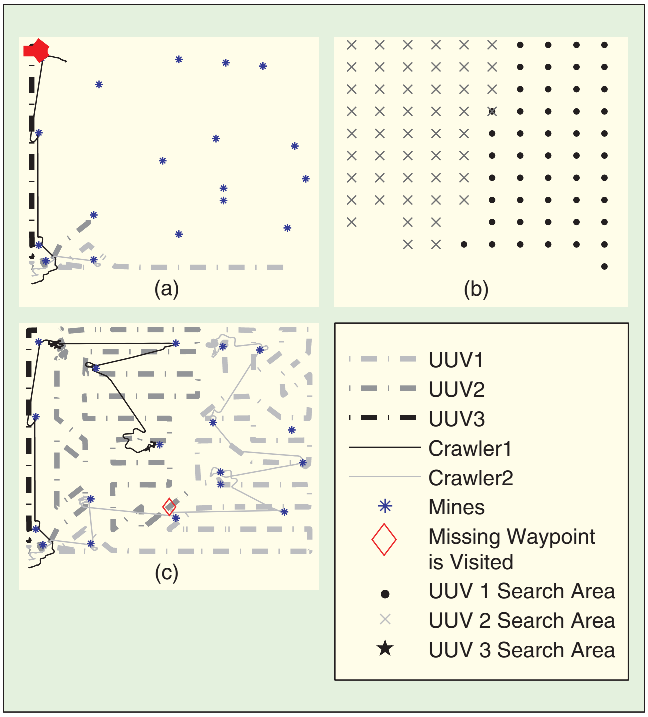


Figure 6. *Scenario 2: (a) Initially, all UUVs begin execution, UUV3 fails, and other UUVs take responsibility of all unvisited waypoints. (b) Region assignments are changed for UUV1–2 after detecting the failure. Because of an uncertainty, one waypoint is left uncovered. (c) UUV2 completes its region coverage task and adds the waypoint missing in (b) to its schedule after detecting that it is not visited.*

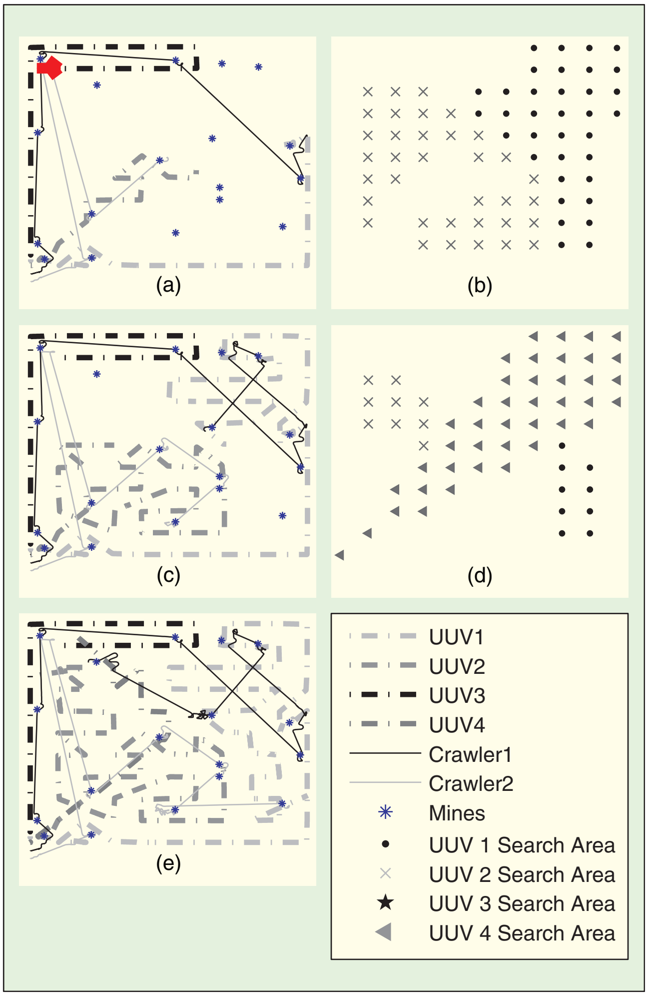


Figure 7. *Scenario 3: (a) UUV3 fails and other UUVs take responsibility of the waypoints initially assigned to UUV3. (b) Region assignments are changed for UUV1–2 after detecting the failure. (c) Another UUV(4) is released from the deployment area. (d) Schedules are changed accordingly after negotiations. However, UUV4 is not informed about visited waypoints and form regions by considering all waypoints. (e) After being informed about visited waypoints, UUV4 only visits unvisited waypoints in its schedule.*

and defines its regions with this incomplete knowledge. After negotiations, the regions are assigned and the schedules are formed. Entering into the communication range, UUV4 redefines its regions by considering incoming information for the visited waypoints.

In the same settings, another experiment is conducted to evaluate the message loss rate effects on the success of the completion of mission. Table 1 illustrates the results $(\mu|\sigma)$ averaged over ten runs. When the message loss rate is different from 0, as expected, the mission completion time performance of the system degrades but linearly. It should be noted that, even for a rate of 0.75, the overall mission (M_C and M_I) by the final identification of the mines is completed. The average of the first visit times of the waypoints increases linearly because of the delays occurring by redundant visits of the targets. The number of waypoint (w) visits increases with high message loss rates. When the message loss rate is one, there is no communication among AUVs, and they cannot correctly reason about the region portions. Therefore, each UUV searches the full area completely. The crawlers detect and identify 12.8% of mines by their local detection in a small area (MLO target information cannot be communicated in this case). Since the identification mission is not complete, the overall mission is not completed. This table illustrates the performance of our framework against message losses. As a final remark, auction generation and clearing in an environment with communication delays desires special attention. Especially, auction deadlines should be determined by considering communication delays that may vary during the run. Plan B precautions can resolve these kinds of problems. Precautions for delayed messages on out-of-date situations prevent the system from getting stuck into further inconsistencies and deadlocks.

Further Extending MCM Mission to Prevent Hostile Attacks

The MCM mission can be further extended with the presence of possible threats from hostile vehicles. We analyze this situation in a dynamic simulation environment where the mission consists of the online tasks, whose generation times are not known in advance by the robots (AUVs). The overall mission is to search a predefined area as a part of the MCM mission and additionally protecting the deployment ship from any hostile intent [1].

The MCM mission is performed undersea and in real time.

Discussion and Conclusions

In this article, we presented the performance of a new framework, DEMiR-CF, in the context of a naval MCM mission in the realistic NAVY simulator ALWSE-MC. DEMiR-CF is a distributed framework for multirobot teams that integrates incremental task selection schemes, distributed allocation methods, and several precaution routines to handle failures and limitations of the real-world task execution. It maintains high solution quality with available resources. Precaution routines can respond to several failures as illustrated in the scenarios presented in this article. Evaluations reveal the high performance of DEMiR-CF on online task and situation handling. Since the framework is a single-item auction method, it can be used for environments with limited, delayed, or unreliable communication. In general, the framework is designed for more complex missions of interrelated tasks. We have implemented the DEMiR-CF framework on Khepera II real robots for the allocation of tasks of the multirobot multitarget exploration mission that can be treated as the classification tasks. Since the proposed approach is computationally cheap, its implementation on even very small robots has been possible, which makes the approach broadly applicable for different robot platforms. Accordingly, as the realistic simulation results reveal, limiting the assumptions in the design of the approach facilitates its porting to the real underwater vehicles. The naval MCM domain has appropriate characteristics to deploy teams of robots and let them cooperate to achieve the overall mission. It should be noted that the objectives and the limitations of this domain are similar to those of both search and rescue and space exploration domains. Therefore, we believe that research in this work can be useful for these domains as well.

Future work on the presented research includes considering the coverage and the detection strategies of the MCM mission together to improve the performance of the system. Especially, if the communication range is known a priori, this information can also be used in region determination and in constructing the paths of the robots to improve the responses of the system to robot failures.

Table 1. Performance results $(\mu|\sigma)$ for different message loss rates.

	0		0.25		0.5		0.75		1	
Message Loss Rate	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
M_C completion (%)	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0
M_I completion (%)	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0	12.8	4.1
M_C completion (t)	3,349.4	60.5	3,683.2	167.1	4,909.0	430.1	5,141.2	938.1	6,304.2	139.0
M_I completion (t)	2,852.8	35.3	3,227.6	205.3	4,205.0	836.9	5,021.2	692.7	N/A	N/A
w first visit	1,380.1	6.1	1,390.0	16.3	1,922.0	92.8	2,256.6	334.5	2,936.0	104.5
w number of visits	1.0	0.0	1.0	0.0	1.01	0.01	1.09	0.04	3.0	0.0

Acknowledgment

We thank Jason R. Stack from Naval Surface Warfare Center, USA, for his constructive comments on this research.

Keywords

Multirobot cooperation, naval mine countermeasures, incremental task selection, robustness.

References

[1] S. Sariel, "An integrated planning, scheduling and execution framework for multi-robot cooperation and coordination," Ph.D. dissertation, Istanbul Technical Univ., Turkey, 2007.

[2] S. Sariel and T. Balch, "Efficient bids on task allocation for multi-robot exploration," in *Proc. 19th Int. FLAIRS Conf.*, 2006, pp. 116–121.

[3] S. Sariel, T. Balch, and N. Erdogan, "Incremental multi-robot task selection for resource constrained and interrelated tasks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2007, pp. 2314–2319.

[4] U.S. Navy. (2008, Jan.). ALWSE. [Online]. Available: http://nswcpc.navsea.navy.mil/analysis/capabilities.asp

[5] S. Botelho and R. Alami, "M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proc. IEEE Intl. Conf. Robotics and Automation (ICRA)*, 1999, pp. 1234–1239.

[6] B. Gerkey and M. J. Mataric, "Sold!: Auction methods for multirobot coordination," *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 758–768, 2002.

[7] M. B. Dias and A. Stentz, "Opportunistic optimization for market-based multirobot control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2002, pp. 2714–2720.

[8] T. Lemaire, R. Alami, and S. Lacroix, "A distributed task allocation scheme in multi-UAV context," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2004, pp. 3622–3627.

[9] M. B. Dias, R. M. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-13, 2005.

[10] S. Sariel, T. Balch, and J. R. Stack, "Empirical evaluation of auction-based coordination of AUVs in a realistic simulated mine countermeasure task," in *Distributed Autonomous Robotic Systems (DARS)*. vol. 7, M. Gini and R. Voyles, Eds. New York: Springer-Verlag, 2006, pp. 197–206.

[11] J. R. Stack and R. C. Manning, "Increased autonomy and cooperation in multi-AUV naval mine countermeasures," in *Proc. Undersea Defence Technology Conf.*, 2004.

[12] G. J. Dobeck, "Image normalization using the serpentine forward-backward filter: Application to high-resolution sonar imagery and its impact on mine detection and classification," in *Proc. SPIE (Special Issue on Detection and Remediation Technologies for Mines and Minelike Targets X)*, 2005, vol. 5794, pp. 381–391.

[13] S. Sariel and T. Balch, "A distributed multi-robot cooperation framework for real time task achievement," in *Distributed Autonomous Robotic Systems (DARS)*. vol. 7, M. Gini and R. Voyles, Eds. New York: Springer-Verlag, 2006, pp. 187–196.

[14] S. Sariel. (2008, Jan.). MCMMovies. [Online]. Available: http://www2.itu.edu.tr/∼sariel/videos/MCM-Movies.html

Sanem Sariel received the B.S. degree in computer and control engineering and M.Sc. degree in computer engineering from Istanbul Technical University (ITU) in 1999 and 2002, respectively. She was awarded a research scholarship for her Ph.D. studies in the United States in 2004, where she worked with Prof. Tucker Balch in the BORG Laboratory at Georgia Institute of Technology. She was jointly advised by Prof. Tucker Balch and Prof. Nadia Erdogan and received the Ph.D. degree in computer engineering from ITU in 2007. She is an instructor in the Computer Engineering Department of ITU, where she served as a research and teaching assistant from 1999 to 2007. Her research interests include distributed problem solving, multirobot coordination or cooperation, and intelligent agents.

Tucker Balch is director of the Institute for Personal Robots in Education (IPRE) and associate professor in the School of Interactive Computing at Georgia Tech. Tucker's robotics research interests include machine learning for robot navigation and large-scale multirobot systems. His recent work with IPRE is focused on understanding how robots can be used to help students learn more effectively. Balch received the B.S. and Ph.D. degrees in computer science from Georgia Tech in 1984 and 1998, respectively. He flew F-15s in the U.S. Air Force from 1988 to 1995. He has published more than 80 technical articles and two books.

Nadia Erdogan received her B.S. degree in electrical engineering and M.Sc. degree in computer science from Bosphorus University in Istanbul, Turkey, in 1978 and 1980, respectively. She received her Ph.D. degree in computer engineering from Istanbul Technical University, Turkey, in 1987, where she is now a professor in the Computer Engineering Department. Her current research areas include distributed computing and execution environments, mobile agent systems, and parallel programming.

Address for Correspondence: Sanem Sariel, Department of Computer Engineering, Istanbul Technical University, Turkey. Phone: +90 212 285 38 52. Fax: +90 212 285 36 72. E-mail: sariel@itu.edu.tr.

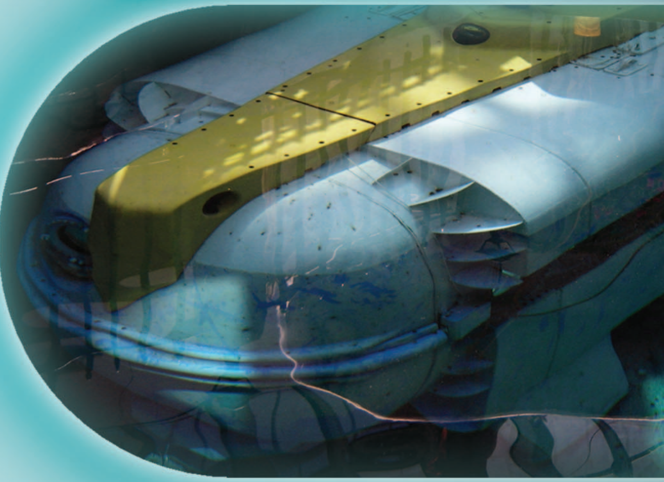
Experimental Testbed for Large Multirobot Teams

Verification and Validation

BY NATHAN MICHAEL, JONATHAN FINK, AND VIJAY KUMAR

© DREAMSTIME

© DREAMSTIME



© ISTOCKPHOTO

Mobile Multirobot Systems

Experimental validation is particularly important in multirobot systems research. The differences between models and real-world conditions that may not be apparent in single robot experiments are amplified because of the large number of robots, interactions between robots, and the effects of asynchronous and distributed control, sensing, and actuation. Over the last two years, we have developed an experimental testbed to support research in multirobot systems with the goal of making it easy for users to model, design, benchmark, and validate algorithms. In this article, we describe our approach to the design of a large-scale multirobot system for the experimental verification and validation of a variety of distributed robotic applications in an indoor environment.

Our research focusses on decentralized multirobot algorithms that rely on an integrated approach to mobility, perception, and communication, with such applications as environmental monitoring, surveillance and reconnaissance for security and defense, and support for first responders in search and rescue operations [1]. In all of these applications, robots must rely on local sensing, computation, and control and exploit the availability of communication links with other robots whenever possible. To enable scaling up to large numbers, computations must be decentralized, and the system must be robust to changes in the numbers of robots and to the dynamic addition and deletion of units. There is also the need to provide some degree of centralization with an interface to one or more human operators for programming, tasking, and monitoring of the system.

These research applications serve as the motivation for our experimental testbed. While there is a rich body of work to build on, there is currently no inexpensive multirobot system that allows users to move easily from conceptual ideas to algorithms and then to experimentation. We begin by motivating design considerations for the testbed in the context of our research and existing multirobot control and experimental architectures. We next arrive at a set of design requirements for the system based on the driving applications as well as practical considerations. Most importantly, we are driven by the pragmatic considerations of ease of use, robustness, flexibility, and scalability to enable the easy inclusion of more robots and sensors with minimal changes to the existing infrastructure. We also review some of the applicable hardware and software options currently available. The experimental testbed is discussed in detail with overviews of the robots, software, and the supporting infrastructure required for multirobot experiments. Since simulation is of great relevance in the experimental process and the testbed design, we discuss its role and detail the transition from simulation to reality. Finally, we present several multirobot experiments for formation control and cooperative manipulation, which demonstrate the capabilities of the system for verification purposes and elucidate the experiment design process with our testbed.

Digital Object Identifier 10.1109/M-RA.2007.914924

1070-9932/08/$25.00©2008 IEEE

Motivating Design Considerations

A number of multirobot control and experimental architectures [2]–[4] have been developed over the years for use with teams of robots on the ground [5], [6], in the air [7], or under water [8], many of which were inspired by behavior-based control paradigms [9]. Often, architectures rely on hierarchy to manage the complexity of the task and the control software [10]. Additionally, the need to have decentralized control to enable scalability to large numbers is clear [3], [11]. However, to command large groups of robots, it is also essential to include an element of centralization to allow humans to interact and task the team.

The design of the experimental testbed was motivated by our interest in multirobot control for the deployment of potentially large numbers of cooperating robots with applications to tasks such as persistent surveillance, object manipulation, and transportation. We have proposed several methodologies in the context of these applications such as formation control [12]–[14], cooperative manipulation [15], and pattern generation [16] with the requirement that the algorithm adheres to three attributes: decentralization, anonymity, and uniform modularity. Decentralization means that the algorithm does not require access to the full global state and all control computations are done locally. Anonymity implies that the algorithm does not require robots to identify each other. Uniform modularity in algorithm implementation extends the idea of anonymity to further promote the notion that each robot executes an instance of the same uniform algorithm module. Modularity permits a higher level of interoperability between different control algorithms and often reduces the complexity of the control algorithm, thus simplifying the implementation. These attributes also improve the efficiency and interoperability of algorithms by permitting computations to execute in parallel across the robot network. Additionally, all robots are considered to be similar if not identical. The algorithm is made robust by ensuring that no single robot plays a role of vital importance or is unique in any way, and each robot is easily replaceable in the case of failure.

In light of these attributes, we advocate an asymmetric broadcast control (ABC) paradigm [17] in which all robots have identical software and receive identical instructions but have the intelligence in the software to differentiate, adopt roles, and perform the required tasks. One or more supervisory nodes serve to provide a degree of centralization by estimating partial global state information about the multirobot system. Such a paradigm is beneficial as we scale up to large numbers of robots, for example, numbers that are characteristic of sensor networks [18]. It becomes necessary to consider approaches to program, command, control, and monitor the robot teams without requiring knowledge of the specifics of the robots and the number of robots in the team. The asymmetry refers to the large volume of information that can be broadcast to the multirobot system relative to the partial state information sensed by or communicated back to the supervisory node.

System Requirements

The motivating design considerations and attributes discussed previously and the need to build a system that is adaptable to a range of multirobot applications lead to the following requirements:

- robust and reliable
- scalable and allows for the easy addition or deletion of agents
- capable of measuring and logging state information (including ground truth) for analysis
- extensible to a variety of applications
- inexpensive
- easy to use and maintain.

Robustness and reliability are of great concern when designing an experimental testbed. Since an assumption is made on the performance of the testbed when evaluating an algorithm, uncharacterized failure modes prevent accurate verification. Scalability is the focus of much of our research and cannot be limited by the system implementation. Measurements, state information, and algorithm status provide insight into the performance of the algorithms being tested and are invaluable during debugging. The ability to access or log such information at run time or for postprocessing is vital to the analysis of any experiment. Extensibility ensures that the testbed can be used to test a wide range of algorithms. By requiring that the system supports applications that demand significant computation, communication, and environmental sensing, we also enable the system to support algorithms that are less demanding but still require verification. With this requirement, we are also able to ensure that we support the many desirable properties previously discussed. The system must be designed to be inexpensive to allow researchers to incrementally increase the size of the system. Ease of use and maintenance is of great concern when the testbed consists of multiple independent units and supports collaborative research with many individuals accessing the system.

Resources for Multirobot Experimentation

Many resources currently exist for multirobot experimentation. We reviewed several hardware and software systems in the context of the system requirements discussed previously for suitability while designing the experimental testbed.

Hardware for Multirobot Experimentation

Robot selection is of crucial relevance when designing an experimental testbed. Since many robots may be used during an experiment, the capabilities, cost, and ease of maintenance are important considerations. The range of applicable algorithms is limited by the capabilities of the robot, particularly in distributed, decentralized, or sensor-rich algorithms, where the robots are expected to perform local computations and manage communication. The cost and ease of maintenance of the robots are relevant when the number of agents is increased or the hardware fails.

We considered many off-the-shelf platforms for indoor experimentation. The solutions we considered were often expensive, commercially unavailable, or did not lend themselves to multirobot experimentation. The three most promising designs were the SwarmBots from iRobot [19], the Khepera III from K-Team [20], and the ER1 from Evolution Robotics [21]. Unfortunately, the SwarmBot is not commercially available. The Khepera III was investigated but was found to have limited computational capabilities. Additionally, the Khepera III requires familiarity with embedded Linux and software that support the necessary cross-compilation requirements. The ER1 was extensively tested but is no longer available as an individual unit. Indeed there were no commercially available mobile robots for less than US$5,000 with the computational capabilities of average laptops, sensors, and networking cards. Recently, iRobot has introduced the economical *Create* robot [22] which comes with actuation and a limited sensor suite. It is the most viable commercially available off-the-shelf solution at present. However, it lacks onboard processing and networking. For this reason, we chose to design and manufacture our custom robot.

Another important element of a multiagent testbed is a localization and ground-truth system. The system must be capable of estimating the pose of tens of robots simultaneously during an experiment. Applicable commercial systems are available including the Vicon MX System [23] from Vicon and Northstar [24] from Evolution Robotics. Although both of these systems were investigated, they were found to be either too expensive or impractical for our needs. Therefore, we developed a custom localization and tracking solution.

Support Software for Robotics

Software for even a single robot is a complex undertaking involving everything from low-level drivers for sensors and actuation up to higher-level computation and reasoning. For systems that integrate large numbers of agents, code modularity becomes even more important as one must also consider communications and networking between many agents. By writing drivers, controllers, and algorithms in a modular fashion, complex systems can be built that reside on a single agent or require the interaction of many modules on many agents.

Given adherence to writing and using modular, reusable code, it is inevitable that some pieces of even a highly customized multirobot system will already exist. This could range from a modern operating system to libraries that provide commonly used algorithms. An attribute by which most available software can be distinguished is licensing; i.e., distributed software is either open or closed source. When considering large teams of agents, the cost of licensing a proprietary operating system and other software can be significant.

Several open- and closed-source software libraries are available that support robotics and generally provide some or all of the following:

- an architecture with commonly defined interfaces so that software modules can be written that encourage good design practices and reuse
- a middleware library that allows both local and networked communication efficiently between modules
- a set of low-level drivers for robotic hardware
- a simulation environment to substitute when hardware is not necessary or available.

As such a system is extremely complex, most choose to not build a home-grown solution. Additionally, selecting an existing system with a large user-base and active development can lead to beneficial collaboration. There are a number of such systems that are currently available.

- *Microsoft Robotics Studio* [25]: Developed recently by Microsoft, this package provides a services-oriented architecture with both a visual programming environment and a physics-based simulator. It relies on proprietary modules to control and connect user-defined software modules in any language supported by Microsoft Visual Studio. This software dictates the use of a closed-source Windows operating system.
- *ORCA* [26]: This project leverages the separately developed Internet Communication Engine [27] middleware, which provides a host of features from a well-supported open-source project including easy interface definitions and tools to manage services, deployment, and event messaging. ORCA is released under the LGPL and GPL licenses and can be compiled on both Linux and Windows operating systems.
- *Open Robot Control Software* [28]: The OROCOS project has focused its development on real-time constraints that are often necessary in industrial robotics applications. OROCOS provides a component system using CORBA as a middleware as well as libraries for kinematics/dynamics and Bayesian filtering.
- *Player/Stage/Gazebo* [29]: Probably the most widely used robotics software package, the Player/Stage/Gazebo (PSG) project consists of libraries that provide access to communication and interface functionality. The robot server Player provides an architecture where many modules (known as drivers) can be independently written and connected through a custom middleware relying on transmission control protocol (TCP) communication. Users are also able to write simpler client applications that can connect to and command modules running on a Player server. Additionally, this project provides a two-dimensional simulator Stage and close collaboration with the three-dimensional physics-based simulation environment Gazebo. These simulators provide the powerful ability to transition transparently from code running on simulated hardware to real hardware. The project is developed for Unix-variant operating systems (e.g., Linux and Mac OS X).
- *Webots* [30]: A simulation environment for mobile robots relying on the open dynamics engine (ODE) [31] for physically accurate models, Webots has the capability of exporting control programs to a few select embedded robotic platforms. It is commercially available for multiple platforms (Windows, Linux, and Mac OS X) in a professional and less-enabled educational version.

We decided to pursue the open-source route, relying on the significant robotics user-base and the potential for growth in this area. Based on this decision, we chose to leverage the existing open-source software developed by the PSG project due to the availability of the three-dimensional physics-based simulation tools and the ability to write and test control software in simulation while moving seamlessly to experimentation with hardware. We also find that this allows us to pursue collaborations with researchers who may not have access to our robots but are able to develop and test software in simulation with models of our robots.

Experimental Testbed Components

The experimental testbed consists of many components that are interfaced together to create the total system. In the discussion that follows, we present the robots, software, and infrastructure of the testbed.

Robots

As stated previously, we chose to design a robot for use in the testbed. The Scarab, a small differential drive robot, serves as the standard platform for multirobot experimentation. Additionally, we designed a cable robot platform, Khepri, which enables interaction with the team of robots as a global observer or aerial vehicle. The design and capabilities of each of these robots are detailed in the following sections.

Scarab Robot

As previously mentioned, we require a robot for indoor experimentation for algorithms that require local sensing, communication, and computation. Additionally, we wish to perform indoor experiments with large teams of robots with a limited experimental space. The robot must also be easily maintained, robust to failures, and economical.

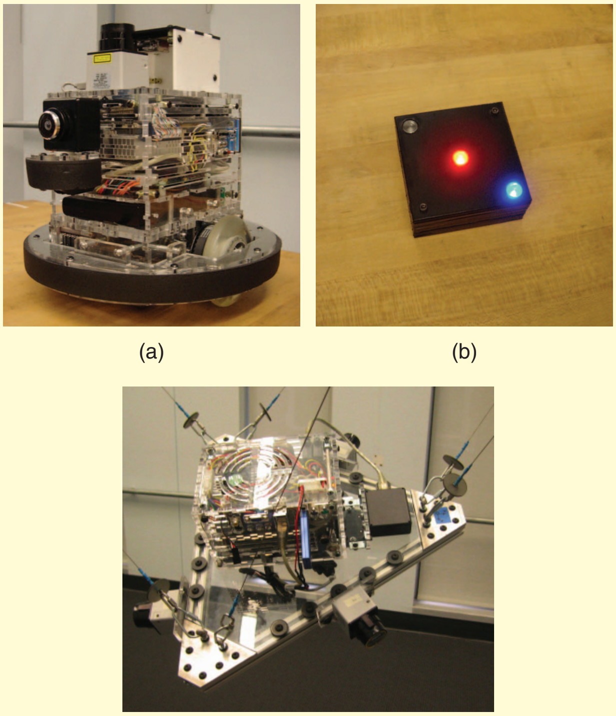

(a)
(b)
(c)

__Figure 1.__ (a) The 20 × 13.5 × 22.2 cm^3 Scarab platform. (b) An LED target is tracked for localization and ground truth on each of the robots. (c) The Khepri robot is controlled by six dc motors via pulleys and cables and has a full suite of sensing and computational abilities, making it well suited for emulation of an unmanned aerial vehicle in indoor environments.

To achieve the above requirements, we developed the differential drive nonholonomic Scarab mobile robot shown in Figure 1(a). The design was completed using computer-aided design tools to be modular, easy to manufacture and assemble, and built from off-the-shelf components (see Figure 2).

Each Scarab is equipped with an onboard computer, power management system, wireless communication, and is actuated by stepper motors. The sensors, actuators, and controllers are modular and connected through the robotics bus [32] (which is derived from the controller area network protocol) or standard interfaces such as the universal serial bus or IEEE 1394. The result is a plug and play system where sensors and actuators can be added or removed from the hardware configuration.

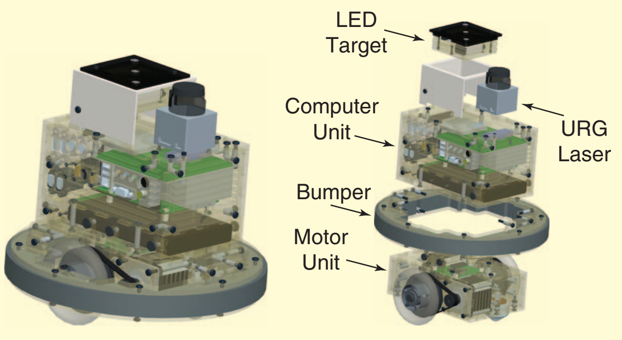
The Scarab robot in Figure 1(a) depicts a typical platform configuration with a Hokuyo URG laser range finder and a Point Grey Firefly IEEE 1394 camera. This image also depicts the robot's foam bumper that protects the robot and allows it to interact with its environment. The physical dimensions of the robot in this configuration (less the bumper) are 20 × 13.5 × 22.2 cm^3 with a mass of 8 kg.

By designing the robot to be manufactured from readily available components and materials, the final cost of the robot shown in Figure 1(a) (without the camera or laser) is less than US$1,500. The end result is modular, easily maintained, and ready for application to a broad range of distributed robotics algorithms.

Khepri Robot

The experimental testbed also includes the Khepri, the aerial robot shown in Figure 1(c). Khepri is a six degree of freedom cable-controlled robot with the same onboard computing module as a Scarab. It is equipped with three Hokuyo URG laser range finders, a three axis inertial measurement unit, and a color Point Grey Dragonfly IEEE 1394 camera. The Khepri's kinematics and actuation system allow it to move in all six directions (positions and orientations), but the workspace is constrained since the cable tensions must be nonnegative [33].

By introducing the Khepri into the testbed, we are able to study interactions between the team of Scarabs and the Khepri



(a)
(b)

__Figure 2.__ Computer-aided design drawings showing the basic components of the Scarab and an exploded view of the robot design with relevant labels.

and consider heterogeneous multirobot applications requiring a supervisor (as in the ABC paradigm). The distributed formation control discussed in the "Experimental Validation" section exemplifies an application requiring a supervisory agent with onboard sensing and computation capabilities.

Software

As discussed previously, we decided to use the open-source software developed by the PSG project. The choice of middleware is crucial in any multirobot testbed. It is the enabling factor that defines the networking and programming frameworks to which all algorithms must adhere or adapt. We have found that the capabilities provided by PSG are sufficiently flexible and transparent that most algorithms are easily accommodated to the framework design. As PSG is open source, modifications to the middleware are straightforward if new features are required.

Two methods exist for interfacing with the robots and sensors within the testbed via PSG (drivers and clients) using a variety of programming languages including C, C++, MATLAB, and Python. A driver is a code module that runs locally on the robot or computers in the testbed and is able to send and receive data to other drivers running locally or across the network. Such a design pattern permits the construction of code modules that run in their own thread and are able to manage both algorithm updates and communications with other robots and other local code modules. By ensuring that algorithms are properly programmed as drivers with strong interfacing, we are able to create identical reusable code modules for use on an individual robot, computer, or a large team of robots. The PSG client is an application that communicates with a driver but does not publish data accessible to other drivers. Generally in our system, clients serve as a simple way to interact with the robots.

Each experiment is defined by configuration files that are loaded by the Player server at runtime. These configuration files determine which code modules each robot or computer runs as well as the communication links required between the agents.

Since the system is distributed across many robots and computers, all information is generated and computed locally. However, a paradigm that requires global information can be implemented by writing a code that uses shared memory (often as a client). As our research interests pertain to distributed and decentralized algorithms, we generally write modules that operate asynchronously across the system without shared memory and with access only to information acquired locally or from network communications.

Infrastructure

Instrumentation for Localization and Ground Truth

We have developed a ground truth verification system consisting of a target with LED markers shown in Figure 1(b) and a network of overhead IEEE 1394 Point Grey Color Dragonfly cameras. Each marker contains three LEDs that flash an 8 B identification pattern that is detected and tracked by the overhead cameras to provide pose information. Measurements from multiple cameras are fused with an extended Kalman filter (EKF) to provide pose and uncertainty estimates for each robot in a global reference frame. For further refinement, an EKF runs on each of the robots, incorporating local odometry motion and the overhead tracking estimates.

The overhead tracking system allows control algorithms to assume pose is known in a global reference frame, thus eliminating the localization problem. Conversely, the tracking system allows the verification of localization algorithms as ground truth. It is also possible to use the tracking system in lieu of sensors that may be unavailable, such as neighbor sensors or collision avoidance sensors.

By the definition of the blinking pattern, the tracking system is theoretically capable of detecting 64 markers simultaneously. While the system has never been tested at its theoretical limit, it has been successfully used to track tens of robots simultaneously with a position error of approximately 2 cm and an orientation error of 5° at 29 Hz in a $9 \times 6 \times 6$ m^3 room. These values are based on raw data without any filtering either at the source or at the robot. While commercial tracking systems exist with higher accuracy [23], it should be noted that the cost difference between our system and commercial systems is significant. The tracking system consists of IEEE 1394 cameras, computers for image processing, and tracking targets that have a unit cost less than US$50.

Network

Since we need a low-latency network to communicate between agents and controllers with reasonable data rates, we use a dedicated 802.11a wireless network in a frequency range not used by adjacent wireless networks to ensure that we have complete control over the bandwidth available to the robots. We have successfully experimented with tens of computers, robots, and sensors performing data intensive experiments without a noticeable impact on the performance or latency of the network.

Data Logging

A requisite component of an experimental system is logging functionality. The system design permits local or networked data logging, depending on the demands of the experiment. Logging to local storage or mounted network drives on each robot is possible, depending on the space and the logging frequency required. Additionally, since we use PSG, a common logging interface exists that permits networked logging. As each robot communicates with other robots in the system, the same messages are sent to a computer that stores the data for postprocessing. With such a design, we are able to log relevant system information without requiring significant computational overhead from the robots.

Additional Considerations

The robots and the supporting computer infrastructure are networked with a dedicated local area network managed by a server with networked storage and a centralized user database. A user remotely accesses the robots in the same way they would access a desktop computer, and all working repositories and code are mounted via network drives. Since the robots and workstations all use the same x86 computer architecture, the same compiled binaries work on all platforms for easy development. Deployment is simple since the same storage is available on both robots and workstations. By viewing the team of robots as a system of networked computers and using off-the-shelf technology, we are able to effortlessly distribute changes in the code base to all of the robots. Additionally, the dedicated server hosts web server capabilities, a repository for software and documentation, and other data to facilitate research and collaboration.

Simulation and Integration

As mentioned in the section on software, we use the software developed by the PSG project, which defines interfaces for our distributed system and provides communication between the robots. Additionally, Player provides a layer of hardware abstraction that permits algorithms to be tested in simulated three-dimensional Gazebo environments. Indeed all algorithm implementations and experiment designs (for example, those discussed in the "Formation Control" and "Cooperative Manipulation" sections) are identical for simulation and experimentation on hardware.

Gazebo incorporates dynamic interactions between models via ODE. Models of the environment of the local laboratory and hardware (discussed in the section on robots) have been reproduced in a simulated world (see Figures 3 and 4). The robot models accurately reflect the geometric, kinematic, and dynamic descriptions of the local robots used in the hardware implementation. Frictional coefficients and contact models (for environment interaction) have been estimated and incorporated into the simulations.

The robotics middleware (discussed previously) is the key to achieving seamless integration between components. The middleware offers clearly defined interfaces that carry out the following functions:

- permit software to be reused for multiple experiments
- allow new hardware or sensors to be rapidly introduced to the system
- enable tight integration between simulation and the real-world
- facilitate collaboration.

The third and fourth points emphasize the benefit of common middleware and interfaces. By defining a common interface structure, simulation environments (such as Stage and Gazebo) may be enabled to support the interfaces. This allows the code written for a simulation environment to be gracefully transitioned to the hardware. The same code that runs on a local computer in simulation will function in the same way on the robots. Additionally, software written using common robotics middleware allows for collaborations by requiring common interfaces between software.

By integrating the simulation environment into the testbed design and ensuring compatibility between the two, we are able to test both the algorithms and the experiment design. Since the same middleware and code base are used during simulation and experimentation, we are able to test the soundness of the experiment design and isolate possible points of failure or weakness that relate to issues not commonly addressed during algorithm verification, such as communication or memory constraints.

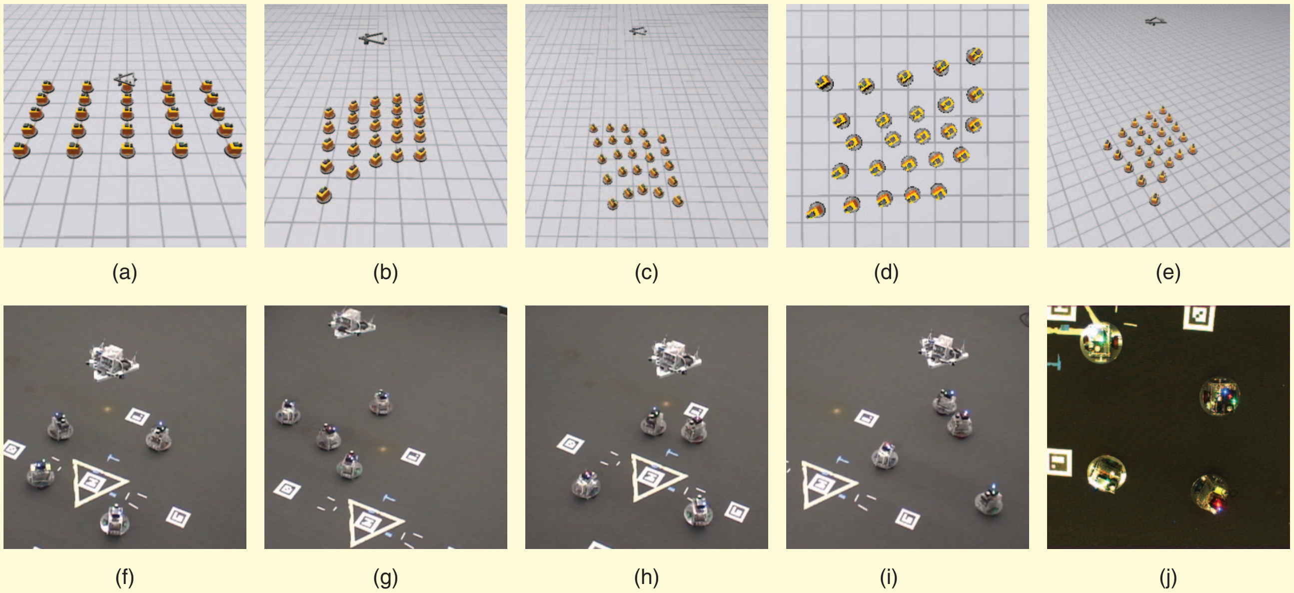


Figure 3. *Results from representative simulation and experimentation runs of formation control: (a)–(e) A representative trial run simulated in Gazebo. (a) The starting formation of twenty-five robots. (b) and (c) The motion of the group given a sinusoidal trajectory on the abstract manifold. (c) and (d) A snapshot of the robots and the corresponding view from the aerial robot's camera. (e) The final formation of the system. (f)–(j) Snapshots that are similar to (a)–(e) but with four Scarab robots. (f) The start configuration. (g) The convergence of the ground robots to* $a^{\mathrm{des}} = \{1, 1, 0.5, 1, 0.5\}$ *(where* $a^{\mathrm{des}} = \{\mu_x, \mu_y, \theta, s_1, s_2\}$*). (h) and (i) The motion of the system to* $a^{\mathrm{des}} = \{1, -1, -0.5, 0.5, 1\}$*. (j) An image from the camera on the aerial robot. The Khepri controls to* $x = \mu_x, y = \mu_y,$ *and* $z = 3.0$ *m or* $z = 1.5$ *m in simulation and experimentation, respectively.*

There are occasions when simulation does not completely capture the behavior of the robots due to differences between reality and the simulated environment. These differences consist of model inaccuracies, simulation approximations, and local rather than distributed communication links. The difference between simulation and experimentation can be particularly significant in experiments involving physical contact between objects where models of frictional contact and the numerical methods for integration need to be more sophisticated than ODE for accurate prediction.

> ***We advocate an asymmetric broadcast control paradigm in which all robots have identical software and receive identical instructions.***

Experimental Validation

In the following discussion, we review recent results in distributed formation control and cooperative manipulation for a team of robots. The discussion emphasizes the implementation of these control algorithms using the experimental testbed.

Formation Control

Formation Control Algorithm

We are interested in controlling the shape, position, and orientation of a formation of a large team of nonholonomic ground robots in a decentralized manner using algorithms that are invariant to the number of ground robots. We briefly present experimental results using the Khepri aerial robot and a team of Scarab robots based on our previous work in [12]–[14]. The central idea is the development of an abstract description of the team of ground robots, which allows the aerial platform to control the team without any knowledge of the specifics of individual vehicles. The abstract description takes the form of a concentration or spanning ellipse defined by its pose $(\mu \in \mathbb{R}^2, \theta \in \mathbb{R})$ and shape $(s_1, s_2 \in \mathbb{R})$ along the major and minor axes. Thus, the pose and shape of the team of ground robots is a point on an abstract manifold. A controller on the abstract manifold yields changes in the abstract state necessary to drive the pose and shape of the formation to its desired value. Consistent with the ABC paradigm, the measured abstract state and the desired changes in the abstract state are broadcast to all of the Scarabs. Individual robot controllers with information about the abstract state (pose and shape) and their own local information ensure that the changes in the abstract state are achieved. Interagent collisions are resolved by constructing local control strategies that do not change the overall abstract state description [14].

Experimental Results and Ramifications

We experimentally validated the control law using the Khepri as a supervisory agent, which estimated the abstract state based on local observations from an onboard camera and performed the necessary computations required to control the abstract state. As seen in Figure 3(j), the Khepri is able to control the gross position and orientation of the formation as well as the shape by simply broadcasting the current observed abstract state, $\mathbf{a} = (\mu_x, \mu_y, \theta, s_1, s_2)$, and the desired abstract state, $\mathbf{a}^{des}$, to the ground robots. The Scarabs receive a broadcast abstract control command from the Khepri derived from its abstract state controller. Each Scarab locally computes its

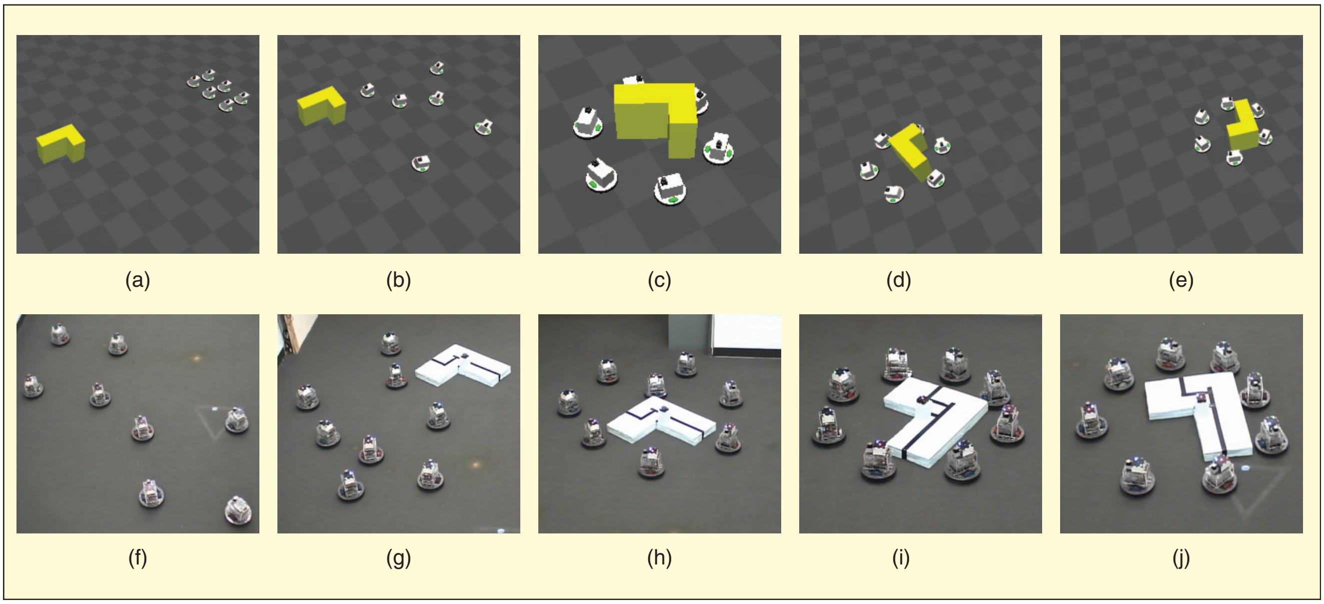

(a) (b) (c) (d) (e)
(f) (g) (h) (i) (j)

Figure 4. *Results from representative simulation and experimentation runs of cooperative manipulation. (a)–(e) The L-shaped cooperative manipulation in Gazebo. The robots all start using the approach controller [(a) and (b)], switch to the surround controller (c), and then to the transport controller [(d) and (e)], thus manipulating the object. (f)–(j) Similar snapshots: approach in (f) and (g), surround in (h), and transport in (i) and (j).*

The Scarab serves as the standard platform for multirobot experimentation.

own control inputs, which are velocities in the horizontal plane, based on this broadcast command as well as its current state and local neighbor measurements (for collision avoidance). A feedback-linearization scheme converts the linear velocities to forward and turning velocities for the nonholonomic Scarab.

These experiments highlight the importance of integrating simulation and experimentation during the implementation process. In simulation, we examined scenarios that required greater local computations and complexity by considering many more agents. We isolated points of fragility in the control algorithm and presented practical solutions to overcome these issues before working with the hardware [14].

Cooperative Manipulation

Cooperative Manipulation Algorithm

In a series of articles and papers, we described our approach to cooperative manipulation, which involves caging the manipulated object and moving while maintaining a condition of object closure [15], [34]. As in the previous subsection and consistent with the ABC paradigm, a geometric description of the manipulated object and the desired reference trajectory for manipulation is obtained by a supervisory agent and is broadcast to the team of Scarabs. Each robot chooses from a suite of controllers (vector fields) each of which is carefully constructed to guarantee properties of interest. For example, an approach controller guarantees that a robot will approach the object to be manipulated, while a surround controller ensures that a robot will go around the object and orbit it [16], [35], [36]. A transport controller allows each robot to move along the reference trajectory while ensuring that the condition of object closure (or caging) is maintained. All controllers guarantee that there will be no collisions. The complexity of the control problem is reduced to the problem of sequentially composing these controllers or vector fields [15]. Since these controllers or vector fields depend only on the object's position and geometric shape and the desired trajectory for the object, the resulting control computations are independent of the number of agents and only require the assumption that the number of agents is sufficiently large to surround the object for caging purposes. The control law is anonymous in that the identification of individual agents is unnecessary and the number of robots can change dynamically.

Experimental Results and Ramifications

While in theory the discrete protocols and continuous controllers are all guaranteed to work, the interaction between the discrete and continuous components and the fact that each robot operates asynchronously necessitates validation through simulation and experimentation. We demonstrated using Gazebo and the testbed that the sequential composition of the three behaviors, approach, surround and transport, which involves switches between these behaviors, is robust to both the type of object being manipulated and the number of robots available for manipulation. On real hardware, we have conducted tens of trials with four to eight robots manipulating an object along linear and sinusoidal trajectories as shown in Figure 4, as well as along trajectories obtained from a navigation function.

Through simulation and experimental trials, we demonstrated that the environment models in Gazebo mirrored reality to a sufficient degree that we returned to simulation and assessed large sets of initial conditions and parameters for testing and analysis. Such hardware and software integration lead to a significant speedup in the experimental process.

Conclusions

In this article, we presented our experimental testbed for a large team of robots and sensors, describing the hardware, software, and infrastructure for experimentation as well as the rationale for the design choices. In addition, we discussed our framework for developing software and some experimental results from recent studies. Our testbed enables us to validate distributed robotics algorithms for large numbers of robots engaged in a variety of tasks including formation control, search and pursuit of targets, and cooperative manipulation. This work also highlights a major benefit of selecting Player and Gazebo as an enabling mechanism to evaluate distributed robotics algorithms in simulation and on real robots. While our main focus in this article was on control algorithms, we intend to develop algorithms and software for distributed estimation and mapping from onboard sensors and look forward to reporting these advances in the future.

The application of multirobot theory to real-world scenarios requires the consideration of many challenging details that increase the complexity of implementation. It is clear that relaxing the assumptions of point models, Euclidean dynamics, and synchrony for multiagent systems is nontrivial. Further, multiagent systems require significant hardware, software, networking, and infrastructure support. To surmount these issues as multiagent systems scale in complexity and size, we advocate a close integration of high-fidelity simulation and experimentation and a carefully designed testbed that is constructed of robust, modular, and inexpensive components.

Acknowledgments

This research was supported by NSF grants CCR02-05336, NSF IIS-0413138; and IIS-0427313; ARO Grants W911NF-04-1-0148 and W911NF-05-1-0219; and ONR Grant N00014-07-1-0829.

Keywords

Multirobot systems, experimental robotics, decentralized control, formation control, cooperative manipulation.

References

[1] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 24–33, Oct. 2004.

[2] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 6, pp. 777–790, Nov. 2001.

[3] L. E. Parker, "Alliance: An architecture for fault tolerant multi-robot cooperation," *IEEE Trans. Robot. Automat.*, vol. 14, no. 2, pp. 220–240, Apr. 1998.

[4] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky, "A decentralized architecture for active sensor networks," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, New Orleans, LA, Apr. 2004, pp. 1097–1102.

[5] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and B. Southall, "A framework and architecture for multi-robot coordination," *Int. J. Robot. Res.*, vol. 21, nos. 10–11, pp. 977–995, Oct. 2002.

[6] D. Cruz, J. Mcclintock, B. Perteet, O. A. A. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control—A multivehicle platform for research in network embedded systems," *IEEE Control Syst. Mag.*, vol. 27, no. 3, pp. 58–78, Jun. 2007.

[7] M. Valenti, B. Bethke, G. Fiore, and J. How, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," presented at the *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Keystone, CO, Aug. 2006.

[8] R. Bachmayer, N. E. Leonard, J. Graver, E. Fiorelli, P. Bhatta, and D. Paley, "Underwater gliders: Recent developments and future applications," in *Proc. Int. Symp. Underwater Technology*, Taipei, Taiwan, Apr. 2004, pp. 195–200.

[9] R. C. Arkin, *Behavior Based Robotics*. Cambridge, MA: MIT Press, 1998.

[10] R. G. Simmons, S. Singh, D. Hershberger, J. Ramos, and T. Smith, *First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly* (Lecture Notes Series in Control and Information Sciences). New York: Springer-Verlag, 2000, vol. 271, pp. 323–332.

[11] J. T. Feddema and D. Schoenwald, "Decentralized control of cooperative robotic vehicles," *Proc. SPIE*, vol. 4364, pp. 136–146, Sept. 2001.

[12] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Trans. Robot*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[13] N. Michael, C. Belta, and V. Kumar, "Controlling three-dimensional swarms of robots," in *IEEE Int. Conf. Robotics and Automation*, Orlando, FL, May 2006, pp. 964–969.

[14] N. Michael, J. Fink, and V. Kumar, "Controlling a team of ground robots via an aerial robot," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, CA, Oct. 2007, pp. 965–970.

[15] J. Fink, N. Michael, and V. Kumar, "Composition of vector fields for multirobot manipulation via caging," presented at *Robotics: Science and Systems*, Atlanta, GA, Jun. 2007.

[16] M. A. Hsieh and V. Kumar, "Pattern generation with multiple robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, FL, May 2006, pp. 2442–2447.

[17] N. Michael, J. Fink, S. G. Loizou, and V. Kumar, "Architecture, abstractions, and algorithms for controlling large teams of robots: Experimental testbed and results," presented at *Int. Symp. Robotics Research*, Hiroshima, Japan, Nov. 2007.

[18] K. Whitehouse, "The design of calamari: An ad hoc localization system for sensor networks," M.S. thesis, Dept. Elect. Eng. Comp. Sci., Univ. California, Berkeley, 2002.

[19] J. McLurkin, "Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots," M.S. thesis, Dept. Elect. Eng. Comp. Sci., MIT, Cambridge, MA, 2004.

[20] K-Team. [Online]. Available: http://www.k-team.com

[21] ER1 Personal Robot System. [Online]. Available: http://www.evolution.com/er1

[22] iRobot Create Programmable Robot. [Online]. Available: http://www.irobot.com/create

[23] Vicon MX Systems. [Online]. Available: http://www.vicon.com/products/viconmx.html

[24] NorthStar. [Online]. Available: http://www.evolution.com/products/northstar

[25] Microsoft Robotics Studio SDK. [Online]. Available: http://msdn.microsoft.com/robotics

[26] A. Brooks, T. Kaupp, A. Makarenko, S. B. Williams, and A. Oreback, *Software Engineering for Experimental Robotics* (Springer Tracts Series in Advanced Robotics). Berlin, Germany: Springer-Verlag, 2007, vol. 30, pp. 231–251.

[27] Internet Communications Engine. [Online]. Available: http://www.zeroc.com/ice.html

[28] H. Bruyninckx, "Open robot control software: The OROCOS project," in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, May 2001, vol. 3, pp. 2523–2528.

[29] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Int. Conf. Advanced Robotics*, Coimbra, Portugal, Jun. 2003, pp. 317–323.

[30] Webots*, Fast prototyping and simulation of mobile robots*. [Online]. Available: http://www.cyberbotics.com/products/webots

[31] Open Dynamics Engine. [Online]. Available: http://www.ode.org

[32] D. Gomez-Ibanez, E. Stump, B. Grocholsky, V. Kumar, and C. J. Taylor, "The Robotics Bus: A local communcations bus for robots," *Proc. SPIE*, Ser. XVII, vol. 5609, pp. 155–163, Dec. 2004.

[33] E. Stump and V. Kumar, "Workspaces of cable-actuated parallel manipulators," *ASME J. Mech. Des*, vol. 128, no. 1, pp. 159–167, Jan. 2006.

[34] G. A. S. Pereira, M. F. M. Campos, and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *Int. J. Robot. Res.*, vol. 23, nos. 7–8, pp. 783–795, July 2004.

[35] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling swarms of robots using interpolated implicit functions," in *Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 2487–2492.

[36] M. A. Hsieh, S. G. Loizou, and V. Kumar, "Stabilization of multiple robots on stable orbits via local sensing," in *Proc. IEEE Int. Conf. Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 2312–2317.

Nathan Michael is currently pursuing a Ph.D. degree in mechanical engineering at the University of Pennsylvania, Philadelphia. His research interests include control and estimation for multirobot systems and experimental robotics.

Jonathan Fink received a dual B.S. degree in electrical and computer systems engineering at Rennselaer Polytechnic Institute, New York, in May 2004. He is currently pursuing a Ph.D. degree in electrical and systems engineering at the University of Pennsylvania, Philadelphia. He is a member of the GRASP Lab of the University of Pennsylvania. His research interests include algorithms for multirobot systems with sensor network and manipulation applications.

Vijay Kumar received his M.Sc. and Ph.D. degrees in mechanical engineering from The Ohio State University, Columbus, in 1985 and 1987, respectively. He has been on the faculty in the Department of Mechanical Engineering and Applied Mechanics with a secondary appointment in the Department of Computer and Information Science at the University of Pennsylvania, Philadelphia, since 1987. He is currently the UPS Foundation professor and the chairman of mechanical engineering and applied mechanics.

Address for Correspondence: Nathan Michael, University of Pennsylvania, Philadelphia, PA 19104-6228, USA. E-mail: nmichael@grasp.upenn.edu.

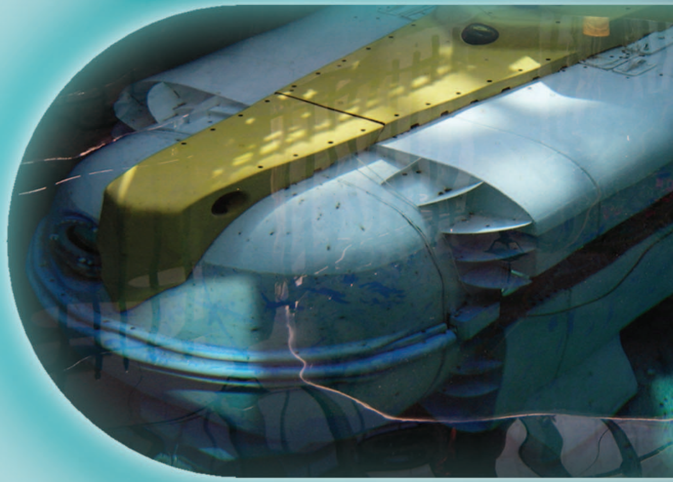
Heterogeneous Wireless Multirobot System

A Platform for Safety and Security in Distributed Communication Computing and Control

BY ANTONIO BICCHI, ANTONIO DANESI,
GIANLUCA DINI, SILVIO LA PORTA,
LUCIA PALLOTTINO, IDA M. SAVINO,
AND RICCARDO SCHIAVI

© DREAMSTIME

© DREAMSTIME



© ISTOCKPHOTO

Mobile Multirobot Systems

The convergence of communication, computing, and control is considered by many the future of information technology [1], [2]. This will provide the ability for a large number of sensors, actuators, and computational units interconnected, wirelessly or over wires, to interact with the physical environment. One of the main expected consequences of such convergence is the possibility to create large systems of many autonomous and interconnected units, which have capabilities of not only sensing [3] but also acting in and on the environment. Several research challenges raised by multiple autonomous mobile systems are stimulating a keen interest in the robotics research community, such as formation control and flocking (e.g., [4]–[6]), coordination (e.g., [7]–[9]), communication problems and protocols (e.g., [10]), and algorithm distribution (e.g., [1], [11]). These advances form the basis for addressing the application of multiagent robotic systems outside the labs in new scenarios, ranging from the exploration of unknown environments to surveillance, patrolling, and so forth.

In this article, we consider a scenario in which a group of vehicles move autonomously in a shared environment. Each vehicle is given a specific task to accomplish, on its own or in collaboration, such as monitoring the environment, reconstructing a map, searching for an object, or detecting light or heat sources. Agents can join or leave the group dynamically. Typical agents are inexpensive, unmanned vehicles equipped with embedded sensor systems with limited onboard processing units and short-range wireless communication capabilities. Contrary to what is often assumed in the current state of art, we accept the realistic requirement that the platform must accommodate for broadly heterogeneous vehicles in terms of different tasks to accomplish, different onboard sensors and computational capabilities, and also different dynamics or dimensions.

In particular, we focus on four crucial requirements on the design of a component-based platform for multiagent systems, which are safety, scalability, security, and reconfigurability. In our context, safety means that motions of the robots are executed so that any collision among them is avoided while they attend to their tasks. The need to manage a possibly large number of vehicles imposes scalability of the platform. An immediate consequence of this requirement is that solutions using a centralized traffic supervisor dispatching detailed instructions to all vehicles are unacceptable.

To fulfill their tasks, including collision avoidance, cooperative vehicles have to communicate, and ad hoc wireless network

Digital Object Identifier 10.1109/M-RA.2007.914925

1070-9932/08/$25.00©2008 IEEE

technologies are apparently good candidates to provide support for such architecture. Protecting wireless communication poses unique challenges. Unlike traditional wired networks, an adversary equipped with a simple radio receiver or transmitter can easily eavesdrop communication and inject or modify packets. Furthermore, to make them economically viable, embedded devices are often limited in their energy, computation, storage, and communication capabilities, and this leads to constraints on the types of security solutions that can be applied. To address these challenges, the platform should support security requirements in terms of secrecy, integrity, and authenticity of communications with respect to a potential active outsider.

A practical use of the platform also imposes the reconfigurability requirement. Because of their tasks, vehicles typically operate in critical environments that are subject to unpredictable changes of operational conditions. This requires the multiagent systems application to be able to reconfigure itself so as to meet the changing conditions. The proposed platform supports reconfigurability at different levels. At a physical level, vehicles may dynamically join and leave the group. Hence, candidate new members of the group can be accepted if safety is not compromised. At a logical level, a vehicle may need to reprogram tasks and the implementation of a given service because of the changed operational conditions. Also, reprogramming must not endanger the security of the vehicle software platform and thus of the whole platform.

The platform described in this article has been designed and implemented to fulfill the requirements described earlier. In particular, for the heterogeneity requirement, the platform must also be accessible to very simple vehicles with possibly low computational and data storage capabilities. To realize a platform that can deal with a larger class of mobile robots (from the very simple to the advanced-technology examples), services have been implemented taking into account possible technological limitations of the vehicles involved in the scenario.

Platform Architecture

In this section, the component-based agent architecture is described. A component is an encapsulated unit of functionality and deployment that provides services through its interface. This architecture abstracts away from the actual platform implementation and provides a general design framework for multiagent systems. Furthermore, the component-based approach supports and promotes encapsulation and modularity of design and implementation and thus makes it possible to integrate vehicle with hardware and software of completely different origin and make them safely and securely coexist and collaborate. In addition, if the basic runtime software allows it, components can be dynamically added and removed. This makes it possible to retask a robot and change the implementation of a service according to the changing operational conditions.

Agent Architecture

In Figure 1, the architecture of an agent in terms of its constituting components and their relationships is reported. Components are of different types. The network component provides network services for sending and receiving packets. The application components implement the task the agent has to fulfill. In the rest of this article, we abstract away application components with the component application in Figure 1 and do not specify them any further.

The control components and the security components allow a vehicle to access to, or even participate to the implementation of, the services described in the "Architecture Implementation" section. More precisely, the control components comprise the collision avoidance component (CAC), the self-localization component, and the neighbor-localization component and deal with collision avoidance and vehicle localization, respectively. The security components comprise the security controller component (SCC), the rekeying component, and the authenticated loading component (ALC) and deal with secure communication, rekeying, and secure software reconfiguration, respectively. Finally, the actuation components deal with the actual actuation of the motion commands issued by the CAC.

In the rest of the section, we provide a detailed description of components, interfaces, and services provided. We describe components interfaces, in a language-neutral interface description language (IDL). For each operation provided by an interface, the IDL allows us to specify the name and the type of both the arguments the operation takes as input (in parameters) and the values the operation returns as output (out parameters).

1) CAC: The CAC coordinates the motion of agents preventing collisions and guaranteeing that each agent eventually reaches the final configuration required by its task, providing to the agent a collision avoidance maneuver.

The component implements the following interface:

```
Interface ICollision{
  setParameters(in Parameters p);
  getParameters(out Parameters p);
  join(in Name n, in Configuration initial, in
Configuration final);
  leave(in Name n);
}
```

The operation `setParameters` initializes parameters necessary for the correct execution of each collision avoidance

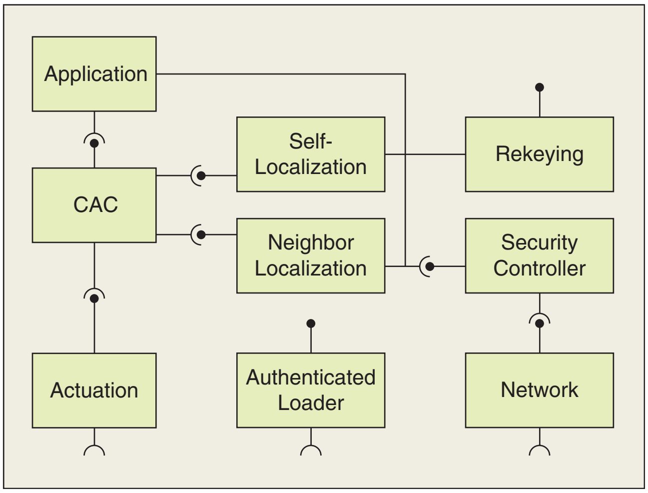


Figure 1. *The software architecture.*

algorithm the component implements. The actual implementation of `Parameters` depends on the specific collision avoidance algorithm (e.g., dimension and speed of the vehicle). The operation `getParameters` returns the current value of parameters. The operation `join` makes a vehicle named `n` to join the group. The operation takes the initial and the final configuration of the vehicle as input arguments. Finally, the operation `leave` makes vehicle named `n` to leave the group.

The CAC provides two services: the collision avoidance service that guarantees that no collision will occur among vehicles belonging to the group and the group membership service (GMS) that guarantees that every new joining does not endanger the safety property.

For every vehicle, the collision avoidance service requires both the current configuration of the vehicle and the configurations of its neighboring vehicles. The self-localization service provides the former information whereas the neighbor-localization service provides the latter. Such services are provided by the two following components, respectively.

2) Self-localization component: The self-localization component provides the agent with data about its own position. The component provides the following interface:

```
Interface ISelfLocalization{
  getSelfPosition(out Configuration c);
}
```

Operation `getSelfPosition` returns the current agent configuration.

3) Neighbor-localization components: The neighbor-localization component provides each vehicle with configurations of neighboring vehicles. The component provides the following interface:

```
Interface INeighbourLocalization{
  getParameters(out Parameters[ ] p);
}
```

The operation `getParameters` returns the parameters of all neighbors necessary for the correct execution of each collision avoidance algorithm.

4) Actuation component: The actuation component sets the desired linear velocity and angular velocity of the vehicle. The implementation of this component is strictly related to the dynamic of the vehicle. The component provides the following interface:

```
Interface IActuation{
  set(in int8 aVelocity, in int8 lVelocity);
}
```

Operation `set` sets as angular and linear velocity the values specified by the `aVelocity` and `lVelocity` arguments, respectively.

The SCC fulfills the communication security requirements in terms of confidentiality, integrity, and authenticity. The component implements the same interface as the network component:

```
Interface INetwork{
  send(in Message m, in Address a)
  receive(out Message m, out Address a)
}
```

By doing so, the SCC can be inserted and removed without affecting the other components. This allows us to reconfigure the software architecture by inserting the SCC only when needed.

Operationally, the SCC intercepts incoming or outgoing messages and applies to them the cryptographic transformations specified by the secure communication protocol. The actual specification and implementation of the protocol depends on several factors including the kind of embedded computing device and the hardware and software platform on which the SCC is deployed. The component can be implemented via software. However, if an hardware cryptographic device is present, the component can encapsulate and abstract the cryptographic services offered by that device.

5) The rekeying component: The rekeying component performs key distribution and revocation and updates the key repository on the vehicle. Usually, the keys are distributed in such a way that both confidentiality and authenticity are guaranteed. Operationally, the rekeying component receives a new key and performs the cryptographic transformation specified by the rekeying protocol to guarantee the key confidentiality. If required, it also verifies that the key comes from a trusted part.

In distributed rekeying protocols, the vehicles could generate the keys and securely transmit them to other nodes. In this case, the rekeying component provides the following interface:

```
Interface IRekeying{
  void renewKey(in Key k, in KeyValue v);
}
```

The operation `renewKey` renews key `k` with the new value v. The implementation of `Key` and `KeyValue` depends on the actual implementation of the rekeying protocol.

6) The ALC: On downloading a new software component through the network, a vehicle needs a proof that the component comes from a trusted source (component authenticity) and that the component has not been modified (component integrity). The ALC downloads a component from a remote trusted source, buffers the component during the downloading, verifies the component authenticity, and finally loads the component into memory from the buffer. The ALC can also guarantee the component confidentiality, if necessary. This component provides the following interface:

```
Interface IAuthLoad{
  load(in String cname, out ComponentType t);
}
```

The operation `load` downloads the component whose name is specified by the string `cname` and returns the component type.

Architecture Implementation

In this section, we present our implementation of the architecture services described in the "Agent Architecture" section.

Collision Avoidance Service

The collision avoidance service is one of the two services offered from the CAC. It coordinates the motion of vehicles within the group, preventing collisions and guaranteeing that each vehicle eventually accomplishes its individual task. The service implementation is based on a decentralized collision avoidance policy, called generalized roundabout policy (GRP), that has been recently proposed for vehicles evolving on the plane [14]. The GRP is now briefly reported for the reader's convenience. However, a complete, formal, and detailed description of it can be found in the cited literature with references to other existing decentralized conflict avoidance approaches.

Since we are dealing with heterogeneous vehicles, we consider vehicles with nontrivial kinematics; for example, they are not able to stop their motion and have nonholonomic constraints. Those assumption are not restrictive since vehicles that are able to stop or are holonomic can always perform trajectories obtained with the proposed policy. Indeed, we consider a number of vehicles moving on the plane at a constant speed, along paths with bounded curvature. The state of each vehicle is represented by the coordinates (x, y) and the heading angle θ. According to the policy, a first circle is assigned to each vehicle, called the safety disk, being the circle centered at the vehicle position (x, y) with heading given by θ. A collision is said to occur whenever two or more safety disks overlap.

As mentioned earlier, the proposed policy also applies to vehicles that cannot stop their motions. For dealing with such a case, the policy defines a reserved disk for each vehicle as the circle that contains the path traveled by the safety disk when its associated vehicle turns right at the minimum allowable curvature. The center of a reserved disk can easily be obtained from its vehicle state, and its heading is directly inherited from that of the corresponding vehicle. In spite of the vehicle constraint, the motion of the reserved disk can be stopped at any time by making the vehicle turn right at the minimum curvature rate.

Referring to Figure 2, suppose that each vehicle has to reach a desired final position and heading to accomplish its task. The motion strategy followed by the vehicle is based on four distinct modes of operation, each assigning a suitable value to the control input (i.e., curvature rate) of the vehicle. Each vehicle enters the straight mode if the motion along the line directed toward the desired configuration is permitted, that is, a motion in that direction does not cause an overlap with other reserved disks. Whenever its reserved disk becomes tangent to the one of another vehicle, a test is made based on the current motion heading θ. If a further movement in the direction specified by θ causes an overlapping, then the vehicle enters the hold mode. Otherwise, the vehicle is able to proceed and remains in the straight mode. When the hold mode is entered, the vehicle's curvature rate is set to the minimum allowable, and the motion of its reserved disk is stopped. As soon as the vehicle heading is permitted but not directed toward the target destination, the vehicle enters the roll mode and tries to go around the other reserved disk. This is achieved by selecting a suitable value for the curvature rate of the vehicle such that the two disks never overlap. An example of possible trajectory of a vehicle that moves according to the GRP is pictorially depicted in Figure 2.

The proposed policy satisfies the safety and scalability requirements [14]. In particular, the decentralized characteristic of the policy allows the CAC to be implemented on board of the vehicle. As a matter of fact, each vehicle is able to make a safe decision about its motion, based only on the locally available information. This information consists of the position and orientation of vehicles that are within a certain sensing or communication radius. For this reason, each vehicle communicates its state via the wireless network, though it is not required to explicitly declare its task or goal. The policy can be easily adapted in case of nonexact information on neighbors by enlarging the reserved disk radius and relaxing switching conditions between modes.

Group Membership Service

The second service offered by the CAC is the GMS. The motion strategy described in the "Collision Avoidance Service" section guarantees that no collision will occur among vehicles belonging to the group (safety property) and that all the vehicles eventually reach their final destinations (liveness property). These two properties are guaranteed provided that initial and final vehicles' configurations satisfy suitable conditions. In particular, safety is obtained if the vehicles' reserved disks do not initially overlap whereas liveness is guaranteed if the vehicles' destinations are not concentrated in the plane. Details on the target sparsity requested for liveness can be found in [14].

Taking into account the fact that vehicles can dynamically join or leave the group, the GMS purpose is to guarantee that such conditions are never violated. Thus, on joining, a new vehicle sends the configuration of its reserved disk to the GMS that verifies whether its entrance may compromise the overall system safety and liveness (join phase). Later, on reaching its final destination, the vehicle leaves the group and alerts the GMS that cancels its data (leave phase). This event may allow new vehicles

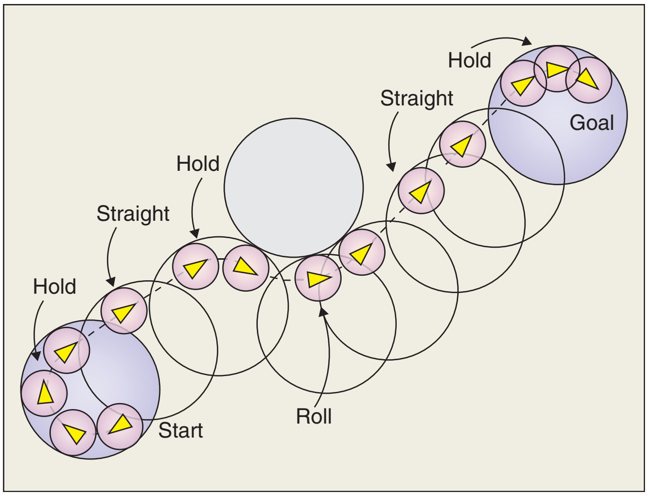


Figure 2. *Example of possible trajectory of a vehicle applying the proposed collision avoidance policy. Smaller circles are safety disks and larger circles are reserved disks.*

to enter the group. The GMS must be managed by a centralized server as the conditions guaranteeing safety and liveness are based on the information provided by all the vehicles.

Self-Localization Service

The self-localization component is responsible for providing the vehicle with the information about its own state. This component can be implemented in several different ways depending on the sensors mounted onboard the vehicle. In case of very simple vehicles, a possible implementation is a set of cameras that monitor the environment and detect position and direction of motion for all the vehicles. The data are collected in a centralized server that will send to any vehicle through the wireless network its own state information.

Otherwise, if some of the vehicles have some localization sensors, the component may have an onboard implementation based on the particular sensors. For example, in case of sonar, the localization component may be implemented as it has been proposed in [15].

Secure Communication Service

The secure communication service is provided by the SCC on board of vehicles that implements a secure communication protocol. Conceptually, a secure communication protocol is defined as a set of rules, each of which consists of a transformation, a cryptographic suite, and a set of selectors. A transformation specifies the set of cryptographic processing to be applied to messages before or after sending or receiving them to/from the network. A transformation can be either a cryptographic primitive or a combination of primitives. Cryptographic primitives can use cryptographic keys. A cryptographic suite specifies the actual cryptographic primitives, and the related keys, to be used in a transformation. Keys are specified by a key unique identifier. Finally, selectors make it possible to specify which messages a transformation has to be applied to. Selectors include at least the type of message (e.g., the port), the destination address, the source address, whether the message is incoming or outgoing, and so forth.

For example, let us assume that both confidentiality and integrity must be guaranteed for messages addressed to port p. One way to achieve these goals is to hash and encrypt the message according to the following transformation $t : \mathrm{E}(m\|\mathrm{H}(m))$, where E specifies symmetric encryption, H specifies hash, and $\|$ is the concatenation operation. For example, if we would like to use the hash function SHA-1 [16] and the symmetric cipher RC5 [17] keyed by the K key, then we specify the cryptographic suite $c : \langle \mathrm{e} = \mathrm{RC5}, \mathrm{keyid} = K; \mathrm{H} = \mathrm{SHA} - 1\rangle$. Finally, selectors specifying that the relevant messages are addressed to port p are $s : \langle \mathrm{port} = p, \mathrm{direction} = \mathrm{outgoing}\rangle$. It follows that the secure communication protocol is specified by the rule $\langle s, t, c\rangle$.

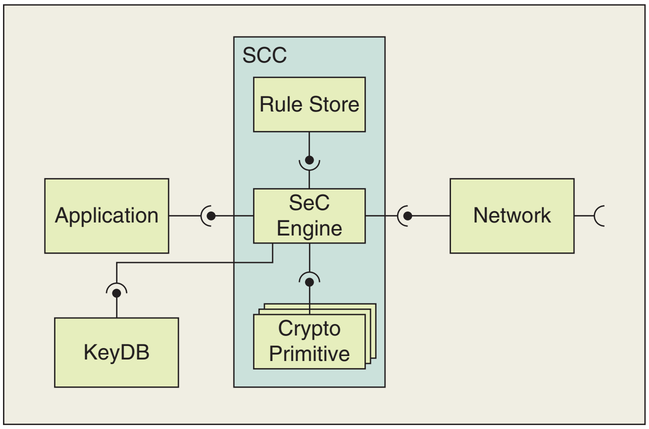


Figure 3. *The secure communication component.*

SCC is itself conceptually structured in components as specified in Figure 3. When the application sends or receives a message through network, SecEngine intercepts the message, retrieves the rule whose selector matches the message from the RuleStore, and applies the corresponding transformations to the message using the specified cryptographic suite implemented by CryptoPrimites. If the performed algorithms need cryptographic keys, SCC retrieves them from the KeyDB component.

Rekeying Service

The rekeying service is managed by the centralized rekeying server (RKS). So, the vehicles only have to verify the authenticity and the freshness of the received keys. That is, they have to verify that the key comes from the RKS and has not been used before, respectively.

Rekeying may occur either periodically, as requested by good cryptographic practices, or on events such as the leaving of a vehicle. So, GMS has to inform RKS that a vehicle leaves the group communication so that RKS distributes the new group key to all vehicles except the leaving one. The scalability of the rekeying service depends on the chosen rekeying protocol.

In our implementation, we chose $\mathrm{S}^2\mathrm{RP}$, the secure and scalable rekeying protocol for devices with low-computational capabilities [18]. $\mathrm{S}^2\mathrm{RP}$ guarantees the key authenticity by using only one-way hash functions that are computationally affordable even by the simplest devices. In short, the key authentication mechanism levers on keychains, a technique based on the Lamport's one-time passwords. A keychain is a set of symmetric keys so that each key is the hash preimage of the previous one. Hence, given a key in the keychain, anybody can compute all the previous keys, but nobody can compute any of the next keys. Keys are revealed in the reversed order with respect to creation. Given an authenticated key in the keychain, the vehicles can authenticate the next keys by simply applying an hash function.

To reduce the communication overhead, RKS maintains a tree structure of keys according to $\mathrm{S}^2\mathrm{RP}$ (Figure 4). Each internal node is associated with a keychain, whereas each leaf is associated with a vehicle. More in detail, a leaf is associated with the symmetric vehicle-key that the corresponding vehicle secretly shares with RKS. Let us refer to the last-revealed key associated with the node j as K_j and to the hash preimage of K_j as K_j^{nxt}. Each vehicle stores the key K_j if the subtree rooted at the node j contains the leaf associated with the vehicle-key. Hence, the key K_1 associated to the tree root is shared by all group members and it acts as the group key. Let us suppose vehicle D leaves the group. All its keys K_j with $j \in \{1, 2, 5\}$ are considered compromised and RKS has to securely broadcast the new keys K_j^{nxt} with $j \in \{1, 2, 5\}$. The rekeying messages are shown in the right-hand side of Figure 4, where $E(K, K^\star)$ is the encryption of key $K^\star$ by using the key K. So, the rekeying protocol is scalable because RKS has to broadcast

$\mathcal{O}(\log n)$ messages where n is the number of vehicles. The rekeying component constitutes the vehicle side of the rekeying service. The component is implemented by an security controller component SCC and an AuthKey component (Figure 5). SCC performs the secure communication and is responsible for guaranteeing the key confidentiality whereas the AuthKey component performs the check required by the keychain for verifying the key authenticity.

Authenticated Loading Service

The authenticated loading service is managed by the centralized authenticated loading server (ALS) that is responsible for guaranteeing the component authenticity.

A typical approach to authenticate a component on downloading consists in authenticating it as a whole. However, this approach requires that the component is entirely received before being verified, and this can be exploited by an adversary to mount a denial of service attack. More in detail, an attacker can make the device waste resources by causing it to buffer long strings of bytes that in the end fail the authenticity verification. An alternative approach is based on the observation that a component is typically transmitted in several packets [12]. If every packet is authenticated, a device stores only authenticated material and reduces the risk of denial of service at minimum. Nevertheless, this solution introduces overhead as now each packet needs to be authenticated.

A trade-off between security and performance can be achieved by authenticating bursts of packets. A burst contains a fixed predefined number N_B of packets. If N is the total number of packets conveying the components, N_B is comprised between 1 and N. Each burst is linked to the one that will be transmitted next by a hash function. ALS computes the hash of each burst and transmits the result with the previous burst. The hash value associated with the last transmitted burst is filled with the null value. It follows that, if the vehicle can authenticate the first burst, then it can sequentially authenticate all the subsequent bursts. On receiving a burst, the vehicle computes the hash and compares it with the hash value conveyed by the previously received burst. If the two values are equal, the received burst is authentic.

The authenticity of the first burst must be proven in a different way. In a scenario with many vehicles equipped with reduced computing capabilities, the digital signature might not be efficient. Therefore, we chose to prove the authenticity of the first burst by means of a message authentication code (MAC) computed with the pairwise key that the vehicle secretly shares with ALS or with the group key (see "Rekeying Service" section). Let us consider the example in Figure 6. Given $N = 6$ and $N_B = 3$, each burst contains two component-packets and the hash of the next burst. The first burst also contains an authenticator constituted by a MAC computed with the current group key. In this method, particular attention must be paid to whether an adversary can capture a device or not. Whenever a device is suspected of being compromised, the rekeying service has to revoke and then redistribute the group key to every device except the suspected one.

It follows that the proposed authentication scheme is efficient in that it requires $1 \leq N_B \leq N$ hash function computations and the authentication of the first burst. It is also flexible in that the value of N_B and the authentication method for the first burst (MAC or digital signature) are design parameters. It is also worthwhile to notice that the proposed scheme does not negatively affect scalability especially if one considers that the choice of the design parameters is not influenced by the number of vehicles to which a component has to be sent. Of course, a component could be potentially broadcast to all vehicles. In this case, the broadcast protocol is crucial for scalability. However, this is a general problem that is beyond our interests and is not addressed in this article. If necessary, we will resort to proposals in the literature [21].

The ALC constitutes the vehicle side of the authenticated loading service. In principle, the component includes an SCC and an AuthComp component (Figure 7). The SCC performs the secure communication protocol aimed at protecting packets

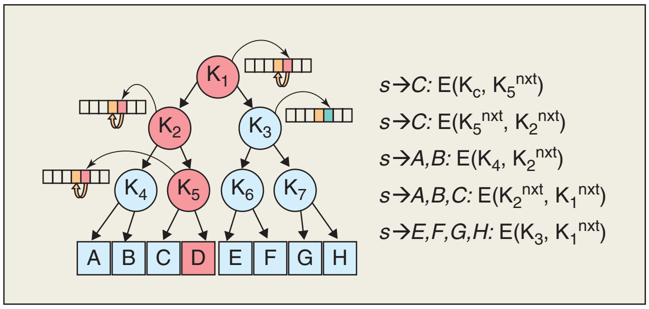


Figure 4. *Hierarchical structure of key chains in* S^2RP.

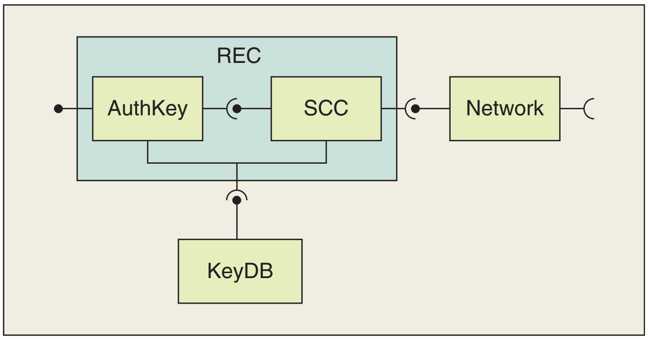


Figure 5. *The rekeying component.*

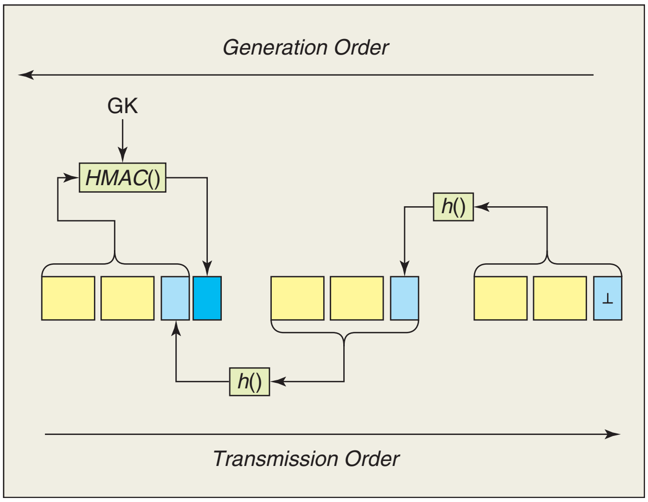


Figure 6. *A chain of bursts.*

carrying the component, whereas the AuthComp component is responsible for component buffering and authentication.

Platform Prototype

This section describes a platform prototype that has been realized according to the proposed architecture. The platform is composed of a fixed main infrastructure and a number of homogeneous mobile robotic vehicles. As already mentioned, our architecture implementation is tailored to a large number of low-cost vehicles equipped with limited sensor systems. Indeed, vehicle prototypes have been developed with such requirements. Details about vehicle hardware and software components are reported in the following subsections.

Vehicle Prototype

Robotic vehicles have been built, consisting of a chassis of 14 cm × 13 cm × 9 cm size that hosts motors, batteries, and electronics. The vehicles are also equipped with a Tmote-Sky sensor board, which enables communications with the 802.15.4 protocol, and some programmable system on chip (PSoC) mixed-signal array controllers that serve as servodriving, odometry, and CAC implementation (Figure 8). The Tmote-Sky board has been adopted for its high compatibility with the Zigbee protocol and low power consumption. An interface between microcontrollers and Tmote-Sky has been developed. To take advantage of every resources offered by all the units, the load of computing algorithms has been divided among the Tmote-Sky and PSoCs CPUs. With such an approach, performances have been improved with respect to the performance achievable only with the Tmote board. Indeed, a 40-Hz CAC computation and a 200-Hz servo driver control have been obtained. Extensive tests have been done on a test bed composed of two and three robotic vehicles (see Figure 9 for a two-vehicle example) as reported in the "Experimental Results" section.

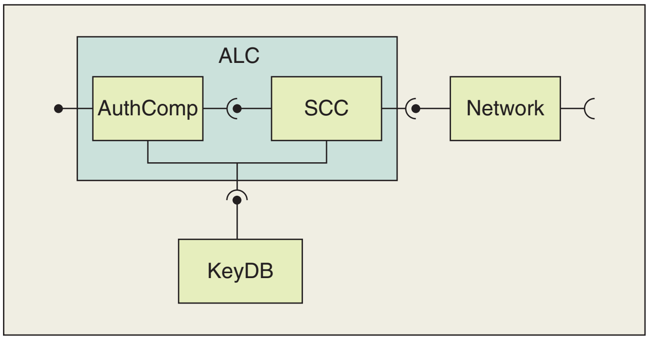


Figure 7. *The ALC.*

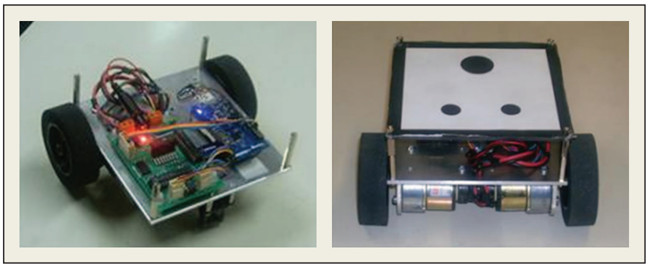


Figure 8. *Prototype of the vehicle.*

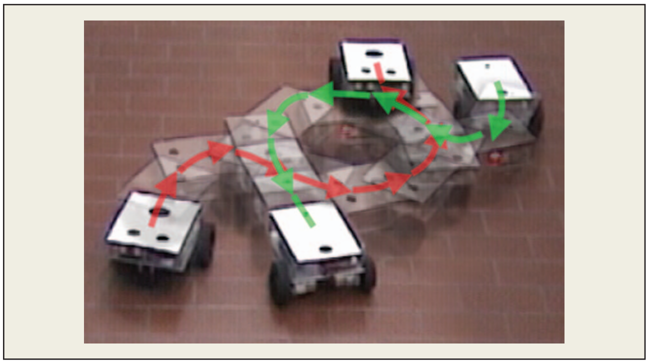


Figure 9. *GRP trajectories of two vehicles with assigned initial and final configurations.*

The chosen operating system is Contiki [19], which was developed for devices with low memory and computational capabilities. The implementation of the component model for Contiki is known as the component runtime kernel [13]. It has been chosen for the Contiki operating system since it is a lightweight and flexible operating system for tiny networked sensors and has a dynamic structure that allows to replace components during runtime.

As a consequence of the low-cost implementation of the vehicle, the peer- and self-localization have not been implemented on board. Indeed, the excessive cost and the insufficient precision of available sensor technologies have induced us to implement the location service (LS) localization service. The self-localization is achieved by means of aperiodic requests to a fixed infrastructure localization service that relies on computer vision to identify the vehicles' states. Furthermore, the peer-localization module is performed by listening to periodic messages of other vehicles communicating data about position and reserved disk radius.

From a security perspective, each vehicle implements an early prototype of the SSC (see "Secure Communication Service" section), the rekeying protocol (see "Rekeying Service" section), and the authenticated loading protocol (see "Authenticated Loading Service" section). The security controller uses Skipjack as a symmetric cipher to encrypt application and rekeying messages; the rekeying protocol (AuthKey component) uses SHA-1 to build and verify keychains; and finally, the authenticated loading protocol (AuthComp) uses a keyed-hash message authentication code based on SHA-1 to authenticate the first slice of a component. Furthermore, in our prototype, we do not consider it necessary to protect the confidentiality of a component during downloading. So ALC contains only the AuthComp component responsible for the authenticity of component slices.

Infrastructure

The infrastructure enables the LS by detecting the states of every vehicles and providing the common reference frame shared within the group. Second, the infrastructure enables the RKS by generating new keys and distributing them when necessary.

Off-the-shelf cameras have been exploited for monitoring the environment. Vision algorithms have been developed to identify the state of every vehicle by means of markers placed over the chassis. By precisely calibrating the cameras, an accurate estimate of the position and orientation of each vehicle has been obtained. Despite the use of the low-cost cameras, the chosen algorithms are robust to illumination changes in an indoor test bed. Cameras and algorithms are hosted on a system composed of three PCs, connected in a LAN.

Communication Protocol

To test the platform, a simple ad hoc wireless communication protocol has been implemented for both periodic (required by the LS) and aperiodic communications (required by other services). The communication protocol realizes a time-division multiple access protocol briefly described in the following.

A central authority is responsible for the temporal synchronization and a time slice-based subdivisions. In large, multihop wireless networks, an accurate distributed time synchronization is a nontrivial problem [20]. Each time slice is composed of $2N + K$ slots, where N is the number of vehicles, and $K \geq 2$ is an integer value used to avoid starvation. Any time the group membership changes, the communication protocol assigns a slot index and the time slice duration to each vehicles. A time slice is composed of two phases: periodic communication phase and aperiodic communication phase. The periodic communication phase starts with a synchronization message, and it is composed of N slots. Every vehicle has its own slot to perform peer localization (broadcast of position and reserved disk radius), and it can submit a request to the central authority for self-localization and permission for further aperiodic communications. At the end of this phase, all vehicles have collected information regarding neighboring vehicles. In the aperiodic communication phase, the central authority replies to requests for self-localization (in no more than N slots), and it gives acknowledgments to vehicles that performed a request for an aperiodic slot.

Experimental Results

Components proposed in previous sections have been separately implemented and tested to verify their effectiveness before the integration of the overall platform. Details on technical data of the implemented components are reported.

In the proposed implementation of the SCC and the ALS, the time required for encrypting a packet of 48 B is 9.92 ms by using by SkipJack whereas the time for applying the hash function SHA-1 on a packet of 28 B is 14.3 ms. Furthermore, the time required for key authentication is 32.2 ms by using SkipJack as symmetric cipher and SHA-1 as hash function. To authenticate a component of 1,264 B, the computational overhead is 1.84 S.

The localization service is provided with resolution of 0.23 cm and 0.03 rd in an environment of 290 cm × 133 cm using two cameras. The truncation error during the transmission process is at most 0.04 mm for lengths and 10^{-5} rd for angles. The average errors measured during experiments is around 1 cm and 0.06 rd for lengths and angles, respectively. On a PC Pentium of 43 GHz with 1 GB RAM, the proposed implementation is able to process ten frames per second.

As reported in the "Platform Prototype" section, a basic component for the wireless network management has been implemented to allow wireless communication between agents. Each localization packet is composed of 12 B where the first is the identifier of the localized vehicle, while the others represent the status of the vehicle (x, y, and θ on two bytes) and the position of its center (x_c and y_c on two bytes). The short time needed to send a localization packet (approximately 0.006 s) allows the system to manage a large number of vehicles under the bandwidth constrains of the 802.15.4 wireless protocol.

Finally, several experiments have been performed to prove the effectiveness and the reliability of the overall platform. As the particular task for each robot involved does not influence the testing of the platform, a scenario in which two or three vehicles were assigned a final configuration without any specific task has been considered. The trajectory performed by two robots during an experiments is shown in Figure 9.

In all the conducted experiments, vehicles have forward velocity of 3cm/s, angular velocity during the hold mode of 0.385rd/s, reserved disk radius of 13 cm, and safety disk diameter of 15 cm. On each vehicle a battery pack of 9.6 V, 1,800 mA has been mounted to provide energy to the Tmote-Sky and the PSoCs. With such a power supply, this kind of experiments can be conducted for around 90min. It is important to notice that during the experiments, intensive use of the wireless communication is required by the architecture. Most of the energy supply is used for vehicles' motion, while the communication and the security protocols are less energy demanding.

Partial overlapping of reserved disks has occurred during experiments for at most 4.1cm because of nonexact integration of motion and delay on data communicated through the network. Indeed, as reported in the "Collision Avoidance Service" section, the GRP policy ensures the safety of the system only theoretically. In the real framework, the system safety can be recovered enlarging the reserved disk size according to estimated errors on the localization system and to a forecast of communication delays.

Conclusions and Future Work

A scalable platform for decentralized traffic management of a multiagent system has been proposed. Safety of the platform is achieved with a cooperative conflict avoidance policy. Security of communications among vehicles with respect to potential external adversaries is obtained through use of cryptographic keys and rekeying policies. A prototypical implementation of the architecture has been described, and some experimental results have been reported. Future work will be devoted to addressing further decentralization of the check-out and security procedures, intrusion detection, and noncollaborative collision avoidance protocols.

Acknowledgments

We thank M. Dell'Unto and G. De Paolis for having contributed in the implementation of the platform prototype. The work has been done with partial support from EC project RUNES (Contract IST-2004-004536) and EC Network of Excellence HYCON (Contract IST-2004-511368).

Keywords

Networked heterogeneous mobile robots, component-based platform, collision avoidance, safe communication.

References

[1] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications," *Proc. IEEE*, vol. 91, no. 8, pp. 1235–1246, 2003.

[2] S. Graham and P. Kumar, Eds., "Proceedings of PWC 2003," in *Personal Wireless Communication* (Lecture Notes in Computer Science), vol. 2775. New York: Springer-Verlag, 2003.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 988–1001, 2003.

[5] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Automat. Contr.*, vol. 52, no. 5, pp. 863–868, 2007.

[6] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 401–420, 2006.

[7] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments," *Proc. IEEE*, vol. 95, no. 1, pp. 234–254, 2007.

[8] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Automat. Contr.*, vol. 49, no. 8, pp. 1292–1302, 2004.

[9] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1465–1476, 2004.

[10] Y. Mostofi, T. Chung, R. Murray, and J. Burdick, "Communication and sensing trade-offs in decentralized mobile sensor networks: A cross-layer design approach," in *Proc. Int. Symp. Information Processing in Sensor Networks*, 2005, pp. 118–125.

[11] N. Lynch, *Distributed Algorithms.* San Mateo, CA: Morgan Kaufmann, 1996.

[12] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems*, Baltimore, MD, 2004, pp. 81–94.

[13] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis, "The runes middleware: A reconfigurable component-based approach to networked embedded systems," in *Proc. 16th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, Berlin, Germany, 2005, vol. 2, pp. 806–810.

[14] L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi, "Decentralized cooperative policy for conflict resolution in multi-vehicle systems," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1170–1183.

[15] P. Alriksson, J. Nordh, K.-E. Arzen, A. Bicchi, A. Danesi, R. Schiavi, and L. Pallottino, "A component-based approach to localization and collision avoidance for mobile multi-agent systems," in *Proc. European Control Conference*, 2007, pp. 4285–4292.

[16] Secure Hash Standard, National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 180-1, 1993.

[17] R. Rivest, "The RC5 encryption algorithm," in *Fast Software Encryption—Second International Workshop*, vol. LNCS 1008. Heidelberg, Germany: Springer-Verlag, 1995, pp. 86–96.

[18] G. Dini and I. M. Savino, "S^2RP: A secure and scalable rekeying protocol for wireless sensor networks," in *Proc. 3rd IEEE Int. Conf. Mobile Ad-Hoc and Sensor systems*, 2006, pp. 457–466.

[19] A. Dunkels, B. Gröonvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. IEEE Workshop on Embedded Networked Sensors*, 2004, pp. 455–462.

[20] R. Solis, V. Borkar, and P. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. 44th IEEE Conf. Decision and Control*, 2005, pp. 2734–2739.

[21] J. Wu and F. Dai, "Efficient broadcasting in ad hoc wireless networks using directional antennas," in *IEEE Trans. Parallel and Distributed Systems*, 2006, vol. 17, no. 4, pp. 335–347.

Antonio Bicchi is a professor of automatic control and robotics at the University of Pisa. His main research interests include dynamics, kinematics, and control of complex mechanical systems, including robots, autonomous vehicles, and automotive systems; haptics and dexterous manipulation; and theory and control of nonlinear systems, in particular, hybrid (logic/dynamic, symbol/signal) systems. He has published more than 200 articles in international journals, books, and refereed conferences.

Antonio Danesi works in software specification for space launchers for European Space Agency. He obtained his Ph.D. in robotics and automation in 2007 at Centro "E. Piaggio," Università di Pisa, with a research on component-based control software on networked mobile agents for visual navigation and mapping. He participated in European project RUNES for the advanced control group.

Gianluca Dini received the Laurea degree in electronic engineering from the University of Pisa in 1990 and a Ph.D. in computer engineering from Scuola Superiore S. Anna, Pisa, in 1995. Since 2000, he has been an associate professor of computer engineering at the University of Pisa. His main research interests are in distributed computing, with particular reference to security and fault tolerance.

Silvio La Porta received the Laurea degree in computer engineering in 2006 from the Faculty of Engineering, University of Pisa. Currently, he is a Ph.D. student in computer engineering at the Department of Ingegneria della Informazione, Pisa. His research interests are in network security and network forensics.

Lucia Pallottino received the Laurea degree in mathematics from the University of Pisa in 1996/97 with a thesis in numerical analysis. She received a Ph.D. in robotics and industrial automation at the Department of Electrical Systems and Automation of the University of Pisa in 2002. She is currently a researcher in the Interdepartmental Research Center "E. Piaggio" in the University of Pisa. Her main research interests within robotics are in motion planning and control for nonholonomic vehicles, air traffic management systems, and quantized control.

Ida M. Savino received the Laurea degree in computer engineering in 2004 from the Faculty of Engineering, University of Pisa. Currently, he is a Ph.D. student in computer engineering at the Department of Ingegneria della Informazione, Pisa. His research interests are in security of networked embedded systems.

Riccardo Schiavi received the Laurea degree in computer engineering in 2004 from the Faculty of Engineering, University of Pisa. Currently, he is a Ph.D. student in robotics automation and bioengineering at the Centro Interdipartimentale di Ricerca "E. Piaggio" and at the Dipartimento di Sistemi Elettrici e Automazione. His research involves the development and control of embedded systems.

Address for Correspondence: Lucia Pallottino, Centro "Enrico Piaggio," via Diotisalvi, 2, 56100 Pisa, Italy. E-mail: l.pallottino@ing.unipi.it.

Consumer Robotic Products

Studying the Factors That Influence Market Adoption

BY JIM WYATT, WILL N. BROWNE,
MARK N. GASSON, AND KEVIN WARWICK

It is a frequent assertion that there is to be a proliferation of robot technology, specifically for use within domestic environments [1], [2]. It is further claimed that such devices will fulfill numerous practical roles [3], [4]. However, despite much research in this area (e.g., [5], [6]), few products of this type have actually been available to consumers.

The rapid expansion of the domestic robotics market in the period 2002–2003 prompted the United Nations Economic Commission for Europe's World Robotics survey to identify the new market for consumer (domestic) robots as booming, with the potential for further expansion in the field of personal care robots for the elderly and disabled as well as home security applications. The survey, carried out in 2004, predicted approximately 6.6 million units to be in service by the end of 2007, with an estimated sales value of US$6.7 billion. "They will not only clean our floors, mow our lawns, and guard our homes, but they will also assist old and handicapped people with sophisticated interactive equipment" [7].

Previous studies have been conducted with regard to the response of subjects toward individual research robots [8] and the responses of the robot to interaction with a human operator [9]. Extensive studies have focused on the use of robotic platforms in education [10], [11]. However, relatively little research has been conducted on the views of individuals with regard to the overall requirements they would have of a domestic robotic product. Such research is important as it helps to establish attitudes toward robots that might serve as either motivation or a barrier for individuals to engage with or purchase robots.

The aim of this article is to identify the key factors that are associated with the adoption of a commercial robot in the home. This article is based on the development of the robot product Cybot by the University of Reading in conjunction with a publisher (Eaglemoss International Ltd.). The robots were distributed through a new part-work magazine series (*Ultimate Real Robots*) that had long-term customer usage and retention. A part-work is a serial publication that is issued periodically (e.g., every two weeks), usually in magazine format, and builds into a complete collection. This magazine focused on robotics and was accompanied by cover-mounted component parts that could be assembled, with instructions, by the user to build a working robot over the series. In total, the product contributed over half a million operational domestic robots to the world market, selling over 20 million robot part-work magazines across 18 countries, thereby providing a unique breadth of insight.

Gaining a better understanding of the overall attitudes that customers of this product had toward robots in the home, their perception of what such devices could deliver and how they would wish to interact with them should provide results applicable to the domestic appliance, assistance/care, entertainment, and educational markets.

Digital Object Identifier 10.1109/M-RA.2007.907358

Background

The majority of robotics research considers social [12] and psychological [13] issues and is orientated toward rehabilitation [14], education [11], [15], and investigative studies [16]. These works are generally not applied to high-volume/low-cost robots that are considered here (LEGO Mindstorm [10] is a notable exception). If the large potential market for robotic products is to be realized, then ongoing research is needed to identify and understand all the important factors in the relationship of the user to a robotic product.

A study of public expectations toward robots [17] showed that respondents expected robots to be humanoid (or to have some human features) and to be able to help with domestic chores (largely female respondents) or could be played with and investigated for recreation (largely male respondents). The limiting factors associated with robot products related to behavior in that a robot should not be a simple pet, spy on users, takeover, or be incompetent.

In 2002, a large-scale survey was conducted of visitors to the Robotics exhibit at the Swiss National Exhibition Expo 2002 [18], in which respondents were asked to give their opinions on robots in a domestic context and with regard to prosthetic devices. The survey showed a strong interest in robots that performed labor-saving tasks or enhanced personal welfare but did not identify factors that would lead to their adoption. The idea that robots would enhance personal happiness was less prevalent in this study, as was guardedness toward robots. Subjects also showed a

1070-9932/08/$25.00©2008 IEEE

lesser disposition toward humanoid features on a robot. The discrepancies between these findings may be, in part, due to the methodology of the study in that, as in the smaller scale studies discussed in [19] and [20], the surveys were conducted immediately after exposure to specific robots.

Studies of children's attitudes toward robots are also scarce but indicate a propensity for subjects to be influenced by themes in contemporary science fiction. In the late 1970s, children aged between 5–11 were asked to draw pictures of and write short stories about robots [21]. These showed humanoid robots, many of which were aggressive in nature and appearance. Two decades later [22], children aged 7–11 also imagined robots as having a humanoid form. However, while in this later study most subjects continued to include human characteristics, such as considering them as male and exhibiting free will, notably fewer drew robots that were aggressive.

In 2004, a study was conducted in preparation for an interactive presentation at a science museum in the United Kingdom [23], and in this study the following trends were established:

- subjects were skeptical of the potential abilities of robots in the future
- most adults consider that the role for robots would be to perform housework and menial tasks, while children consider them to be a source of recreation
- most subjects were unaware of the extent to which robots were already used in areas such as space exploration and warfare
- adult subjects did not believe that robots would ever achieve a level of intelligence comparable with humans, while younger subjects believed that they would
- children's views of robots are heavily determined by their physical appearance
- some children differentiate robots by their ability to perform tasks, such as walking and talking.

The results of this study are perhaps a little surprising in that there is no mention of the "Robot Wars" television series (and by extension, aggressive robots), which had been highly successful only one or two years earlier. ("Robot Wars" was a popular television series in which contestants built their own radio-controlled battling robots. The robots were required to battle each other and were neither autonomous nor able to perform useful tasks.)

Data Collection and Methodology

This article represents a five-year study of data from the entire lifespan of the Cybot product (described in the next section) in the United Kingdom from its inception through a test launch (from May 2001) and full national launch (from September 2001 to April 2005). The aim of the market research was commercial, whereas the aim of this work is academic.

Focus Group Interviews

All focus group interviews were arranged by the publisher and conducted by professional market research companies. Due to the commercial nature, some restrictions were placed on the subject recruitment criteria (e.g., all subjects interviewed prior to the product launch were male due to the publisher's assertion that the product would predominantly appeal to a male audience).

Four rounds of focus group interviews were conducted prior to the launch of the product. The initial round of interviews was used to establish what factors were important to potential customers in the subject of robots. Later interviews were used to refine initial concepts as well as the design of the product.

Each of the four rounds of interviews was conducted with 30–36 subjects split into groups of not more than eight (segregated by age). Subjects were male between the ages of 8 and 25 (with additional groups made up of parents of boys aged 8–14). Interviews were conducted across England at either purpose-built research facilities (rounds one and two) or in a school environment (rounds three and four). Candidates were selected either on the street, by phone, or through schools using questionnaires. Subjects were required to have some interest in science and technology and in building model kits and to have purchased a part-work product at some time. Thus, conclusions drawn from this stage of research will be from a subset of the population with an affinity toward robotics.

Product Test Launch

It is common practice for the publisher to launch the product in a restricted test area a few months prior to a national rollout to assess product performance and the effectiveness of the television advertising campaign.

Customer Survey Questionnaire

Customer survey questionnaires were supplied in all test copies of Issues 2 (May 2001) and 13 (December 2001). Customers were given an incentive to complete the survey, although prizes were selected so not to bias the responses.

Customer Correspondence and Interaction

Online forums enabled customers to communicate with the publisher and with each other. Customers were also invited to participate in competitions, with the entries providing an insight into how customers responded to Cybot and robots in general and were then used to determine the desired functionality and appearance.

Robotic Product

Robot

The original Dwarf robots, upon which Cybot is based (Figure 1), were created in the early 1990s so members of the public could gain hands-on experience with a functional robot as part of the University of Reading's public understanding of science and recruitment programs.

The robots represent a simple autonomous robot system consisting of an array of forward facing sonar sensors and two driven wheels linked by a control system (Figure 2). This allowed for the creation of simple reactive behaviors based on input states, such as following and object avoidance. Later versions of the Dwarf robots included increased processing capabilities as well as radio and infrared communication systems, facilitating the implementation of flocking and group learning behaviors [24].

The drives and sensors on the Cybot robot (Figure 3) are similar to those used on the Dwarf robots. The final design was based on the identified factors that influence the user, with focus on the appearance, materials, and construction.

The electronic and mechanical design of the robot was such that inexperienced users could assemble the robot themselves and undo any errors without permanent damage. The design was required to be modular so that functionality could be released throughout the product's lifetime to retain customer interest (Table 1). By monitoring the sales of the magazine as functionality of the robot increased, the importance of specific functionality could be determined.

The Magazine

The magazine was not only intended to support the assembly and use of the cover-mounted robot but also to retain customer interest in the product in the two-week period between issues. Each strand of the magazine was intended to cover a different aspect of the customers' interest in robots and related subjects (Table 2).

Marketing and delivery of the product was designed to reinforce aspects of the robot identified as desirable during focus group research. For the first six issues (the period during which the magazine was on display in stores), the magazine and components were presented on a high visibility backing board (Figure 4). Components were blister packaged for high visibility such that customers could examine the contents for quality and value.

Secondary Support Material

Due to the diversity of the market relating to robots, the importance of supporting the product with additional media (apart from the primary magazine) that would help explain the functionality and overall nature of the robot to potential customers was identified early in the research. This material included an eight-page introductory supplement, two television commercials, a promotional video, and the Website.

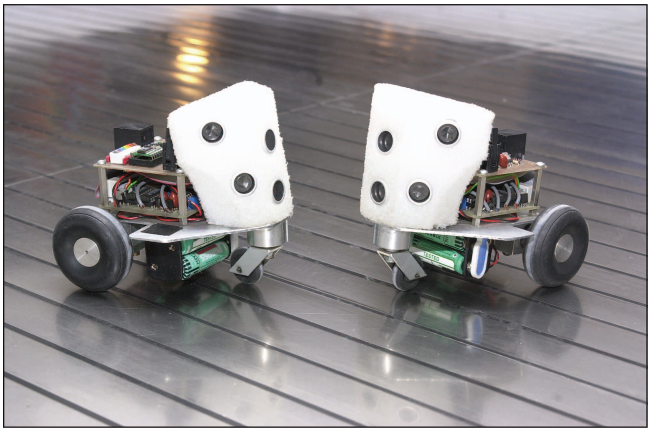


Figure 1. *A pair of second-generation Dwarf robots used as the inspiration for the Cybot robot kit.*

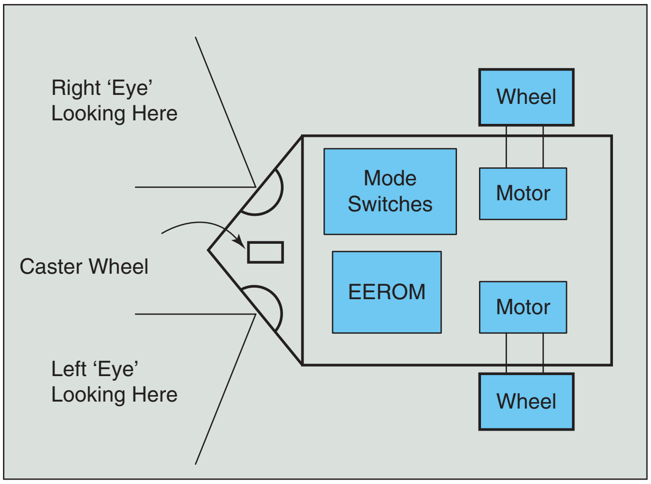


Figure 2. *Basic topology of the Dwarf robot: a simple look-up table stored in the memory allowed the mapping of sensor data to wheel movements for simple reactive behaviors.*

Figure 3. *Fully constructed Cybot (25 × 18 × 17 cm) from the parts provided with Issues 1–17 of* Ultimate Real Robots Magazine. *Autonomous capabilities include line following, basic flocking, and obstacle avoidance.*

Table 1. Schedule for the delivery of functionality.

Issue Number	Description of Functionality
4	The chassis, drive train, power distribution, and motor control board are assembled. The robot can be made to move forward.
9	The first microcontroller board is installed enabling the robot to perform simple phototaxis and photophobic behavior.
14	Installation of a second microcontroller enables modes on the robot to be changed via switches and facilitates line-following behavior, light avoid mode, and follow mode.
16	The robot is capable of functioning in sonar avoidance and follow modes.
17+	Additional functionality: remote control, voice recognition, learning algorithms, infrared localization, communication between robots and programmability.

Results and Discussion

Through the successive focus group interviews, themes were identified and categorized as follows (Table 3): terminology, functionality, status/pride, appearance and construction, value, personalization, and support material. These factors were verified during the test launch through further rounds of interviews, customer surveys, and correspondence and were validated by the analysis of the sales figures.

The results of the product's test launch indicated that its appeal to consumers was very strong with the sales of Issue 1 exceeding 25,000 copies (Figure 5), with higher customer retention levels than were found at the launch of other partwork series. It is therefore considered a valid assertion that the product (which went on to sell a total of 20 million copies worldwide) was highly successful (see Table 4).

Influencing Factors

Terminology

From early focus group sessions, it was clear that the word *robot* was itself sufficient to generate interest and instilled a desire to find out more about the product. This was most noticeable among preteens, in whom the subject area frequently caused an excited response.

Table 2. Example content of a typical magazine.

Strand Title (pages)	Strand Description
Front cover (1)	To maintain magazine identity, the cover image was taken from one of the narrative strands. Cybot had an ongoing presence on the front cover of every magazine.
Inside front cover (2)	Editorial such as contents, contributors, publisher's address, technical support, etc.
Cybot: Step by step (3/10)	A highly detailed guide to assembling the components supplied with the issue. It was anticipated that the user would obtain an increased ownership over a prebuilt robot.
Cyber science (11/13)	Articles introducing the science and technology behind robots. This could affect the kudos of owning the robot.
Workshop (14/17)	Robot designs by members of the general public: Rex's Robot Challenge in which a robot engineer guided readers through the construction of different robot types. This helped link Cybot with other robot styles.
Robostars (18/19)	Intended to link fictional robots with real robots to affect user perception.
Robots in action (20/22)	The application of robots in the real world to tasks, thereby placing Cybot in context.
Network (23/24)	A collection of short articles on two pages, including news, comic strips, puzzles, robot facts, jargon explanation, Websites, and information on forthcoming magazines.

When first asked to define what a robot was, the majority of younger subjects described a device similar to those featured in the "Robot Wars" television series, while adult subjects referred to science fiction robots. Although their exposure to robots in popular culture helped to develop their interest in the subject matter, the majority were also aware of the existence of what was termed *real robots*. Research tools (such as the Honda androids) or commercial products (such as Aibo and robot vacuum cleaners) were identified as being real robots based on having a useful purpose as well as possessing some artificial intelligence, a grouping which, notably, did not include products that were considered to be toys (e.g., Furby and Poo-Chi).

Real robots were considered by many of the focus group subjects to be highly desirable and valuable. Subjects' knowledge of these devices was less comprehensive than that of fictional robots and the machines featured in "Robot Wars," but their interest in these real robots was comparable, if not greater.

Customers in the test market were predominantly 9–16 years of age (Figure 6), which is a likely result of there being a greater enthusiasm for robots among this age group. That said, visual cues in the television commercial and magazine were targeted toward this demographic. The large-scale acceptance of the product suggests the flexibility of the term *robot* as the majority of subjects in the previous focus group interviews first associated robots with science fiction and the "Robot Wars" series.

Functionality

Despite a strong interest from subjects in the completed robot, it became apparent that to sustain interest over an extended period of time, the functionality of the robot would need to be released at regular intervals during the series. It was clear that subjects' expectations of the robot were realistic, e.g., the rejection of a legged robot by subjects as being too slow. A concern was that the robot should be fun to use or play with but at the same time should be a serious endeavor. Any aspects of the robot's functionality or design that suggested that it was a child's toy prompted a negative response. Some responses were as follows:

- "Toy robotics is not as appealing." (Group age: 14–15)
- "When you build it, it's a learning aid, when it's finished it's a toy." (Group age: 14–15)

Two aspects of the robot's functionality that were often raised as being important were that it should be useful and intelligent. Most subjects accepted that it would be difficult to make a small robot particularly useful. However, it was often suggested that if the user was given control over the robot, either directly (by means of a remote control or voice activation) or indirectly (by making it programmable), then the user could dictate how the robot was used. This would indicate that an appropriate application for robots is to entertain simply through interaction with the user in a manner that provides not only autonomy but also a means by which the user may exert control.

From Issues 2 and 13 of the customer surveys, the intelligence of the robot was of high importance to customers and gave rise to threads in the online forums, indicating that the more intelligent behavior of the robot was seen to be the better (see Table 5). These discussions also highlighted that the customers' understanding of the terms *intelligence* and *artificial intelligence* were relatively loose, leading to passionate discussion as to whether Cybot was intelligent or not.

The overall approval of the Cybot robot was shown to be high (Table 6). There was a strong interest among focus groups in battling functions, which was resisted by the development team, who were concerned about safety of users and the robot. This was alleviated by Rex's Robot Challenge in the magazine and, as a result, there was little discussion among forum contributors regarding the inclusion of this feature on Cybot itself. Forum contributors and survey respondents also indicated that few of them were building the robots featured in the Rex's Robot Challenge strand (Table 7).

Status and Pride

The original Dwarf robots were recognized by many of the subjects in the initial focus group interviews from their numerous television appearances, in which they had been introduced as advanced research robots. This provided the robots with a credibility that was not afforded to other products demonstrated, causing them to be seen as unique, sophisticated, and serious (rather than as a toy) and thereby providing customers with a means to raise their personal status among their peers. This sense of customer pride would be magnified by allowing customers to build the robot themselves, a task that most subjects initially considered to be beyond their capabilities. "You can show off to people . . . 'I made this!' . . . and I'd put it on show" (Group age: 11–12).

Delivery of a product that is seen by consumers to be truly robotic raises the status of the owner as an early adopter of new and exciting technology. This position is strengthened if the owner has had the ability to build the device themselves.

Appearance and Construction

The highly technical appearance of the Dwarf robots, featuring exposed circuit boards and wires and the obviously hand-engineered mechanical components, again reinforced their uniqueness and credibility as serious robots.

During demonstrations of the Dwarf robots, subjects regularly anthropomorphized the robot's behavior. However, they did not wish the robot to appear overly animal-like as this again suggested that the product was intended to be a child's toy. "Something that looks futuristic . . . non-tacky" (Group age: 19–25).

Cybot's final design was influenced strongly by three main contributing factors: 1) manufacturing and budget constraints influenced the materials used in production, 2) many of the subjects insisted that the robot should be fully encased so as to provide protection for the electronic components, and 3) design trends used in contemporary consumer electronic products influenced the design.

To avoid negative association with toy products, the robot's outward appearance was such that all features of the design served a practical purpose in the function of the robot, e.g.,

(a)

Welcome back to the Real Robots website roboteers! We hope you're enjoying the magazine and will continue to put Cybot and Tom through their paces!

Check out Cybot's development

ROBOT SUPPORT is now closed but Any Questions? is full of useful technical advice.

Catch up with the best of Hubble's photographs – space the final frontier! Click here.

Click here for our brilliant Cyball Goal offer!

An Eaglemoss Publication. © Eaglemoss International Ltd 2001–2005
Privacy Statement

(b)

Figure 4. (a) Issue 1 pack: the context of the robot is an important influence on the perception of users of robot technology. (b) The Real Robots Website provided additional information and support as well as a means of interaction between customers. Status and pride as well as personal expression occurred through this channel.

Table 3. Factors contributing to the success of the product at each stage of the research.

Round	Identified Factors from the Interviews
1	Predictable issues regarding cost, product design, and usage were raised, but deeper core factors affecting an individual subject's likelihood to adopt the product were identified. *Value*: Most subjects were aware of part-work products, viewed them with cynicism, and doubted that a robot delivered in this manner could meet with their expectations. *Support material*: It became clear that the magazine component of the product would carry almost equal importance to the robot. Subjects had a deep interest in most aspects of robots and as such the magazine should serve as more than an instructional guide for assembly of the robot. *Status and pride*: It became clear that the robot would be something that subjects would wish to show off to their friends not only because of what it did but also because they had built it themselves. *Appearance and construction*: Subjects were mostly averse to a product that appeared pet-like and showed a preference for more technical designs. Subjects also expressed concern that their level of skill would not be sufficient to assemble and operate the robot. *Functionality*: Functions identified as desirable were those that demonstrated the advanced technology of the robot. Subjects also expressed a desire to have direct control over the robot. Pet-like behaviors were rejected.
2	*Terminology*: Subjects made a distinction between fictional/battling robots and real robots. *Appearance and construction*: Subjects held reservations as to their own abilities and were concerned that they could damage the robot. The metal chassis of the prototype robot helped to reassure them that the robot would be tough and resilient. *Personalization*: The robot was required by subjects to be versatile enough that they could use it in a way that reflected their own interests. This was further emphasized by their desire to customize the robot's appearance. This round of interviews also served to identify the target age group of the core customer base (aged 8–14). These subjects had a highly developed interest in robots and few demands on their time and money.
3	Although primarily used to verify the findings of previous interviews, this round of interviews highlighted opinions that the product should be treated seriously by both the publisher and the user and that this should be reflected in the appearance and functionality of the product as well as in the television advertising campaign.
4	This round of interviews primarily served to verify the results of the previously conducted research.

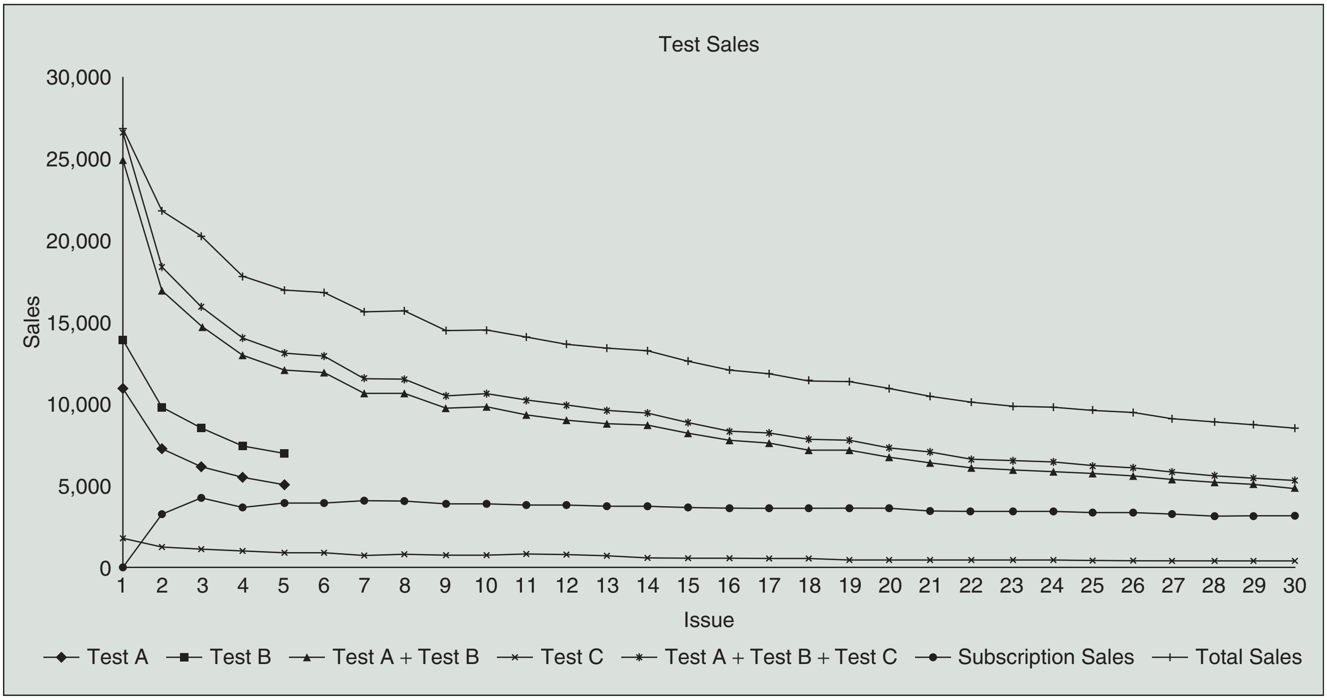


Figure 5. *Sales figures for the test launch of the product in the United Kingdom. These provide an indication of the initial and long-term (customer retention) market acceptance of the product.*

status indicator lights in the antennae. As the robot did not conform to a stereotypical design (real or fictional), this enabled users to approach the product with an open mind in terms of its functionality and level of performance. The device was accepted as a robot as it performed functions that were accepted as robot-like.

During the focus group interviews, subjects had expressed a lack of confidence in their ability to build a complex robot. Construction of this robot was atypical for a domestic robot product as it relied on the user to complete the assembly, so

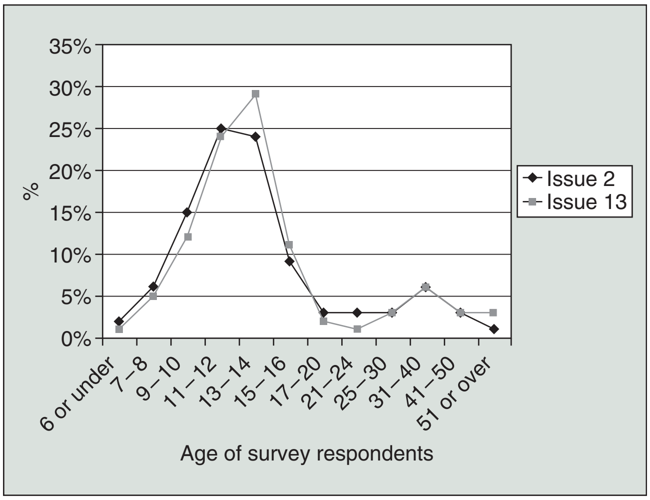


Figure 6. *Age distribution of the surveyed customers; an indication of the magazine customer base.*

Table 4. Likelihood to continue purchasing the series.

Survey Conducted	Issue 2	Issue 13
Total (% of issue sales)	2103 (9.7%)	1006 (7.5%)
I definitely will	84%	85%
I am likely to	15%	13%
I am unlikely to	0%	1%
I definitely will not	0%	0%

Table 5. Reasons given for survey respondents purchasing the product.

Survey Conducted	Issue 2	Issue 13
Total	2103	1006
I love anything to do with robots.	13%	17%
I enjoy "Robot Wars" and want to build my own battling robot.	25%	20%
I want to learn more about robots and how they work.	11%	11%
I just enjoy building things.	19%	19%
It is something I can do with my dad or brother.	4%	2%
My friends are into it too so we can compete with our robots.	2%	2%
I've wanted to build my own intelligent robot for a long time.	19%	20%
Not answered.	6%	8%

Table 6. Survey respondents' responses to robot benefit statements.

Benefit Statement	Cybot Is the Best Robot Available to Buy		The Magazine and Robot (Cybot) Are Really Good Together		Cybot Is Better than Remote Controlled Robots (as on "Robot Wars")		Cybot Is Just Another Toy	
Survey Conducted	**Issue 2**	**Issue 13**	**Issue 2**	**Issue 13**	**Issue 2**	**Issue 13**	**Issue 2**	**Issue 13**
Total	2103	1006	2103	1006	2103	1006	2103	1006
Agree strongly	53%	51%	69%	71%	32%	26%	2%	3%
Agree slightly	27%	28%	23%	22%	24%	24%	5%	6%
Neither agree nor disagree	15%	17%	5%	4%	26%	28%	9%	10%
Disagree slightly	2%	2%	1%	1%	10%	14%	18%	20%
Disagree strongly	1%	1%	0%	1%	5%	7%	62%	59%
Not answered	2%	2%	2%	1%	3%	2%	3%	2%

Benefit Statement	Cybot Is the Only Intelligent Robot I Can Build Myself		There Are Better Robots Around Than Cybot		Cybot Is a Sophisticated Electronics Robot	
Survey Conducted	**Issue 2**	**Issue 13**	**Issue 2**	**Issue 13**	**Issue 2**	**Issue 13**
Total	2103	1006	2103	1006	2103	1006
Agree strongly	54%	51%	9%	11%	54%	57%
Agree slightly	18%	20%	14%	16%	25%	28%
Neither agree nor disagree	14%	14%	31%	31%	12%	10%
Disagree slightly	6%	7%	14%	13%	2%	1%
Disagree strongly	5%	6%	27%	26%	1%	1%
Not answered	3%	2%	4%	3%	6%	3%

that it had to be robust, user-maintainable, reversible (if mistakes were made), and expandable. Constructing the robot from semifabricated component parts allowed the user to assemble the robot with a minimum level of skill and resources and strengthened the users' confidence in their abilities. Although a minority of users encountered problems that they could not resolve themselves, the degree of user interaction strengthened the user's affinity with the robot.

Value

In addition to the value of the product provided by its functionality and uniqueness, the electronic components and materials used in the construction of the Dwarf robots' chassis suggested that it represented good value for the money. This was especially important if it was to be delivered with a partwork publication, which is commonly considered to have low value. When customers perceived that they were being supplied with components that did not help advance the functionality of the robot, unfavorable views were expressed in the online forums that the publisher was treating the robot as a toy. "I wouldn't mind but the team Cybot shell is or was just a waste of money" (Harvey 51, message board, 26/03/03).

Overall the product was seen to represent a good value for the money. However, it was clear that failure to deliver exactly on the promises made in the product's advertising and in the magazine resulted in a loss of the product's credibility.

Personalization and Personal Expression

Although the idea that the robot could be personalized was very well received in both the focus group interviews and in the customer surveys (Table 8), there was little indication that customers had attempted to customize the appearance of the robot themselves. It was also observed that little discussion was made of the programming languages developed for Cybot and that very few forum members exchanged programs that they had written.

Both the lack of independent customization and programming indicates that while the provision for such activities is seen as important and reinforces the individuality and seriousness of the product, few customers had the time or skill to invest in such endeavors.

Support Material

Subjects indicated that they did not want an educational product (perceived as boring) nor one that was too technically orientated (perceived as intimidating). Instead the magazine should present robots in a manner that they could easily identify with—fun to be with yet nontrivial.

The high product sales and retention figures (Figure 5) indicate good customer awareness (through the television commercials and the promotional video) and overall satisfaction with the delivered product.

Acceptance of the magazine itself was high among customers, many of whom saw it as having equal importance with the cover-mounted robot (46%, compared with 53% for building the robot as main appeal). Although the tone was not overtly educational, from the topics of discussion raised in the online forums as well as competition entries and fan mail received by the publisher, it was observed that both Cybot and the magazine had influenced customers' perceptions of what a robot was in terms of appearance and functionality.

The research has shown the importance of the support material created to accompany the robot. This material provided the product with credibility as a real robot and therefore a serious endeavor, while maintaining that it would be interesting and fun to use. The material (specifically the magazine) also provided a means by which users' interest in the general subject area could be maintained over a prolonged period as well as providing additional suggestions as to how the robot could be used through projects printed in the magazine and by suggestions from other users in online forums.

Conclusions

Although limited to a niche area of the United Kingdom robot product market with an identified user-base, this work has highlighted several important factors that have a bearing on the potential adoption of a robotic product. Attitudes toward robot technologies and the term *robot* are dynamic and malleable and tend to stem from contemporary media rather than academic/historical works. However, the relationship with robotic products is seemingly fragile, and, as such, users can become quickly disengaged if the robot product contravenes their expectations.

The key factors that affect consumers' propensity to interact with (and adopt in the long term) robot products have been identified as functionality, appearance and construction, terminology, status and pride, real and perceived value, personalization and personal expression, and support material accompanying the product. While these findings may well be specific to this product and user-base, the key factors are likely

Table 7. Survey respondents constructing the robot detailed in Rex's Robot Challenge.

Survey Conducted	Issue 2	Issue 13
Total	2103	1006
Yes	57%	28%
No	43%	70%

Table 8. How survey respondents wished to customize their robot.

Survey Conducted	Issue 13
Total	1006
I have already personalized my Cybot using my own materials and ideas.	2%
I would like ideas for customizing my Cybot without spoiling it.	48%
I would only adapt my Cybot using materials supplied with the magazine.	41%
I would only adapt my Cybot using body parts supplied with the magazine.	41%
I am not interested in customizing my Cybot at all.	6%

to be largely transferable to other robotic products aimed at the consumer market.

Keywords

Consumer electronics, human factors, robotics.

References

[1] G. A. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA: MIT Press, 2005.

[2] R. A. Brooks, *Robot: The Future of Flesh and Machines*. London: Allen Lane, 2002.

[3] P. Menzel and F. D'Aluisio, *Robosapiens*. Cambridge, MA: MIT Press, 2003.

[4] S. Turkle, *Life on the Screen: Identity in the Age of the Internet*. New York: Simon and Schuster, 1997.

[5] C. Breazeal, "Robot in society: Friend or appliance," in *Proc. 1999 Autonomous Agents Workshop on Emotion-Based Architectures*, Seattle, WA, 1999, pp. 18–26.

[6] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artif. Intell.*, vol. 114, no. 1–2, pp. 3–55, 1999.

[7] UNECE and IFR (United Nations Economic Commission for Europe and The International Federation of Robotics), "World Robotics 2004," United Nations Publications, New York, 2004.

[8] H. Ishiguro, "Android science: Toward a new cross-interdisciplinary framework," in *Proc. 12th Int. Symp. Robotics Research*, 2005, pp. 1–6.

[9] C. Breazeal and B. Scassellati, "How to build robots that make friends and influence people," in *Proc. 1999 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Korea, 1999, pp. 858–863.

[10] M. Resnick, F. Martin, R. Sargent, and B. Silverman, "Programmable bricks: Toys to think with," *IBM Syst. J.*, vol. 35, no. 3, pp. 443–452, 1996.

[11] P. Frei, V. Su, B. Mikhak, and H. Ishii, "Curlybot: Designing a new class of computational toys," in *Proc. Computer-Human Interaction 2000,* New York, pp. 129–136.

[12] C. Bartneck and J. Forlizzi, "Shaping human-robot interaction: Understanding the social aspects of intelligent robotic products," in *Proc. Computer-Human Interaction 2004,* New York, pp. 1731–1732.

[13] S. Turkle, *The Second Self: Computers and the Human Spirit*. New York: Simon and Schuster, 1984.

[14] K. Dautenhahn, "Roles and functions of robots in human society—Implications from research in autism therapy," *Robotica*, vol. 21, no. 4, pp. 443–452, 2003.

[15] S. Turkle, "Cyborg babies and cy-dough-plasm: Ideas about self and life in the culture of simulation," in *Cyborg Babies: From Techno-Sex to Techno-Tots*, R. Davis-Floyd and J. Dumit, Eds. New York, NY: Routledge, 1998, pp. 317–329.

[16] C. Breazeal, *Designing Sociable Robots*. Cambridge, MA: MIT Press, 2002.

[17] Z. Khan, "Attitudes towards intelligent service robots," NADA, KTH, Stockholm, Sweden, Tech. Rep. TRITA-NA-P9821, 1998.

[18] K. O. Arras and D. Cerqui, "Do we want to share our lives and bodies with robots? A 2000-people survey, " Autonomous Systems Lab, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland, Tech. Rep. 0605-001, June 2005.

[19] J. Hammel, K. Hall, and D. Lees, "Clinical evaluation of a desktop robotic assistant," *J. Rehab. Res. Dev.*, vol. 26, no. 3, pp. 1–16, 1989.

[20] P. Dario, E. Guglielmelli, C. Laschi, and G. Teti, "MOVAID: A personal robot in everyday life of disabled and elderly people," *Technol. Disabil.*, vol. 10, no. 2, pp. 77–93, 1999.

[21] J. Reichardt, *Robots: Fact, Fiction and Prediction*. Harmondsworth: Penguin, 1978.

[22] K. Bumby and K. Dautenhahn, "Investigating children's attitudes towards robots: A case study," in *Proc. 3rd Int. Conf. Cognitive Technology,* 1999, pp. 359–374.

[23] K. Bultitude, B. Johnson, F. Burnet, D. Evans, and A. Winfield, "Robot thought—A dialogue event for family audiences," in *Proc. AISB 2005 Creative Robotics Symp.*, Hertfordshire, UK, 2005, pp. 14–21.

[24] I. Kelly, D. Keating, and K. Warwick, "Mutual learning by autonomous mobile robots," in *Proc. First Workshop on Teleoperation and Robotics, Applications in Science and Arts*, Austria, 1997, pp. 103–115.

Jim Wyatt received a B.Sc. in cybernetics and control engineering and a Ph.D. in cybernetics from the University of Reading, United Kingdom in 1997 and 2006, respectively. Until 2007, he worked with the University of Reading, both in the United Kingdom and in Asia, to develop university-held intellectual property for use in the consumer market, of which Cybot and *Real Robots Magazine* was a part. Currently, he is the creative director of Kablamm Ltd. He is a visiting research fellow at the School of Systems Engineering, University of Reading, where he continues his research on public attitudes toward robots as well as promoting robotic engineering to schools through the Androids Advance initiative.

Will N. Browne received the B.Eng. degree in mechanical engineering from the University of Bath, United Kingdom, in 1993. He received his M.Sc. in energy and Ph.D. from the University of Wales, Cardiff, in 1994 and 1998, respectively. From 1998 to 2001, he worked as a postdoctoral research associate in the Control and Instrumentation Research Group, University of Leicester, United Kingdom. In October 2001, he was appointed to a lectureship in the Cybernetic Intelligence Research Group, University of Reading. His research focusses on cognitive robotics through enhancing genetics-based machine learning techniques with concepts from neuroscience such as abstraction.

Mark N. Gasson is a senior research fellow at the University of Reading, United Kingdom, and was awarded a Ph.D. in cybernetics in 2005. His research interests primarily focus on novel methods of human-machine interaction, most notably invasively implantable technologies and their application in the robotics domain. He also conducts collaborative, cross-discipline research on the impact that such emerging technologies have in the broad context of identity. He frequently delivers invited public lectures and workshops internationally.

Kevin Warwick is a professor of cybernetics at the University of Reading, England, where he carries out research in artificial intelligence, control, robotics, and cyborgs. He is also the director of the university's Knowledge Transfer Centre. In addition to publishing 500 research papers, Kevin is best known for his experiments in implant technology. He has been awarded higher doctorates (D.Sc.) both by Imperial College and the Czech Academy of Sciences, Prague. He was presented the Future of Health Technology Award at MIT. He is an honorary member of the Academy of Sciences, St. Petersburg, In 2004, he received the IEE Achievement Medal.

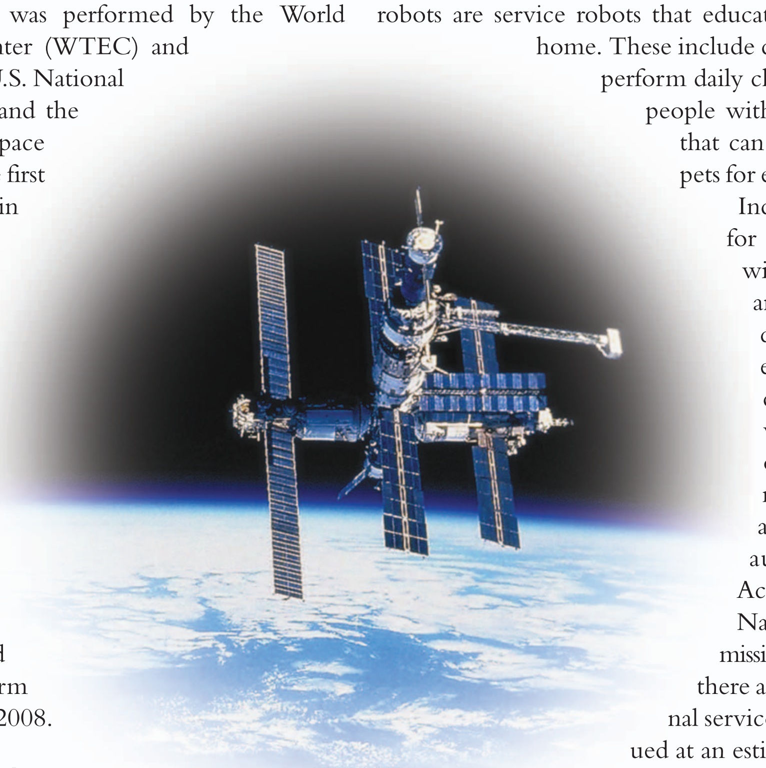
Address for Correspondence: Kevin Warwick, Cybernetics, University of Reading, Whiteknights, Reading, Berkshire RG66AY UK. E-mail: k.warwick@reading.ac.uk.

The Status of Robotics

Report on the WTEC International Study: Part II

BY GEORGE BEKEY AND JUNKU YUH

This article is the second part of a summary report on the status of robotics in the United States, Western Europe, Korea, Japan, and Australia. This report is based on visits to over 50 laboratories in 2004 and 2005. The study was performed by the World Technology Evaluation Center (WTEC) and supported primarily by the U.S. National Science Foundation (NSF) and the National Aeronautics and Space Administration (NASA). The first part of the report, published in *IEEE Robotics and Automation Magazine* in December 2007, concentrated on robotic vehicles, space robotics, and humanoid robots. This article summarizes the findings of the survey in industrial, service, and personal robots, biological and medical applications and networked robots. The full report may be accessed at http://wtec.org/robotics and will be published in book form by Imperial College Press in 2008.



© PHOTODISC

Industrial, Service, and Personal Robots

Robots can be classified into different categories depending on their function and the market needs for which they are designed. Here, we identify two major classes of robots: industrial robots and service robots. Within the latter class of robots, we will divide service robots into personal service robots and professional service robots, depending on their function and use. According to the Robotic Industries Association, an industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes that may be either fixed in place or mobile for use in industrial automation applications. The first industrial robot, manufactured by Unimate, was installed by General Motors in 1961. Thus, industrial robots have been around for over four decades. According to the International Federation of Robotics, another professional organization, a service robot is a robot that operates semiautonomously or fully autonomously for performing services useful to the well being of humans and equipment, excluding manufacturing operations. Personal robots are service robots that educate, assist, or entertain at home. These include domestic robots that may perform daily chores, assistive robots for people with disabilities, and robots that can serve as companions or pets for entertainment.

Industrial robots account for a US$4 billion market with a growth rate of around 4%. Most of the current applications are either in material handling or in welding. Spot welding and painting operations in the automotive industry are almost exclusively performed by robots. According to the United Nations Economic Commission for Europe (UNECE), there are over 20,000 professional service robots in use today valued at an estimated US$2.4 billion. If personal entertainment robots and domestic robots such as vacuum cleaners are included, this number is well over US$3.5 billion. The UNECE estimates that the value of service robots (both professional and personal) sold in 2005 was about US$5 billion.

Most of the industrial robotics industries are based in Japan and Europe. This is despite the fact that the first industrial robots were manufactured in the United States. At one time, General Motors, Cincinnati Milacron, Westinghouse, and General Electric made robots. Now, only Adept, a San Jose-based company, makes industrial robots in the United States. However, there are a number of small companies developing service robots in the United States. Companies such as iRobot (Figure 1), Mobile Robotics, and Evolution Robotics are pioneering new technologies.

The two largest manufacturers of industrial robots, ABB and Kuka, are in Europe. Over 50% of ABB is focused on automation products, and industrial robots are a big part of

Digital Object Identifier 10.1109/M-RA.2007.907356

their manufacturing automation, with an annual revenue of US$1.5 billion. ABB spends 5% of their revenue on research and development, with research centers all over the world. An ABB pick-and-place robot capable of performing two complete operations per second is shown in Figure 2. As in the automotive and other businesses, European companies outsource the manufacture of components (motors and sensors), unlike Japanese companies, which emphasize vertical integration. As in the United States, service robots are made by small companies, which include spin-offs launched from university research programs.

FANUC in Japan is the leading manufacturer of industrial robots, with products ranging from computer numerical control (CNC) machines with 1 nm Cartesian resolution and 10^{-5} degrees angular resolution to robots with 450 kg payloads and 0.5 mm repeatability. FANUC has 17% of the industrial robotics market in Japan, 16% in Europe, and 20% in North America. Kawasaki and Yaskawa follow FANUC as industry leaders. FANUC is also the leading manufacturer of CNC machines, with Siemens as its closest competitor. A Fujitsu household watchman robot, controllable from a cell phone, is shown in Figure 3. Another household assistant robot, with Internet connections and various modes for interaction with humans, was recently announced by Mitsubishi Heavy Industries under the name Wakamaru. Unlike the United States and Europe, the service robotics industry in Japan includes big companies such as Sony, Fujitsu, Mitsubishi, and Honda. The industry is driven by the perceived need for entertainment robots and domestic companions and assistants. In Korea, there are small robot companies, such as Yujin and Hanool, making vacuum cleaner robots and household assistant robots with Internet connections, while big companies such as Samsung also invest in robotics.

Figure 1. *Roomba vacuum cleaner (courtesy iRobot, Inc.).*

Figure 2. *ABB Flexpicker pick-and-place robot, the fastest robot produced by ABB (courtesy ABB).*

Biological and Medical Applications

The primary purpose of robotics in biology is to achieve high throughput in experiments related to research and development in the life sciences. These experiments involve the delivery and dispensation of biological samples/solutions in large numbers, each with very small volumes, for example in DNA sequencing (Figure 4). Another purpose of robotics for biological applications is for effective handling and exploration of molecular and cell biology. It is interesting to note that robotics-inspired algorithms are being used for molecular and cellular biology. Robotics for medical applications includes robotic surgery, diagnostic systems and devices, and rehabilitation. The latter includes robotic assistance to physical therapists as well as development of prosthetic and orthotic devices.

At the present time, the United States is leading other countries in both biological and medical applications. Among the leading biologically oriented laboratories are those of Deirdre Meldrum (Arizona State University, formerly with University of Washington), Lydia Kavraki (Rice University), and Yuan Zheng (Ohio State University) dealing with robots and robotics-inspired algorithms for molecular and cellular biology. The Engineering Research Center for Computer-Integrated Surgical Systems and Technology at Johns Hopkins University, supported by the NSF and directed by Russell Taylor, is a leader in the field. Intuitive Surgical Corp., is the developer of the

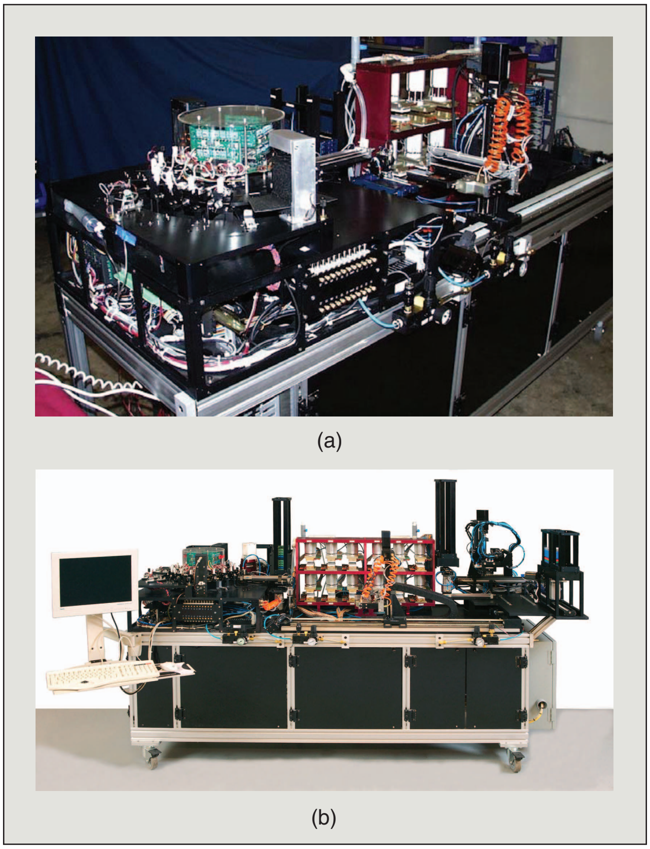
Figure 3. *MARON Robotic Watchman [courtesy PFU Limited (a subsidiary of Fujitsu)]. The sales of the MARON have been discontinued.*

highly successful Da Vinci robotic surgical system (Figure 5) designed to assist surgeons with complex medical operations. The system has been purchased by many hospitals throughout the world. As noted below, activity in this field is rapidly increasing in other countries.

In Japan, researchers at Nagoya University study noncontact cell manipulations using lasers and intravascular surgery based on a three-dimensional reconstructed cerebral arterial model using computed tomography images and an in vitro model of human aorta. Waseda University is well known for its research on legged locomotion. In recent years, Waseda University has also been active in the research on robotic surgery and walking-assistance devices for elderly people. For example, a new type of powered walker is capable of sensing pressure from both the left and right arms (Figure 6). Advanced Telecommunications Research (ATR) Computational Neuroscience Laboratories study brain function using a special computational approach called *understanding the brain by creating one*. In Korea, there are a few institutions for biological and medical applications, such as the Seoul National University for microelectromechanical systems (MEMS) and nanotechnologies for bioapplications, Korea Institute of Science and Technology for advanced techniques for cell handling, and Hanyang University for medical applications.

Research in biological and medical applications is growing rapidly in Europe. We cite here only a few examples. At the ETH, Zurich, Brad Nelson (formerly with the University of Minnesota) leads state-of-the-art research in nano and microrobotics devices for biological and medical applications. A major group at Scuola Superiore Sant'Anna in Pisa, Italy, works on a variety of applications from robot hands to miniaturized sensor capsules that can be swallowed to provide information on the state of the gastrointestinal tract (Figure 7). This project is part of the joint research with Korea Institute of Science and Technology funded by the Korean Government. Humboldt University in collaboration with the Fraunhofer Institute for Production and Design (IPK) has major programs in medical and surgical robotics, as illustrated in Figure 8.

Figure 5. *Da Vinci C surgical system (courtesy Intuitive Surgical, Inc.).*



(a)

(b)

Figure 4. *High-throughput system for DNA sequencing (University of Washington).*

Figure 6. *Walking assistance device (courtesy Waseda University).*

While the United States is currently the leader in biological and medical applications of robotics, activity is increasing in other countries. Progress is heavily dependent on advances in MEMS and nanotechnologies. Advances in this area depend on collaboration between life scientists, physicians, and engineers; such collaboration is challenging for all parties. We expect that future development will have a major economic impact.

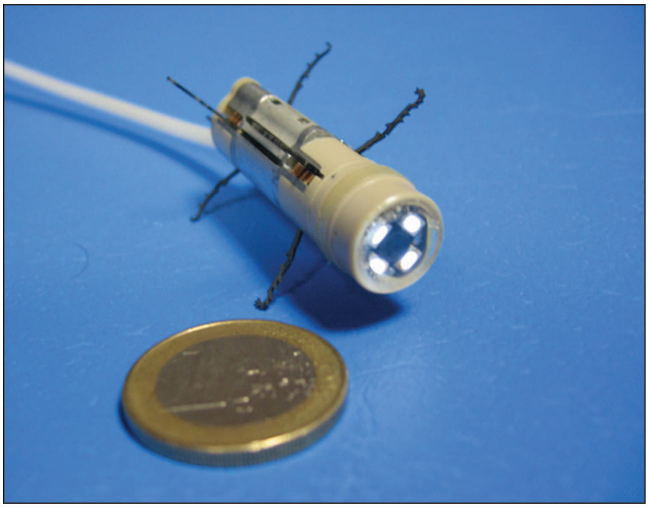


Figure 7. *Robotic capsular endoscope (courtesy Sant'Anna School of Advanced Studies).*

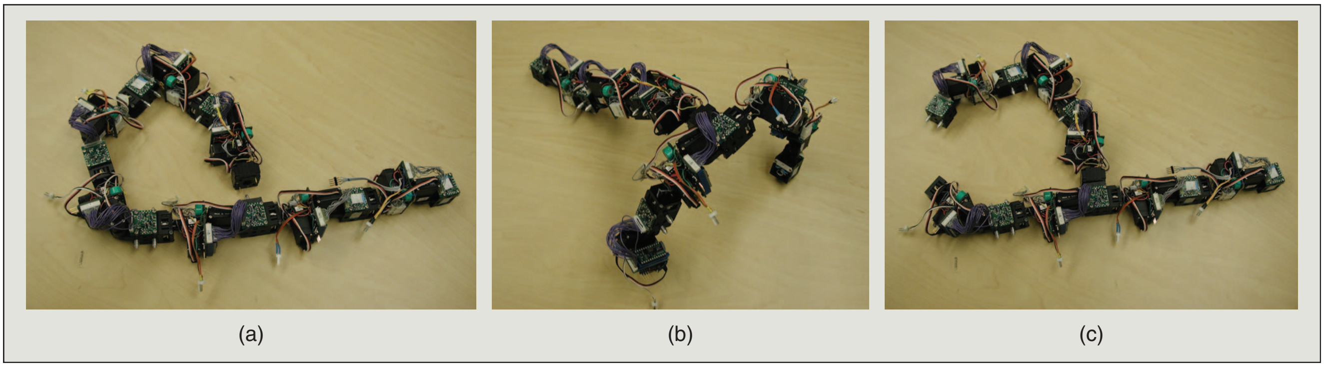
Figure 8. *Robotic surgical suite at Humboldt University/IPK, Germany (courtesy Tim Lüth, Humboldt University).*

Networked Robots

Networked robots refer to multiple robots operating together in coordination or cooperatively with sensors, embedded computers, and human users. Cooperation entails more than one entity working toward a common goal, while coordination implies a relationship between entities that ensures efficiency or harmony. Communication between entities is fundamental to both cooperation and coordination and hence the central role of the network. Embedded computers and sensors are now ubiquitous in homes and factories, and wireless ad hoc networks or plug-and-play wired networks are increasingly becoming commonplace. Robots are functioning in environments while performing tasks that require them to coordinate with other robots, cooperate with humans, and act on information derived from multiple sensors. In many cases, these human users, robots, and sensors are not collocated, and the coordination and communication happens through a network.

Networked robots allow multiple robots and auxiliary entities to perform tasks that are well beyond the abilities of a single robot. Robots can automatically couple to perform locomotion tasks and manipulation tasks that either a single robot cannot perform or would require a special purpose larger robot to perform. They can also coordinate to perform search and reconnaissance tasks, exploiting the efficiency that is inherent in parallelism. They can also perform independent tasks that need to be coordinated (for example, fixturing and welding) in the manufacturing industry. Networked robots also result in improved efficiency. Tasks such as searching or mapping, in principle, are performed faster with an increase in the number of robots. An increase in speed in manufacturing operations can be achieved by deploying multiple robots performing operations in parallel but in a coordinated fashion.

In the United States, there are numerous multirobot projects that may be termed *network robots*. Multiple robots can automatically connect themselves to each other to create a variety of locomotion systems. As shown in Figure 9, robotic modules can be reconfigured to morph into different systems, such as the "snake" in Figure 9(a) or the two-legged locomotion system in Figure 9(b). A transitional phase is shown in Figure 9(c).



(a) (b) (c)

Figure 9. *Reconfigurable robotic modules.*

The U.S. military routinely deploys unmanned vehicles that are reprogrammed remotely based on intelligence gathered by other unmanned vehicles, sometimes automatically. A major program of the Software for Distributed Robotics project from the U.S. Department of Defense demonstrated the ability to deploy 70 robots to detect intruders in an unknown building (University of Tennessee, University of Southern California, and Science Applications International Corp.) (Figure 10).

Home appliances now contain sensors and are becoming networked. As domestic and personal robots become more commonplace, it is natural to see these robots working with sensors and appliances in the house while cooperating with human users.

Japan has many national research and development programs related to this area. The five-year Ubiquitous Networking Project, established in 2003, has paved the way for a five-year Network Robots Project in 2004. The Network Robot Forum was established in 2003 and now has over a hundred prominent members from industry, academia, and government. Currently, Dr. Norihiro Hagita at the ATR in Japan leads a national project, the Networked Robot Project. The Korean Ministry of Information and Communication sponsors a large national research project, Ubiquitous Robotic Companion (URC), using network-based intelligent robots.

The European Union (EU) has several EU-wide coordinated projects on collective intelligence or swarm intelligence. The I-Swarm project at Karlsruhe and the Swarm-bot project at Free University of Brussels and École Polytechnique Fédérale de Lausanne (EPFL) are examples of swarm intelligence (Figure 11). The Laboratory for Analysis and Architecture of Systems in France has a strong group in robotics and artificial intelligence. This group has had a long history of basic and applied research in multirobot systems. The recent focus of this group is the COMET project, which integrates multiple unmanned vehicles for applications such as terrain mapping and firefighting (Figure 12).

Figure 10. *Demonstration from Software for Distributed Robotics project.*

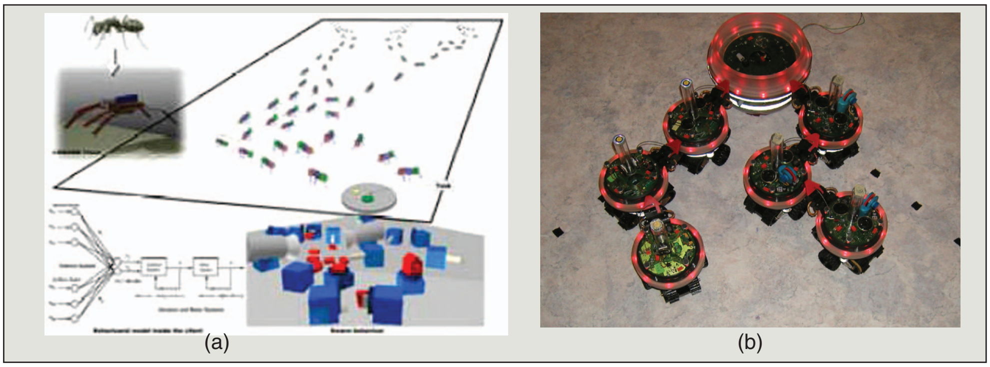


Figure 11. *EU project on Swarm Intelligence: (a) the I-Swarm project in Karlsruhe and (b) two Swarm-Bots comprising three s-bots each are cooperating in the transport of a red object. [courtesy Marco Dorigo, Swarm-bots project (www.swarm-bots.org)]*

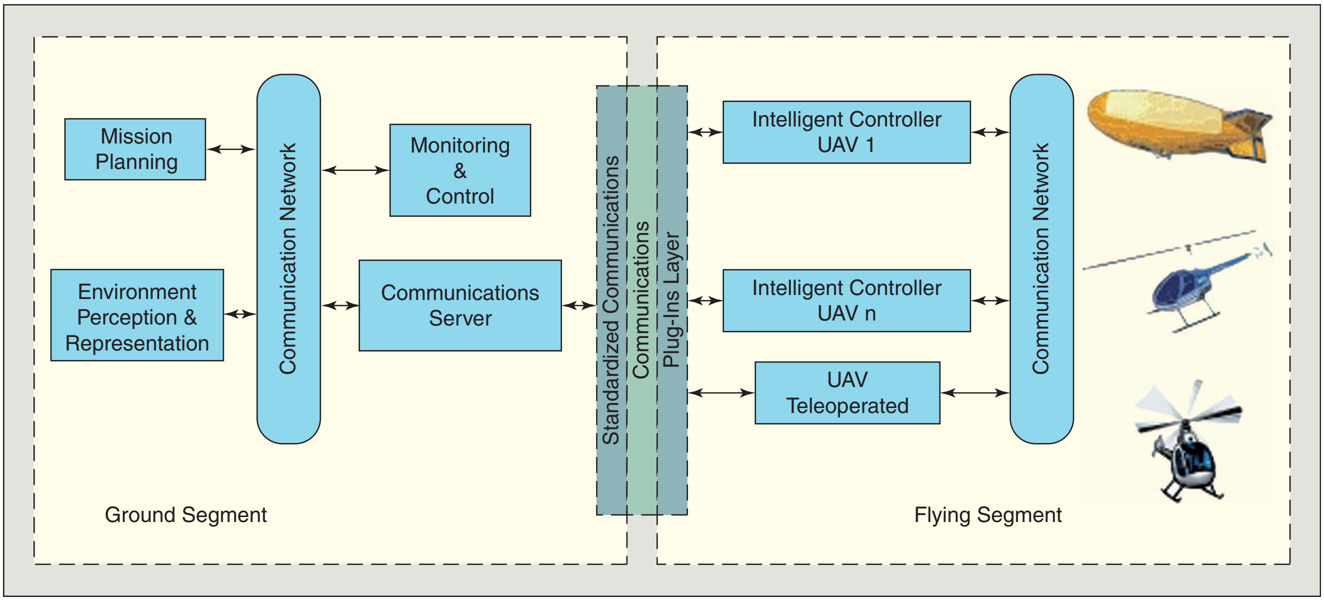


Figure 12. *The COMETS project at Institut National de Recherche en Informatique et en Automatique seeks to implement a distributed control system for cooperative detection and monitoring using heterogeneous unmanned aerial vehicles.*

Table 1. Qualitative robotics comparison chart.

Area	Degree or level of activity			
	United States	Japan	Korea	Europe
Input				
Basic, university-based research (individual, groups, and centers)	*****	***	***	***
Applied, industry-based research (corporate and national labs)	**	*****	****	****
National or multinational research initiatives or programs	**	*****	*****	****
University-industry-government partnerships and entrepreneurship	**	*****	*****	****
Output				
Robotic vehicles: military and civilian	****	**	**	**
Space robotics	***	**	N/A	***
Humanoids	**	*****	****	**
Industrial robotics: manufacturing	**	*****	**	****
Service robotics: nonmanufacturing	***	***	****	***
Personal robotics: home	**	*****	****	**
Biological and biomedical applications	****	**	**	****

The number of stars indicates the team's qualitative assessment of the relative strength in each technology and region.

Figure 13. *Australian unmanned aerial vehicle fleet with 45-kg aircrafts of 3-m wing spans with reconfigurable sensor payloads (University of Sydney).*

Pioneering work on decentralized state estimation, localization, and tracking has been done by the Australian Center for Field Robotics at the University of Sydney in Australia. They also have exceptional strength in coordinated unpiloted aerial vehicles, having demonstrated many impressive capabilities with multiple vehicles (Figure 13).

While there are more mature efforts in Japan and Europe to develop better sensors and robot hardware for robot networks, the United States has more impressive embodiments and imaginative applications of networked robots. Although it is hard to make such sweeping generalizations, the United States arguably still maintains the lead in control and networking, while Europe and the United States may have an edge over Japan in perception. Japan has a bigger investment in network robots and has done a better job of creating national agendas that will affect the development of networked robots for service applications and eventually for domestic assistance and companionship.

Summary and Conclusions

Robotics is a very active field worldwide. Japan, Korea, and the European community (EC) invest significantly larger funds in robotics research and development for the private sector than the United States. There are numerous start-up companies in robotics, both in the United States and abroad. Venture capital appears to be available.

The United States currently leads in such areas as robot navigation in outdoor environments, robot architectures (the integration of control, structure, and computation), and applications in space, defense, underwater systems, and some aspects of service and personal robots. Japan and Korea lead in technology for robot mobility, humanoid robots, and some aspects of service and personal robots (including entertainment). Europe leads in mobility for structured environments, including urban transportation. Europe also has significant programs in the care of the elderly and home service robotics. Australia leads in commercial applications of field robotics, in such areas as cargo handling and mining, and in the theory and application of localization and navigation.

In contrast to the United States, Korea and Japan have national strategic initiatives in robotics; the EC has EC-wide programs. In the United States, there is coordination only in military robotics. The United States lost its preeminence in industrial robotics at the end of the 1980s, so nearly all robots for welding, painting, and assembly are imported from Japan or Europe; it may lose its leading position in other aspects of robotics as well.

Some examples of funding disparities include the following: In the United States, NSF funding for robotics is about US$10 million per year, while annual funding for military robotics in the United States is estimated at more than US$200 million per year. In Japan, robotics useful in the home and urban environment was selected as one of the 62 priority technologies by the Japanese Government's Council for Science

The industry is driven by the perceived need for entertainment robots and domestic companions and assistants.

and Technology Policy for Japan's Third S&T Basic Plan (JFY2006-2010). In Korea, robotics has been selected as one of the ten areas of technology for economic growth; the total funding for robotics is about US$80 million per year. In Europe, a new program called Advanced Robotics has been funded at about US$100 million for three years.

A summary of the areas of major strength in various aspects of robotics in the United States, Asia, and Europe is given in Table 1. Entries under "Input" refer to the kinds of resources and organizations that are involved in research and development, while those under "Output" refer to the outcomes of research as key robotic products or applications.

A number of trends in technology are expected to have a major impact on robotics in the near future. The DARPA Grand Challenge in the United States in 2006 demonstrated the ability of autonomous vehicles to travel at average speeds in excess of 30 m/h over unknown terrain and in the presence of number of hazards and obstacle. The winning vehicles integrated sensors (including the global positioning system), complex and intelligent vision systems, and sophisticated navigation algorithms to accomplish the task. These and other aspects of the so-called intelligent vehicle technology are expected to influence the development of autonomous robotic vehicles in the near future. Developments in nanotechnology may lead to nanorobotic systems capable of self-assembly or perhaps manipulation of individual molecules for research in genetics and related areas. We have cited robotic surgery as a major current area of application. We expect that in the future increasingly autonomous systems will be able to operate within the body to identify and perhaps remove tumors. New imaging techniques, such as fMRI, combined with nanorobotics, may make possible dramatically new and different studies of brain function. Networks of sensors distributed throughout the environment may allow distributed robotic systems to interact and function as a collective system in the solution of environmental and other problems. These are just a sampling of the exciting potential of robotics. Clearly, the age of robotics is here, and we expect it to have an increasingly important effect on our lives, both as individuals and as societies.

Keywords

Robotics, robots, automation, autonomous systems.

George Bekey is an emeritus professor of computer science and founder of the Robotics Research Laboratory at the University of Southern California (USC). His research interests include autonomous robotic systems, applications of robots in biology and medicine, and human-robot interaction. He received his Ph.D. in engineering from the University of California, Los Angeles (UCLA). During his UCLA career, he was chairman of the Electrical Engineering Department and later of the Computer Science Department. From 1997 to 2001, he served as associate dean for Research of the School of Engineering. He has published over 240 papers and several books in robotics, biomedical engineering, computer simulation, control systems, and human-machine systems. His latest book, titled *Autonomous Robots*, was published by MIT Press in 2005. He is the founding editor of the *IEEE Transactions on Robotics* and the journal *Autonomous Robots.* He is a member of the National Academy of Engineering and a Fellow of the IEEE, the AAAI, and the AAAS. He has received numerous awards, including the IEEE Robotics and Automation Award in 2006, the IEEE Robotics and Automation Society Pioneer in Robotics Award in 2002, and the Engelberger Award for Robotics Education from the Robotics Institute of America in 2001. He officially retired from UCLA in 2002 but continues to be active on a part-time basis at the University, as well as in consulting and service on the advisory boards of several high technology companies. He is also affiliated with California Polytechnic University in San Luis Obispo, California, where he is a research scholar in residence and an adjunct professor of biomedical engineering. He is also connected with Patient Patents, Inc., a startup company in San Luis Obispo specializing in intellectual property development for medical devices.

Junku Yuh is the President of Korea Aerospace University. He served as program director of the Robotics and Computer Vision programs at the NSF in Washington, District of Columbia, and also as Head of NSF Tokyo Regional Office. He is an elected IEEE Fellow and received several prestigious awards, including a Lifetime Achievement Award from the World Automation Congress (2004) and an NSF Presidential Young Investigator Award (1991). He has published more than 120 technical articles and edited/coedited ten books in the area of robotics. He serves as editor-in-chief for the *International Journal of Intelligent Service Robots* (Springer). He also served as program chair of the 2003 IROS and program cochair of the 2001 and 2006 IEEE ICRA.

Address for Correspondence: George Bekey, Computer Science Department, University of Southern California, 941 W. 37th Place SAL 218, Los Angeles, CA 90089-0781 USA. E-mail: bekey@usc.edu.

A Software Tool for Hybrid Control

From Empirical Data to Control Programs

BY FLORENT DELMOTTE, TEJAS R. MEHTA, AND MAGNUS EGERSTEDT

When humans instruct each other on how to solve complex navigation tasks, they typically use statements like "Do A until B, then go toward C until you see D," and so on. Such statements contain only high-level descriptions of the task at hand, whereas lower-level issues concerning the exact path to follow or what muscle groups to use are implicitly assumed. This situation is in contrast with the classic control theoretic idea of prescribing the exact inputs or actuation signals that are needed for solving the task.

In this article, we present a software tool that bridges this gap by providing a framework in which robots and other dynamic systems can be controlled using automatically generated, high-level, symbolic control programs. In particular, the automated tools extract high-level control programs from observed behaviors (possibly biological) and then produce symbolic control laws that can be executed on mobile robots to mimic the observed behavior (Figure 1).

The main question investigated here can be summarized as follows: given the assumptions about what features and measurements are relevant to the original system, can we produce hybrid control strategies that mimic the observed behavior? The resulting hybrid strategies would, for example, allow us to generate control laws for teams of mobile robots that behave similarly to groups of ants or schooling fish. In other words, can we produce multimodal control strategies in an automated fashion from observed empirical data? These empirical data can be generated by nature (as in the group of ants) or from human-operated robots. From the standpoint of naturally occurring data, the short-term aim of such a research agenda would be to learn from nature, but a more lofty, long-term goal would be to understand naturally occurring control mechanisms based on hybrid control theory. From the human-operator standpoint, the goal would be to learn effective control strategies from examples.

Similar ideas have been pursued in [3] and [10], which are based on presegmented data or predefined collections of potential control laws. Alternative approaches can also be found in literature on motion captioning [17] or in the hybrid systems identification area [12]. However, these research programs are focused mainly on fitting piecewise linear, autonomous systems to the data. In this article, we take a different view, wherein the dynamics are given and the problem is to find control laws, together with the conditions for transitions between them, defined with respect to the system dynamics. At this point, it should be noted that some of the technical results presented here have appeared in [6] and [7], but in this article, we combine these results with new work and unify them into a tool for automating the process of extracting executable control strategies from empirical data. The tool is mode optimization and data extraction (MODEbox), which is available as a MATLAB toolbox [20]. The graphical user interface for MODEbox is depicted in Figure 2. MODEbox consists of four major modules: preprocessing, motion description language (MDL) capturing, MDL to automata, and simulation. Each of these modules and their functionality is discussed in this article, and further information is available on the MODEbox Web site [20].

The outline of this article is as follows: In the section on motion description languages, a brief introduction to MDLs is given. In the next section, the MODEbox and its basic

Digital Object Identifier 10.1109/MRA.2008.919022

1070-9932/08/$25.00©2008 IEEE

functionality is introduced. This is followed by a detailed explanation of the key modules that comprise MODEbox. In particular, the section on recovering MDL strings from data details how to recover MDL strings from data. The next two sections introduce a method for reducing the size of the recovered mode set to lower complexity and show how to construct a finite automaton that reproduces the recovered MDL strings. Finally, some examples are presented in the section on robots and ants.

Motion Description Languages

The main idea behind multimodal control is to define the different modes of operation, e.g., with respect to a particular task, operating point, or data source. These modes are then combined according to some discrete switching logic, and one attempt to formalize this notion is through the concept of an MDL [5], [8], [14].

Each string (or word) in an MDL corresponds to a control program that can be operated by the control system. Slightly different versions of MDLs have been proposed, but they all share the common feature that the individual atoms (or letters), concatenated together to form the control program, can be characterized by control-interrupt pairs. In other words, given a dynamic system

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^k$$
$$y = h(x), \quad y \in \mathbb{R}^p, \tag{1}$$

together with a control program $(k_1, \xi_1), \ldots, (k_z, \xi_z)$, where $k_i : \mathbb{R}^p \to \mathbb{R}^k$ and $\xi_i : \mathbb{R}^p \to \{0, 1\}$, the system operates on this program as $\dot{x} = f(x, k_1(h(x)))$ until $\xi_1(h(x)) = 1$. At this point, the next pair is read, and $\dot{x} = f(x, k_2(h(x)))$ until $\xi_2(h(x)) = 1$, and so on. Loosely speaking, the system evolves under triggers, i.e., it is controlled by the feedback law $k_i(y)$ until interrupt $\xi_i(y)$ goes from 0 to 1, at which point the next feedback law $k_{i+1}(y)$ is used. Note that the interrupts can also be time-triggered, but this can be incorporated by a simple augmentation of the state space.

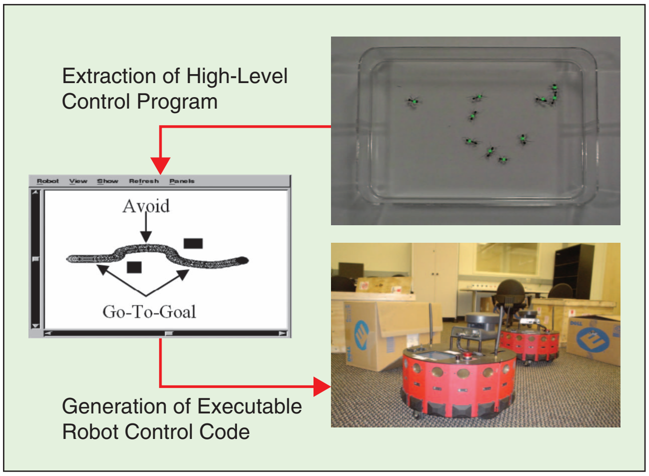
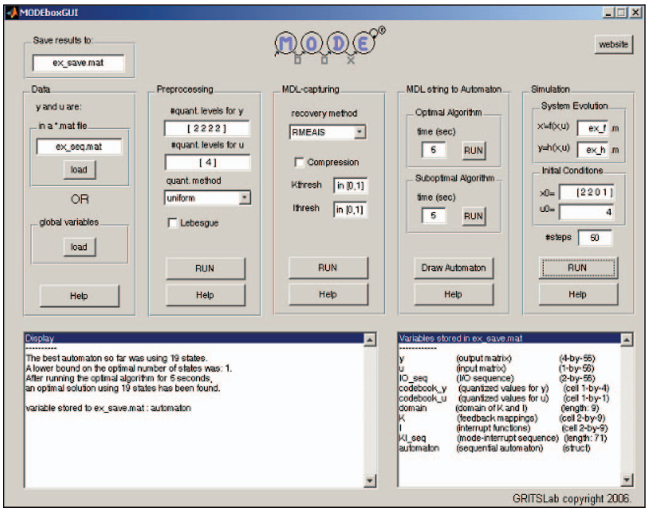


Figure 1. An example is shown in which ten roaming ants in a tank are tracked. Their behavior is analyzed and high-level control programs are extracted for controlling teams of mobile robots.



```
Save results to: ex_save.mat
y and u are:
in a *.mat file: ex_seq.mat
OR
global variables
#quant. levels for y [ 2 2 2 2 ]
#quant. levels for u [ 4 ]
quant. method uniform
☐ Lebesgue
recovery method RMEAIS
☐ Compression
Kthresh in [0,1]
lthresh in [0,1]
Optimal Algorithm time (sec) 5
Suboptimal Algorithm time (sec) 5
System Evolution
x'=f(x,u) ex_f .m
y=h(x,u) ex_h .m
Initial Conditions
x0= [2 2 0 1]
u0= 4
#steps 50
```

```
Display
----------
The best automaton so far was using 19 states.
A lower bound on the optimal number of states was: 1.
After running the optimal algorithm for 5 seconds,
an optimal solution using 19 states has been found.

variable stored to ex_save.mat : automaton
```

```
Variables stored in ex_save.mat
------------
y            (output matrix)             (4-by-56)
u            (input matrix)              (1-by-56)
IO_seq       (I/O sequence)              (2-by-56)
codebook_y   (quantized values for y)    (cell 1-by-4)
codebook_u   (quantized values for u)    (cell 1-by-1)
domain       (domain of K and I)         (length: 9)
K            (feedback mappings)         (cell 2-by-9)
I            (interrupt functions)       (cell 2-by-9)
KI_seq       (mode-interrupt sequence)   (length: 71)
automaton    (sequential automaton)      (struct)
```

Figure 2. Graphical user interface for MODEbox [20].

We first assume that the input-output (I/O) spaces ($\mathbb{U}$ and $\mathbb{Y}$, respectively) in (1) are finite, which can be obtained through a quantization function. This assumption can be justified by the fact that all physical sensors and actuators have finite range and resolution. Under this assumption, the set of all possible modes $\Sigma_{\text{total}} = \mathbb{U}^{\mathbb{Y}} \times \{0, 1\}^{\mathbb{Y}}$ is finite as well. Moreover, we can assume that a data point is measured only when the output or input changes value. This corresponds to the so-called Lebesgue sampling, in the sense of [2], which allows us to study only the I/O strings of finite length (given that the data were generated over a finite time horizon). Under this sampling policy, we can define a mapping $\delta : \mathbb{R}^n \times \mathbb{U} \to \mathbb{R}^n$ as $x_{q+1} = \delta(x_q, k(h(x_q)))$, given the control law $k : \mathbb{Y} \to \mathbb{U}$, with a new time update occurring whenever a new I/O value is encountered. For such a system, given the input string $(k_1, \xi_1), \ldots, (k_z, \xi_z) \in \Sigma^*$, where Σ^* is the free-monoid over a particular mode set $\Sigma \subseteq \Sigma_{\text{total}}$, the evolution of x is given by

$$\begin{cases} x_{q+1} = \delta(x_q, k_{l_q}(y_q)), & y_q = h(x_q) \\ l_{q+1} = l_q + \xi_{l_q}(y_q), \end{cases} \tag{2}$$

where $l_q \in \{1, \ldots, z\}$ is the position of the active mode within the input string at time q. (As an example, consider the situation in Figure 3, where a mobile robot is executing an MDL string consisting of alternating avoid-obstacle and go-to-goal modes.)

One of the objectives of MODEbox is to recover such strings of feedback-interrupt pairs from observed data, which will be discussed in more detail later.

MODEbox Basics

The basic functionality of MODEbox consists of four major modules (Figures 2 and 4). We will briefly describe each of these building blocks below and then present the theory behind their operation in later sections. Overall, MODEbox takes in a string of observed data (I/O pairs) and produces MDL strings consistent with the observed data. Next, MODEbox constructs a finite automaton capable of producing these MDL strings as a sample path, which can then be used as a control law to simulate similar trajectories or to control real systems.

The description of each module will be accompanied by a simple maze example to better illustrate the MODEbox operation. Note that this example is overly simplistic, but it is merely to be thought of as a vehicle for making certain operational aspects explicit. Let us assume that data are collected from the observations of a robot going through a maze as depicted in Figure 5(a). These data will be given by an I/O string, where the inputs are variables relevant to the system's control decisions and the outputs are signals possibly used to control the observed system. For this particular maze example, we choose the output to be $y = (y^1, y^2, y^3, y^4)$, where y^1, y^2, y^3, and y^4 correspond to the colors (i.e., 0 is black and 1 white) of the cells in front, to the left, behind, and to the right of the robot, respectively. The input u is the corresponding action (1 stands for go straight; 2, turn left; 3, U-turn; or 4, turn right) taken by the robot in response to the outputs. These data are sent to the preprocessing block.

Block 1: Preprocessing

In the first block in Figure 4, a data string consisting of I/O pairs is being read by MODEbox. The assumption is that the data are generated by a dynamic system $\mathbf{x}_{q+1} = f(\mathbf{x}_q, \mathbf{u}_q)$, $\mathbf{y}_q = h(\mathbf{x}_q)$, and the data string is given by $(\mathbf{y}_1, \mathbf{u}_1), \ldots, (\mathbf{y}_N, \mathbf{u}_N)$, where the outputs $\mathbf{y}_i \in \mathbb{R}^p$ and the inputs $\mathbf{u}_i \in \mathbb{R}^k$. Here, we use boldface to denote variables before they have been operated on by the preprocessing block. In fact, this string is operated on by the preprocessing block using three different, sequential components, namely, `Quantize`, `Encode`, and `Lebesgue`. `Quantize` produces a finite precision representation of the data string, `Encode` maps the quantized data strings to symbols, and `Lebesgue` reduces the length of the data string by making sure that no consecutive, symbolic I/O pairs are the same [2]. As a result, the output of this block is a new string $(y_1, u_1), \ldots, (y_r, u_r)$, where $r \leq N$ and $y_i \in \mathbb{Y}$, $u_i \in \mathbb{U}$. The user can specify how many regions (`quant.numbers`) the quantization should produce and what quantization method to use (`quant.method`). The user can select between four quantization methods: uniform, equidistributed, optimal pulse code modulation (PCM), and optimal differential pulse code modulation (DPCM). The choice of a quantization method

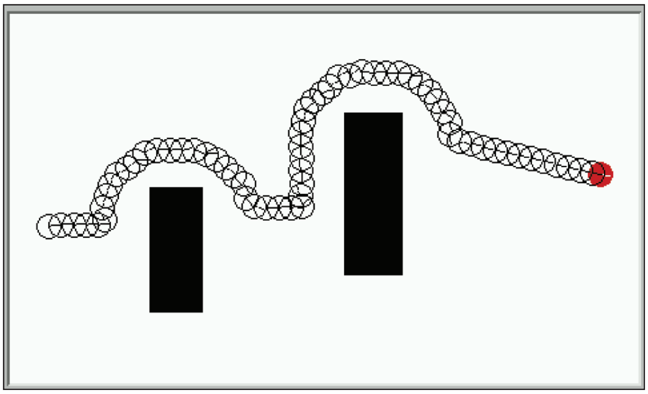


Figure 3. *A multimodal input string is used for negotiating two rectangular obstacles.*

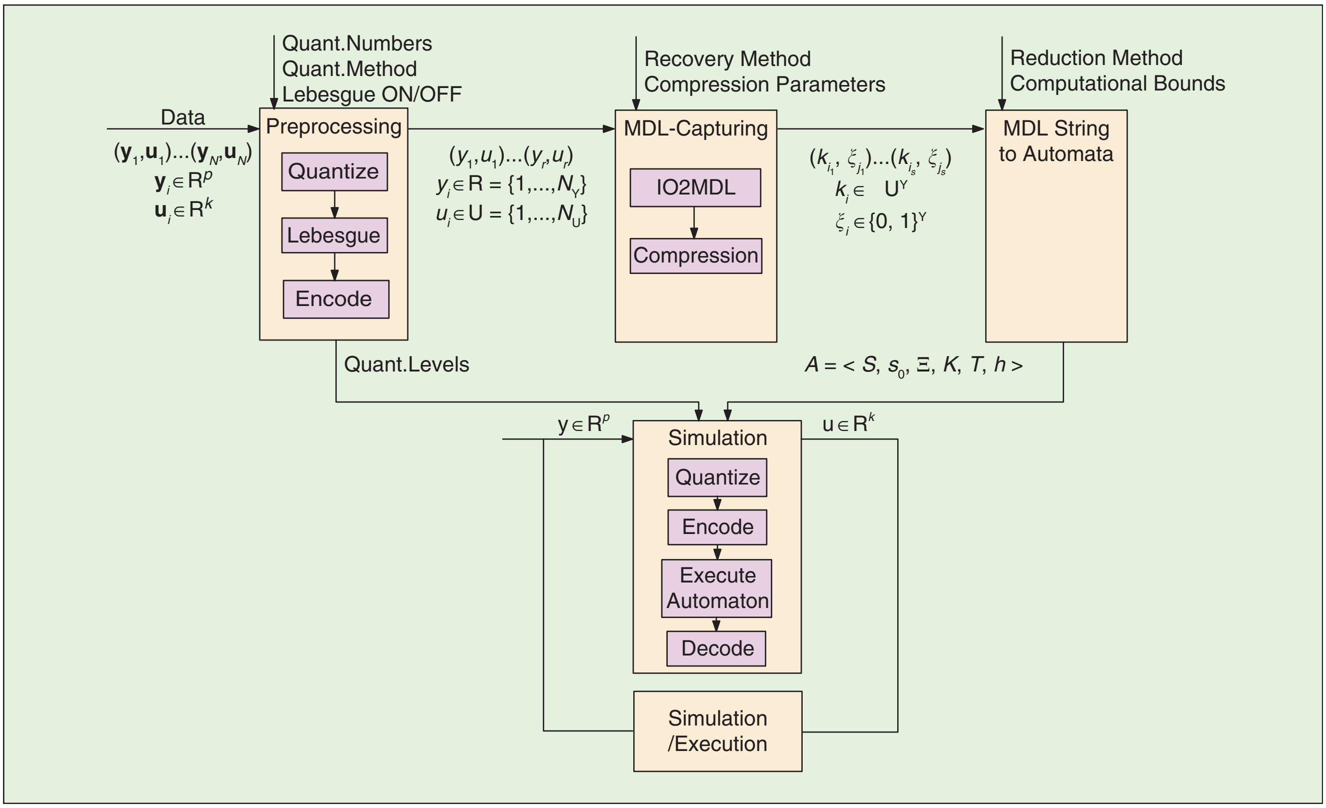


Figure 4. *Overview of the MODEbox operational units.*

The main idea behind multimodal control is to define different modes of operation in different situations.

and quantization levels is motivated by which one of these two opposite entities is to be preferred: the final model's complexity or the model's performance at approximating the original system. The user can also choose whether or not Lebesgue sampling should be employed (`Lebesgue ON/OFF`).

For the maze example, each data point comes from a small discrete set $Y \times U$, where $Y = \{0, 1\}^4$ and $U = \{1, 2, 3, 4\}$. Generally, the data could belong to countably or uncountably infinite sets. Here, data already belong to a small discrete set; therefore, we do not need to quantize the data any further. These data are now encoded into a discrete set of symbols $\mathbb{Y} \times \mathbb{U}$, where $\mathbb{Y} = \{1, 2, \ldots, 16\}$ and $\mathbb{U} = \{1, 2, 3, 4\}$, and the resulting I/O string is sent to the MDL-capturing block.

Block 2: MDL Capturing

The output from the preprocessing block is now fed into the MDL-capturing block. The method for recovering MDL strings from I/O data, `IO2MDL`, has been developed by the authors in [7], and different strategies for minimizing certain objectives have been devised. Although minimizing the so-called empirical specification complexity is the preferred objective, this is not always achievable, as noted in [7]. Instead, the user can choose from one of four methods that manage this complexity (this will be addressed in more detail in the next section). Once an MDL string is produced, the result is fed to `Compression`, where similar feedback laws and interrupt functions are identified and combined, which are based on user-specified `compression parameters` that set the thresholds (between zero and one) for how similar they need to be in order to be considered the same. The similarity measure is a normalized average entropy quantifying the uncertainty in the random variable $k_i(y)$ [and, respectively, $\xi_i(y)$], where i can be any of the modes under consideration. The resulting output from this block is a string $(k_{i_1}, \xi_{j_1}), \ldots, (k_{i_s}, \xi_{j_s})$, where $s \leq r$.

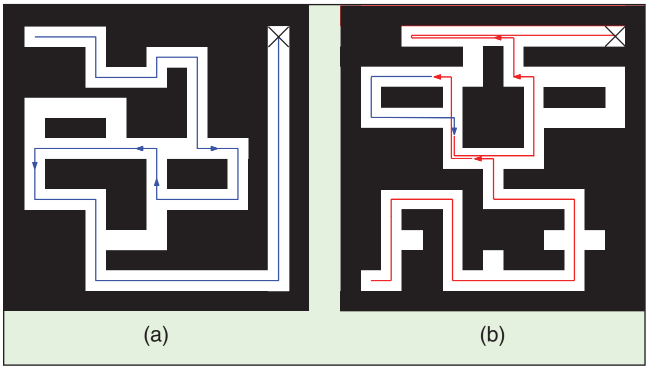


Figure 5. *(a) Trajectory of a robot going through a maze, which can be used as input data for MODEbox. The arrow and cross indicate the starting and finishing locations, respectively. (b) Example of a recovered mode sequence. Each arrow corresponds to the execution of one mode. Note that at the same moment an interrupt is triggered, a last action is taken by the active mode. For this reason, interrupts are located one step before the head of an arrow, i.e., when the robot faces a wall with two openings on its left and on its right.*

For the maze example, a close look at the trajectory in Figure 5(a) shows that the robot's behavior is almost always predictable. Indeed, the robot goes straight whenever possible (i.e., $u = 1$ when $y = (1, -, -, -)^T$) and turns left or right when it is the only possible choice (i.e., $u = 2$ when $y = (0, 1, 1, 0)^T$ and $u = 4$ when $y = (0, 0, 1, 1)^T$). The only unpredictable situation is when the robot is facing a wall, with two openings on the left and right (situation that we will denote $y_\perp = (0, 1, 1, 1)^T$). In this case, we see that the robot sometimes chose to turn left ($u = 2$) and sometimes right ($u = 4$). With this in mind, a recovery method minimizing the number of distinct modes could return the sequence $k_1\xi_1k_1\xi_1k_2\xi_2 k_1\xi_1k_1\xi_1k_1$, where mode 1 (respectively, mode 2) makes the robot turn left (right) when $y_\perp$ happens (i.e., $k_1(y_\perp) = 2$ and $k_2(y_\perp) = 4$), where the two modes behave the same for all other situations (i.e., whenever $y \neq y_\perp$, $k_1(y) = k_2(y)$) and where the interrupts functions would be the same, only triggering when $y_\perp$ happens (i.e., $\xi_1(y_\perp) = \xi_2(y_\perp) = 1$). This particular mode sequence is depicted in Figure 5(b). Because the two modes are almost identical, it is conceivable to merge them into a single mode (k_{12}, ξ_{12}). In this case, $k_{12}(y_\perp)$ is now a random variable (and not deterministic as before) as $k_{12}(y_\perp) = 2$ or 4. The resulting compressed mode sequence would be $k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}$. Now this mode sequence will be sent to the MDL to automata block.

Block 3: MDL to Automata

The resulting MDL string can be thought of as a sample path generated on a finite automaton, where the output function $h(s) = k$ returns the feedback law that the system should use in state s. Transitions in the automaton are triggered by the corresponding interrupts. If we let $\mathcal{K}$ and Ξ denote the set of feedback laws and interrupt functions, respectively, the MDL to automata block produces a finite automaton $A = \langle S, s_0, \Xi, \mathcal{K}, T, h \rangle$, where S is the state space, s_0 the initial state, $T : S \times \Xi \rightarrow S$ is the transition relation, and $h, \mathcal{K}$, and Ξ are as previously defined. Moreover, A should not only be such that the MDL string is a sample path of A but should also be small in the sense of state-space cardinality. This subject is considered in the "From MDL Strings to Finite Automata" section, where algorithms for finding such automata are discussed. The user is free to choose between an optimal and a suboptimal algorithm through the user input `reduction method`.

For the simple maze example, the optimal automaton is easily recovered, and the results are shown in Figure 6. An automaton corresponding to the recovered mode string without compression ($k_1\xi_1k_1\xi_1k_2\xi_2k_1\xi_1k_1\xi_1k_1$) is shown in Figure 6(a), and an automaton corresponding to the compressed mode string ($k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}\xi_{12}k_{12}$) is shown in Figure 6(b).

Block 4: Simulation or Execution

Once A has been produced, it can be used as a hybrid control law to mimic observed behavior or control real systems in the simulation block. The last block in the MODEbox flow diagram represents this situation, where externally obtained measurements $\mathbf{y} \in \mathbb{R}^p$ (either through simulation or a real experiment) are quantized and encoded [with the same quantization levels (`quant.levels`) used in the preprocessing block] to produce symbolic measurements $y \in \mathbb{Y}$. These measurements are then used for driving the finite automaton through `ExecuteAutomata`, and the corresponding control symbols $u \in \mathbb{U}$ are computed and decoded to produce executable control signals $\mathbf{u} \in \mathbb{R}^k$. This is the only block in the toolbox that requires any significant user input since each simulation is application specific.

The control procedure recovered in the previous block is now applied to a robot to navigate through a new maze depicted in Figure 7. We assume that the control automaton used in this example is the one depicted in Figure 6(b). First, we define two functions $f(\mathbf{x}, \mathbf{u})$ and $h(\mathbf{x})$ reflecting the state evolution and observation of the robot traversing the maze. At each time increment, the real measurement $\mathbf{y}_q = h(\mathbf{x}_q)$ is quantized and encoded into a symbolic measurement $y_q \in \mathbb{Y}$. A symbolic input $u_q \in \mathbb{U}$ is then computed. It corresponds to the value $k_{12}(y_q)$ of the feedback mapping of the active mode. This symbolic input is decoded into an executable control $\mathbf{u} \in \mathbb{R}^k$. By applying this control to the robot, the state evolves according to $\mathbf{x}_{q+1} = f(\mathbf{x}_q, \mathbf{u}_q)$. Figure 7 shows how the simulated system exhibits trajectories or behaviors similar to those of the system used for training.

The remainder of this article is devoted to presenting the theory related to the MODEbox modules in detail and showcasing the MODEbox operation through some examples.

Recovering MDL Strings from Data

In [7], we presented different methods for recovering multimodal control strings from empirical data in a theoretical setting. In particular, the problem was to produce strings in a given MDL that were consistent with the empirical I/O strings. At the same time, the control programs were viewed as having an information-theoretic content, i.e., they could be coded more or less effectively. For this, we define a complexity measure, the empirical specification complexity, which corresponds to the number of bits needed to specify a mode string σ with an optimal coding scheme:

$$\mathcal{S}^e(\sigma) = |\sigma|\mathcal{H}^e(\sigma),$$

where $|\sigma|$ is the length of the mode sequence σ and $\mathcal{H}^e(\sigma)$ is its entropy.

The minimization of $\mathcal{S}^e(\sigma)$ is, in fact, very hard to address directly and is still an open problem. However, the easily established bound $\mathcal{S}^e(\sigma) \leq |\sigma| \log_2(M(\sigma))$, where $M(\sigma)$ is the number of distinct modes in σ, allows us to focus on the following more tractable subproblems:

- minimizing the length of the mode sequence $|\sigma|$, which was solved using dynamic programming in [1]
- minimizing the number of distinct modes $M(\sigma)$, which was solved in [6] and relies on the initial construction of mode sequences, where the interrupts always trigger and are referred to as always interrupt sequences (AIS).

> ***Application of the MODEbox tool is illustrated on two different examples involving robots and ants.***

It is important to note that the solutions to these problems are not unique; hence, they can be further processed to reduce complexity. In particular, [7] presents additional algorithms to minimize entropy and reduce the length of mode strings given by the AIS solution while preserving consistency. MODEbox supports four different methods for recovering MDL strings, and the users can select among them based on their preference. The supported recovery methods are as follows: `MinLength` (minimizes the string length), `AIS` (minimizes the distinct number of modes), `RAIS` (reduces the length of a string produced by the `AIS` to further reduce the specification complexity), and `RMEAIS` (reduces the length of the lowest entropy `AIS` string). Although the `RMEAIS` produces the lowest complexity strings among `AIS`, `RAIS`, and `RMEAIS`, it does not necessarily produce strings with complexity lower than the `MinLength`.

Reducing the Size of the Motion Alphabet

The conversion from mode sequences to executable I/O automata requires splitting the motion alphabet Σ into two

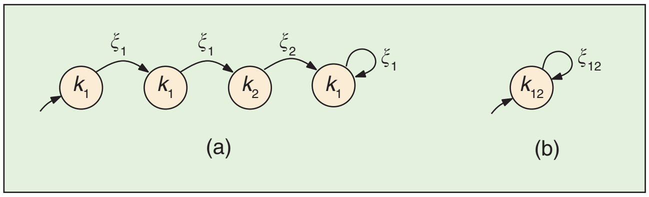


Figure 6. *(a) Automaton corresponding to the noncompressed recovered mode string. (b) Automaton corresponding to the compressed mode string.*

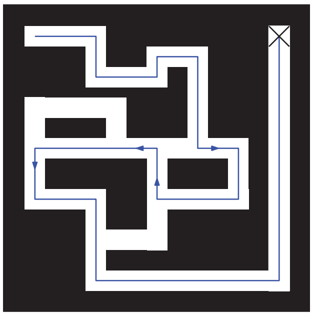


Figure 7. *Trajectory of a robot navigating through a maze using a controller derived by MODEbox.*

alphabets: the input alphabet Ξ of interrupt functions and the output alphabet $\mathcal{K}$ of feedback mappings. To produce small automata (in terms of the number of states), a preliminary task consists of reducing the size of Ξ and $\mathcal{K}$ by merging the combinations of elements that look similar. To do so, we first need to define a measure of similarity. Consider merging n distinct feedback mappings $k_{i_1}, k_{i_2}, \ldots, k_{i_n}$, resulting in the creation of a macro-feedback mapping K_I, where we let $I = \{i_1, \ldots, i_n\}$. Here, we choose to merge feedback mappings, but the same ideas, definitions, and algorithm apply for interrupt functions. Note that for a given $y \in \mathbb{Y}$, two distinct feedback mappings may map y to two different inputs, i.e., $\exists(\alpha, \beta) \in I^2$ such that $k_\alpha(y) = u$ and $k_\beta(y) = v$ with $u \neq v$. For this reason, we choose to represent $K_I(y)$ as a random variable. Consequently, the macro-feedback mapping K_I is a random process defined on $\mathbb{Y}$.

Now, for a given $y \in \mathbb{Y}$, the probability mass function of $K_I(y)$ can be recovered by

$$\begin{aligned} p_{K_I(y)}(u) &= \Pr\{K_I(y) = u\} \\ &= \frac{\operatorname{card}\{q | y_q = y, u_q = u, \text{ and } m_q \in I, q = 1, \ldots, N\}}{\operatorname{card}\{q | y_q = y \text{ and } m_q \in I, q = 1, \ldots, N\}}, \end{aligned}$$

where m_q refers to the active mode at time q, card denotes cardinality, and N is the number of data points in the I/O string. Next, we define the entropy of the random variable $K_I(y)$:

$$H(K_I(y)) = -\sum_{u \in \mathbb{U}} p_{K_I(y)}(u) \log(p_{K_I(y)}(u)),$$

with the following bounds

$$0 \leq H(K_I(y)) \leq \log(n).$$

Finally, we define an entropy measure for the random process K_I. It is the normalized average of the entropies of all random variables $K_I(y)$, $y \in \mathbb{Y}$:

$$H(K_I) = \frac{1}{\log(n)} \sum_{y \in \mathbb{Y}} p_Y(y) H(K_I(y)).$$

Note that in this definition, the output is also considered as a random variable Y. We propose three methods for establishing its probability mass function $p_Y : y \rightarrow p_Y(y) = \Pr\{Y = y\}$:

1) y is a uniform random variable. In this case, $p(y) = 1/|\mathbb{Y}|$
2) we look at the proportion of y when any mode in I is active, i.e., $p_Y(y) = (\operatorname{card}\{q | y_q = y \text{ and } m_q \in I, q = 1, \ldots, N\})/(\operatorname{card}\{q | m_q \in I, q = 1, \ldots, N)\}$
3) we look at the proportion of y in the whole observed output sequence, i.e., $p_Y(y) = (\operatorname{card}\{q | y_q = y, q = 1, \ldots, N\})/N$.

The total entropy $H(K_I)$ is a measure of the average uncertainty in the random process K_I. It varies from zero to one, where

- $H(K_I) = 0$ means that there is no uncertainty in K_I, i.e., for all $y \in \mathbb{Y}$, all modes in I map y to the same input value
- $H(K_I) = 1$ means that the uncertainty in K_I is maximal, i.e., for all $y \in \mathbb{Y}$, all modes are equally active, and they all map y to a distinct input value.

In other words, the two extreme values are reached when the n feedback mappings are either equal ($H(K_I) = 0$) or completely different ($H(K_I) = 1$). We propose to define a threshold value $\gamma_k \in [0, 1]$ so that if $H(K_I) \leq \gamma_k$, the feedback mappings are considered similar enough to be merged. On the basis of this idea, we suggest the following alphabet reduction algorithm.

Given an alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ and a reduction threshold γ, we reduce $\mathcal{A}$ in the following manner:

```
$\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$
$p = n$
while $p>1$
  find the combination $C^*$ of $p$ elements from $\mathcal{A}$ with minimum entropy $H(C^*)$
  if $H(C^*)<\gamma$
    merge the elements of $C^*$
    update $\mathcal{A}$
  else
    $p = p - 1$
  end
end
```

This alphabet reduction algorithm serves many purposes. First, as mentioned earlier, this algorithm splits the motion alphabet (Σ) obtained from the earlier step into two alphabets (Ξ and $\mathcal{K}$) so that the recovered hybrid strings can be viewed as I/O strings of a hybrid automaton. Second, this process facilitates noise reduction, which is naturally occurring when dealing with empirical data. Additionally, the reduction in the size of the alphabet makes the construction of automata more tractable. Finally, this process also leads to a stochastic interpretation of the feedback and interrupt functions, which is sometimes more natural than a deterministic interpretation. However, it should be noted that the computational burden associated with this algorithm may be quite high in that every mode combination must be computed.

From MDL Strings to Finite Automata

After applying the alphabet reduction algorithm presented earlier, our recovered string is of the form $k_{i_1}\xi_{j_1}k_{i_2}\xi_{j_2}\cdots$, where we can think of k_{i_q} as the output from the underlying finite automaton in state s_q, and ξ_{j_q} as the corresponding event that triggers a transition from state s_q to s_{q+1}. The question then becomes, can we recover this underlying automaton? And, is it unique? The answer to the second question is "no," and we will focus our attention on trying to recover minimal automata, but first we need to establish some notation.

An output automaton is a sextuple $\langle S, \Sigma, Y, s_0, T, h\rangle$, where S is the finite set of states, Σ is the input alphabet, Y is the output alphabet, $s_0 \in S$ is the initial state, $T \subseteq S \times \Sigma \times S$ is the set of allowable transitions, and $h : S \rightarrow Y$ is the output function. For our purposes, the input and output alphabet will be the finite set of interrupts (Ξ) and feedback laws ($\mathcal{K}$), respectively. We define a path π as a finite alternating sequence, $s_{i_1}\sigma_{j_1}s_{i_2}\sigma_{j_2}s_{i_2}\cdots\sigma_{j_{n-1}}s_{i_n}$, of states and inputs, starting and ending with a state. We say that a path is executable on A if $s_{i_1} = s_0$, and each input transitions the state preceding it to the one following it, i.e., $(s_{i_q}, \sigma_{j_q}, s_{i_{q+1}}) \in T$for all q. An I/O path π_y is an alternating sequence, $y_{\ell_1}\sigma_{j_1}y_{\ell_2}\sigma_{j_2}\cdots\sigma_{j_{n-1}}y_{\ell_n}$, of outputs and inputs, starting and ending with an output. We say that an I/O path is executable on A if there exists an executable path $s_{i_1}\sigma_{j_1}s_{i_2}\sigma_{j_2}\cdots s_{j_n}$ such that for all q, $h(s_{i_q}) = y_{\ell_q}$. Now, given an I/O path $\pi_y = y_{\ell_1}\sigma_{j_1}\cdots y_{\ell_n}$, the problem under consideration here is to find the smallest deterministic output automaton $A = \langle S, \Sigma, Y, s_0, T, h\rangle$ on which π_y is executable.

Note that for a given I/O path $\pi_y = y_{\ell_1}\sigma_{j_1}\cdots y_{\ell_n}$, there always exists at least one output automaton A on which π_y is executable. It is the automaton that jumps to a new state at each transition. This sequential output automaton has exactly n states. The set of automata that can execute π_y is thus nonempty, and there always exists a solution to the problem defined earlier.

In fact, this problem is related to the problem of producing minimal equivalent automata since one could consider applying a state reduction algorithm to the sequential output automaton derived earlier. However, this automaton is not necessarily complete, which is a necessary condition for applying such algorithms [4]. There is, however, an abundance of literature pertaining to the reduction of incompletely specified automaton. This problem is known to be NP-complete [18]. The various approaches for solving this problem can be categorized as either exact or heuristic based. The standard approach for this problem is based on enumeration of the set of compatible states and the solution of a binate covering problem [15]. A different approach for exact minimization not based on enumeration is presented in [16], and [19] presents heuristic-based algorithms that significantly reduce run time while obtaining correct results in most cases. Finally, what we are aiming for can also be cast in terms of finding the smallest automaton that simulates a particular sequential output automaton by using the terminology in [9].

As mentioned earlier, MODEbox allows the user to select between an optimal and suboptimal algorithm.

The optimal algorithm is an exhaustive search algorithm. The set of all consistent automata is progressively constructed by reading the I/O path from left to right. At the end of this search, the automaton with the fewest number of states is the optimal solution. As the length of the I/O path increases, the number of possible candidates quickly increases as well. Indeed, it is easy to show a superexponential relation for a worst-case scenario. To contain this explosion, we apply two heuristic modifications to the algorithm.

- We limit the memory resources so that only a fixed number of automata M can be stored. When this number has been reached, new candidate automata are automatically discarded.
- We set a maximum size $c_{\max}$. If an automaton has more than $c_{\max}$ states, it is discarded.

In MODEbox, we encode these heuristics in a high-level iterative algorithm that slowly increases the bounds until a solution is reached. Although this modification reduces the computation time, the problem remains NP-complete and quickly becomes numerically intractable for long I/O paths. For this reason, the user can specify a time limit after which, if no solution has been found, the algorithm returns a lower bound on the size of the optimal solution.

MODEbox allows the user to select between an optimal and suboptimal algorithm.

Instead of keeping up with all consistent automata (as done in the optimal algorithm), the suboptimal algorithm constructs a single automaton consistent with the given I/O path by greedily selecting one of them randomly. As before, the automaton is constructed by progressively reading the I/O path from left to right. At each iteration, we identified potential (consistent) candidates for the next state and chose one of them randomly. A new state is added only when previously created states cannot be used. This results in a small consistent automaton, but the random process in this selection cannot guarantee that the solution is optimal. However, this algorithm is significantly faster than the previous one. In fact, it has cubic complexity with respect to the length of the I/O path.

Robots and Ants

We now illustrate how MODEbox can be put to use in two particular examples, involving mobile robots and ants. In the first example, the system should recover a multimodal control strategy, given I/O data obtained when controlling a mobile robot using two distinct dynamic behaviors, whereas the other example is given by the observation of a biological system.

Control of Mobile Robots

For this example, originally reported in [1], we use a unicycle-type robot, i.e., its kinematics are given

$$\begin{aligned}\dot{x} &= v\cos\phi\\ \dot{y} &= v\sin\phi\\ \dot{\phi} &= \omega,\end{aligned}$$

where (x, y) is the position and ϕ is the heading of the robot. The translational and angular velocities (v, ω) are the control variables, and we quantize them according to $u \in \{(v, \omega) | v \in V, \omega \in \Omega\}$, where

$$\begin{aligned}V &= \{\text{slow},\ \text{medium},\ \text{fast}\},\\ \Omega &= \{\text{fast left},\ \text{slow left},\ \text{straight},\ \text{slow right},\ \text{fast right}\}.\end{aligned}$$

In a similar manner, the measurements made by the robot are sampled and quantized to produce an output string. We let

The robot is driven manually, and the resulting input-output string serves as input to MODEbox so that we can control the robot through the recovered control strategies.

$y \in \{(y^1, y^2, y^3) | y^1 \in Y^1, y^2 \in Y^2, y^3 \in Y^3\}$, where y^1 gives the distance to the closest obstacle, y^2 gives the relative angle to the closest obstacle, and y^3 gives the relative angle to the goal. By letting the angular quantization be given by the positions of the sensors on the sonar ring, as shown in Figure 8(a), we get

$$Y^1 = \{\text{close}, \text{ medium}, \text{ far}\},$$
$$Y^2 = \{1, 2, \ldots, 8\}, \quad Y^3 = \{1, 2, \ldots, 8\}.$$

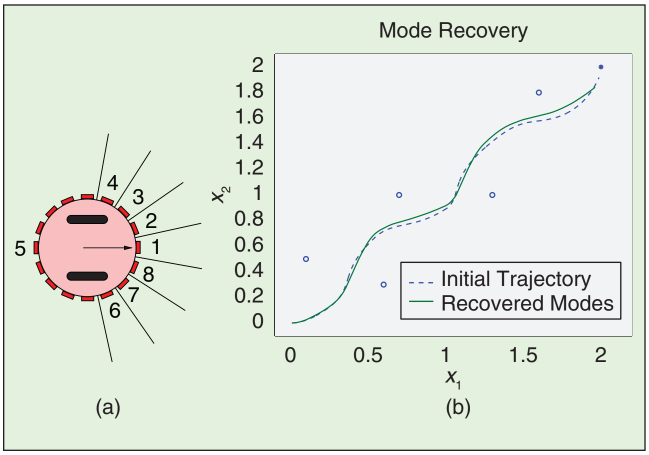


Figure 8. *(a) Angular quantization. (b) Effect of controlling the robot using a recovered mode string.*

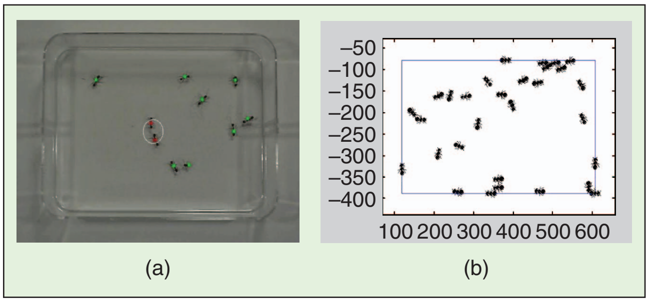


Figure 9. *(a) Ten ants are moving around in a tank. The circle around two ants means that they are docking or exchanging information. (b) Simulation environment depicting 30 simulated ants.*

In the experiment, we let the actual robot be controlled using

$$v = v_0 \min\{1, (d_{ob}/D)^2\}$$
$$\omega = C_{ob}(d_{ob})(\phi_{ob} + \pi - \phi) + C_g(\phi_g - \phi),$$

where D is a specified safety distance, d_{ob}, ϕ_{ob} is the distance and direction to the closest obstacle, ϕ_g is direction to the goal, and the gain $C_{ob}(d_{ob}) = 0$ if $d_{ob} \geq D$ and $(d_{ob} - D)/d_{ob}^3$ otherwise.

The robot is driven manually, and the resulting I/O string serves as input to MODEbox so that we can control the robot through the recovered control strategies. The results are shown in Figure 8(b), where the real execution (dashed line) is shown together with the effect of controlling the robot using the MODEbox tool.

Ants in a Tank

We now consider an example from [6], where ten ants (*Aphaenogaster cockerelli*) are placed in a tank with a camera mounted on top [Figure 9(a)]. A 52-s movie is used to extract the Cartesian coordinates x and y and the orientation θ of every ant every 33 ms using a vision-based tracking software. This experimental setup is provided by Tucker Balch and Frank Dellaert at the Georgia Institute of Technology BORG Lab [13], [21].

From this experimental data, an I/O string at each sample time k is found for each ant i as follows:

- the input u_k is given by (u_k^1, u_k^2), where u_k^1 is the quantized angular velocity, and u_k^2 is the quantized translational velocity of the ant i at time k
- the output y_k is given by (y_k^1, y_k^2, y_k^3), where y_k^1 is the quantized angle to the closest obstacle, y_k^2 is the quantized distance to the closest obstacle, and y_k^3 is the quantized angle to the closest goal of ant i at time k.

An obstacle is either a point on the tank wall or an already visited ant within the visual scope of ant i, and a *goal* is an ant that has not been visited recently.

These data are fed to MODEbox, and the resulting control programs can be used to simulate ant behavior. Figure 9(b) shows an example of this when 30 ants are simulated based on the recovered hybrid control strategy.

Conclusions

In this article, we introduce the MODEbox tool for automatically producing hybrid, multimodal control programs from data. In particular, given an I/O string, four distinct operational units are introduced.

- *Preprocessing*: The real-valued I/O strings are transformed into strings of symbols through quantization, Lebesgue sampling, and encoding operations.
- *MDL capturing*: Low-complexity strings of symbolic feedback-interrupt pairs are produced that are consistent with the empirical data.
- *MDL to automata*: Small finite automata are produced in such a way that outputs correspond to feedback mappings, and transition events correspond to interrupts.
- *Simulation or execution*: Simulated or real systems can be controlled using the obtained hybrid control strategy.

The application of the MODEbox tool is illustrated on two different examples involving robots and ants.

Acknowledgments

We thank Tucker Balch and Frank Dellaert for the ant data that were made available through the BORG Lab at the Georgia Institute of Technology. Part of the work on which this article is based was conducted by Adam Austin and Johan Isaksson, and their participation is greatly appreciated. This work was supported by the National Science Foundation through the programs EHS NSF-01-161 (Grant 0207411) and ECS NSF-CAREER award (Grant 0237971).

Keywords

Multimodal control, hybrid systems, mobile robots.

References

[1] A. Austin and M. Egerstedt, Mode reconstruction for source coding and multimodal controlin *Hybrid Systems: Computation and Control*, O. Maler and A. Pnueli, Eds. Prague, The Czech Republic: Springer-Verlag, 2003, pp. 36–49.

[2] K. J. Åström and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, 2002, pp. 2011–2016.

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 926–939, Dec. 1998.

[4] R. G. Bennetts, J. L. Washington, and D. W. Lewin, "A computer algorithm for state table reduction," *Radio Electron. Eng.*, vol. 42, no. 4, pp. 513–520, 1972.

[5] R. W. Brockett, "On the computer control of movement," in *Proc. IEEE Conf. Robotics and Automation*, New York, 1988, pp. 534–540.

[6] F. Delmotte and M. Egerstedt, "Reconstruction of low-complexity control programs from data," in *Proc. IEEE Conf. Decision and Control*, Atlantis, Bahamas, 2004, pp. 1460–1465.

[7] F. Delmotte, M. Egerstedt, and A. Austin, "Data-driven generation of low-complexity control programs," *Int. J. Hybrid Syst.*, vol. 4, no. 1-2, pp. 53–72, 2004.

[8] M. Egerstedt and R. W. Brockett, "Feedback can reduce the specification complexity of motor programs," *IEEE Trans. Automat. Contr.*, vol. 48, pp. 213–223, Feb. 2003.

[9] A. Girard and G. J. Pappas, "Approximate bisimulations for nonlinear dynamical systems," in *Proc. 44th IEEE Conf. on Decision and Control*, 2005, pp. 684–689.

[10] D. Grünbaum, S. Viscido, and J. K. Parrish, "Extracting interactive control algorithms from group dynamics of schooling fish," in *Proc. Block Island Workshop on Cooperative Control*, V. Kumar, N. E. Leonard, and A. S. Morse, Eds. New York: Springer-Verlag, 2004.

[11] J. E. Hopcroft and G. Wilfong, "Motion of objects in contact," *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 32–46, 1986.

[12] A. Lj. Juloski, W. P. M. H. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. H. G. Niessen, "Comparison of four procedures for the identification of hybrid systems," in *Proc. 8th Int. Workshop, Hybrid Systems: Computation and Control*, Mar. 2005, pp. 354–369.

[13] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," Georgia Inst. of Technology, Atlanta, GA. Tech. Rep. GIT-GVU-03-35, Oct. 2003.

[14] V. Manikonda, P. S. Krishnaprasad, and J. Hendler, "Languages, behaviors, hybrid architectures and motion control," in *Mathematical Control Theory,* J. Willems and J. Baillieul, Eds. New York: Springer-Verlag, 1998, pp. 199–226.

[15] M. Paull and S. Unger, "Minimizing the number of states in incompletely specified state machines," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 356–367, Sept. 1959.

[16] J. M. Pena and A. L. Oliveira, "A new algorithm for exact reduction of incompletely specified finite state machines," in *Proc. Int. Conf. Computer Aided Design*, 1998, pp. 482–489.

[17] V. Pavlovic, J. M. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *Advances in Neural Information Processing Systems 13* (NIPS*2000), pp. 981–987.

[18] C. F. Pfleeger, "State reduction in incompletely specified finite state machines," *IEEE Trans. Comput.*, vol. C-22, pp. 1099–1102, 1973.

[19] J. K. Rho, G. Hachtel, F. Somenzi, and R. Jacoby, "Exact and heuristic algorithms for the minimization of incompletely specified finite state machines," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 2, pp. 167–177, 1994.

[20] *GRITS lab* (2007). [Online]. Available: http://gritslab.ece.gatech.edu/MODEbox.html

[21] *BORG lab* (2006). [Online]. Available: http://borg.cc.gatech.edu

Florent Delmotte received a Diplôme d'Ingénieur from the Ecole Supérieure d'Electricité (Supélec), Gif-sur-Yvette, France, in 2003. Since then, he has been a graduate research assistant in the Department of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, Georgia, where he received an M.S. degree in 2003. He is now a Ph.D. candidate. His research interests include hybrid systems, multimodal estimation, linguistic control of mobile robots, and optimal control.

Tejas R. Mehta is pursuing his Ph.D. degree at the Georgia Institute of Technology, where he received his M.S. degree in 2004 and his B.S. degree in 2002 in electrical and computer engineering. He is currently working as a graduate research assistant at the Georgia Robotics and Intelligent Systems Laboratory. His research interests include optimal control of hybrid systems, multimodal control, and linguistic control of mobile robots.

Magnus Egerstedt is an associate professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, where he has been on the faculty since 2001. He received the M.S. degree in engineering physics and the Ph.D. degree in applied mathematics from the Royal Institute of Technology, Stockholm, in 1996 and 2000, respectively. He also received a B.A. degree in philosophy from Stockholm University in 1996. From 2000 to 2001, he was a postdoctoral fellow at the Division of Engineering and Applied Science at Harvard University. His research interests include optimal control and modeling and analysis of hybrid and discrete event systems, with an emphasis on motion planning and control of (teams of) mobile robots.

Address for Correspondence: Magnus Egerstedt, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. Phone: +1 404 894 3484. E-mail: magnus@ece.gatech.edu.


©DREAMSTIME

Optic-Flow-Based Collision Avoidance

Applications Using a Hybrid MAV

BY WILLIAM E. GREEN AND PAUL Y. OH

Recent terrorist attacks on the United States have exposed the need for better surveillance and situational awareness technologies. Organizations created to address these needs are aggressively funding research in the use of micro air vehicles (MAVs) for homeland security missions. Such missions have been occurring in caves, tunnels, and urban areas. By mimicking flying insects, which navigate in these complex environments regularly, an optic flow collision avoidance system for MAVs was prototyped. However, there were certain instances (e.g., flying directly into a corner) where this system failed. To address this, a new MAV platform was prototyped, which enabled a quick transition from cruise flight into a hovering mode to avoid such a collision. The hybrid MAV offers the endurance superiority of a fixed-wing aircraft along with the hovering capabilities of a rotorcraft. This article details the applications and design of a hybrid MAV in conjunction with sensing and control techniques to perform autonomous hovering and collision avoidance. This is, to the best of our knowledge, the first documented success of hovering a fixed-wing MAV autonomously.

The Novel MAV Platform

More often, homeland security and disaster mitigation efforts have taken place in unforeseen environments, which include caves, tunnels, forests, cities, and even inside urban structures. Performing various tasks such as surveillance, reconnaissance, bomb damage assessment, or search and rescue within an unfamiliar territory is dangerous and also requires a large, diverse task force. Unmanned robotic vehicles could assist in such missions by providing situational awareness without risking the lives of soldiers, first responders, or other personnel. Although ground-based robots have had many successes in search-and-rescue situations [6], they move slowly, have trouble traversing rugged terrain, and can still put the operator at risk. Alternatively, small unmanned aerial vehicles (UAVs) can provide soldiers and emergency-response personnel with an eye-in-the-sky perspective (Figure 1). On an even smaller scale, tiny, bird-sized aircrafts or MAVs can be designed to fit in a backpack and can be rapidly deployed to provide around-the-corner or over-the-hill surveillance. Navigating in urban environments, however, remains a challenging problem for UAVs. In [7], promising results are shown for a rotorcraft equipped with a SICK laser scanner. Because lift decreases with platform size, carrying this type of sensor on MAVs is not feasible.

To design an MAV that can fly autonomously in and around buildings, inspiration came from looking at nature. Flying insects, such as honeybees and fruit flies, use optic flow to navigate in complex and dynamic environments [2], [9]. By mimicking insect behaviors, we were the first to demonstrate tasks such as collision avoidance and landing inside an urban

Digital Object Identifier 10.1109/MRA.2008.919023

1070-9932/08/$25.00©2008 IEEE

structure [4]. More recently, optic flow has been used outdoors to avoid collisions with a tall building and to navigate through canyons [5]. Although using optic flow outdoors in rich texture areas seems promising, there are some limitations when using this technique as the only sensing modality inside buildings (e.g., flying directly at a wall with no texture). To address these sensor limitations, we prototyped a fixed-wing MAV that is capable of a quick transition into the hovering mode to avoid collisions directly in front of the aircraft. This article illustrates how integrating optic flow sensing, for lateral collision avoidance, with a novel MAV platform results in a vehicle that is well suited for flight in urban areas. The article also discusses optic flow and reactive control experiments mimicking flying insects as well as the fixed-wing MAV with hovering capabilities. The autonomous control of the aircraft's attitude during a hover is detailed later along with near-future goals.

Optic Flow

Insects perform tasks such as collision avoidance and landing by perceiving the optic flow of their surroundings. Optic flow refers to the apparent motion of texture in the visual field relative to the insect's body. On the basis of several experiments with honeybees [8] and fruit flies [10], it is suggested that flying insects avoid collisions by turning away from regions of high optic flow (Figure 2). To mimic these navigation techniques, a 30-g flying testbed was prototyped. Figure 3 shows the prototype that was designed to be small and fly at 2 m/s for extended reaction times to avoid detected obstacles.

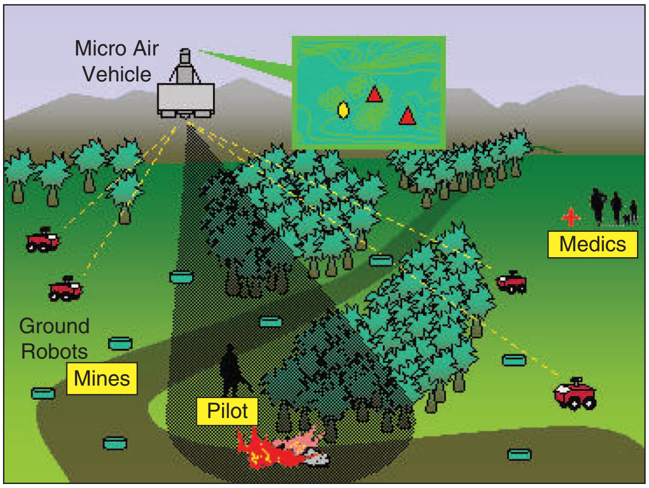


Figure 1. *A small UAV is hovering above to acquire and distribute situational awareness to command and control personnel.*

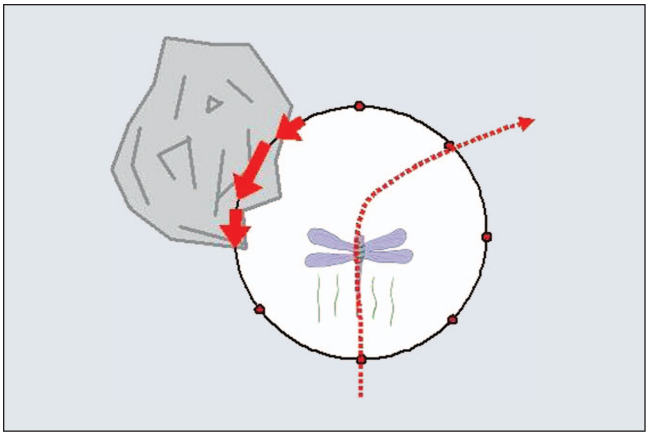


Figure 2. *A dragonfly saccading away from regions of high optic flow to avoid a collision.*

Insects perform tasks such as collision avoidance and landing by perceiving the optic flow of their surroundings.

Collision Avoidance

Mimicking behaviors of flying insects required optic flow to be measured in front of the aircraft to detect oncoming collisions (Figure 4). Figure 5 shows a one-dimensional (1-D) optic flow sensor, developed by Centeye, that was used in the experiments. It comprises a mixed-mode vision chip that images the environment and performs low-level processing using analog very large scale integration (VLSI) circuitry [1].

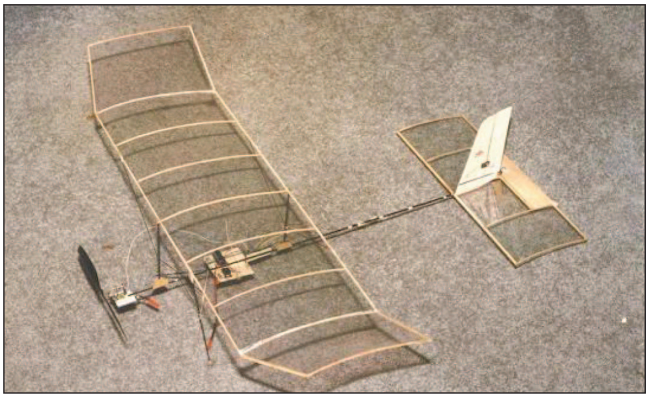


Figure 3. *Our 30-g prototype with a 60-cm wingspan flies at speeds of 2 m/s.*

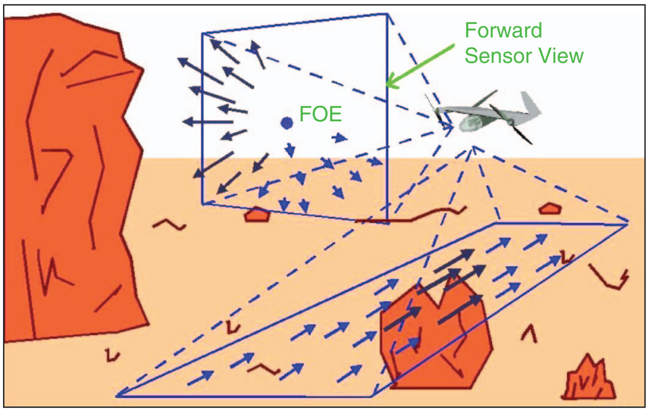


Figure 4. *Optic flow as seen by an aerial robot flying above the ground.*

Navigating in urban environments remains a challenging problem for UAVs.

Then, an off-the-shelf microcontroller performs mid- and high-level processing using standard digital techniques. The resulting sensor, including optics, imaging, processing, and input-output (I/O), weighs 4.8 g. This sensor grabs frames at up to 1.4 kHz and measures optic flow up to 20 rd/s.

Using two of these sensors angled at ±45° from the fuselage, optic flow fields were detected on each side of the aircraft. Optic flow is measured in rd/s and is a function of the MAV's forward velocity, V, angular velocity, ω, distance, from an object, D, and the angle, α, between the direction of travel and the sensor's optical axis (Figure 6). The formula, originally derived in [12],

$$OF = \frac{V}{D}\sin\alpha - \omega \tag{1}$$

was used to set an optic flow threshold that corresponded to D being twice the turning radius of the aircraft. The threshold assumed cruise conditions (i.e., $V =$ constant and $\omega = 0$) and was preset experimentally.

The aircraft was then flown toward different obstacles, and an approaching object on either side of the MAV would generate an increase in optic flow as seen in (1). The output of each of these sensors was fed into an onboard microcontroller. If the values from either of the sensors exceeded the threshold, the processor would apply full deflection to the rudder to avoid the collision. By implementing this reactive-based method, autonomous collision avoidance was successfully demonstrated (Figure 7).

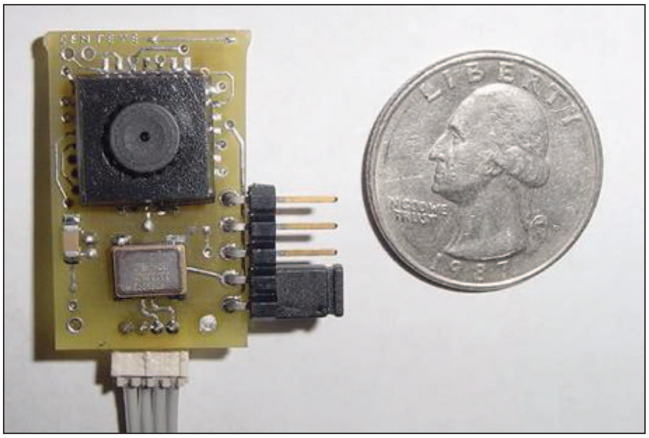


Figure 5. *The mixed-mode VLSI optic flow microsensor is slightly bigger than a U.S. quarter.*

Optic Flow Limitations

The proof-of-concept experiments showed promising results for using optic flow for lateral collision avoidance. However, there are some limitations when flying directly toward an object. For example, when two optic flow sensors are aligned at 45° from the fuselage, as shown in the experiments discussed previously, smaller objects such as poles could remain outside the sensor's field of view [see Figure 8(a)]. This is most likely why honeybees never fly in a straight line toward a target but rather make a slight zigzag pattern. This generates an artificial parallax that will yield optic flow values for smaller oncoming obstacles.

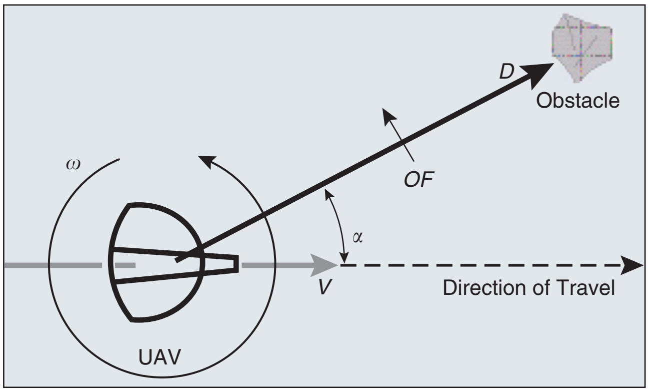


Figure 6. *1-D optic flow during the MAV's steady-level flight.*

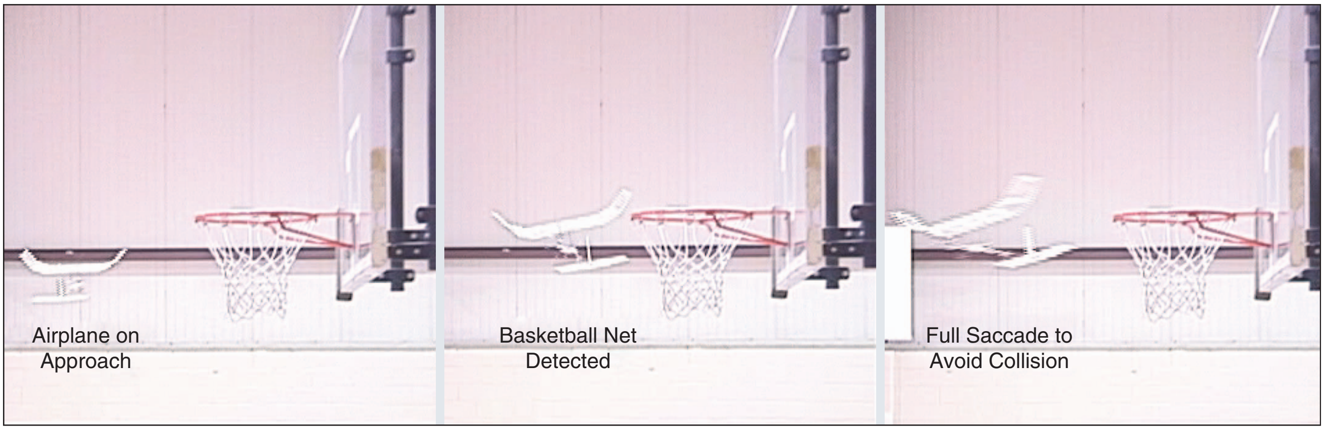


Figure 7. *Optic flow is used to sense when an obstacle is within two turning radii of the aircraft. The aircraft avoids the collision by fully deflecting the rudder.*

Similarly, optic-flow-based collision avoidance is also insufficient when flying directly toward larger obstacles such as walls. Figure 8(b) shows an example of this scenario. In [14], the diverging optic flow field generated by the wall was used to trigger a warning 2 m before the collision. However, the experiment was performed in an artificially textured environment (i.e., alternating white and black sheets were used as walls). Realistically, walls are often homogeneous and have little texture. Therefore, this method will most likely fail, especially since the wall will be the only object in the sensor's field of view. When fruit flies are presented with this scenario in [11], they stick out their legs in preparation for landing. Landing on a wall is obviously not feasible for MAVs. However, a quick transition to a stationary attitude is possible; i.e., a fixed-wing MAV can be designed to quickly transition to a hover to avoid collisions in these instances.

Fixed-Wing Hovering MAV

Integrating the endurance of a fixed-wing aircraft with the hovering capabilities of a rotorcraft have recently been realized in the radio-controlled (RC) community through a maneuver known as prop-hanging. During a prop-hang, the longitudinal axis of the fuselage is completely vertical, and the thrust from the motor balances the weight of the aircraft. Leveraging this maneuver, we were able to prototype a fixed-wing platform with an additional flight mode for hovering [3]. Figure 9 shows the prototype in its hovering attitude. The prototype is constructed with a 3-mm depron foam core laminated with carbon fiber cloth. It has a 1-m wingspan, weighs 600 g, and could fly in cruise mode for 30 min on a 11.1-V, 1,320-mAh lithium polymer battery. With a 6.6:1 gear ratio and a brushless motor, which yielded 900 g of thrust, the MAV has a thrust-to-weight (T/W) ratio of 1.5. This high T/W ratio was required to balance the weight of the aircraft and an extra 100-g payload when in hover mode. In cruise flight (i.e., wings parallel to the ground), it has a speed range of 5–20 m/s.

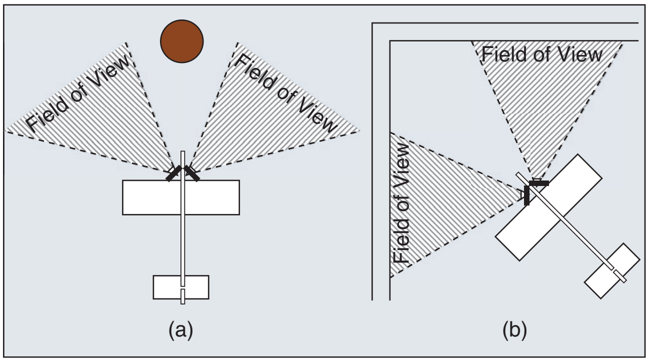


Figure 8. *Limitations of using optic flow for navigation.*

Tiny, bird-sized aircrafts or MAVs can be designed to fit in a backpack.

Transition Between Flight Modes

The most critical aspect of the hybrid design is the transition from cruise to hover flight, which will be used as a secondary collision avoidance maneuver (Figure 10). During this phase,

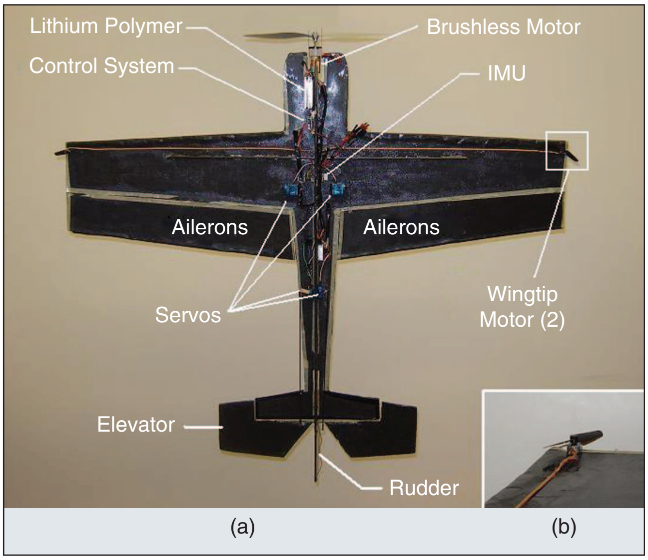


Figure 9. *(a) Our hybrid prototype. (b) The wingtip motors are added to counter the rotation about the roll axis during a hover (i.e., torque roll).*

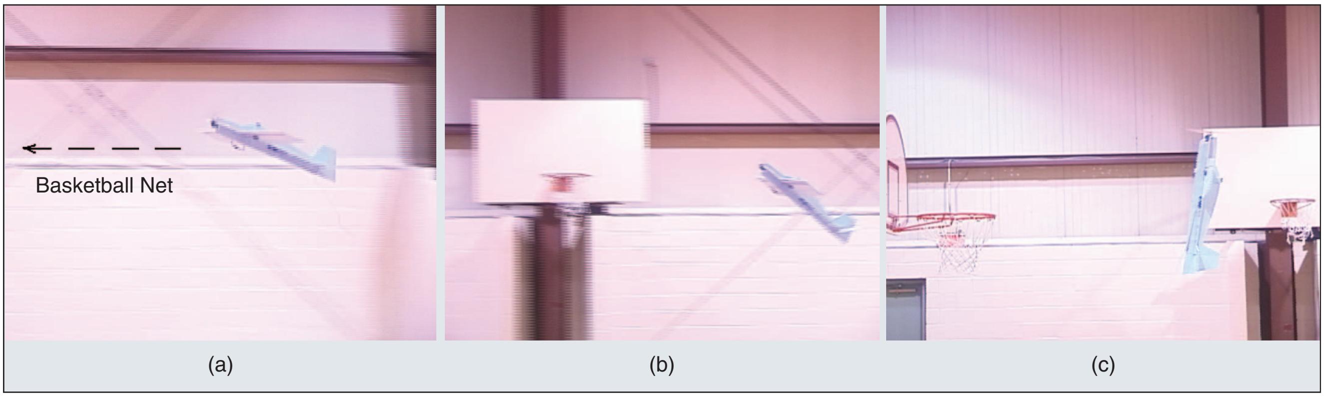


Figure 10. *Our MAV prototype with a 1-m wingspan manually transitions from (a) cruise flight through (b) the stall regime and into (c) a hovering position to avoid collision with a basketball net.*

> ***Flying in and around caves, tunnels, and buildings demands more than one sensing modality.***

there exists an angle-of-attack, α, for which the wings are no longer a contributing factor to the lift component (i.e., stall). To achieve the transition, the aircraft has to leverage its momentum and essentially overpower its way through the stall regime. This requires a high T/W ratio so that the momentum is not lost through the transition. Furthermore, as the aircraft is transitioning from cruise flight (minimum thrust) to the hovering flight mode, the throttle must be increased to balance the weight of the aircraft. The transition back to cruise mode is less complex. Vertical acceleration is required first to give the plane some momentum, and then the elevator is deflected to pitch the aircraft forward into cruise mode.

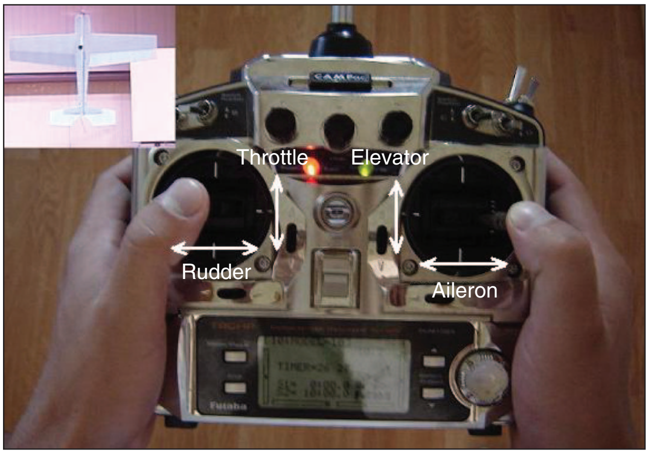


Figure 11. *Manual hovering demands the control of all four transmitter channels.*

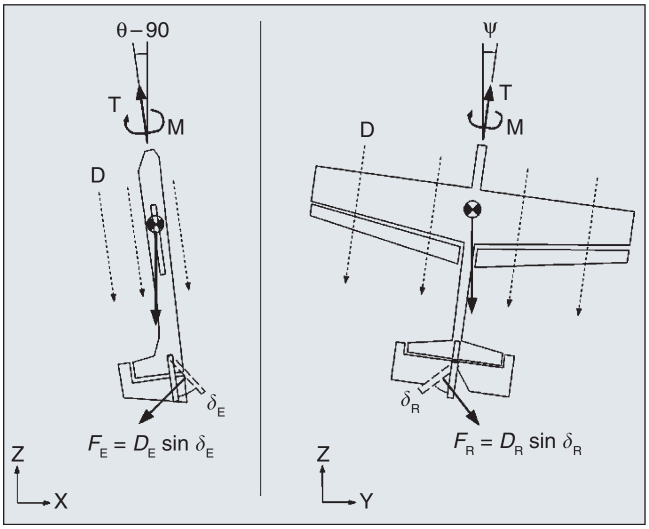


Figure 12. *When in a hovering attitude, the elevator and rudder control surfaces are used to regulate the pitch and yaw angles, respectively.*

Hovering

After transitioning into the hovering mode, the attitude must be sustained by constantly adjusting four channels of an RC transmitter (Figure 11). Assuming the aircraft is in or close to the hovering attitude (i.e., fuselage is vertical), an expert human pilot must ensure the following: 1) increase or decrease the throttle if the plane begins to lose or gain altitude, 2) apply left or right rudder deflection if the plane begins to yaw to the left or right, 3) administer the up or down elevator if the aircraft starts to pitch forward or backward from the nose-up position, and 4) counter the moment created by the motor torque by deflecting the ailerons. Steps 1–3 are shown in more detail in Figure 12, which shows the forces acting on the MAV during a hover. The forces generated by the rudder and elevator deflection angles regulate the aircraft's attitude, while the thrust force balances the aircraft weight. Summing the forces in the vertical direction yields

$$\begin{aligned}&(T - D - F_{\mathrm{E}} \sin \delta_{\mathrm{E}} - F_{\mathrm{R}} \sin \delta_{\mathrm{R}})\\ &\quad \cos \psi \cos (\theta - 90) - W = \mathrm{m}a_z,\end{aligned} \tag{2}$$

where F_{E} and F_{R} are the elevator and rudder restoring forces, respectively, and are functions of the drag force, D, and control surface deflection angle, δ. When the aircraft is in a perfect hover (i.e., $\theta = 90$, $\psi = \delta_{\mathrm{E}} = \delta_{\mathrm{R}} = a_z = 0$), the thrust must equal both the weight and drag forces.

Autonomous Hovering

To autonomously avoid a collision by transitioning into the hover mode, both the transition into hover and the hover itself must be automated. To regulate the attitude during a hover, data from a small and lightweight inertial measurement unit (IMU) are fed into an onboard control system. These data are captured during both manual and autonomous hovering and are used to compare the controller performance with that of an expert human pilot.

Sensing and Control

Autonomous attitude control of this aircraft requires a sensor that can measure the vehicle's orientation when pitch angles approach and exceed ±90°. Figure 13 shows an IMU by MicroStrain that outputs a gyroscopically stabilized four-component quaternion describing the MAV's orientation with respect to the fixed earth coordinate frame. It weighs just 30 g (out of its protective casing) and is composed of three triaxial accelerometers and angular rate gyros as well as three orthogonal magnetometers. The IMU, using RS232 communications, will transmit orientation data to the host computer at a clock cycle of around 10 ms. Therefore, embedding the sensor on the MAV platform will enable an onboard microcontroller to obtain the aircraft's orientation at a rate of 100 Hz.

An onboard control system was designed using a PIC16F87 microcontroller and an RS232 converter chip to communicate serially with the IMU. The microcontroller pings the IMU for the measured quaternion, q_{m}, which corresponds to the MAV's attitude at that instant. The commanded quaternion, q_c, which describes the MAV's orientation during a hover, is

$$q_{1c} = e_1 \sin(\Theta/2) = 0.000i \tag{3}$$

$$q_{2c} = e_2 \sin(\Theta/2) = 0.707j \tag{4}$$

$$q_{3c} = e_3 \sin(\Theta/2) = 0.000k \tag{5}$$

$$q_{4c} = \cos(\Theta/2) = 0.707, \tag{6}$$

where e_i (for $i = 1, 2, 3$) represents the direction cosines of the Euler axis and Θ gives the scalar angle of rotation about that axis. The error quaternion can be found using the following formula [13]:

$$q_e = q_c^* \times q_m, \tag{7}$$

where q_c^* represents the conjugate of the commanded quaternion. The yaw and pitch error can be extracted from q_e, and the proportional-derivative control is used to send pulse-width modulated signals to the rudder and elevator servos. This, in turn, drives the aircraft orientation back to the hovering attitude. Figure 14 shows the control loop that repeats continuously and is synchronized with the IMU clock cycle (i.e., every 10 ms).

The hybrid MAV offers the endurance superiority of a fixed-winged aircraft along with the hovering capabilities of a rotorcraft.

Experiments

The first autonomous hovering experiments were conducted inside an urban structure with limited flying space (i.e., 3 × 3 m area) to demonstrate that hovering can be sustained within small areas. The MAV's attitude is under full autonomous control through rudder and elevator inputs, while the height is adjusted manually through throttle commands via the pilot until the aircraft's weight is balanced. Initial experiments demonstrated that the MAV was able to successfully hover in hands-off mode for several minutes before draining the battery (Figure 15).

Another experiment was performed to contrast hovering under both manual and autonomous control. The metrics used were the duration of the hover before losing control and the stability of the aircraft while in the hovering mode. A skilled human pilot was initially given control of the aircraft and was instructed to fly around a gymnasium in cruise configuration, then transition from cruise to hover flight and attempt to hover the aircraft for as long as possible. The video stills in Figure 16(a)–(c) show the pilot struggling to keep the fuselage vertical but able to keep the aircraft positioned over a small area. (The video sequence shows three images extracted once per second for a period of 3 s. With the plane rotating at a rate of 0.25 r/s, this is enough to show two quarter rotations.) After a few trials, the human pilot was able to sustain a hover for several minutes before draining the battery. However, the aircraft's pitch and yaw angles oscillated significantly as the pilot tried to maintain the hover. This is supported with a portion of the captured flight data, labeled human-controlled, in Figure 17. Next, the pilot was instructed to again fly in cruise configuration and manually transition from cruise to hover flight. However, instead of trying to hover the aircraft manually, the pilot flicked a switch on the transmitter, which

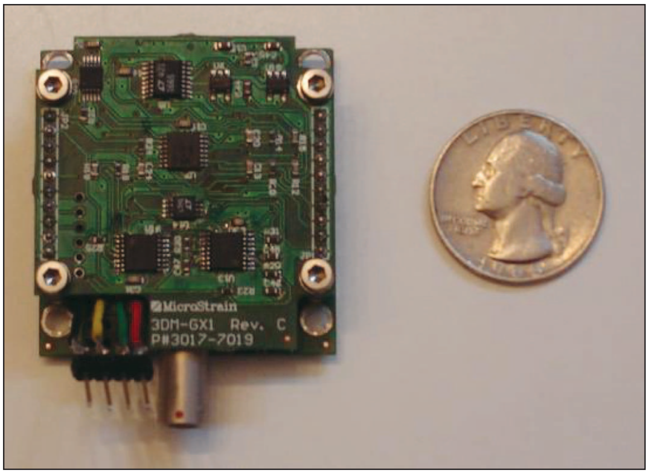


Figure 13. *MicroStrain's 30-g IMU sensor was used to obtain attitude information on the onboard control system.*

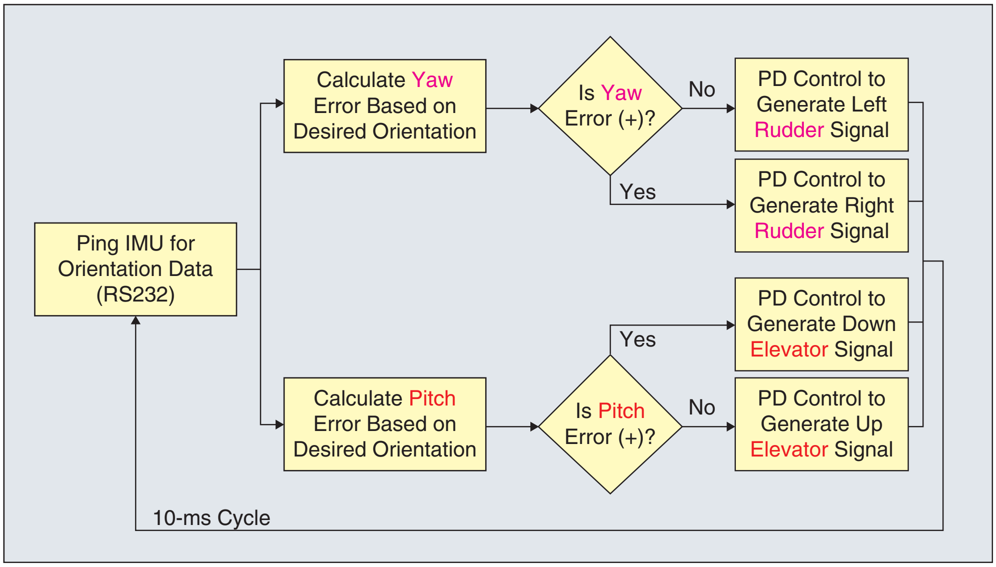


Figure 14. *Flow chart describing the autonomous hovering code.*

enabled the onboard control system. This time, the aircraft was fixed in a vertical position and was able to hover for more than 5 min before exhausting the battery [see Figure 16(d)–(f)]. Again, flight data were captured and a fraction of it is shown in Figure 17.

As originally thought, the torque-roll did not affect the stability of the aircraft during a hover; i.e., the MAV was still able to remain in the vertical position despite the rotations resulting from the motor torque. However, if this MAV were to be used in the field for surveillance and reconnaissance purposes, the view from the wireless camera onboard would have a dizzying effect as the plane was rotating at a rate of 15 r/min.

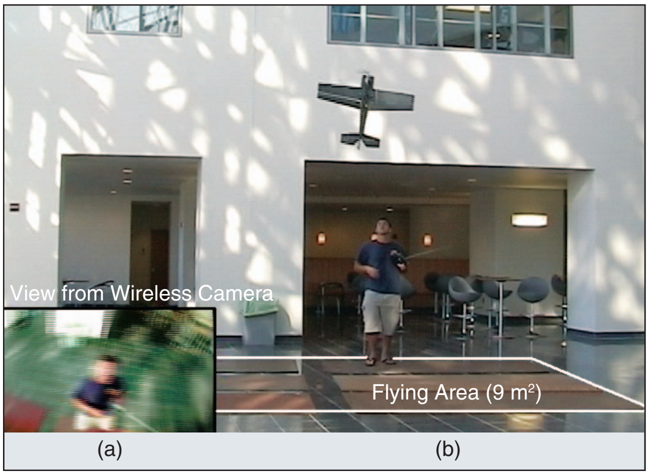


Figure 15. *(a) A photograph from the MAV's bellycam is shown. (b) MAV performing a hands-off autonomous hover in an urban structure.*

As the original aileron surface area did not create enough torque to counter the rotation, other alternatives had to be investigated. The first and most obvious was to increase the aileron surface area by lengthening them in the direction of the wing chord. However, this did not work because 1) the prop wash during a hover only flowed over about 30% of the aileron and 2) a longer aileron when fully extended caused some airflow to completely miss the tail, which greatly affected attitude regulation during a hover. The second approach was to mount miniature dc motors on each wingtip, which blow in opposite directions to create a rotational force opposite that of the motor torque (see Figure 9). Original experiments showed promising results as the torque rolling rate was decreased by more than 75%. Slightly more powerful motors are currently being investigated.

Conclusions

Flying in and around caves, tunnels, and buildings demands more than one sensing modality. This article presented an optic-flow-based approach inspired by flying insects for avoiding lateral collisions. However, there were a few real-world scenarios in which optic flow sensing failed. This occurred when obstacles on approach were directly in front of the aircraft. Here, a simple sonar or infrared sensor can be used to trigger a quick transition into the hovering mode to avoid the otherwise fatal collision. Toward this end, we have demonstrated a fixed-wing prototype capable of manually transitioning from conventional cruise flight into the hovering mode. The prototype was then equipped with an IMU and a flight control system to automate the hovering process. The next step in this research is to automate the transition from cruise to hover flight.

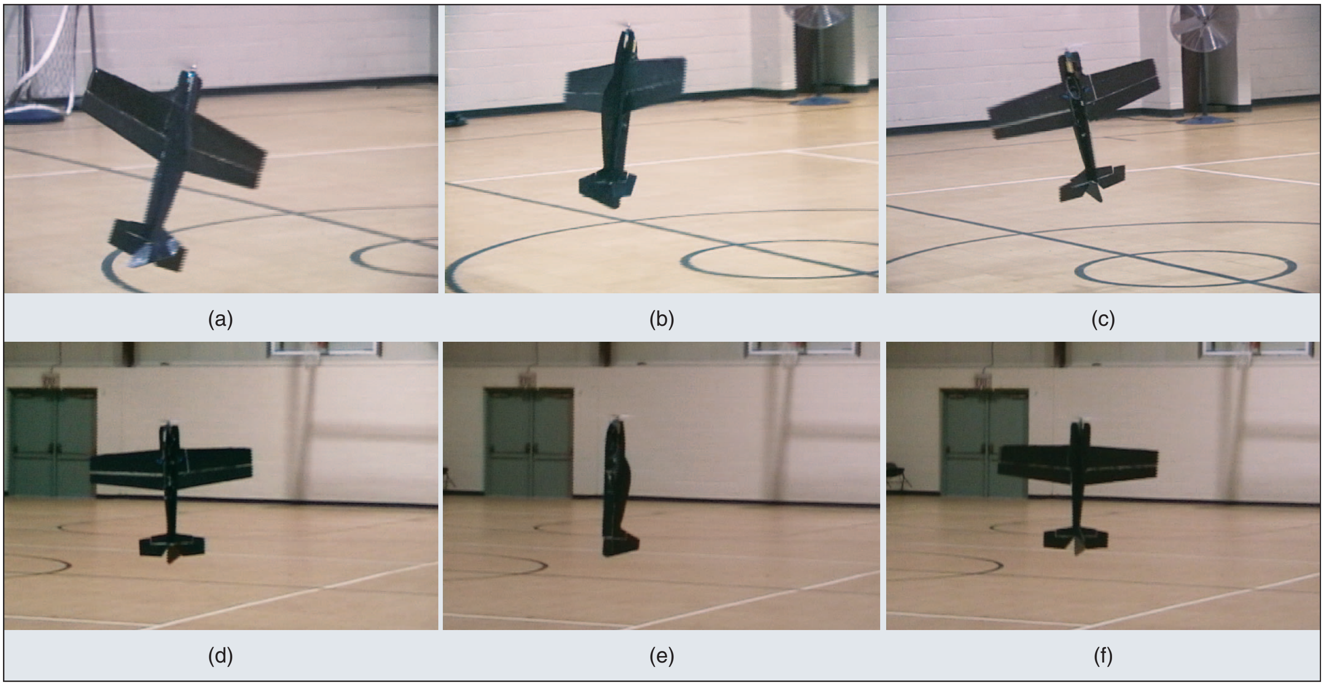


Figure 16. *(a)–(c) A skilled-human pilot hovers a fixed-wing aircraft in a small gymnasium and struggles to maintain a vertical orientation. (d)–(f) Under autonomous control, the same aircraft is able to sustain a hover while remaining fixed in the vertical position.*

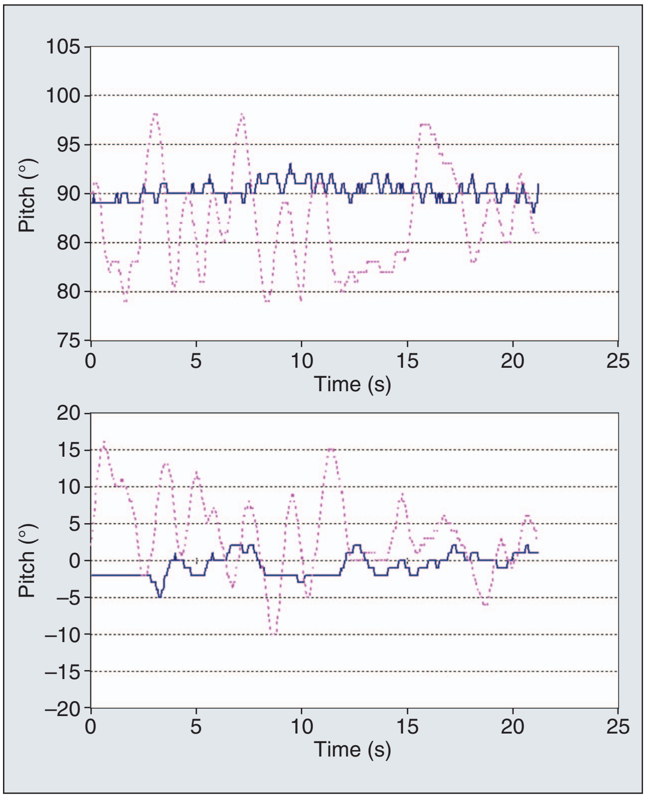


Figure 17. *Pitch and yaw angles captured during both human-controlled and autonomous hovering.*

Acknowledgments

This study is based on the work supported by the National Science Foundation under Grant No. 0347430. Any opinions, findings, conclusions, or recommendations expressed in this study are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Keywords

Unmanned aerial vehicles, collision avoidance, hovering aircraft, autonomous, flight control.

References

[1] G. Barrows, “Mixed-mode VLSI optic flow sensors for micro air vehicles,” Ph.D. dissertation, Dept. Elect. Eng., Univ. Maryland, College Park, Dec. 1999.

[2] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin, 1950.

[3] W. E. Green and P. Y. Oh, “An MAV that flies like an airplane and hovers like a helicopter,” in *Proc. IEEE/RSJ Int. Conf. Advanced Intelligent Mechatronics*, Monterey, CA, July 2005, pp. 699–704.

[4] W. E. Green, P. Y. Oh, and G. Barrows, “Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments,” in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, Apr. 2004, pp. 2347–2352.

[5] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, “Maximizing miniature aerial vehicles,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 34–43, 2006.

[6] R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten, “Mobility and sensing demands in USAR,” in *Proc. IEEE Industrial Electronics Conf. (IECON)*, 2000, vol. 1, pp. 138–142.

[7] D. H. Shim, H. Chung, and S. S. Sastry, “Conflict-free navigation in unknown urban environments,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 27–33, Sept. 2006.

[8] M. V. Srinivasan, S. W. Zhang, M. Lehrer, and T. S. Collett, “Honeybee navigation en route to the goal: Visual flight control and odometry,” *J. Exp. Biol.*, vol. 199, no. 1, pp. 237–243, 1996.

[9] M. V. Srinivasan, J. S. Chahl, K. Weber, S. Venkatesh, M. G. Nagle, and S. W. Zhang, “Robot navigation inspired by principles of insect vision,” *Robot. Auton. Syst.*, vol. 26, no. 2, pp. 203–216, 1999.

[10] L. F. Tammero and M. H. Dickinson, “The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*,” *J. Exp. Biol.*, vol. 205, pp. 327–343, 2002.

[11] L. F. Tammero and M. H. Dickinson, “Collision avoidance and landing responses are mediated by separate pathways in the fruit fly, *Drosophila melanogaster*,” *J. Exp. Biol.*, vol. 205, pp. 2785–2798, 2002.

[12] T. C. Whiteside and G. D. Samuel, “Blur zone,” *Nature*, vol. 225, pp. 94–95, 1970.

[13] B. Wie, H. Weiss, and A. Arapostathis, “Quaternion feedback regulator for spacecraft eigenaxis rotations,” *J. Guid. Control Dynam.*, vol. 12, no. 3, pp. 375–380, 1989.

[14] J. C. Zufferey and D. Floreano, “Fly-inspired visual steering of an ultralight indoor aircraft,” *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 137–146, 2006.

William E. Green received his B.Sc., M.Sc., and Ph.D. degrees in mechanical engineering from Drexel University in 2002, 2004, and 2007, respectively. He is currently working with ATK in Maryland, USA.

Paul Y. Oh is an associate professor of mechanical engineering at Drexel University. He received his B.Eng. degree from McGill University in 1989. He received the M.Sc. degree from the Seoul National University in 1992 and his Ph.D. from Columbia University in 1999, both in mechanical engineering. He received faculty fellowships from the NASA Jet Propulsion Lab in 2002 and the Naval Research Lab in 2003. He received the NSF Career Award in 2004, the SAE Ralph Teetor Award for Engineering Education Excellence in 2005, and was named a Boeing Welliver Fellow in 2006. He is the director of the Drexel Autonomous Systems Lab, and he chairs the IEEE Technical Committee on Aerial Robotics.

Address for Correspondence: William E. Green, Drexel Autonomous Systems Laboratory, Drexel University, Philadelphia, USA. E-mail: william.edward.green@drexel.edu.

TUTORIAL

Do It Yourself Haptics: Part II

Interaction Design

BY KARON E. MACLEAN AND VINCENT HAYWARD

This article is the second of a two-part series intended to be an introduction to haptic interfaces, their construction, and application design. Haptic interactions employ mechanical, programmed physical devices that can be used for human-computer communication via the sense of touch. In Part I of this series, we focused on the devices themselves: the classes of hardware schemes currently available or envisioned, the software components that drive them, and specific examples that can be built on the kitchen table. Here in Part II, we broach a topic that is coming into its own; between the vision of a particular utility that haptic feedback theoretically should enable and the hardware capable of delivering the required sensations is the problem of designing the interaction in a usable way.

Introduction

Haptic technology has hit the mainstream. In 2000, there weren't that many people who knew that the word *haptic* definitely did not refer to a liver dysfunction. By 2004, any self-respecting gamer had it in a joystick at home, and cell phones buzzed. Today, these devices already show the potential to transform many specialized tasks, and the vision of embedded, haptically enabled devices soon dominating our everyday existence is shared by a guru of human–computer interaction (HCI) [71]. It is an inevitable development, despite considerable technological challenges. Our information age has taken the path of networking and abstractions; yet, evolutionarily, we are physical animals dependent on touch to function and communicate. As information technology matures and continually becomes more complex and intrusive, its intangibility and remoteness (action at a distance) become more obvious flaws. Haptic technology offers a solution—if we do it right.

The haptic sense, comprising taction (mediated by the skin) and proprioception (our conscious or unconscious experience of body movements and forces), is often observed to be special in its close association with motor channels—one perceives and acts in tight integration. Today, it has another imputed virtue: that of not simply not being either vision or audition. Contemporary computational interfaces have saturated our eyes and ears. There's not much communication bandwidth left there, whether one is an automobile driver, an urban pedestrian, or a medical professional in the operating room. It is therefore common to suggest that beyond its role in providing tangibility and real-world fidelity, the touch sense is another potential information conduit. Thus, we see at least two distinct and major role types for haptics in

- restoring tangibility to digital interactions, with functional and aesthetic potential
- offering an additional communication conduit, providing we recognize the importance of attentional design and the overall user environment and its loading.

We'll be going into these aspects, which have many facets and can overlap, in more detail.

Why Interaction Design Matters

There are not many computer users today without a collection of stories of user interfaces (UIs)—generally graphical, as that is what we are surrounded with—that have annoyed, confused, or stymied them. The frequency of these incidents has unfortunately not diminished with time and experience nor are they, in most cases, due to limits in the extraordinary graphical display and back-end hardware available today. They

Digital Object Identifier 10.1109/M-RA.2007.914919

are, rather, the intersection of bad UI design by untrained and unsupported application creators and paying customers who clamor (or respond to marketing) for features and style rather than recognizing and valuing usability. These problems are exacerbated by the remarkable number of technologically supported tasks that we now tend to do at the same time. It is like being treated for multiple ailments by several specialist doctors who cannot or will not coordinate with one another—leaving the patient/user to sort out the impossible conflicts alone.

As some forms of haptic technology depart research labs as commodities, it is exhibiting a similar phenomenon. It is becoming technically feasible to integrate haptic feedback into everyday devices, but it is also easy to misuse it—far easier, in fact, than to use it well. Good UI design is hard. It's not just a need for formal training and experience, which helps, but much of what is taught is really just a codification of common sense. The tough part is taking the time, space, and money in a given design cycle to

1) truly understand the user's experience, problems and needs—the whole context of the interaction; this happens by observing and talking to said users
2) base design prototypes (ideally, a few very different approaches) on thorough knowledge of relevant human capabilities, in terms of perceptual, cognitive, and motor attributes. These, again must be related to the entire context; if a user is doing many things at once, that means their resources are not fully available for your task
3) verify and iterate on a design prototype through user testing, rather than relying on a designer's guess of what will work
4) allow the UI design to influence the rest of the system's design to support an optimization of the user's experience (as opposed to, say, a feature list created by the marketing group, which is longer than the list of a competitor's product). Sometimes, a good UI design will indicate a change in a device's physical form factor. If the UI has been slapped on as a final step, this will probably be impossible.

These basic principles of good HCI design are all the more important when the modality is one that people are not accustomed to using in this way and, furthermore, one that is often being layered on top of whatever else the user is already seeing or hearing. It's a perfect storm for sensory and cognitive overload.

This article's primary goal is to provide some basic heuristics and examples for avoiding that storm, and instead offer a path for integrating haptic feedback into the mix of the user experience in a way that will help.

Overview

In the remainder of this article, we'll start by considering the mapping between the crosscutting roles, which haptic feedback is thought to serve, in many different kinds of application spaces and, conversely, the human abilities and limitations that must be recognized, targeted, or supported as these roles are developed (see the "Usable Roles for Haptic Feedback" section). We'll provide some design guidance, which is especially relevant to haptic interactions (see the "Haptic Interaction Design Practices" section), and then close with a pair of case studies that illustrate contrasting approaches to actually doing it (see the "Design Case Studies" section).

Usable Roles for Haptic Feedback

Previously, we listed some very broadly defined several potential *roles* for haptic feedback. On a closer look, here, we take a different cut. In each of the several categories (the list is certainly not exhaustive), we will consider haptic value in terms of functionality, emotion and aesthetics, in search of ways in which it can improve task performance or expand capabilities, allow us to communicate through technological conduits, or make an interaction more pleasurable and satisfying. Some of the categories relate to control, i.e., the closely coupled perception-motor action loop referred to earlier. Others are more sensory in nature, e.g., tactile messaging where the skin is used as a display surface, but the user's response might be less direct—e.g., a thought or a directed look. For additional background, we refer the reader to some recent comprehensive reviews of human sensory, cognitive, attentional, and motor abilities, which [63] summarizes in the context of interaction design.

Naturalistic Interactions

A common theme in the following discussion is to relate new potential functionality to natural, i.e., ecological, touch interactions in the nontechnological world. Our sensorimotor equipment and social wiring are likely to be well evolved or conditioned to handle the things we do naturally, comfortably, and with easy precision in this domain.

This is not an adage to follow slavishly, however. There are many examples of humans picking up new technological skills with apparent ease, despite a lack of obvious evolutionary preparation (driving a car, typing, and perhaps most remarkably, text messaging on tiny cell phone keyboards). We already see evidence of this here, e.g., in human acuity in abstract tactile message decoding, an unnatural act that will come back to haunt us with stress and damaged thumbs? Perhaps only time will tell.

Multimodality of Haptic Interactions

Haptic design is nearly always a multimodal design. The touch sense is generally used in conjunction with other sensory modalities, whether their roles are to reinforce the same task or to handle different tasks performed at the same time. Touch-derived input plays a unique role in this context, and theories continue to develop on how sensory information is integrated and how conflicting information is resolved. The emerging answer is that relevance of the source to the task matters along with the source's trustworthiness [30].

Precise Control: Force Versus Position

We will start with a low-level attribute of coupled perception-action applications (usually involving force feedback), because of its far-ranging and often overlooked consequences. The sensation and control of absolute position is easily perturbed—try to reach out and touch a specific point in space with your hand while turning your gaze away and without groping for landmarks. Conversely, we're quite skilled at detecting and producing small variations in force resistance. This is seen in a comparison of natural, ungrounded human gestures (conversational emphasis, demonstrating emotion, or indicating a relatively discrete-valued command—stop, come, look over there) with those that entail resistance (almost any kind of tool use, from chopping vegetables to writing and painting, maintaining a desired pressure on an automobile throttle, precisely controlling a violin string's vibration). For humans, precision requires resistance.

The implication for design is that grounded resistance—something solid to push against—is desirable for most kinds of precise tasks. It is imperative to remember this when choosing what will be displayed, and the tasks best suited to haptic augmentation. To implement this principle, resistance could be provided by a programmed force feedback system or, alternatively, by a passive ground (e.g., a tabletop) with nongrounded feedback (such as impulses or vibrations) supplying the programmed feedback. In this latter case, in pushing against a stiff surface, the user's input will be isometric (without measurable motion), and so position sensing cannot be used to measure user intent. However, pressure might be more suitable.

When precision is not needed, and broad expansive gestures are appropriate, then nongrounded systems (such as a limb-mounted tactile display) might be more appropriate.

High-Fidelity Rendering and Model Creation

The role for haptic feedback, which has received the greatest research attention to date, is the creation and literal haptic rendering of what we see on a graphical display. These efforts have been dominated by surgical simulation and remote surgical procedures. Because of their substantial coverage elsewhere ([14], [22], [47], [56], [87]; see also Part I of this tutorial), we will not discuss them in detail here but place them in context with other uses and relate this role to human attributes.

A dominant and fairly unique aspect of these applications is their need for high fidelity to real-world analogs, so as to recreate a specific task environment—e.g., for training, or for actually conducting a remote or virtualized version of a task, which was once performed physically. Because of this direct tie, high fidelity rendering obviously borrows heavily from haptic interactions in the real world. In some cases, the real world case can be improved upon (for example, a tool geometry that is awkward or misscaled in reality can be reconfigured or magnified).

Obtaining satisfactory fidelity is one challenge, as discussed in Part I. The turing test of haptic rendering would be a user's inability to distinguish it from the real thing. In fact, this is currently possible for only a small subset of possible rendering targets, usually the more squishy ones, and thus usability can mean identifying and exploiting the limitations of the perceptual system to reduce the negative impact of system constraints. Another design direction is in augmentation, e.g., reconfiguring an operation or layering information atop a rendering such as signals or virtual fixtures (more about these are discussed later).

An additional element is the creation of the models themselves, which can be done through a variety of empirical and analytical, automated and manual approaches (a brief review is available in [63]). In particular, it is necessary to understand a user's perceptual attributes to specify the resolution, stiffness, and many other aspects of the model. In general, highly detailed and stiff renderings—exactly what you'd need to recreate many interesting physical systems—are difficult to stabilize, and the resulting artifacts destroy the illusion of realism [19]. Thus, the designer is often faced with a tradeoff between overall realism versus fidelity in shape detail, texture, hardness, dynamic response, and other rendering parameters. Alleviating this tradeoff drives much of the research in rendering techniques [56].

Finally, multimodal issues are almost always critical to attaining a realistic simulation result, in particular for renderings that need to convey high stiffness. In these cases, achieving visual-haptic and audio-haptic synchrony to perceptual limits will allow perceptual fusion of the information arriving on the different sensory modalities. Furthermore, the presence of the visual and auditory stimuli can significantly modify the user's interpretation of what they feel, allowing the use of less expensive or slower haptic hardware (e.g., [23], [44], [55], [105]).

Physical Guidance

Both force and tactile feedback can be used to provide direct spatial guidance to a user, either by leading with forces or orienting attention in a particular direction. Attentional orientation usually takes the form of applying a discrete signal to a body location, which then draws visual attention in the same direction, or providing an information-containing signal at a single location (which is discussed more in the following section). Guidance, on the other hand, implies a more continuous engagement that is usually delivered through grounded force feedback for motor skills or, with lower resolution, via distributed tactors on the body for applications such as vehicle steering. It can vary in precision and subtlety, for example, steering a car or aircraft, drawing a calligraphic character, or learning a surgical procedure. Force feedback guidance applications tend to vary across the spectrum of control sharing with the intelligent system (i.e., equally shared versus dominated by one or the other).

Training

In teaching applications, the user is expected to exactly follow the intelligent system's lead. The teacher or another human could be an expert system, and the latter is an instance of shared control or remote collaboration, which is also discussed more in the next section. These methods have been tested in applications ranging from calligraphic writing and surgical tasks to rehabilitation therapy for stroke patients. Haptic feedback has been shown to have value in the training of sensorimotor tasks, with improved performance in a real version of the task following inclusion of haptic feedback in a virtual-reality training segment [1], [69], when the real task has a force component. It has been further observed that visual training is better for teaching trajectory shape, but haptic guidance is more effective for temporal aspects [31].

There are many variants of implementing the construction of training forces. These include guiding the user along a predefined trajectory [2], displaying both the activating pressure and position of the teacher to the student (one indirectly) [51], and requiring the student to cancel a reversed target force [84]. More long-term learning strategies include monitoring the student's resistance and backing off as the need for guidance decreases. This also allows a simultaneous assessment capability [38], [53], [99]. These methods have not been directly compared with one another, and so at this point, it is difficult to evaluate their relative appropriateness in different situations. However, there seems little debate that the creation of motor programs requires realistic resistance to fully develop.

Shared Control

The notion of shared control refers to a cooperative balance of control between the user and the machine. An expert system has knowledge of the sensed and networked environment and the databases but does not know the user's goals. In this case, the system and user can jointly exert the forces that control the system. This concept is especially natural in steering contexts, where there is a single locus of control (e.g., a steering wheel or aircraft stick) that is intuitive to specify in a physical manner.

Telerobotics: Force sharing lies on a continuum of abstraction, which has at one end bilateral force-reflecting telerobots. These systems consist of a remote robot located in the work environment, connected through a network to a local robot of compatible kinematic design, which an operator moves, often wearing it as an exoskeleton and feeling forces sensed remotely and redisplayed locally. This scheme allows the local user to be sensitive to the impedance of the remote environment, with consequently improved dexterity and error management (an early instance is [48]; the beginnings of force sharing during teleoperation is illustrated in [98]).

Virtual Fixtures: The most common basis for shared control derives from the idea of a physical template for guiding a task by keeping it within specified constraints (e.g., a ruler for drawing a straight line). In a virtual environment, programmed forces provide the constraint [82]. Softening the guiding constraint turns this concept into mixed-initiative guidance: the user can choose to be guided or punched through to do something else. Many variants of control sharing using this concept have been tried ([34], [41], [52], [57], [73]; see [63] for a more thorough discussion). A sought-after metric is improvement in task performance while reducing visual demand, thus freeing attention for other tasks, and this has indeed been shown.

In extending these ideas to less predictable, real-world scenarios, however, there are additional complications. In particular, the reflexive dynamics introduced by the user can make them tricky to implement, e.g., oscillations can result from certain kinds of system disturbances [34]. Usable solutions depend on the task, but ideally they will build upon an as yet incomplete knowledge base deriving from both modeling of the user's reflexive and cognitive responses to control actions that are perceived as intrusive, and user testing in both abstract and reasonably realistic contexts.

Cognitive Factors: The user's mindset and awareness of the control balance is a variable to be managed. There are potentially negative side-effects, for example due to the operator's either over- or under-trusting the control suggestions or not understanding who is in charge at a given time [27], [40]. For this reason, it is crucial to manage the reliability of the expert system's signals. The idea of tuning the ratio of hits and misses for an expert system's detection and communication of crucial environmental events (e.g., dangerously close following of the car ahead [27]) and its effect on operator utility of those signals as well as overall efficiency has roots in multiple resource theory, recently updated in [25].

Remote Collaboration

When force communication is important, remote collaboration with another human in a physical task becomes a special case of shared force control (where the automatic controller potentially still plays an important role). This case is particularly interesting because, beyond the demonstrated need to feel the forces to perform a physical task, the existence of another human in the loop introduces social factors as well; and feeling ones' partner's forces appears to be an important parameter in facilitating this. It enhances the sense of presence and togetherness in the mutual effort [6], [85] and conveys the momentary degree of control balance between the partners [72]. In an explicitly social context, the nature of the force sharing impacts the sense of an interpersonal emotional connection [88].

Tactile Signaling in Multitasking Environments

Passive touch cues (which are presented to the observer's skin, rather than felt in response to active movements [36]) can be used for notification of events and to create relatively nonintrusive, ambient background awareness. Such cues can be delivered through a tactile display or overlaid on a force feedback signal being used for another function.

Typically, this kind of functionality targets *multitasking environments* where the user's primary attention, as well as visual resources and possibly hands, are engaged in another task (in fact, this benefit was foreseen very early on in the technology's development [79]). In this section, we'll therefore first mention issues relating to tactile design for multitasking, as well as typical methods and sites of delivery. We will then look at two major categories of tactile signals themselves: simple signals whose message comprises its on/off state (sometimes coordinated with its location), versus informative signals (haptic icons) that can vary in other parameters, e.g., amplitude or feel, and thereby encode additional meaning. Analogous auditory signals are a simple, consistent beep (perhaps directional) versus the diverse auditory icons we hear on modern computers whose specific sound means something—like an application opening, a device ejecting, or an e-mail arriving. Design in these cases is best based on some understanding of human multisensory attention. An overview, including references to other relevant recent work, can be found in [63].

Design for Multitasking Environments

To manage intrusiveness, tactile signals must be designed with variable salience: important events or urgent events/changes should register as louder than less important ones [16]. Furthermore, the user's interruptibility is not a constant, sensory adaptation aside. In the car, pulled-over versus engaged in a turn differ substantially in what kind of additional distractions the driver can safely deal with. In the office, some tasks require protection from routine interference, and yet certain events might always be important enough to come through. This entails two different needs, both active research areas.

Controlling Tactile Signal Salience: It is most desirable to control signal salience independently of potential content. In different contexts, a given event might be more or less important; and in some cases, context may be identifiable.

Parameters used to encode content may also vary inherently in salience. For example, in some schemes and for some display hardware, higher frequencies and/or amplitudes are perceived as louder than lower ones, yet these are the best parameters to vary to indicate different meanings—the change in output is easy to produce precisely and is clearly detectable by a human. Therefore, salience can be inadvertently confounded with meaning, with an unimportant signal more detectable and intrusive than a critical one. This incidence can be minimized with an up-front awareness of the stimulus salience and detectability patterns for a given display. While it is easy to determine relative salience (by itself) for a group of signals, e.g., using simple subjective ranking tests, due to this confound there is a need for design tools that efficiently aid this task *at the same time* as optimizing design of meaning.

Context Detection: The other part of the problem is detecting the user's momentary environment so that the appropriate salience can be used. The active field of sensor-based computing is devoted in part to detecting various aspects of the user context (e.g., location) [68], [76] and in modeling and detecting user mental/emotional state and interruptibility [32], [46].

Ambient Tactile Displays and the Human Body

Physical Configuration and Body Site: It is necessary for ambient tactile displays to be in continual contact with the stimulus site, so that signals will not be missed. Because the hands are often needed for more dexterous roles, the glabrous skin of the fingertips not always convenient as the delivery site, which leaves the less sensitive hairy skin [35]. Past examples, usually for simple signals, have used vests and belts [50], [95], [102], back [95], [106], and tongue [3], and relied on spatial encoding of meaning.

Applications and contexts where hands can be used for background display include the steering wheel [26], track point [15], mouse [16], [17], and increasingly, mobile devices [54], [60], [81].

Active and Passive Touch: More fundamentally, Gibson has argued that "passive touch ... is atypical of normal tactile perception and it leads the person to focus on the body surface" [36], whereas active touch is predominant in naturalistic environments where people are seeking information [86]. Considering that convenient ambient tactile delivery sites are generally less sensitive skin and that the information is intended to be nonattentive, it will be an experimental challenge to test the implication that passively received information display will be less effective.

Simple Tactile Signals

Simple (binary and/or directional) tactile signals are already commonplace in the form of mobile phone vibrotactile alerts for incoming calls; these are useful in many contexts where auditory signals are socially undesirable. Use of spatially distributed tactile signals has also been shown to speed up orientation of spatial attention, with a potential to aid in situational awareness [9], [90]. While signal complexity can be viewed as a continuum (defined either by information capacity in individual signals or by the number of uniquely recognizable signals achievable in a set), we are here defining simple signals as sitting at the far end of this continuum.

Value: The research to date suggests that simple signals are preferable to complex signals when 1) they are all that can be reliably detected, due to limitations of either hardware or context of use (e.g., when a cell phone is sitting in a pocket, details of the signal will be harder to make out), 2) only limited information need be conveyed, or 3) a strong, fast, and accurate user response is needed. By analogy, if visual attention is to be captured by a flashing light, response will be enhanced if that type of stimulus is only used for one event, rather than many different events indicated by variants in flash frequency or color, thus engaging a cognitive component in the response.

Choice of Hardware: For existing vibrotactile display hardware, there is a direct tradeoff between signal richness (potential complexity) and strength, particularly for power-starved mobile applications. For example, solenoid vibration is capable of much stronger stimuli, which can be noticed through clothing, as compared to more expressive configurations of piezo actuators; but it cannot create as many distinguishable signals, even when touched directly. Simple signals are also the more feasible option for less sensitive, nonglabrous skin delivery sites.

Abstract Communication and Information Display: Haptic Icons

The idea of using tactile signals to display abstractions has roots in communication aids for the blind, with the Optacon particularly notable [58]. A recent review of this application space can be found in [96], backed by reviews of relevant aspects of tactile psychophysics [35], [49], [77]. Abstract tactile information transmission has centered on haptic icons or their equivalent: brief informative haptic stimuli (usually vibratory) to which information has been attached.

Symbolic or Abstract: Haptic signals can be based on metaphorically derived symbols or more arbitrarily assigned associations. The likely pros and cons are fairly obvious. Symbolic notations intuitively seem easier to learn and remember, but there are obstacles to using this approach for large but usable sets of icons, particularly when the rendering palette is limited (imagine how well symbolic graphics would work using a few grayscale pixels to cover all possibilities). These challenges include independent control of signal salience and of perceptual spacing (some signals might feel very similar, others quite different, with no semantic pattern); and the fact that individuals are rarely consistent in their interpretations anyway—so one notation will not work for everyone. Both of these problems are handled relatively easily when the need for semiotic connection is dropped, e.g., using a process of *perceptual optimization* on a proposed signal set (see [61] and the discussion later in this article).

One approach to increasing the controllability of the semantic approach is to carefully ascertain a set of basic primitives with the goal of then using them across contexts in a variety of situations [97]. Another is for designers to create codes by drawing on an existing user knowledge base [13], [16]. Alternatively, we see that users are well able to create their own semantic mappings when given the means in both emotive [12], [18], [33] and informative [28] examples. In the last, we see what may be a cue for how to join the two approaches. A designer inflicted completely arbitrary links on his subjects, then discovered post hoc that most users created their own semantic mnemonics when learning the links, and typically found these personally derived interpretations just as logical (and learned them as well) as when they chose the stimulus-meaning associations themselves. That is: perhaps we can make anything behave as a semiotic link.

Learning Haptically Represented Abstractions: Regardless of the approach used to construct a stimulus-meaning link, in deploying the haptic channel for this kind of abstracted information transmission, we are asking individuals to use their touch sense in a manner they do not encounter in the natural world. Psychophysical evidence for tactile acuity with respect to this kind of information transmission is summarized in [63]. There is some neural evidence of brain plasticity for users asked to pick up this skill after early childhood [35], [43].

What learning techniques will best exploit this plasticity? Taking encouragement from human ability to learn Braille after childhood [39] and guidance from how it is taught, we note that a first step is generally to develop the learner's tactual acuity. Barraga and Errin describe a five-step process that moves from simple to complex, beginning with awareness and attention to tactile details, moving through recognition of structure and shape, part-to-whole relationships, then abstracted graphic representations and finally the learning of Braille symbols [5]. Immersion in rich and guided haptic experiences are the key in early stages [10], with Braille labeling introduced later [5].

Individual Differences: There appears to be significant individual variation in tactile acuity and ability to learn abstract associations, including both hyperacuity [21] and our own informal observations of a "haptically challenged" group among our typical experiment recruits. We do not yet know whether this range arises through basic perceptual function or learned cognition, and if the latter, what the indicators could be. Differences in how individuals organize their perceptual space have also been noted, with strong dimensions being held in common but different weaker dimensions employed differently [45]. Both types of difference (ability and organization) have implications on the widespread introduction of haptic information displays. An important area of future work is to better attribute the causes of both poor and exemplary haptic perceptual function, and to ascertain whether training and awareness can improve the former [66].

Identifying the Perceptual Dimensions of a Device Display Space: To create a set of learnable haptic icons, there are two linked challenges. One of these is creating learnable stimulus-meaning associations. Techniques for this are today largely ad hoc. The other is to ensure that the stimuli in the set are perceptually discernable, and furthermore to understand people's preference for organizing them, for later leverage in choosing appropriate patterns for association. For this, methods are more straightforward and there already exist the beginnings of a practical cataloging of the dimensionality and recognizable resolution available for various types of display hardware [13], [100], [103]. The current status on dimensionality that has been found for various types of stimuli and display hardware is summarized in [63].

Here, we will mention the one systematic tool of which we are aware, which uses Multidimensional Scaling (MDS) to "perceptually optimize" a group of stimuli. In a 20–60-min session (depending on the set size), a few users can provide enough dissimilarity data about a stimulus set to reliably create a map that reveals the dimensions along which the subjects perceive the stimuli relative to one another [61], [78], [100]. This map can be used to 1) guide iterative revision of the stimulus set until a renewed map indicates that the desired perceptual spacing (not too close or too different) has been achieved [16], [61]; and 2) choose a subset of stimuli for actual use in an application, again according to their desired perceptual organization and spacing. This method can be used both for independent creation of stimuli intended for arbitrary mapping to meanings, and for adjustment of a prototype set of representational icons whose meanings are chosen a priori [16].

Learning Stimulus-Meaning Associations: Glossing over the current sketchy state of affairs on creating learnable stimulus-meaning associations, the next step is for users to learn the associations. Because learning generally works best when information is absorbed from different sources (observed for tactile stimuli as well, e.g., [67]), a multisensory reinforcement learning process is probably advantageous even to learn a stimulus that might later be invoked purely through the haptic channel.

In efforts to date, users have already demonstrated a good ability to learn associations that are metaphorically matched by the designer [13], [16], [97], deliberately arbitrary [28], [29], or chosen by the user. In these instances, training took the form of repeated exposure/testing cycles of stimulus-meaning pairs until a given performance is demonstrated. We have also taken a further step of testing and continuing to optimize the icons under realistic environmental stress testing, adjusting the stimuli for relative distinctiveness and salience as needed. For example, in some circumstances, a controlled degradation in noticing performance is desired on response to workload, with some important icons still being noticed but less critical ones washing out when more urgent tasks are in play [17].

Expressive Control

"Expressive" refers to the quality or power of expressing an attitude, emotion, or other communicative information. Based on how we use touch in the real world, physicality seems a completely natural, indeed essential property for control tasks requiring emotiveness or precision, and in particular, both at once. We propose some heuristics and a brief summary of haptic potential in this realm.

Expressive Capacity

We use this term to broadly describe the richness of a communication channel for any purpose: its dimensionality, continuousness, the degree of control it affords the user, and the ease and naturalness with which desired acts can be completed [61]. This can refer both to tools that support artistic or interpersonal communication, i.e., emotional expression; and more prosaically, sheer information capacity. This can be specifically articulated as:

a) *Density*: number of bits of information that can be transmitted
b) *Controllability*: accuracy of conveyance (expression by sender, transmission, and interpretation by recipient)
c) *Directness*: direct versus encoded nature of the required actions (in analogy to direct-manipulation versus command-line interfaces)
d) *Responsiveness*: the immediate confirmatory and/or aesthetic feedback to the user
e) *Emotiveness*: the number, range, and subtlety of emotions that can be expressed.

By this measure, a computer keyboard is highly expressive on the first two counts but fails miserably in the third and fourth. The fifth is tricky, for the product of typing (the printed word) can be highly emotive in every way, both visually (ask a typesetter) and semantically. However, the *act* of typing is not particularly emotive. This raises the interesting question of whether an input device should be classified as expressive (based on its output) if using it doesn't *feel* expressive.

Role for Haptics

An ungrounded gestural interface works well for purely emotive control (low controllability). A keyboard is hard to beat when you wish to indirectly but exactly specify the greatest possible range of actions (high controllability). Physicality seems key when you need to do both at the same time. For example, in the highly studied topic of computer music controllers, many argue that the resistance and feedback of forces or vibrations are essential to controllability [20], [83], [104]. This is further linked to a consistency or closing of the control loop—a mechanical interaction between the subject and the sound source [37], [59]. However, computer-controlled grounded forces bring constraints, such as tethering and a loss of workspace, weight, motors and electrical power, a lack of generality in the control actions and handles that can be used, and a need for extremely tight synchronization between action and sound [7].

Some recent resources give guidance in how to accomplish this, from the standpoint of both the fundamental interactions themselves and their mechatronic implementation [11], [20], [62], [74]. Recent literature applying haptics to both music control and other expressive uses—ranging from the feel of a bristled paintbrush to gaming, control of under-actuated systems, and surgical simulation—is reviewed in [63]. A common feature is strong individuation of instrument to application, i.e., type of music to be created and the gestures employed. These are not general-purpose devices.

Haptic Effect

Affective design addresses the subjective emotional response to and relationship between users and interfaces. Although related, it is distinct from and more personal than expressive control: the latter is about achieving a desired result, although this does include the satisfaction and aesthetics of doing so. In the last decade, subjective response has been recognized as an important, if difficult-to-quantify aspect of everyday interfaces that impacts stress and usability [70]. It also forms the basis of a new, sophisticated type of interface based on *affective computing* [80], where the computer sensors and displays are used to determine and elicit particular emotional experiences from the user.

Haptic affective design has not received a lot of attention to date, despite recognition of the crucial role of touch in human communication and development [63]. Here, we mention two potential roles for effect in haptic design.

Design for Feel

Consider the direct affective response that feeling produces on the user: haptically speaking, what feels good, bad, or neutral? To what extent is this shaped by the task at hand? Is it consistent across people and does it impact performance? Preliminary efforts have explored mechanisms for measuring haptically induced effect (with a combination of biometric study and self-reports), is able to find some consistency in response, and suggests that haptic preference is not always linked to superior performance—i.e., sometimes people prefer controls that don't particularly aid in their task [93]. Eventually, this line of research will deliver heuristics that can guide interface aspects such as the choice of feel for a given control action. For now, the best practice is to routinely include subjective questionnaires in any performance-oriented user test during the design process, and consider this response in design iterations.

More broadly, we need clearer metrics to establish how important it is to get this right. The cost of negative affective response to an interface (whether the reaction is to ugly graphics, sound, or feel) is subtle and probably cumulative. One would expect the impact to be indirect but potentially far reaching, e.g., heightened tension and a lack of well being.

Emotional Communication

How can a haptic channel support human emotional communication? As noted in [92], current collaborative systems demand explicit communication—symbolic, focused, and overt, with an emphasis on transferring information in support of a goal. The overall situation hasn't changed much in the intervening decade, despite many experimental efforts aimed at understanding nonverbal human communication and attempting to support it remotely.

Mediated social touch is "the ability of one actor to touch another over a distance by means of tactile or kinesthetic feedback technology" (for an excellent review, see [42]). A number of examples using haptics have been explored, using a variety of direct force connections or tactile taps and with purposes ranging from emotional connectedness to therapy and ambient communication (summarized in [63]). They are provocative and insightful, but together demonstrate that we need a more systematic investigation of how, exactly, we communicate emotion through touch alone. Early evidence is that we can do so [4], [88] in at least simplified contexts. In another work, we are building a touch sensing-and-display platform to study this in a less-constrained environment [107].

Haptic Interaction Design Practices

There is a wealth of information on the best practices for UI design—textbooks [8], [24], [91], courses, conferences, and journals. There is also a growing literature on the principles for multimodal interface design, which is relevant here [75], [86]. Nevertheless, what is special about the process of designing haptics into interfaces? Or, even better, designing the interface itself around the idea of physical interaction?

Technocentric Versus User Centric Design

Because this article appears in a robotics magazine, it is a good guess that most readers have a technical background and are highly skilled at making machines do things. This can be a big problem when it comes to creating systems that work well for people, for a couple of reasons. The comments that follow are in no way limited to the design of haptic interfaces. But we are particularly vulnerable: haptic feedback started with robotics, and arises out of a culture of respect (reverence?) for complexity and automation. Although for nearly a decade now it's been possible to design haptic applications without building your own device, the ones you can buy mostly don't do quite what you need them to, and the technology is young and demanding enough that it still attracts practitioners of a tinkering mentality.

Have Need, Then Seek Technology: If you're looking for an application that will show off your device's special features, you 1) might have a fruitless search or 2) could find a good match, but then fail to do a good job of integrating it. Human problems are usually better solved by looking closely at the need, then surveying technologies to find the best match. This isn't much help if you're the engineer and have spent a lot of time building a cool gadget. It's important to watch and listen to people and notice where they struggle, and to hold an open mind. Perhaps your original solution isn't the right one, but the problem is real and understanding it will guide you to a different and better one.

Multidisciplinary Teamwork: Cultivate friends and associates who aren't engineers. By far the most productive design teams we've worked with, whether professionals or students, contain technologists, interaction designers, end users (including those with special needs or profession), and artists, freely and respectfully sharing ideas and possibilities. The most effective individual designers are empowered to observe, envision, and build—all within one brain and set of hands. So, leave your own comfort zone and learn to do what your partners are doing too.

Define Requirements in Solution-Independent Terms: When you do identify what seems to be a good problem-technology match, don't just jump in. This means studying the people you hope to help, and what they do without the proposed fixes. Talk to them, understanding that they won't always be able to articulate problems or envision hugely different solutions. Identify what's needed in *solution-independent terms*. Then, and only then, is it time to formulate specific designs with their enabling technology and begin to refine them.

Designing for an Unfamiliar Modality

Haptic design does differ in a significant way from visual and auditory design, in that most users will be initially unfamiliar with most possible uses for haptic technology. This is difficult enough when you're trying to simulate reality in some way, but becomes even harder when you create sensations or interactions that don't occur at all in the natural world.

Lost in Translation: It is difficult to predict how a programmed sensation will feel or whether an interaction will help until you build it and compare it against other possibilities. This is partly a matter of unmodeled device dynamics and partly of uncatalogued perceptual sensitivity. When will a sensation be masked or attenuated by another? Design iteration needs to include feedback from humans (perceptual questions) and sample end users (interaction questions).

Difficulty of Status Quo Comparisons: We often wish to know whether a haptic version or augmentation of a traditional visual interface helps people do something better, and seek a way to compare them. However, it can be difficult or pointless to create comparable versions. They are likely to be different in *many* ways, and so you must choose between a highly controlled comparison where one version is not optimally configured, or a poorly controlled comparison where it's hard to identify causal factors. We believe the most informative compromise is often to compare the best-of-breed versions and focus on collecting and analyzing rich observational data, in contrast to a hypothesis-testing approach, which emphasizes quantitative performance measures and statistical differences.

Evaluation in the Middle of the Learning Curve: The playing field isn't always level. For example, our subjects have been using vision for the kinds of tasks we test since early childhood, and they've been using the tactile version for perhaps a 3–30-min training period. It can be difficult to determine whether an innovation has intrinsic value, or extrapolate where it will go with experience. Longitudinal studies where subjects have more opportunity to become familiar with the use of haptics are expensive but clearly necessary.

Haptic Representations and Verbalizing Sensations: People aren't accustomed to processing haptic representations of abstractions, and they don't have a vocabulary to describe or help them remember detailed haptic distinctions the way they do for sounds and colors. As designers, we don't have a clear idea of the design dimensions. We've made a small start at correcting this [94].

Importance of Rapid Prototyping and Haptic Representations

Regardless of detail, a well-recognized principle of prototyping is to iterate at increasing levels of detail, whether creating a piece of software or a mechanical linkage. You don't start by building a refined, feature-complete instantiation of your vision, because it is likely to be wrong in many ways; and then you will have wasted a lot of effort. It is far more expensive to make changes late in the process when details become

Haptic interactions employ mechanical, programmed physical devices.

rigidified than in early, conceptual stages. For UIs, the truth of this maxim grows. Although there are some trustworthy heuristics, it is difficult to predict user response to any kind of novelty—modeling, simulation, or established rules are sparse and can be difficult to apply. For haptic UIs, the unfamiliarity and the combination of hardware and software design further amplifies this.

Minimalism: Prototyping UIs is an activity that lies somewhere between art, psychology, and science. Little can be described or left to the imagination, since users don't have a useful reference point. However, when prototypes are too high-fidelity early in the design cycle, they can appear to be finalized to a user; who will then be less likely to challenge or suggest modifications.

Modular Prototyping: The primary objective of a prototype is to get your design question answered with as little effort as possible, starting with big ones and proceeding to more detailed ones. It can then be discarded when you move on. If you have an engineering feasibility question, then implement exactly the degree of functionality needed to test that. If you need to figure out if a physical configuration is going to work for a user, then a nonactuated mockup might allow you to get this feedback from a user for a lot less work than a functional model. If you need to test look-and-feel or aesthetics, a conceptual or even a graphical rendering could be sufficient.

Later in the process, it makes sense to prototype multiple aspects together. It's more expensive but by now major directions are confirmed, and risk is gradually being reduced. You'll continue to make new discoveries as more of the system comes online, and you are able to observe real users interacting with increasingly realistic and functional mockups. This modularity is illustrated in the first case study presented later.

Brainstorming and Multiple Approaches: Pursuing a single path to a design goal is unlikely to give the best result. Brainstorming (the wild, absurd kind) helps to generate creative, far-flung approaches which when recombined, toned down, and refined can open up new directions. Whenever possible, advance two or three different paths that are as different as possible. In the end, you'll likely combine elements of different approaches, and you'll have more understanding of the design landscape.

Tools: The principal danger of tools is their introduction of an insidious obstacle to innovation in alternate directions. Having a choice of tools and being aware of their constraints is helpful.

Triangulation in Prototype Creation and Evaluation: Each prototype is built to be evaluated in some way, whether mechanically or in terms of comprehensibility or aesthetics. Any kind of evaluation is flawed, in part because you're only prototyping and observing part of the whole experience. *Triangulation* refers to coming at each evaluative point from multiple directions, using techniques whose strengths and weaknesses complement one another. For example, performance-based and observational evaluations provide different views. For more on user evaluation, see an introductory HCI textbook (e.g., [8], [24], [91]).

Prototyping Things That Can't Be Built: As for any novel technology, to advance we often need to prototype the future. Today's hardware limits us, but if we can show real value for a technology we can't yet build, this can inspire development effort in that direction. For example, our group has put tactile displays into hand-held devices that cannot yet be built with sufficient compactness and power efficiency to actually be untethered. However, we won't know if it's worth finding a way to make this technical advance or be ready for it when it comes, if we haven't by then found a way to use it effectively.

Some Ideas for Getting Started

You have your real human problem, a technology that seems like it should help, and you're prepared to prototype. How do you start?

Each design problem is unique, and we're not at the point of recipes. Nevertheless, we can suggest some ways to get going, which may even end up as useful design approaches.

Use of Metaphor

When an information or control task has roots in predigital interactions, exploiting these roots by building metaphorical interactions around them can aid control and make it comprehensible. An example of this is introduced in the first case study, which describes a mediating virtual physical metaphor for interacting with media. The haptic representation is not of the media itself but of a virtual tool which has similarities to one that users might have once used in the real world [62], [89].

Navigating Modes

Haptic feedback is often proposed as a solution for *modal interfaces* in which the interface can be in different states, and a command thus means different things depending on the state. Problems arise with modal interfaces when the current state is not evident, or when it's hard to move between them. A haptic display (say, a knob with an embedded liquid crystal display) has possibilities here because unlike a physical knob, it can be reprogrammed appropriately for the current mode, just like the graphical display. However, when the graphical display goes away—or the user can't look at it for a while—then the haptic display must be able to transparently indicate the mode. The state of the art in our current hardware is point-based interaction for force feedback. This means that usually you have to explore an environment serially to deduce the state. This is undesirable, and you might inadvertently alter the state in the process. How can we get around this?

One approach is to redefine the interaction in a manner that either gets rid of modes altogether or allows the user's active, deliberate motions to alter or navigate through them in an intuitive way. At the same time, the interface can supply ongoing physical feedback about the state, without requiring continual system interrogation. Physical metaphor is a good way to enter into this idea, because it is how real hand-held tools work: e.g., you might shift the position of a tool in your hand or switch tools entirely (deliberate physical act) and then continue to receive feedback through the shape of the tool in your hand and the sensations transmitted during its use (think about how different writing and cutting implements feel, in terms of shape, heft, and transmitted forces and vibrations). It is hard to change the shape of a handle, but you might be able to change its virtual weight or center of inertia and certainly the vibrations.

Modal Continuums: Discrete and Continuous Control

We think of interface modes as being discrete states, but sometimes this is an artificial construct, and, in fact, the desired control shifts along a continuum. Again using the digital media example, observe how when traversing a media stream we move between discrete and continuous forms of the material, its content, and aggregations. Video is a succession of frames—discrete when played slowly but merged into a fluid when sped up. Spinning the virtual video reel of the first case study allows one to move seamlessly between these states: the ticking of individual frames speeds into a texture while the frame rate fuses visually. A collection of voice mail messages, music tracks, or cable TV channels are discrete objects; when played, individual items are continuous streams. If the set is represented in the right way, you can skim over the discrete items themselves like a texture, feeling for the variation that indicates the item property you're looking for.

Design Case Studies

We conclude with a pair of case studies that illustrate ways in which haptic feedback can be explicitly designed for an application context, chosen to span a broad space of application areas and a variety of principled design mechanisms. For authenticity and detail as well as brevity and focus, they are chosen from the authors' own experience.

Force Feedback Knob: Continuous and Discrete Hand-Held Media Control

Along with digitization of once-tangible tasks and microcomputers everywhere, comes the frequent necessity of managing information or controlling systems through very simple input devices. When hapticized, this generally comes down to knobs and sliders.

In this first example, we relate key points of a design sequence that relied on metaphor to create generalizable but experience-grounded interactions for a hand-held media controller [64], [65], [89], beginning with some relevant principles and observations. The first stages of this project were performed at Interval Research Corp. (Palo Alto, CA) during 1998–1999 by a design team lead by the first author. Later stages were conducted as student projects at University of British Columbia, Canada. This case also illustrates the modular prototyping principle described in the previous section. Starting from the ideas of metaphor-based design and discrete/continuous media modes, we set out to build a hand-held home media controller that would leverage the utility of modal interaction for different media in a consistent way, while making the state transparently clear.

Inspiring Metaphor: We tried a lot of metaphors! And ended up using several. One which felt good and aided navigation was a virtual "clutch" through which the user interacted with a heavy "reel" of film that runs on the computer screen as the reel spins (Figure 1). The inspiration for the bit of applied tangibility used here came from discussions with videographers who missed some aspects of traditional mechanisms for handling celluloid film. It allows a far more fluid handling of the information than cursor clicks or stop/start buttons.

Technical Grounding: A technical path for this was suggested by *tangible interfaces*, where tagged arbitrary objects (e.g., using radio frequency or RFID) can be used to issue commands to a computer [101]. Observing that tagged objects are well suited for issuing digital commands but not for exerting continuous control, we combined the two through the principle of *tagged handles* [64], [65].

Prototype-Driven Design Steps

Figure 2 illustrates several successive prototypes in an iterative conceptual and engineering evolution. In this process, exploration of the prototypes themselves drove further designs, and there was an emphasis on lightweight prototyping where possible. These began with an engineering exercise, shared informally with users, to see whether the combination of discrete (tagged handle) and continuous (force feedback knob) would be compelling [see the prototype in Figure 2(a)]. Each of the handles contained a unique RFID tag, which when installed on the force feedback knob caused the system to browse (and give appropriate force feedback for) a particular kind of content or functionality—e.g., a particular music track, selection of radio versus recorded content, or volume versus navigational control. No attempt was made at usability—for example, the handles did not suggest their function.

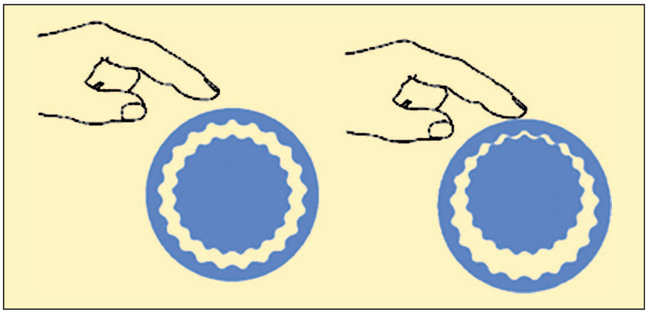


Figure 1. *Virtual clutch metaphor for the force feedback media controller. The knob is equipped with a crude pressure sensor. When the user presses down on the actual knob (which is associated with the outer wheel in this figure), the heavy inner wheel (virtual) is engaged and can be spun up. When the actual knob is released, the inner wheel continues with its imparted inertia. The video displayed on the screen is linked to the rotational speed of the virtual inner wheel. The bumps displayed here correspond to frames and are haptically rendered as small detents that fuse into a texture as the speed increases.*

The prototype in Figure 2(b) is an example of the many ideas explored mostly at a conceptual level in Figure 3. It is a nonfunctioning prototype showing one way that discrete handles (inspired by a charm bracelet) could informatively indicate their function and solve the practical problem of getting lost—the handle is selected from a wheel instead of being picked up and attached. Sadly, these protruding little handles would take a finger off when it rotated under active control, and several more nonfunctional prototypes (not shown) led to the next step.

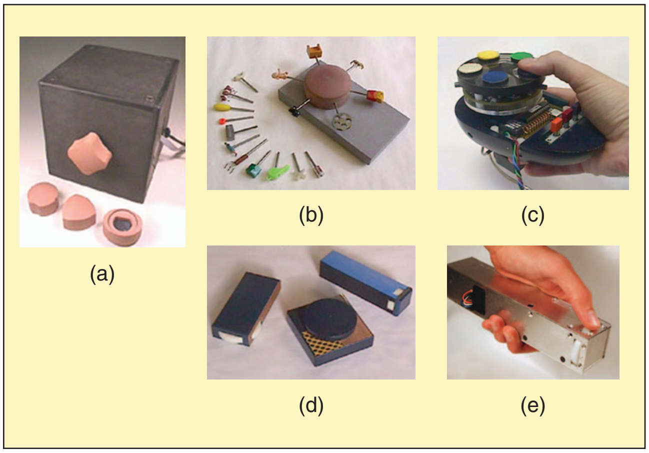


***Figure 2.** Representative haptic media controller design iterations: (a) The initial tagged handles engineering concept prototype. (b) A representative conceptual prototype. (c) A later technical prototype (oversized). (d) A set of nonfunctional concept prototypes that addresses the problems of (c). (e) Another engineering mockup.*

The prototype in Figure 2(c) is a fully functional implementation of a safer variant of the same idea—handles are replaced by texturally marked buttons on a rotating wheel mounted on a hand-held base. In using this mockup, we discovered a problem of disorientation. When the face rotated, the buttons moved, and they were hard to find again; spatial constancy turns out to be critical. The next refinement [Figure 2(d)] inverted this idea. A four-sided object with texturally marked sides and an active thumbwheel knows which face is active by measuring where the thumbwheel was pressed from and changing the function and feedback of the continuous interaction accordingly—e.g., turning from one face could change volume, and turning from another could select channel. Finally, Figure 2(e) is another engineering prototype of this final idea [64].

In Summary

This case study showed a prototype-dominated process, where user feedback was obtained informally at each stage. The use of varied, focused, and stage-appropriate prototypes allowed us to identify key strengths and weaknesses with minimal effort. This example did not make use of extensive, formally controlled user studies for feedback on the prototypes because the concept clearly had many bugs to be worked out before we even reached that stage. However, it was inspired and informed by parallel efforts at the host company, consisting of extensive ethnographic studies of target user groups in their uses of home media, and interviews focusing on their difficulties with currently available models. That is, the user-centered component was upfront observation, and the next step would have been a usability study . . . if the host company hadn't vaporized in the 2000 tech bust.

***Figure 3.** Early prototype for the hand-held media controller project. Found objects and state-of-art examples, Lego + rubber band transmissions, whimsical and serious nonfunctioning concepts, and narrowly targeted functional engineered prototypes.*

Vibrotactile Background Signals

Our second example, in contrast, is heavy on the user studies. Its goal was a first deployment of a set of haptic icons in an application concept. It began with devising an initial set of icons using a symbolic approach based on metaphors thought to intuitively represent the concepts being represented. The icon set was then systematically refined in an iterative, user-centered process mentioned previously (see the "Tactile Signaling in Multitasking Environments" section) and culminating in an *observational user study.* These steps are more fully described in [16] and [17], and we summarize some key points here.

Application: When noncolocated and collaborating users wish to jointly modify a shared object displayed on their local screens—whether a text document, a computer-aided design drawing or a Photoshop file—current technology (e.g., virtual network protocol or VNC) allows only one of them to control the cursor at a time. Somehow, they need to negotiate turn-taking, but in the absence of the nonverbal cues that are so important in colocated situations. (Our own guess is that even the usual nonverbal cues available in colocated meetings could use help too. Could tactile cues discretely remind someone who's impervious to coughs, raised hands, and squirming, that it's really time to stop monopolizing the floor?)

We began with the proposition that tactile feedback could provide a background awareness of others' wish to participate. It could indicate both turn-request queue and urgency of items in the queue, in a less intrusive manner than visual or auditory methods could support—because the latter were also being used in the collaborative task. We further wondered if the ability to make a request gently or urgently would support more equitable control sharing. A quiet or shy team member might be more comfortable asking for control "whenever you're ready," as opposed to "right now!." It was problematic for visual or auditory protocols to support this. Requests not dealt with right away couldn't easily persist, because they'd either be in the way or forgotten.

The only way we could test this idea (which we hoped was representative of a whole class of applications) was to build up a set of icons and try it out on users in a realistic situation.

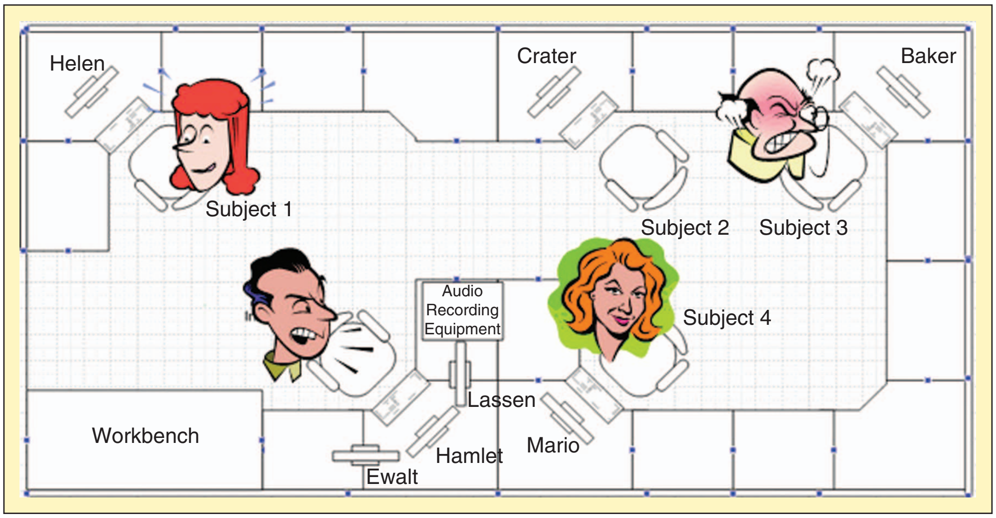


Figure 4. *Experimental setup for the observational study of the turn-taking protocol. The four group members were placed out of direct eyeshot and wore noise-canceling headphones; all vocal communications occurred through a sound system.*

Experiment Paradigm and Display Hardware: The climactic observational study involved groups of four friends who were placed out of direct eye- and earshot of one another (Figure 4) and given voice links and a shared screen view of a common application (a furniture-layout task using Visio). They received tactile feedback through modified tactile mice (Logitech IFeel; Figure 5). Although more expressive displays were available, we wanted to see how far we could get with commodity hardware. Groups performed the room layout task three times: with only tactile mediation, with only visual mediation (following state-of-art visual protocols), and using both modalities. Each member was given responsibility for a subset of the criteria that had to be followed in the solution, and the group collectively got a bonus if they did particularly well. Their interactions were closely monitored.

Haptic feedback has been shown to have value in the training of sensorimotor tasks.

Protocol and Initial Icon Creation: With this scenario in mind, we designed the turn-taking protocol and the initial set of haptic stimuli that would support it, as well as the analogous visual signals. In essence, the protocol recognized three classes of users—those in control, those waiting for control, and those just observing; two types of requests—urgent and gentle; and two types of events—an urgent or a gentle request and a self-removal from the queue. Seven icons were needed to display the current context as relevant to a given user. For example, the user who was in control would experience a different signal than one who was in the queue. The haptic stimuli which were eventually used, are shown in Figure 6. We

Figure 5. *Vibrotactile mouse used to display the haptic icons used in the turn-taking protocol. Two buttons were added to the side to enable special protocol features; such buttons were available in other mouses at the time but not the vibrotactile one.*

The presence of the visual and auditory stimuli can significantly modify the user's interpretation of what they feel.

used a metaphor-based design on the assumption that it would make this small set easier to learn. For example, the change of control states were suggestive of the be-BEEP, BE-beep of the common auditory cue indicating the insertion or removal of a hardware device from your computer.

Process: User-Focused Icon Set Refinement and In Situ Observation

We were too experienced with haptic icons to think we were ready for prime time, though. Would users actually be able to learn them? Would they be confused with one another? Was their salience correctly adjusted? We thus commenced on a multistep refinement process. The initial icon set design described above was Step I (we're currently working on alternatives to its fairly ad hoc nature).

In Step II, we perceptually adjusted the icon set using the MDS technique mentioned previously, testing the most likely candidates along with a lot of others. A few iterations of this served to ensure that all the icons in the set were well distributed within the engineering design space.

In Step III, we "stress-tested" the icon set in realistic conditions, by requiring subjects to learn associations, then abstractly simulating various aspects of the anticipated workload (with appropriate visual and auditory load), and examining how icon detection and identification degraded. We wanted some icons to be less detectable under workload, while others should always get through. For example, an *in control* user should always perceive and recognize an urgent request, but while concentrating hard, he shouldn't be bothered with a gentle request—that was the whole point of the urgency-based protocol. Following this test, we adjusted some of the signals even more to get the desired salience patterns. Subjects learned the seven mediating icons easily in three minutes and maintained 97% accuracy of identification under substantial multimodal workload.

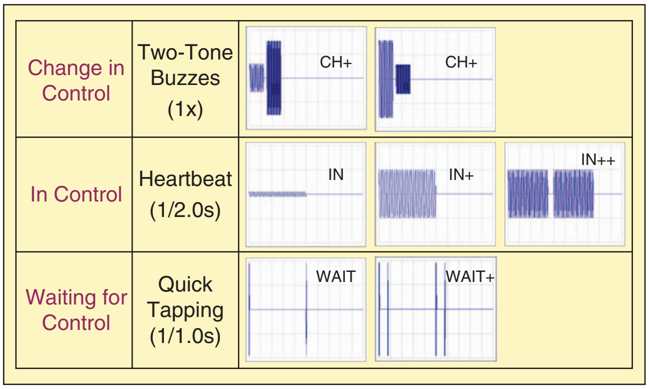


Figure 6. *Final set of haptic icons used in the turn-taking protocol.*

Unfortunately, we did not then return to Step II to readjust their perceptual spacing; next time we will! The salience adjustment did, we later learned, make some pairs harder to distinguish.

Finally, in Step IV, we mounted the group observational study, and learned quite a lot (read the article). Through a combination of performance and subjective measures we did confirm that the haptic signals were utilized in a graded (i.e., appropriate) way, and collaboration dynamics seemed to be positively affected in comparison to the visual cue case. Users, however, preferred having *both* visual and haptic cues available to them.

In Summary

This case exemplifies a quite user-intensive design process. The hardware itself was simple, but what we did with it would fail or succeed based on subtle details, and this could only be determined by trying it out while watching closely. The final endeavor was an observational rather than tightly controlled, performance-oriented study, out of a combination of necessity and design. Because each session was a lot of work, we could only run four groups of varied background, and thus there wasn't enough data to give statistical results. However, by observing and logging everything and following up with detailed interviews (and a second set of interviews a month later after looking over the data) we obtained a great deal of complex and nuanced feedback on the strengths and weaknesses of the approach. Given that there are many ways to implement this general concept, observational data were more valuable at this stage than hard performance data.

Summary

In this second part of our series, we have introduced the concept of and argued the need for explicit, user-centered interaction design for applications using haptic interfaces. We elaborated on a number of potential interface roles where haptic feedback is well suited to provide value, on the basis of the technology's alignment with human capabilities and modern needs, and we suggested some high-level principles to be followed and the pitfalls to be avoided during the application design process. Finally, we illustrated these with two case studies, chosen for their different approaches to the interaction design process.

Readers who are interested in learning more should start by learning about HCI practices in general, through textbooks and courses. Many aspects of user-centered design practices apply here but are unfamiliar to the engineering world. A working knowledge of haptic perception is essential as well. Because this frontier is advancing so rapidly, simply following these articles in haptics conferences will get you far, as well as the survey material mentioned earlier.

In Part I, we introduced the haptic devices themselves, their construction, and operating principle and placed special emphasis on some simple display variants that can be constructed and employed with little special expertise. We hope that our comments in Part II, in tandem with the electromechanical design principles in Part I, will lower the barrier to entry for this exciting young field, and foment many new ideas—usable ones!

Haptic design is nearly always a multimodal design.

Keywords

Haptic interfaces, interaction design, ubiquitous computing, force feedback, tactile feedback, human computer interaction.

References

[1] R. J. Adams, D. Klowden, and B. Hannaford, "Virtual training for a manual assembly task," *Haptics-e*, vol. 2, no. 2, 2001.

[2] C. A. Avizzano, J. Solis, and M. Bergamasco, "Teaching to write Japanese characters using a haptic interface," presented at the *10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Washington, DC, 2002.

[3] P. Bach-y-Rita, K. A. Kaczmarek, M. E. Tyler, and M. Garcia-Lara, "Form perception with a 49-point electrotactile stimulus array on the tongue: A technical note," *J. Rehab. Res. Dev.*, vol. 35, no. 4, pp. 427–430, 1998.

[4] J. N. Bailenson, N. Yee, S. Brave, D. Merget, and D. Koslow, "Virtual interpersonal touch: Expressing and recognizing emotions through haptic devices," *Hum. Comput. Interact.*, to be published.

[5] N. Barraga and J. Erin, *Visual Handicaps and Learning*. Austin, TX: ProEd, 1992.

[6] C. Basdogan, C. H. Ho, M. Srinivasan, and M. Slater, "An experimental study on the role of touch in shared virtual environments," *ACM Trans. Comput. Hum. Interact. (TOCHI: Special Issue on Human Computer Interaction and Collaborative Virtual Environments)*, vol. 7, no. 4, pp. 443–460, 2000.

[7] T. Beamish, K. MacLean, and S. Fels, "Manipulating music: Multimodal interaction for djs," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI'04), CHI Lett.*, Vienna, Austria, 2004, vol. 6, no. 1, pp. 327–334.

[8] D. Benyon, P. Turner, and S. Benyon, *Designing Interactive Systems: People, Activities, Contexts, Technologies*. Reading, MA: Addison Wesley, 2005.

[9] P. Bertelson and B. de Gelder, "The psychology of crossmodal attention," *Crossmodal Space and Crossmodal Attention*, C. Spence and J. Driver, Eds. Oxford: Oxford University Press, 2004, pp. 141–177.

[10] S. J. Blake. (2007). *Teaching Reading in Braille*. [Online]. Available: http://blindness.growingstrong.org/ed/aa010401a.htm

[11] A. J. Bongers, "Interactivation: Towards an e-cology of people, our technological environment, and the arts," Proefschrift, Vrije Universiteit, Amsterdam, The Netherlands, 2006.

[12] S. Brave and A. Dahley, "Intouch: A medium for haptic interpersonal communication," in *Proc. Conf. Human Factors in Computing Systems (CHI'97)*, pp. 363–364.

[13] L. M. Brown, S. A. Brewster, and H. C. Purchase, "Multidimensional tactons for non-visual information presentation in mobile devices," in *Proc. 8th Conf. Human-Computer Interaction with Mobile Devices and Services*, Helsinki, Finland, 2006, pp. 231–238.

[14] D. B. Camarillo, T. M. Krummel, and J. K. Salisbury, "Robotic technology in surgery: Past, present, and future," *Am. J. Surg.*, vol. 188, no. 4A, pp. 2S–15S, 2004.

[15] C. S. Campbell, S. Zhai, K. W. May, and P. Maglo, "What you feel must be what you see: Adding tactile feedback to the trackpoint," in *Proc. Interact'99*, Edinburgh, UK, pp. 383–390.

[16] A. Chan, K. E. MacLean, and J. McGrenere, "Learning and identifying haptic icons under workload," in *Proc. 1st Joint Eurohaptics Conf. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC 2005, IEEE-VR2005)*, Pisa, Italy, pp. 432–439.

[17] A. Chan, K. E. MacLean, and J. McGrenere, "Designing haptic icons to support collaborative turn-taking," *Int. J. Hum. Comput. Stud.*, to be published.

[18] A. Chang, S. O'Modhrain, R. Jacob, E. Gunther, and H. Ishii, "Comtouch: Design of a vibrotactile communication device," in *Proc. Conf. Designing Interactive Systems (DIS'02)*, London, pp. 312–320.

[19] S. Choi and H. Z. Tan, "An analysis of perceptual instability during haptic texture rendering," in *Proc. 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Los Alamitos, CA, 2002, pp. 129–136.

[20] P. Cook, "Principles for designing computer music controllers," in *Proc. Conf. New Interfaces for Musical Expression*, Seattle, Washington, 2001, pp. 1–4.

[21] J. C. Craig, "Vibrotactile pattern perception: Extraordinary observers," *Science*, vol. 196, no. 4288, pp. 450–452, 1977.

[22] S. Delp, P. Loan, C. Basdogan, and J. M. Rosen, "Surgical simulation: An emerging technology for emergency medical training," *Presence Teleoper. Virtual Environ.*, vol. 6, no. 2, pp. 147–159, 1997.

[23] D. DiFilippo and D. K. Pai, "Contact interaction with integrated audio and haptics," presented at *Int. Conf. Auditory Display (ICAD'00)*.

[24] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2003.

[25] S. R. Dixon, C. D. Wickens, and J. S. McCarley, "On the independence of compliance and reliance: Are automation false alarms worse than misses?" *Hum. Factors*, vol. 49, no. 4, pp. 564–572, 2007.

[26] M. Enriquez, O. Afonin, B. Yager, and K. MacLean, "A pneumatic tactile notification system for the driving environment," in *Proc. Workshop on Perceptive User Interfaces (PUI'01)*, Orlando, FL, pp. 1–7.

[27] M. Enriquez and K. MacLean, "Impact of haptic warning signal reliability in a time-and-safety-critical task," in *Proc. 12th Annu. Symp. Haptic Interfaces for Virtual Environments and Teleoperator Systems (IEEE-VR2004)*, Chicago, IL, pp. 407–415.

[28] M. Enriquez and K. E. MacLean, "The role of choice in longitudinal recall of meaningful tactile signals," presented at the *IEEE Symp. Haptic Interfaces (HAPTICS 2008)*, Reno, NV.

[29] M. Enriquez, K. E. MacLean, and C. Chita, "Haptic phonemes: Basic building blocks of haptic communication," in *Proc. 8th Int. Conf. Multimodal Interfaces (ICMI 2006)*, Banff, Canada, pp. 302–309.

[30] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, pp. 429–433, 2002.

[31] D. Feygin, M. Keehner, and F. Tendick, "Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill," in *Proc. 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Los Alamitos, CA, 2002, pp. 40–47.

[32] J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM Trans. Comput. Hum. Interact.*, vol. 12, no. 1, pp. 119–146, 2005.

[33] B. Fogg, L. D. Cutler, P. Arnold, and C. Eisbach, "Handjive: A device for interpersonal haptic entertainment," in *Proc. Conf. Human Factors in Computing Systems*, 1998, pp. 57–64.

[34] B. Forsyth and K. E. MacLean, "Predictive haptic guidance: Intelligent user assistance for the control of dynamic tasks," *IEEE Trans. Visualizat. Comput. Graph.*, vol. 12, no. 1, pp. 103–113, 2006.

[35] A. Gallace, H. Z. Tan, and C. Spence, "The body surface as a communication system: The state of the art after 50 years," *Presence Teleoper. Virtual Environ.*, vol. 16, no. 6, pp. 655–676, Dec. 2007.

[36] J. J. Gibson, "Observations on active touch," *Psychol. Rev.*, vol. 69, no. 6, pp. 477–490, 1962.

[37] B. Gillespie, "The touchback keyboard," in *Proc. Int. Computer Music Conf.*, San Jose, CA, 1992, pp. 447–448.

[38] B. Gillespie, S. O'Modhrain, C. P. Tang, and D. Zaretsky, "The virtual teacher," in *Proc. ASME Dynamic Systems and Control Division*, 1998, vol. 64, pp. 171–178.

[39] A. Grant, M. Thiagarajah, and K. Sathian, "Tactile perception in blind Braille readers: A psychophysical study of acuity and hyperacuity using gratings and dot patterns," *Percept. Psychophys*, vol. 62, no. 2, pp. 301–12, 2000.

[40] W. B. Griffin, W. R. Provancher, and M. R. Cutkosky, "Feedback strategies for shared control in dexterous telemanipulation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, vol. 3, pp. 2791–2796.

[41] P. Griffiths and R. B. Gillespie, "Shared control between human and machine: Haptic display of automation during manual control of vehicle heading," in *Proc. 12th Int. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*, pp. 358–366.

[42] A. Haans and W. Ijsselsteijn, "Mediated social touch: A review of current research and future directions," *Virtual Reality*, vol. 9, no. 2, pp. 149–159, 2006.

[43] R. H. Hamilton and A. Pascual-Leone, "Cortical plasticity associated with Braille learning," *Trends Cogn. Sci.*, vol. 2, no. 5, pp. 168–174, 1998.

[44] V. Hayward, "A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store," *Brain Res. Bull. (Special Issue on Robotics and Neuroscience)*, to be published.

[45] M. Hollins, S. Bensmaïa, K. Karlof, and F. Young, "Individual differences in perceptual space for tactile textures: Evidence from multidimensional scaling," *Percept. Psychophys.*, vol. 62, no. 8, pp. 1534–1544, 2000.

[46] E. Horvitz, C. Kadie, T. Paek, and D. Hovel, "Models of attention in computing and communication: From principles to applications," *Commun. ACM*, vol. 46, no. 3, pp. 52–59, 2003.

[47] R. D. Howe and Y. Matsuoka, "Robotics for surgery," *Annu. Rev. Biomed. Eng.*, vol. 1, pp. 211–240, 1999.

[48] S. C. Jacobsen, F. M. Smith, D. K. Backman, and E. K. Iversen, "High performance, high dexterity, force reflective teleoperator II," in *Proc. ANS Topical Meeting on Robotics and Remote Systems*, Albuquerque, NM, 1991, pp. 1–10.

[49] L. Jones and N. Sarter, "Tactile displays: Guidance for their design and application," *Hum. Factors: J. Hum. Factors Ergon. Soc.*, to be published.

[50] L. A. Jones, M. Nakamura, and B. Lockyer, "Development of a tactile vest," in *Proc. Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*, Chicago, IL, pp. 82–89.

[51] R. Kikuuwe and T. Yoshikawa, "Haptic display device with fingertip presser for motion/force teaching to human," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2001, pp. 868–873.

[52] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human machine collaborative systems for microsurgical applications," *Int. J. Robot. Res.*, vol. 24, no. 9, pp. 731–742, 2005.

[53] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neuro-rehabilitation," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 1, pp. 75–87, 1998.

[54] R. Leung, K. E. MacLean, M. B. Bertelsen, and M. Saubhasik, "Evaluation of haptically augmented touch screen GUI elements under cognitive load," presented at the *9th Int. Conf. Multimodal Interfaces (ICMI'07)*, pp. 374–381.

[55] D. J. Levitin, K. E. MacLean, M. V. Mathews, L. Chu, and E. R. Jensen, "The perception of cross-modal simultaneity," *Comput. Anticipat. Syst.*, vol. 517, pp. 323–329, 2000.

[56] M. Lin and M. Otaduy, *Haptic Rendering: Foundations, Algorithms and Applications*. Wellesley, MA: A.K. Peters, to be published.

[57] M. Lin and R. H. Taylor, "Spatial motion constraints in medical robot using virtual fixtures generated by anatomy," in *Proc. IEEE Int. Conf. Robotics Automation (ICRA'04)*, vol. 2, pp. 1270–1275.

[58] J. G. Linvill and J. C. Bliss, "A direct translation reading aid for the blind," in *Proc. IEEE*, 1966, vol. 54, pp. 40–51.

[59] A. Luciani, J. Florens, and N. Castagne, "From action to sound: A challenging perspective for haptics," in *Proc. 1st Joint Eurohaptics Conf. and Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'05)*, pp. 592–595.

[60] J. Luk, J. Pasquero, S. Little, K. MacLean, V. Levesque, and V. Hayward, "A role for haptics in mobile interaction: Initial design using a handheld tactile display prototype," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI'06)*, Montreal, Canada, vol. 8, no. 1, pp. 171–180.

[61] K. MacLean and M. Enriquez, "Perceptual design of haptic icons," presented at *Eurohaptics*, Dublin, Ireland, 2003.

[62] K. E. MacLean, "Designing with haptic feedback," in *Proc. IEEE Robotics and Automation (ICRA'00)*, San Francisco, CA, vol. 1, pp. 783–788.

[63] K. E. MacLean, "Haptics in the wild: Interaction design for everyday interfaces," *Reviews of Human Factors and Ergonomics*, M. Carswell, Ed. Santa Monica, CA: Human Factors and Ergonomics Society, to be published.

[64] K. E. MacLean, M. J. Shaver, and D. K. Pai, "Handheld haptics: A USB media controller with force sensing," in *Proc. IEEE VR2002 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, FL pp. 311–318.

[65] K. E. MacLean, S. S. Snibbe, and G. Levin, "Tagged handles: Merging discrete and continuous control," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI'00), CHI Lett.*, The Hague, Netherlands, vol. 2, no. 1, pp. 225–232.

[66] S. Millar, *Reading by Touch*. London: Routledge, 1997.

[67] S. Millar, "Memory in touch," *Psicothema*, vol. 11, no. 4, pp. 747–767, 1999.

[68] T. Moran and P. Dourish, "Introduction to this special issue on context-aware computing," *Hum. Comput. Interact.*, vol. 16, no. 2, pp. 87–95, 2001.

[69] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, "Haptic feedback enhances force skill learning," in *Proc. IEEE World Haptics*, 2007, pp. 21–26.

[70] D. A. Norman, *Emotional Design: Why We Love (or Hate) Everyday Things*. New York: Basic Books, 2003.

[71] D. A. Norman, "The next UI breakthrough, Part 2: Physicality," *ACM Interact.*, vol. 14, no. 3, pp. 44–45, 2007.

[72] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 588–592, 2005.

[73] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li, "Shared control in haptic systems for performance enhancement and training," *J. Dyn. Syst. Meas. Control*, vol. 128, no. 1, pp. 75–85, 2006.

[74] D. O'Sullivan and T. Igoe, *Physical Computing*. Boston, MA: Thomson Course Technology, 2004.

[75] S. Oviatt, "Multimodal interfaces," *Handbook of Human–Computer Interaction*, J. Jacko and A. Sears, Eds. Hillsdale, NJ: Lawrence Erlbaum Associates, 2002, pp. 286–304.

[76] M. Pantic, A. Pentland, A. Nijholt, and T. Huang, "Human computing and machine understanding of human behavior: A survey," in *Proc. 8th Int. Conf. Multimodal Interfaces (ICMI'06)*, Banff, Alberta, Canada, pp. 239–248.

[77] J. Pasquero. (2006). Survey on communication through touch, TR-CIM 06.04. Center for Intelligent Machines, McGill University, Montreal, Canada. [Online]. Available: http://www.cim.mcgill.ca/~haptic/pub/JP-CIM-TR-06.pdf

[78] J. Pasquero, J. Luk, S. Little, and K. Maclean, "Perceptual analysis of haptic icons: An investigation into the validity of cluster sorted MDS," in *Proc. 14th Symp. Haptic Interfaces for Virtual Environments and Teleoperator Systems (IEEE-VR'06)*, Alexandria, VA, pp. 437–444.

[79] J. Payette, V. Hayward, C. Ramstein, and D. Bergeron, "Evaluation of a force-feedback (haptic) computer pointing device in zero gravity," in *Proc. 5th ASME Symp. Haptic Interfaces for Virtual Environments and Teleoperated Systems*, 1996, vol. 58, pp. 547–553.

[80] R. W. Picard, *Affective Computing*. Cambridge, MA: MIT Press, 1997.

[81] I. Poupyrev and S. Maruyama, "Tactile interfaces for small touch screens," in *Proc. 16th Annu. ACM Symp. User Interface Software and Technology (UIST'03)*, pp. 217–220.

[82] L. Rosenberg, "Virtual fixtures: Perceptual overlays enhance operator performance in telepresence tasks," Ph.D. thesis, Stanford Univ., CA, 1994.

[83] J. Rovan and V. Hayward, "Typology of tactile sounds and their synthesis in gesture-driven computer music performance," *Trends in Gestural Control of Music*, M. Wanderley and M. Battier, Eds. Paris: IRCAM, 2000, pp. 297–320.

[84] S. Saga, N. Kawakami, and S. Tachi, "Learning effect of haptic teaching using opposite force presentation," *Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu*, vol. 2005, p. 1P2-N-038, 2005.

[85] E. Sallnas, K. Rassmus-Grohn, and C. Sjostrom, "Supporting presence in collaborative environments by haptic force feedback," *Trans. Comput. Hum. Interact. (TOCHI)*, vol. 7, no. 4, pp. 461–476, 2000.

[86] N. B. Sarter, "Multimodal information presentation: Design guidance and research challenges," *Int. J. Ind. Ergonom.*, vol. 36, no. 5, pp. 439–445, 2006.

[87] R. M. Satava, "Emerging technologies for surgery in the 21st century," *Arch. Surg.*, vol. 134, no. 11, pp. 1197–1202, 1999.

[88] J. Smith and K. E. MacLean, "Communicating emotion through a haptic link: Design space and methodology," *Int. J. Hum. Comput. Stud. (IJHCS) (Special Issue on Affective Evaluation—Innovative Approaches)*, vol. 65, no. 4, pp. 376–387, 2007.

[89] S. S. Snibbe, K. E. MacLean, R. Shaw, J. B. Roderick, W. Verplank, and M. Scheeff, "Haptic metaphors for digital media," in *Proc. ACM Symp. User Interface Software and Technology (UIST'01)*, Orlando, FL pp. 199–208.

[90] C. Spence, M. E. R. Nicholls, N. Gillespie, and J. Driver, "Cross-modal links in exogenous covert spatial orienting between touch, audition, and vision," *Percept. Psychophys.*, vol. 60, no. 4, pp. 544–557, 1998.

[91] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha, *User Interface Design and Evaluation*. (*Interactive Technologies Series*). San Matio, CA: Morgan Kaufmann, 2005.

[92] R. Strong and B. Gaver, "Feather, scent, and shaker: Supporting simple intimacy," in *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'96)*, New York, NY pp. 29–30.

[93] C. Swindells, K. E. MacLean, K. S. Booth, and M. Meitner, "Exploring affective design for physical controls," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI'07), CHI Lett.*, San Jose, CA, vol. 9, no. 1, pp. 933–942.

[94] C. Swindells, J. D. Smith, and K. E. MacLean, "An exploration of representations to aid design of haptic behaviours," in *Proc. CHI 2005 Workshop—Hands-on Haptics: Exploring Non-Visual Visualisation Using the Sense of Touch*, Portland, OR, pp. 5–8.

[95] H. Z. Tan, R. Gray, J. J. Young, and R. Traylor, "A haptic back display for attentional and directional cueing," *Haptics-e:Electron. J. Haptics Res.*, vol. 3, no. 1, 2003.

[96] H. Z. Tan and A. Pentland, "Tactual displays for sensory substitution and wearable computers," *Fundamentals of Wearable Computers and Augmented Reality,* W. Barfield and T. Caudell, Eds. Mahwah, NJ: Lawrence Erlbaum Associates, 2001, pp. 579–598.

[97] A. Tang, P. McLachlan, K. Lowe, R. S. Chalapati, and K. E. MacLean, "Perceiving ordinal data haptically under workload," in *Proc. 7th Int. Conf. Multimodal Interfaces (ICMI'05)*, Trento, Italy, pp. 244–251.

[98] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. Dejuan, and L. Kavoussi, "A steady-hand robotic system for microsurgical augmentation," *Int. J. Robotics Res.*, vol. 18, no. 12, pp. 1201–1210, 1999.

[99] C. L. Teo, E. Burdet, and H. P. Lim, "A robotic teacher of Chinese handwriting," in *Proc. 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2002*, pp. 335–341.

[100] D. Ternes, "Building large sets of haptic icons: Rhythm as a design parameter, and between-subjects MDS for evaluation," M.Sc. thesis, Univ. British Columbia, Vancouver, BC, Canada, 2007.

[101] B. Ullmer, H. Ishii, and D. Glass, "Mediablocks: Physical containers, transports, and controls for online media," in *Proc. Siggraph'98*, pp. 379–386.

[102] J. B. van Erp, H. A. Veen, C. Jansen, and T. Dobbins, "Waypoint navigation with a vibrotactile waist belt," *ACM Trans. Appl. Percept.*, vol. 2, no. 2, pp. 106–117, 2005.

[103] J. B. F. van Erp and M. M. A. Spapé, "Distilling the underlying dimensions of tactile melodies," in *Proc. Eurohaptics*, Dublin, Ireland, 2003, pp. 111–120.

[104] R. T. Verillo, "Vibration sensation in humans," *Music Percept.*, vol. 9, no. 3, pp. 281–302, 1992.

[105] W.-C. Wu, C. Basdogan, and M. A. Srinivasan, "Visual, haptic and bimodal perception of size and stiffness in virtual environments," in *Proc. 8th Annu. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME/IMECE*, Nashville, TN, 1999, vol. 67, pp. 19–26.

[106] Y. Yanagida, M. Kakita, R. W. Lindeman, Y. Kume, and N. Tetsutani, "Vibrotactile letter reading using a low-resolution tactor array," in *Proc. 12th Int. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*, pp. 400–406.

[107] S. Yohanan, M. Chan, J. Hopkins, H. Sun, and K. E. MacLean, "Hapticat: Exploration of affective touch," in *Proc. 7th Int. Conf. Multimodal Interfaces (ICMI'05)*, Trento, Italy, pp. 244–251.

Karon E. MacLean received the B.S. degree in biological sciences and mechanical engineering from the Stanford University, California, in 1986 and the M.S. degree in mechanical engineering from MIT in 1988. She served as an engineer at the Center for Engineering Design, University of Utah, from 1988 to 1990. She received her Ph.D. in mechanical engineering from MIT in 1996. She was a member of the research staff of Interval Research Corp. from 1996 to 2000. She is now an associate professor at the Computer Science Department of the University of British Columbia. She is concerned with what haptic interfaces do; how they work; the way they feel, sound, and look; and how users will perceive them. She leads the Sensory Perception and Interaction Research Group (SPIN Lab) at the University of British Columbia.

Vincent Hayward received the Diplôme d'Ingénieur from École Centrale de Nantes in 1978. He received his Ph.D. in computer science from the University of Paris in 1981. He was Chargé de recherches at CNRS, France, from 1983 to 1986. Currently, he is professor of electrical and computer engineering at McGill University, Quebec, Canada. Hayward is interested in haptic interfaces design as well as applications, perception, and robotics. He is leading the Haptics Laboratory at McGill University. He is the recipient of the NASA Space Act Tech Brief Award (1991) and the E. (Ben) & Mary Hochhausen Award for Research in Adaptive Technology for Blind and Visually Impaired Persons (2002). He has several publications in the area of robotics and haptic interfaces, has served on several editorial boards, and has participated in the creation of spin-off companies.

Address for Correspondence: Karon MacLean, Department of Computer Science, The University of British Columbia, ICICS/CS Building, 201-2366 Main Mall, Vancouver, B.C., V6T 1Z4. E-mail: maclean@cs.ubc.ca.

Revised *JCR* Data Bring *T-RO* Back to the Top

The *IEEE Transactions on Robotics* (*T-RO*) is the top-ranking journal in the robotics category in the latest *2006 Journal Citation Reports* (*JCR*), with an impact factor of 1.763, the highest in this category.

The impact factor of a journal is a calculated figure indicating the average number of times articles published in that journal in the previous two years are cited by any publication in the current year (2006 in this case). Collecting these data (for nearly 6,200 highly cited titles in a wide variety of disciplines) requires some time; this is why the *JCR* edition of a specific year is published by mid-June of the following year.

When the 2006 *JCR* was published in June 2007, *T-RO* appeared to have slipped to third place in the impact factor in the robotics category. Editor-in-Chief Alessandro De Luca, with the help of *T-RO* Associate Editor Juan Tardos, carefully analyzed the data and concluded that a number of 2006 citations to *T-RO* were missing. This information was conveyed to the *JCR* publisher, Thomson Scientific.

After doing its own analysis, the publisher concurred and confirmed that the error identified by De Luca and Tardos was the specific cause for the lower and erroneous 2006 citation count for *T-RO*. Thomson usually revises erroneous data in the fall. Correct data for *T-RO* appear in the reissued version of the 2006 *JCR* that became available in late October 2007.

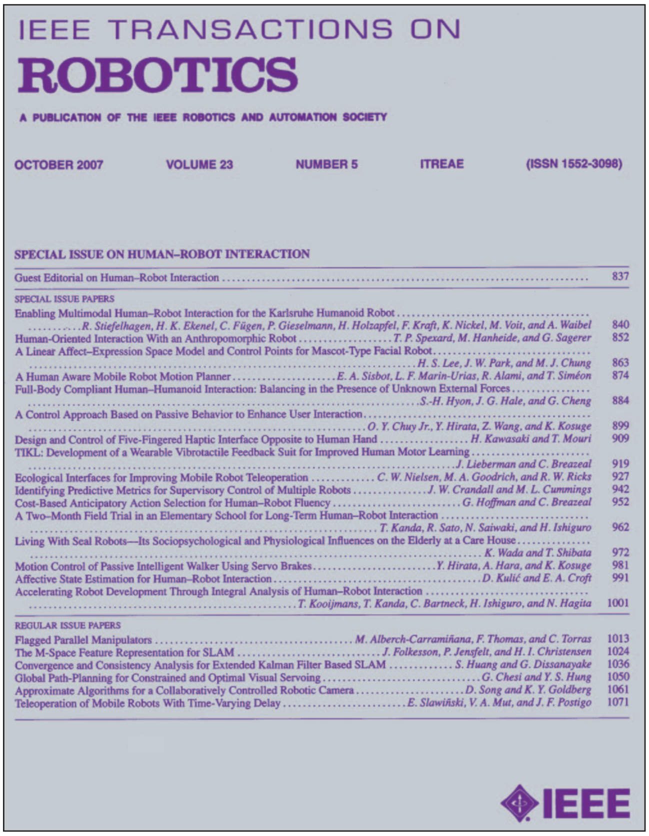
IEEE Transactions on Robotics and the preceding journal, *IEEE Transactions on Robotics and Automation*, have been ranked number one in the robotics category for several years and specifically for the five-year period of 2002–2006. *IEEE Robotics and Automation Magazine* ranks sixth in the 2006 *JCR* for this category, which includes a total of 12 journals.

Besides being the leading journal in impact factor, *T-RO* is now also number one for the immediacy index in Robotics (0.208, almost doubled with respect to the previous year). This index is the average number of times articles published in a specific journal are cited over the course of the same year, an indicator of how a journal is publishing in emerging areas of research.

Many university tenure and promotion committees consider the impact factor and other citation indices of publications when they evaluate the quality of journals where faculty members publish their work.

The 2007 edition of the *JCR* will be published in June 2008.

(continued on page 124)



IEEE TRANSACTIONS ON
ROBOTICS
A PUBLICATION OF THE IEEE ROBOTICS AND AUTOMATION SOCIETY

OCTOBER 2007 VOLUME 23 NUMBER 5 ITREAE (ISSN 1552-3098)

SPECIAL ISSUE ON HUMAN–ROBOT INTERACTION

IEEE

Digital Object Identifier 10.1109/M-RA.2007.914922

T-RO Has a New Paper Review System

Starting with 1 January 2008, the *T-RO* has a new fully Web-based paper review and management system, which is powered by PaperCept. New submissions should go to http://ras.papercept.net/journals/tro. There are three categories of submissions: regular papers, short papers, and communication items. The first two also allow multimedia attachments (e.g., videos) to be uploaded. Information for authors can be found in the new IEEE Robotics and Automation Society (RAS) *T-RO* Web site http://www.ieee-ras.org/tro.

For submission, all authors of a paper should possess a PIN number from the RAS PaperPlaza conference management system. If you have submitted a paper to the 2007 IEEE International Conference on Robotics and Automation (ICRA'07) or ICRA'08, you already have one. To find your own or someone else's PIN, update your personal information (strongly recommended before submitting) or register a new PIN if you never obtained one before; just follow the link in the log-in page of the submission site.

Please note that the resubmissions of papers originally handled through the current *T-RO* review system and were not accepted should go to the new system. Papers currently in review (submitted up to 31 December 2007) or resubmissions of conditionally accepted papers will instead continue to use the current *T-RO* system throughout the final decision, including submission of final material for publication. For these papers, author instructions can still be found at the old *T-RO* Web site http://www.dis.uniroma1.it/ieeetro.

For further inquiries, please contact the *T-RO* Editor-in-Chief Alessandro De Luca (E-mail: deluca@dis.uniroma1.it).

CALENDAR

2008

2–4 April CogSys 2008: International Conference on Cognitive Systems. Karlsruhe. http://www.cogsys2008.org

19–23 May ICRA 2008: IEEE International Conference on Robotics and Automation. Pasadena, California, USA. http://www.icra2008.org/

12 June IERA'08: IFR/IEEE Industry Forum on Innovation and Entrepreneurship in Robotics and Automation. Munich, Germany. http://www.ieee-ras.org/industrial, E-mail: klas@ieee.org

12–15 June 5th Annual International RoboGames. San Francisco CA. http://www.robogames.net/

20–23 June ICIA 2008: IEEE International Conference on Information Automation. ZhangJiaJie, Hunan, China. http://www.ieee-icia.info

25–28 June RSS 2008: Robotics: Science and Systems. Zurich Switzerland. http://www.roboticsconference.org

2–5 July AIM 2008: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Xi'an, China. http://www.aim2008.info

1–3 Aug. RO-MAN 2008: IEEE International Symposium on Robot and Human Interaction. Munich, Germany. http://www.ro-man2008.org

5–8 Aug. ICMA 2008: IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan. http://www.icma2008.org/

20–22 Aug. MFI 2008: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Seoul, Korea. http://www.mfi2008.org/

22 Aug. SICE 2008: Annual Conference. Chofu, Japan. http://www.sice.or.jp/sice2008/

23–26 Aug. IEEE-CASE 2008: 4th International Conference on Automation Science and Engineering. Washington, District of Columbia, USA. http://www.ieee-case.org

1–3 Sept. ICAL 2008: IEEE International Conference on Automation and Logistics. Qingdao, China. http://myweb.dal.ca/jgu/ical08/

16–17 Sept. IEEE-CYBER: IEEE International Conference on Automation, Control, and Intelligent Systems. Shenyang, China. (Contact: E-mail: tarn@wuauto.wustl.edu)

22–26 Sept. IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France. http://www.iros.org

19–22 Oct. BioRob 2008: International Conference on Biomedical Robotics and Biomechatronics. Scottsdale, Arizona, USA. http://www.biorob2008.org

28–31 Oct. SSRR 2008: IEEE International Workshop on Safety, Security, and Rescue Robotics. Sendai, Japan. http://www.rm.is.tohoku.ac.jp/ssrr2008/cfp.html

10–12 Nov. TePRA 2008: IEEE International Conference on Technologies for Practical Robot Applications. Woburn, Massachusetts, USA. http://www.ieeerobot-tepra.org/

17–19 Nov. DARS 2008: 9th International Symposium on Distributed Autonomous Robotics Systems. Tsukuba, Japan.

2–5 Dec. ICARV: 10th International Conference on Control, Automation, Robotics and Vision. Hanoi, Viet Nam. http://www.icarcv.org/2008/

4 Dec. SI International 2008: IEEE/SICE International Symposium on System Integration. Nagoya, Japan. http://www.rm.is.tohoku.ac.jp/SIInt08/

14–17 Dec. ROBIO'08: IEEE International Conference on Robotics and Biomimetics. Bangkok, Thailand. http://www.ee.cuhk.edu.hk/~qhmeng/robio/ROBIO2008-CFP.pdf

2009

11–13 Mar. HRI 2009: ACM/IEEE International Conference on Human-Robot Interaction. San Diego, California, USA. http://www.hri2009.org/

Digital Object Identifier 10.1109/M-RA.2008.917790

INDUSTRY/RESEARCH NEWS

(continued from page 120)

Ed Colgate Named First *IEEE T-Haptics* EIC

Prof. J. Edward Colgate of Northwestern University in Evanston, Illinois, was just named to be the founding editor-in-chief of the new *IEEE Transactions on Haptics*. Prof. Colgate has worked extensively in the areas of haptic interface and telemanipulation. He is known for his work on passivity and collaborative robots (cobots) and has recently developed a strong interest in variable friction haptic displays.

Paper submission will be done through Manuscript Central. See http://www.ieee-ras.org/toh/index.php for submissions information. The first issue is scheduled for late 2008. The *IEEE Transactions on Haptics* is jointly sponsored by the Robotics and Automation Society, the IEEE Computer Society, and the IEEE Consumer Electronics Society.

Call for Papers

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

Special Issue on Scientific Workflow Management and Applications

http://www.swinflow.org/si/t-ase.htm

Scientific workflow is a new special type of workflow that often underlies many large-scale complex e-science applications such as climate modeling, structural biology and chemistry, medical surgery or disaster recovery simulation. Compared with business workflows, scientific workflow has special features such as computation, data or transaction intensity, less human interaction, and a large number of activities. Some emerging computing infrastructures such as grid computing with powerful computing and resource sharing capabilities present the potential for accommodating those special features. Currently, many efforts are being on this new workflow area, and workshops such as WaGe07, WORKS07, WSES07, SWF07 and NSF funded workshop on challenges of scientific workflows have been or are being held to explore scientific workflow issues. Gradually, research results are published and several scientific workflow management systems such as SwinDeW-G, Kepler and Taverna are developed or evolved from existing systems. However, in general, research and development in scientific workflow management are still in their infancy with obscure knowledge of scientific workflow specific features and techniques. This special issue aims to systematically investigate and shape the special features, challenges and new techniques of scientific workflows as well as corresponding applications and underlying computing infrastructures. Original and unpublished high-quality research results are solicited to explore and boost the new area. The topics for contributions include, but are not limited to:

- Special features of scientific workflows and their hints on new techniques
- Scientific workflow modeling, execution and scheduling
- Formal representation, scientific workflow patterns
- Control flows and data flows in scientific workflows
- Web/grid services based scientific workflows
- Application programming interface and Graphical user interface
- Scientific workflow verification and validation
- Exception handling, Quality of Service, performance and security issues in scientific workflows
- Underlying infrastructures targeting scientific workflow support
- Real-world scientific workflow applications

Important Dates

March 31, 2008	Paper submission deadline
July 31, 2008	Completion of the first round review
November 30, 2008	Completion of the second round review
January 15, 2009	Final manuscript due
July 2009	Tentative publication date

Guest Editors

Jinjun Chen
Swinburne University of Technology
Email: jchen@ict.swin.edu.au

W.M.P. van der Aalst
Eindhoven University of Technology
Email: w.m.p.v.d.aalst@tue.nl

Bertram Ludäscher
University of California at Davis
Email: ludaesch@ucdavis.edu

Carole Goble
University of Manchester
Email: carole.goble@manchester.ac.uk

Rajkumar Buyya
The University of Melbourne
Email: raj@csse.unimelb.edu.au

Michael Rosemann
Queensland University of Technology
Email: m.rosemann@qut.edu.au

Yun Yang
Swinburne University of Technology
mail: yyang@ict.swin.edu.au

Paper Submission

All papers are to be submitted through the **Manuscript Central** for T-ASE at http://mc.manuscriptcentral.com/t-ase. Please select "Special Issue" under Manuscript Category of your submission. All manuscripts must be prepared according to the IEEE T-ASE publication guidelines http://www.engr.uconn.edu/~ieeetase/. Papers will be reviewed following the standard IEEE T-ASE review process.

Please address inquiries to jchen@ict.swin.edu.au.

Digital Object Identifier 10.1109/M-RA.2007.914991

By Kostas Kyriakopoulos

The European Robotics Research Network (EURON) started in 1999 as a Network of Excellence under the Future and Emerging Technologies Fifth Framework Program (FP-5) with the purpose of bringing together the best groups and resources in research, industry, and education in Europe and for demonstrating Europe's world class position in robotics. EURON II was launched in 2004 as an FP-6 Network of Excellence. Under the active and firm leadership of its early coordinator, Henrik Christensen, and his successor Herman Bruyninckx, EURON has grown to include approximately 210 members in 28 countries. Any group working in robotics in the European Union may apply to join, and new members do so regularly.

Scientists, industrialists, and educators in EURON work together toward the dream of the next generation of robots. It is a forum for members to meet and exchange news and results so that new ideas and collaborations are born, and old ideas are reviewed and extended. EURON was keen on making European research efforts focus toward more productive goals. EURON's educators worked toward developing and training the new, skilled robotics workforce through general science promotion activities and advanced summer schools. Active encouragement and exchange of ideas between the research and industrial communities was pursued, and continuous efforts to cooperate with the European Robotics Platform (EUROP) were made. EURON sponsored a prize for the successful transfer of good ideas from the research world to the robotics industry. As a result, through EURON, the world sees the scope and quality of European robotics.

Digital Object Identifier 10.1109/M-RA.2008.915556

EURON II is finishing as a funded Network of Excellence by the end of April 2008. Under its current scheme, its final annual meeting will be taking place 26–28 March 2008, in Prague, Czech Republic, collocated with the European Robotics Symposium (EUROS) 2008 (http://www.action-m.com/euros2008/). However, most of the current members have expressed their interest in letting EURON live on, in the form of a true community-driven organization. The concrete form and mission of this new EURON will be the result of an open discussion within the community, but, most likely, the major success stories of the past will be continued in one way or another: the Ph.D. and tech transfer awards, the summer schools, and the electronic dissemination of robotics information (http://www.euron.org), including job offers and drafts of research papers.

The "EURON Report," in *IEEE Robotics and Automation Magazine* (*RAM*), as part of EURON's publicity activities, has constantly been reporting on European robotics research activities with the contribution of Prof. Bruno Siciliano, who is now acting as the new president of the IEEE Robotics and Automation Society (good luck, Bruno!), and Prof. Kostas Kyriakopoulos. The 17 EURON columns reported between 2003 and 2007 were on the robotics summer schools in Europe, EURON meetings, EUROS, the EURON/European Robotics Foundation Technology Transfer Award, the Joint Program of Research, research coordination activities of EURON, the European Commission—Future and Emerging Technologies' Beyond Robotics initiative-funded projects, the Georges Giralt Ph.D. Award in Robotics in Europe, and the European Commission funding activities.

This column introduces the new format for regional perspectives in robotics and automation, in which leaders in robotics and automation from around the world will report on new and ongoing initiatives in our technology. I would like to thank EURON coordinators Henrik Christensen and Herman Bruyninckx, who supported this effort, my coauthor Bruno Siciliano, for sharing the load, and all those EURON members who provided me with material to support the column. I will keep on reporting from Europe under the new scheme.

In the future, parties interested in promoting activities via the "Regional" column can contact Kostas Kyriakopoulos at http://users.ntua.gr/kkyria or E-mail: kkyria@central.ntua.gr.

IEEE BioRob 2008 **http://www.ieee-biorob.org** **biorob08@www.fulton.asu.edu**

IEEE BioRob 2008

IEEE International Conference on Biomedical Robotics and Biomechatronics

October 19-22, 2008
Scottsdale, Arizona, U.S.A.

Sponsored by IEEE Robotics and Automation Society
& IEEE Engineering, Medicine, and Biology Society

Call for Papers

GENERAL CO-CHAIR
Deirdre Meldrum
Arizona State University, USA

GENERAL CO-CHAIR
Oussama Khatib
Stanford University, USA

PROGRAM CHAIR
Jaydev Desai
University of Maryland, USA

CONFERENCE ADVISORY BOARD

George Bekey
U. of Southern California, USA
Alain Berthoz
Collège de France, France
Emilio Bizzi
MIT, USA
Paolo Dario
Scuola Sup. Sant'Anna, Italy
Blake Hannaford
U. of Washington, USA
Koji Ikuta
Nagoya U., Japan
Rolf Johansson
Lund U., Sweden
Yoshihoko Nakamura
U. of Tokyo, Japan
Atsuo Takanishi
Waseda U., Japan
Russ Taylor
Johns Hopkins U., USA

The second IEEE RAS / EMBS International Conference on Biomedical Robotics and Biomechatronics – **BioRob 2008** – is a joint effort of the two IEEE Societies of Robotics and Automation – **RAS** – and Engineering, Medicine, and Biology – **EMBS**. BioRob covers both theoretical and experimental challenges posed by the application of robotics and mechatronics in medicine and biology. The primary focus of Biorobotics is to analyze biological systems from a "biomechatronic" point of view, trying to understand the scientific and engineering principles underlying their extraordinary performance. This profound understanding of how biological systems work, behave and interact can be used for two main objectives: to guide the design and fabrication of novel, high performance bio-inspired machines and systems, for many different applications; and to develop novel nano-, micro-, macro- devices that can act upon, substitute parts of, and assist human beings in diagnosis, surgery, prosthetics, rehabilitation and personal assistance.

BioRob is a highly interdisciplinary conference that brings together scientists and engineers from different backgrounds to share and learn about research activities in this fast growing field. The IEEE RAS and EMBS share this vision and thus jointly sponsor this conference.

The technical program of IEEE BioRob2008 will consist of invited talks, special sessions, posters, and paper presentations. Submitted papers must describe original work, in the form of modeling abstractions, algorithms, theoretical analysis, case studies, and experiments. Papers can cover areas of biorobotics including:

- Biologically-inspired systems
- Biomechatronic systems
- Biorobotics
- Exoskeletons and augmenting devices
- Human-machine interaction
- Locomotion and manipulation in robots and biological systems
- Micro/nano technologies in medicine and biology
- Modeling interactions between robots and biological systems
- Neuro-robotics
- Protheses
- Rehabilitiation and assistive robotics
- Surgery and diagnosis

Paper Submissions: Author(s) should submit full papers electronically in double column IEEE-compliant PDF format. All papers will be peer-reviewed. Accepted papers will have a choice for oral and/or poster presentations, and will be published in CD-ROM. Posters will be displayed throughout the conference and Poster Sessions will be scheduled for author/audience interaction. Six pages are allowed per paper, and detailed instructions for paper preparation and submission will be available on the conference web site: http://www.ieee-biorob.org.

Venue: The venue for the conference is FireSky Resort (www.fireskyresort.com). This luxurious hotel is a five minute walk to historic Old Town Scottsdale. Camelback Mountain, Taliesan West Frank Lloyd Wright School of Architecture, Desert Botanical Gardens, and more are close by. FireSky is ~9 miles from Arizona State University and ~13 miles from Phoenix Sky Harbor International Airport.

IMPORTANT DATES!

February 29, 2008	March 31, 2008	June 30, 2008	July 31, 2008
Proposal for special sessions	**Submission of full papers**	**Paper acceptance notification**	**Final paper submission**

CALENDAR

2008

2–4 April CogSys 2008: International Conference on Cognitive Systems. Karlsruhe. http://www.cogsys2008.org

19–23 May ICRA 2008: IEEE International Conference on Robotics and Automation. Pasadena, California, USA. http://www.icra2008.org/

12 June IERA'08: IFR/IEEE Industry Forum on Innovation and Entrepreneurship in Robotics and Automation. Munich, Germany. http://www.ieee-ras.org/industrial, E-mail: klas@ieee.org

12–15 June 5th Annual International RoboGames. San Francisco CA. http://www.robogames.net/

20–23 June ICIA 2008: IEEE International Conference on Information Automation. ZhangJiaJie, Hunan, China. http://www.ieee-icia.info

25–28 June RSS 2008: Robotics: Science and Systems. Zurich Switzerland. http://www.roboticsconference.org

2–5 July AIM 2008: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Xi'an, China. http://www.aim2008.info

1–3 Aug. RO-MAN 2008: IEEE International Symposium on Robot and Human Interaction. Munich, Germany. http://www.ro-man2008.org

5–8 Aug. ICMA 2008: IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan. http://www.icma2008.org/

20–22 Aug. MFI 2008: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Seoul, Korea. http://www.mfi2008.org/

22 Aug. SICE 2008: Annual Conference. Chofu, Japan. http://www.sice.or.jp/sice2008/

23–26 Aug. IEEE-CASE 2008: 4th International Conference on Automation Science and Engineering. Washington, District of Columbia, USA. http://www.ieee-case.org

1–3 Sept. ICAL 2008: IEEE International Conference on Automation and Logistics. Qingdao, China. http://myweb.dal.ca/jgu/ical08/

16–17 Sept. IEEE-CYBER: IEEE International Conference on Automation, Control, and Intelligent Systems. Shenyang, China. (Contact: E-mail: tarn@wuauto.wustl.edu)

22–26 Sept. IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France. http://www.iros.org

19–22 Oct. BioRob 2008: International Conference on Biomedical Robotics and Biomechatronics. Scottsdale, Arizona, USA. http://www.biorob2008.org

28–31 Oct. SSRR 2008: IEEE International Workshop on Safety, Security, and Rescue Robotics. Sendai, Japan. http://www.rm.is.tohoku.ac.jp/ssrr2008/cfp.html

10–12 Nov. TePRA 2008: IEEE International Conference on Technologies for Practical Robot Applications. Woburn, Massachusetts, USA. http://www.ieeerobot-tepra.org/

17–19 Nov. DARS 2008: 9th International Symposium on Distributed Autonomous Robotics Systems. Tsukuba, Japan.

2–5 Dec. ICARV: 10th International Conference on Control, Automation, Robotics and Vision. Hanoi, Viet Nam. http://www.icarcv.org/2008/

4 Dec. SI International 2008: IEEE/SICE International Symposium on System Integration. Nagoya, Japan. http://www.rm.is.tohoku.ac.jp/SIInt08/

14–17 Dec. ROBIO'08: IEEE International Conference on Robotics and Biomimetics. Bangkok, Thailand. http://www.ee.cuhk.edu.hk/~qhmeng/robio/ROBIO2008-CFP.pdf

2009

11–13 Mar. HRI 2009: ACM/IEEE International Conference on Human-Robot Interaction. San Diego, California, USA. http://www.hri2009.org/

Digital Object Identifier 10.1109/M-RA.2008.917790

INDUSTRY/RESEARCH NEWS

(continued from page 120)

Ed Colgate Named First *IEEE T-Haptics* EIC

Prof. J. Edward Colgate of Northwestern University in Evanston, Illinois, was just named to be the founding editor-in-chief of the new *IEEE Transactions on Haptics*. Prof. Colgate has worked extensively in the areas of haptic interface and telemanipulation. He is known for his work on passivity and collaborative robots (cobots) and has recently developed a strong interest in variable friction haptic displays.

Paper submission will be done through Manuscript Central. See http://www.ieee-ras.org/toh/index.php for submissions information. The first issue is scheduled for late 2008. The *IEEE Transactions on Haptics* is jointly sponsored by the Robotics and Automation Society, the IEEE Computer Society, and the IEEE Consumer Electronics Society.

ICRA 2008

2008 IEEE International Conference on Robotics and Automation

May 19-23, 2008
Pasadena, California

Honorary General Chair:
George Bekey (USC)
bekey@usc.edu

General Chair:
Maja J Mataric' (USC)
mataric@usc.edu

General Co-Chair:
Paul Schenker (JPL)
schenker@helios.jpl.nasa.gov

Program Chairs:
Stefan Schaal (USC)
sschaal@usc.edu
Gaurav S Sukhatme (USC)
gaurav@usc.edu

Local Organization Chair:
Edward Tunstel (JPL)
tunstel@helios.jpl.nasa.gov

Finances Chair:
Ayanna Howard (GaTech)
ayanna.howard@ece.gatech.edu

Finances Advisor:
Yi Guo (Stevens)
yguo1@stevens.edu

Publications Chair:
Jochen Triesch (FIAS, Goethe)
triesch@fias.uni-frankfurt.de

Publications Advisor:
Max Meng (CUHK)
max@ee.cuhk.edu.hk

Exhibits and Publicity Chairs:
Ashley Stroupe (JPL)
ashley.w.stroupe@jpl.nasa.gov
Paolo Pirjanian (Evolution Robotics)
paolo@evolution.com
Dan Kara (Robotics Trends)
dk@roboticstrends.com

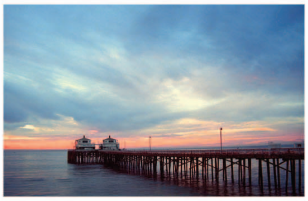
Workshop and Tutorial Chair:
Monica Nicolescu (U Nevada-Reno)
monica@cs.unr.edu

Awards Chair:
Nikolaos Papanikolopoulos (UMN)
npapas@cs.umn.edu

Video Proceedings Chairs:
Nancy Pollard (CMU)
nsp@cs.cmu.edu
Adriana Tapus (USC)
tapus@usc.edu

Robotic Challenge Chairs:
Bill Smart (WUSTL)
wds@cse.wustl.edu
Paul Rybski (CMU)
prybski@cs.cmu.edu

Region Chairs:
Europe: Raja Chatila (LAAS-CNRS)
raja.chatila@lass.fr
Asia:Sukhan Lee (Sungkyunkwan)
lsh@ece.skku.ac.kr
Oceania:Peter Corke (CSIRO)
peter.corke@csiro.au



Call for Participation

www.icra2008.org

The 2008 IEEE International Conference on Robotics and Automation **(ICRA 2008)** will be held in Pasadena, California, USA, during May 19 - 23, 2008. The theme of the conference is Human-Centered Robotics, the movement toward robotics technology that aids in the course of human everyday life.

The conference will feature technical presentations of papers, invited sessions, special video sessions, and workshops and tutorials. A diverse array of exhibits (both industrial and academic) is also planned. The conference will feature plenary talks by **Prof. Andrew Blake**, **Prof. Mitsuo Kawato**, and **Prof. Naomi Leonard**.

A new event is being introduced at **ICRA 2008**: The ICRA Robot Challenge.The Challenge will consist of a number of specific robot events with the overall theme of "**space robotics**".

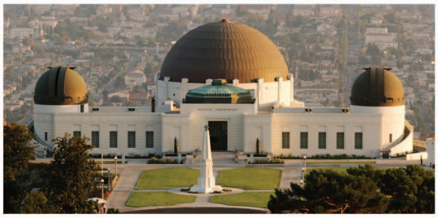
ICRA 2008 will be held at the **Pasadena Conference Center**. The recently renovated Center lies within walking distance of four major hotels and Old Pasadena's 200 shops and restaurants.

Registration for ICRA 2008 will open in January 2008. To register and make hotel bookings please visit the ICRA 2008 website.

IMPORTANT DATES

Submissions of all contributions:	September 14, 2007
Notification of acceptance:	January 7, 2008
All final contributions due:	February 4, 2008

www.icra2008.org



Digital Object Identifier 10.1109/M-RA.2007.914992

International Federation of Robotics

Call for Applications

Robotics & Automation Society

IEEE/IFR Invention & Entrepreneurship Award

for Outstanding Achievements
in Commercializing Innovative Robot and Automation Technology

in conjunction with
IEEE/IFR Joint Forum on Innovation and Entrepreneurship in Robotics and Automation
June 11, 2008 – 12:30-16:30
International Congress Centre Munich, Germany

co-located with
Robotik 2008 – Germany's largest robotics conference – June 11-12, 2008
International Congress Centre Munich, Germany
and
Automatica 2008 – International Trade Fair for Automation – Assembly, Robotics, Vision – June 10-13, 2008
New Munich Trade Fair Centre, Germany

Sponsoring Organizations:
International Federation of Robotics (http://www.ifr.org)
IEEE Robotics and Automation Society (http://www.ieee-ras.org)

Deadline for applications: April 4, 2008

http://www.ieee-ras.org/industrial

Honorary Chair

Sukhan Lee
(Sungkyunkwan University)

General Chairs

Stefan Müller
(KUKA, IFR President)
Alex Zelinsky
(CSIRO, IEEE RAS VP Industrial Activities)

Organizing Committee

Program Chair:
Rainer Bischoff (KUKA)
Awards Chair:
Klas Nilsson (Lund Univ.)
Finance Chair:
Erwin Prassler (FH BRS)
Local Arrangements:
Helga Rosenzweig (VDI)

Awards Committee

IEEE:
Klas Nilsson (Lund Univ.)
Erwin Prassler (FH BRS)
Alex Zelinsky (CSIRO)
IFR:
Rolf-Dieter Schraft (IPA)
Tokuo Iikura (JARA)
Ake Lindqvist (ABB, RIA)

Announcement

The purpose of this award is to highlight and honor the achievements of the inventors with value creating ideas and entrepreneurs who propel those ideas into world-class products. This is a key element to the continuing success of robotics and automation today. Active infusion of innovation and entrepreneurship into technological advancement is regarded critical at this juncture to strengthen a healthy balance between research and practice as well as a healthy growth of industrial and commercial sectors in robotics and automation.

In a joint event the IEEE Robotics and Automation Society and the International Federation of Robotics will therefore recognize and honor outstanding achievements of entrepreneurs in the commercialization of innovative robotic and automation technology.

These achievements will be recognized in a specially organized IEEE/IFR Joint Forum on Innovation and Entrepreneurship in Robotics and Automation, which is being held in conjunction with the conference "Robotik 2008", Germany's largest bi-annual robotics conference. The selected finalists will have the opportunity to present their story of the genesis of a successful innovative product in robotics and automation from its very inception to the final state of commercialization in a series of plenary lectures. The ultimate winner will be chosen by an evaluation board, consisting of distinguished individuals from industry and academia. A prestigious plaque will be awarded to each finalist and a US$ 2,000 prize will be awarded to the winner.

Applications should describe the original work that has been translated into a commercial success. The application must include statements regarding:

- description of the innovation/product/application
- stages of the product genesis
- novelty/uniqueness of the product
- market analysis, economic viability and pathway for commercialization
- sustained competitive advantage
- current and future impact on and relevance to industry

Applications should not exceed a maximum length of 5 pages. Product descriptions and public relation material will not be accepted as an application. References and links to online material are permitted.

Submission of Applications

Please send as PDF document (< 6MB) no later than April 4 (any time zone) to klas@ieee.org

Schedule

April 4, 2008	Submission of applications
April 28, 2008	Evaluation of applications and selection and notification of finalists
June 11, 2008	Award Ceremony with plenary lectures of the finalists at the IEEE/IFR Joint Forum on Innovation and Entrepreneurship in Robotics and Automation

Contact

Klas Nilsson
Dept. of Computer Science, Lund University, Box 118, 221 00, Lund, Sweden
phone: +46-46-2224304 / fax: +46-46-131021 / skype: klas_nilsson
http://www.cs.lth.se/~klas e-mail: klas@ieee.org

Digital Object Identifier 10.1109/MRA.2008.918686

MONASH University

IEEE ICIT'09 AUSTRALIA

International Conference on Industrial Technology

10-13 February 2009
Monash University – Gippsland

Sponsor: IEEE Industrial Electronics Society

Call for Papers

General Chair
Yousef Ibrahim
Monash University
Yousef.Ibrahim@eng.monash.edu.au

Technical Co-chairs
Karel Jezernik
University of Maribor, Slovenia
karel.jezernik@uni-mb.si

Chandan Chakraborty
Indian Institute of Technology, India
chandan@ee.iitkgp.ernet.in

Special Sessions Co-chairs
Luis Gomes
Universidade Nova de Lisboa, Portugal
lugo@uninova.pt

Maria Valla
Universidad Nacional de La Plata, Argentina
m.i.valla@ieee.org

International Program Committee
Kouhei Ohnishi, Japan
Leopoldo G. Franquelo, Spain
John Hung, USA
David Doral. Australia
Koorosh Khodabandehloo, UK
Paul Ford, Australia
Ekkehard Bolte, Germany
Bimal K. Bose, USA
Stefan Brock, Poland
Xinghuo Yu, Australia
Carlo Cecati, Italy
Tom Byerley, USA
Andy Russell, Australia
Gerhard P. Hancke, South Africa
Takamasa Hori, Japan
Charles Einolf, USA
Yoichi Hori, Japan
Bijan Shirinzadeh, Australia
Jon Billingsley, Australia
Bogdan (Dan) Wilamowski, USA
Werner Schmidt, Australia
Peter N. Brett, UK
Ju-Jang Lee, Korea
Issa Batarseh, USA
James Trevelyan, Australia
Imre J. Rudas, Hungary
Asif Sabanovic, Turkey
Frantisek Solc, Czech Republic
Istvan Nagy, Hungary
Javier Uceda, Spain
Samir M. El-Ghazaly, USA
Andre Vandenput, The Netherlands
Saman Halgamuge, Australia
Martin Wollschlaeger, Germany

International Publicity Co-chairs
Antonio Estepa, Spain
Manukid Parnichkun, Thailand
Tx Ma, China

Finance Co-chairs
Terry Martin (USA)
Yousef Ibrahim (Australia)

Local Arrangements Committee
Linda Brock
Tim Wills
Maree McPherson
Dushmanta Dutta
Alan Scarlett
Ray Beebe
Lisa wood
Abdullah Mamoun
Gerald May

GRACe GIPPSLAND REGIONAL AUTOMATION CENTRE

Invitation to Participate in an Intellectually Stimulating Conference & Experience the True Australian Culture

Main Theme:

FUTURE TECHNOLOGY
in Service of
Regional Industry

The purpose of the conference is to provide a forum for presentation and discussion of emerging industrial technology. Also, it serves as a conduit for channelling advanced technology to the regional industry. It brings together researchers, from industry and academia, active in industrial technological fields to discuss current developments and future perspectives.
Therefore, the topics of this conference include, but are **not limited to**:

Mechatronics and Robotics
(Chairs: Chris Cook, Australia; William Hamel, USA; Hideki Hashimoto, Japan)

Mechatronics applications in rural and food industry, micro/nanomechatronics, flexible structures; manufacturing technology; intelligent inspection, diagnosis and classification systems; robotic automation; special service robots; modular robots; spherical robots; flexible-joint robots; aerial vehicles; grippers; Intelligent sensors and actuators, multi-sensor fusion, microsensors and microactuators, instrumentation electronics, industrial antennas and electromagnetic devices, MEMS and system integration.

Factory Automation & Industrial Informatics
(Chairs: Elizabeth Chang, Australia; Juan Jose Rodriguez, Spain)

Industrial vision, motion control, autonomous mobile robots, electrical vehicles, intelligent transportation, factory communications, flexible manufacturing systems, industrial automation, process automation, CAD/CAM/CAT/CIM and LANs, industrial application of internet technologies, multimedia, wireless communications, data privacy and security, authentification, authorization and federation.

Computer and Control Systems
(Chairs: Farhat Fnaiech, Tunisia; Sandrine Moreau, France)

Advanced control and measurement, computer and microprocessor-based control, signal processing, estimation and identification techniques, application specific IC's, automotive electronics, chaos control, non-linear control systems, industrial applications of neural networks, fuzzy algorithms, evolutionary computing.

Power Generation and Distribution
(Chairs: Seddik Bacha, France; Gerald Ledwich, Australia)

Power system analysis and control, HVDC and FACTS, Power market, Power distribution network and automation, Power protection, SCADA, EMS, WAMS and substation automation, Green power.

Power Electronics and Energy Conversion
(Chairs: Marcian Cirstea, UK; Kamal Haddad, Canada; Marco Liserre, Italy)

Power electronics devices and systems, high frequency power converters, digital control of power electronics, energy systems, electrical machines and drives, static VAR and harmonic compensations, power management, power quality, electric vehicle, analytical and simulation methods, and e-learning in education.

Keynote Speakers: Distinguished speakers will be invited to give plenary talk on various technical topics.

Special Sessions: The conference will include Special Sessions on highly specialised topic areas, within the scope of the conference and its theme. Special Sessions are organised at the initiative of one or more individuals, who must adhere to specific procedures published at the conference website.

Working Language: English

Pre- and post-conference tours: Assistance with organizing pre- and post-conference touring of Australia's best kept secret is provided to all conference delegates and their partners by Latrobe visitor centre: http://www.visitlatrobevalley.com

Submission Schedule:

Deadline for submission of full paper:	15 July 2008
Notification of acceptance with electronic publication instruction:	15 October 2008
Deadline for final manuscript:	1 December 2008

Prospective authors are invited to submit full papers (double column, six pages maximum) in English using the online paper submission system located at conference website http://www.ieee-icit09.org/

Selected Papers will be published in the Fall 2009 Issue of the IEEE *Transaction on Industrial Electronics.*

State Government Victoria AUSTRALIA