An Overview of Emerging Results in Networked Multi-Vehicle Systems

Anouck Renée Girard, João Borges de Sousa and J. Karl Hedrick

Abstract— Autonomous vehicle systems have been the topic of much research due to their ability to perform dangerous, repetitive and automated tasks in remote or hazardous environments [12]. The potential for multi-vehicle systems cooperating together to accomplish given tasks is starting to draw together researchers from several fields, including robotics, control systems, and computer science.

Multiple vehicles can be more effective than a single one, for example in information gathering tasks. By spreading out over the terrain to be searched, a cluster of autonomous helicopters, for example, can locate a target quite rapidly, or a group of coordinated autonomous underwater vehicles can search a coastal area for mines. In other cases, the coordinated operation of multiple vehicles can provide new capabilities. This is the case, for example, of the PATH strategy of platooning several vehicles as they travel along the highway, which may yield up to a four-fold increase in transportation capacity while enhancing safety. Another example is the Mobile Offshore Base, where semi-submersible modules are aligned to form a military base and runway at sea. The unprecedented length of the at-sea runway (up to a mile long) warrants the use of several modules.

In each of these cases, there is a need for inter-vehicle communications so that each vehicle can know the status of the operation, the position of its counterparts, and whether the specific mission goals have changed. Thus the control and communication problems become inexorably tied. However, few results are available to analyze performance and stability of a closed loop system where some of the loops are closed by communicated variables.

Using the above examples as a motivation, this paper examines emerging results in networked multi-vehicle systems. Recent work has taken many different approaches, such as hybrid systems, distributed control, differential games, control architectures, and artificial intelligence. The focus of this paper is on the control systems perspective. We attempt to present some current issues common to networked multi-vehicle systems, and to show how they have been solved to date in the perspective of the case studies.

Index Terms — networked multi-vehicle systems, communications and control, hybrid systems, control architectures, multi-agent systems.

I. INTRODUCTION

There are many applications where “coordinated” control of multiple vehicles or systems is desirable, e.g. automotive vehicles in various stages of automation ranging from automated highway systems to coordinated adaptive cruise control systems, to “platooning” of passenger and military vehicles.

Also, there is a trend in the military towards autonomous air and underwater vehicles; these vehicles perform coordinated missions and require some communicated information among them. Some of these applications include coordinated ocean platform control for the Mobile Offshore Base (MOB), coordinated operation of several Autonomous Underwater Vehicles (AUVs), and/or of Unmanned Combat Air Vehicles (UCAVs).

Currently, there exists no unified theory to aid in the design of networked multi-vehicle systems. Analytical techniques to deal with crucial issues such as coordination mechanisms, maneuver design, control strategies and performance, communication system, overall architecture design and implementation are not readily available to the control or communication systems designer. It is currently not possible to specify performance and stability requirements for a closed loop system where some of the loops are closed by communicated variables. The goal of this paper is to discuss some emerging results in the area of networked multi-vehicle systems. It is clear that the number of these types of systems will be increasing exponentially as the wireless revolution continues and new control and communications techniques are developed.

This paper is organized as follows: in section II, we describe the problem domain and the assumptions that have been made. In section III, we cover some case studies that are used as a motivation. Section IV contains an overview of the current analytical practice in networked multi-vehicle systems. Finally, section V deals with the lessons learned, some concluding remarks, and the open problems in networked multi-vehicle systems.

II. PROBLEM DOMAIN AND ASSUMPTIONS

A major difficulty in the study of networked multi-vehicle systems is the size of the space of possible configurations and designs problems. Some classification procedures exist in the field of multi-agent systems; we refer the interested reader to [1]. Our goal in this paper is not to provide a general classification procedure for multi-vehicle systems. We simply wish to emphasize some characteristics of the types of problems that have been tackled.

In this paper we emphasize the control of multi-vehicle systems, such as the modules forming a MOB, the cars forming a platoon, the helicopters forming a cluster etc… However, there is nothing in the framework that restricts it to transportation applications. For example, we may also

The authors are with the University of California at Berkeley.

consider the coordinated operation of ground robots, or of clouds of satellites. We further assume that the vehicles can have nonlinear dynamics, actuators or sensors.

The vehicles evolve in a given environment. We assume that this environment is not accessible (sensors are not perfect, do not cover the whole world), non-deterministic (vehicles may break, there is uncertainty) and dynamic. There are many types of vehicle constraints in networked multi-vehicle problems. For the sake of simplicity, in this paper we have assumed four main types of vehicle constraints: dynamics, costs, computational requirements, and information requirements. Environmental constraints may also arise, such as in obstacle avoidance problems.

We are interested in vehicle formations, by contrast to problems in which all vehicles are independent. The vehicles are organized in a particular geometrical configuration. Typical problems that arise are how to maintain the geometrical configuration, and/or track desired changes of the configuration with time. Associated problems involve how to join and form the configuration, and how to separate from it. These different problems can be seen as modes of operation, and the switching between modes suggests setting the problem in the hybrid system formalism [16].

The dynamic nature of the problem stems from the existence of multiple vehicles whose roles, relative positions, and dependencies change during operations. These changes in the geometry of the formation are sometimes termed dynamic reconfiguration, meaning vehicles can be dynamically added or removed from the geometrical formation. To meet these complex system description requirements, the control architecture is generally modeled as a dynamic network of hybrid systems.

In the problems that have been tackled, it is usually assumed that the vehicles are homogeneous and have similar properties. By contrast, heterogeneous vehicle formations could be made of several types of robots or helicopters, or by a string of cars of different makes and models.

III. CASE STUDIES

In this section we explore some case studies upon which we shall base our subsequent analysis. These case studies serve as a catalyst and motivation for our research.

A. The Mobile Offshore Base

The Mobile Offshore Base (MOB) is a very large floating ocean structure meant to provide the same capabilities as an on-land army base. It must accommodate the landing and take-off of C-17 conventional aircraft, host 3000 troops, carry 10 million gallons of fuel and provide 3 million square feet of internal configurable storage.

The modules forming the MOB must be able to perform long-term station keeping at sea, in the presence of waves, winds and currents.

In order to achieve support air and sea operations, the MOB is required to assemble at sea, remain aligned and assembled to allow for landing of aircraft and cargo transfer from ships, align in the wind to facilitate the landing of aircraft, and disassemble if the environmental conditions become to severe or in case of emergency.

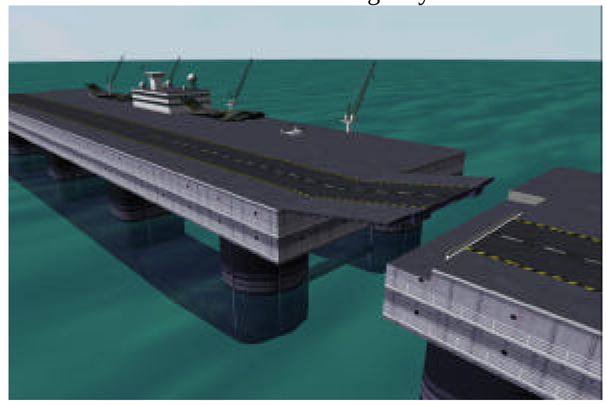

Figure 1. A Possible MOB Configuration [28].

The Mobile Offshore Base can be viewed as a string of homogeneous modules that have to be kept aligned. All modules are homogeneous, that is they are assumed to have the same dynamics and properties, except in specialized failure handling modes.

The most significant requirement is that the modules have very precise relative position control with respect to each other. The string, however, is allowed to drift in terms of its global position. This allows for a reduction in the power consumption (cost) in lower sea states, and focuses all the control effort on maintaining the relative alignment in high sea states.

Because of the relatively slow dynamics and large size of the modules, it has been assumed that communication constraints between modules will be met at all times, and that the communication problem can be decoupled from the control problem.

The (decentralized) coordinated control problem for the MOB was separated into two sub-problems: reference generation and coordination. Reference generation deals with selecting a string control strategy, maximizing the string alignment, and minimizing the global fuel consumption. Coordination deals with the implementation of a string control strategy, and the stability and control of neighboring modules with respect to one another. A meticulous description of the architecture can be found in [3], and the control algorithms are presented in [6].

B. Automotive Applications

The PATH Program at UC-Berkeley has proposed a strategy for Automated Highway Systems (AHS) that yields up to a four-fold increase in transportation capacity while enhancing safety [2,7]. The architecture proposes a strategy of platooning several vehicles as they travel along the highway. The separation of vehicles within a platoon is small (2m) while separation of platoons from each other is large (60m). The movement of vehicles is realized through simple maneuvers---join, split, lane change, entry and exit---that are coordinated.

Early in the PATH platooning [19] work it was noticed that strings of automatically controlled vehicles exhibited "string instabilities", i.e., disturbances in the front of the platoon were amplified as they were propagated upstream.

Ref. [18, 19, 20] showed through linear transfer function analysis that these instabilities could be eliminated by the introduction of a common reference trajectory for all of the vehicles. It was shown in Ref. [20] that if all of the vehicles in the platoon had knowledge of the lead vehicle's absolute velocity then "weak string stability" could be achieved, i.e., no disturbance would ever be amplified as it traveled upstream in the platoon. It was also shown in [20] that if all of the vehicles in the platoon had knowledge of the relative position error between themselves and the lead vehicle, then "strong string stability" could be achieved, i.e., all downstream disturbances could be geometrically attenuated as they traveled upstream in the platoon. The lead vehicle information needs to be communicated to all of the vehicles via a wireless communication link.

C. Unmanned Combat Air Vehicles (UCAV)

The goal of the BEAR - Berkeley Aerobot [17] project is to develop a fleet of autonomous aerial robots that are capable of performing navigation functions, flying autonomously, and recognizing and locating target objects. Several different model helicopters are used as airframes for the robots, which have various flight modes such as hovering, forward and sideward flight, vertical climb/descent, take-off and landing, etc.

One of the project goals is to implement a pursuit-evasion game, with a cluster of (possibly heterogeneous) autonomous helicopters searching for and locating autonomous ground robots.

When doing cooperative work, the physical configuration of the helicopters with respect to one another is dictated by several constraints: 1) Communications, 2) Navigation, 3) Sensing capabilities, 4) Physical constraints of the terrain, 5) Adversaries. The helicopters are supposed to operate in a potentially hostile environment where the adversary can be broadly characterized in terms of vehicles and capabilities. The nature of the operations imposes severe constraints on communications

The software is implemented in a four-layer architecture; at the lowest level, the regulation layer handles stability and control functions. It is followed by a trajectory planning layer, and a tactical planner. Finally, at the top layer, the strategic planner handles discrete events and logic, and deals with high-level mission goals.

D. Autonomous Underwater Vehicles

Autonomous Underwater Vehicles (AUV) are small, unmanned, untethered submersibles. They are intended to provide researchers with a simple, long-range, low-cost rapid response capability to collect pertinent environmental data. There are numerous applications for AUV, such as oceanographic surveys [8], operations in hazardous environments [23], underwater structure inspection [24], and military applications [5].

Admittedly, AUV present many difficult challenges. However, recent advances in navigation, power and communication systems offer the appropriate technology to use AUV for data gathering in the coastal and open ocean very feasible. Acoustic communications (by underwater modem) now allow for cooperation and coordination between several AUV, hence permitting cooperative adaptive sampling.

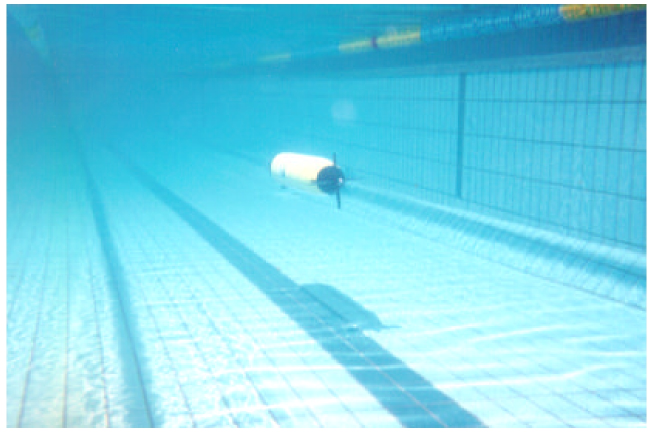

Figure 2. The AUV ISURUS [24].

Here, and in this context, we consider the problem of adaptive sampling of oceanographic phenomena with heterogeneous Autonomous Underwater Vehicles with limited capabilities. The fundamental idea underlying adaptive sampling is to increase the survey efficiency by concentrating measurements in regions of interest. Thus, to map an oceanfront, one might first run a very coarse survey to localize the front, then concentrate operations in the front vicinity. Substantial savings can be realized both in terms of expended energy and in terms of time required to characterize the front. Several AUV must also be able to coordinate to accelerate the process.

A recent paradigm for ocean presence is the Autonomous Ocean Sampling Network (AOSN) [10]. The AOSN is based on small, low cost vehicles supported by a sophisticated communication and control infrastructure. The diversity of vehicles, sensor packages, communication links, control software, and data processing/visualizations tools creates tremendous opportunities, but only if elements of the system are compatible with each other. This provides the motivation to develop an architecture and infrastructure which a) provides science users with a straightforward yet flexible set of tools for interacting with deployed AOSN assets, b) substantially eases the introduction of new capabilities to AOSN such as new vehicles or new software tools, c) ensures compatibility of elements of the AOSN "tool kit", and d) provides a re-configurable "on-the-fly" capability to support real-time operations.

Different teams may use different architectures and control strategies at the vehicle level [5, 8, 10, 23-24]. However each vehicle has interfaces to the common coordination and control architecture. Communications are of paramount importance, and yet must be kept at a minimum, since underwater communications typically offer much less bandwidth than their on-land counterparts. More information about Autonomous Underwater vehicles, including an extensive literature review, can be found in [8].

IV. CURRENT ISSUES IN NETWORKED MULTI-VEHICLE SYSTEMS

A. Similarities/Differences between Case Studies

For our purposes, the most important similarity of the case studies presented in section II is that they all involve experimental validation of systems and concepts. This requires early prototyping, simulation and testing, and multi-disciplinary teams. Usually, a "living" structure (or architecture) is developed early in the project, which can accommodate change and reconfiguration, and from which lessons can (and must) be learned. The design of this architecture is perhaps the most important factor of success in multi-vehicle systems. Hardware/software co-design must occur early, and good software architecture design from the start of the project allows for maximum code re-use and compatibility of software at all levels.

In networked multi-vehicle systems, it is essential to have a consistent and uniform architecture, which accommodates integrated simulation and real-time deployment. Initially the focus of the architecture is on the lower level of controls, and it gradually increases in scope to include more complexity and higher levels of control and communications.

B. Coordination and Control Strategies

There are many different control strategies that have been considered for the coordinated control of multiple vehicle systems. It was discussed in Section III B that string instabilities could occur if only relative position error information is used in the vehicle control laws and that some common information such as the position and velocity of a "leader" is required for string stability. Ref [22] extended the concept of string stability to 3-D configurations and used the term "mesh stability" to denote the property of disturbance attenuation in multi-dimensions. Ref [22] concentrated on minimizing the communication requirements to achieve mesh stability and analyzed systems with "look-ahead" sensor information that could be communicated or sensed directly. It should be noted that the "leader" in the mesh or string could be a "virtual" leader, e.g., a desired trajectory that would take a cluster of vehicles and move them to a given location. In this case each vehicle would have to know its position with respect to the virtual trajectory.

C. Formalization of the Control Architectures

By control architecture we mean a specific way of organizing the motion control and navigation functions performed by the coordinated vehicles. It is convenient to organize the functions into hierarchical layers. This way, a complex design problem is partitioned into a number of more manageable sub-problems that are addressed in separate layers. The decomposition also allows for modular design and testing and the incorporation of plug-and-play components. A vast majority of multi-vehicle systems are organized into hierarchical control architectures. "*The fact of the matter is that the control of every large scale system is organized in a distributed hierarchy*" [11]. Examples can be found in [2-5]. The building blocks that form the architecture should be modular if they are to be used in several different vehicle formation control problems.

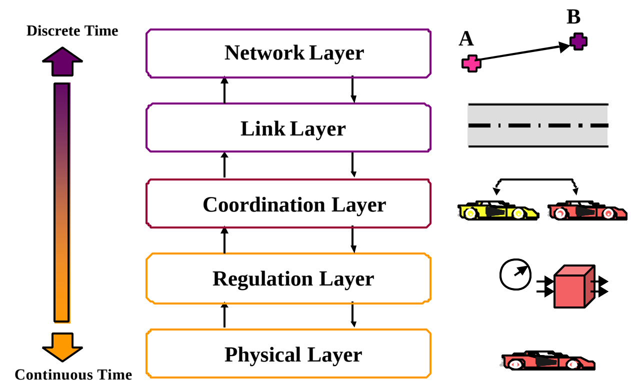

Figure 3. PATH Control Architecture.

As an example, we present the PATH hierarchical control architecture for Automated Highway Systems (AHS) [2]. The automation strategy of the PATH AHS architecture is organized in a control hierarchy with the following layers:

- Physical Layer--- the automated vehicles. The vehicle dynamical models are given in terms of nonlinear ordinary differential equations.
- Regulation Layer--- control and observation subsystems responsible for safe execution of simple maneuvers such as join, split, lane change, entry, and exit. Control laws are given as vehicle state or observation feedback policies for controlling the vehicle dynamics.
- Coordination Layer--- communication protocols that vehicles and highway segments follow to coordinate their maneuvers for achieving high capacity in a safe manner. The protocols are given in terms of finite state transition systems.
- Link Layer--- control strategies that the highway segments follow in order to maximize throughput. Control laws are given as traffic state and observation feedback policies for controlling the highway traffic using activity flow models.
- Network Layer--- end-to-end routing so that vehicles reach their destinations without causing congestion. Control laws are given in terms of queuing models.

The physical, regulation and coordination layers reside on each vehicle and the link and network layers reside on the roadside. To avoid single-point failures and to provide maximum flexibility, the design proposes distributed multi-agent control strategies. Each vehicle and each highway segment is responsible for its own control. However, these agents must coordinate with each other to produce the desired behavior of high throughput and safety.

Control architectures and their software implementations – the structure of interacting systems that enables those to

exhibit properties not to be found in the constituent modules -- are the single most important piece of design for multi-vehicle systems. Those are the elements that provide consistency and unity and where knowledge -- the main source of reuse that reflects the underlying design and implementation principles -- is encapsulated. Yet, a formal representation of this knowledge is not available for now.
At this level of design, and having in mind that these projects mainly involve some sort of middle-ware design, most problems share the same structure, one additional reason for reuse. As an example, the control architecture for the MOB project [3], inherited most of its conceptual design from the PATH architecture.
Architectures are more than interfaces -- although interfaces are definitely quite important for interacting systems -- and exhibit structural properties that, in the end, determine not only the performance of the overall design, but that of the overall project. Yet, this level of design lacks some formal expression. In the field of software design, the Architecture Description Languages (ADLs) provide formal representations of software architectures [9]. This is not the case with control architectures. Moreover, there is no formal basis to study properties at this level.

D. Maneuver Design

A sound design principle for maneuvers is that complex maneuvers should be formed using “basic”, simpler maneuvers as building blocks. The basic maneuvers can be verified for safety, consistency, and performance guarantees. More complex maneuvers are then built incrementally.

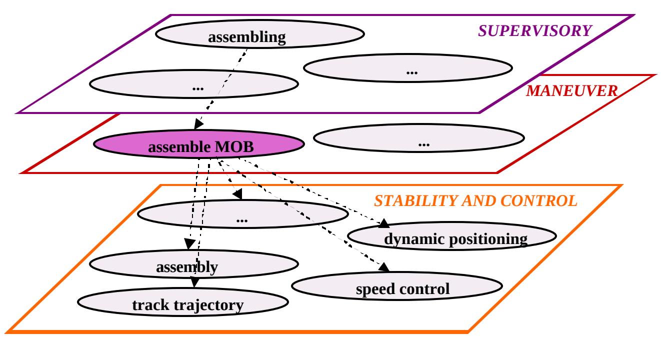

Figure 4. Composition of complex maneuvers from simpler ones. Example from the MOB project [3].

Consider, for example, the implementation of the “assemble a MOB” maneuver for the MOB case study. An order to assemble a MOB formed of two modules is received at the highest level in the hierarchy (supervisory layer). The supervisor invokes some communication protocols that ensure smooth cooperation between the modules.
If there are no obstacles, no other modules trying to join etc…, the assemble MOB maneuver is called (maneuver coordination layer). It uses a combination of different controllers, communication protocols and logic that switches between controllers.
These controllers generate reference trajectories or set points. The desired position is then sent to the low-level controllers (stability and control layer). These low-level controllers have been verified and shown to be safe. They deal with such things as station-keeping, position control, speed control, trajectory tracking etc… Details are given in [3].

E. Communications

The use of wireless communicated sensors in closed loop systems is an emerging new topic of importance in many distributed control system applications [13]. Analytical tools need to be developed to address concerns of performance degradation due to bandwidth limitations, packet losses and packet delay [21]. There are many alternative communication architectures that are used in wireless applications. Current controller designs assume that communication between the sensors and actuators and the central logic device is done over hard-wired lines. Thus, the controller design typically assumes perfect communication with no rate constraints, delays, or loss of information. The problem is that in networked multi-vehicle systems communications most often take place over wireless links. However, there currently exists no unified theory for the design of closed loop control systems where some of the loops are closed by imperfect communication links (links that are constrained in rate, introduce random delays, and occasionally drop packets).
For applications with wired communications, a good comparison of protocols and communication options appears in [14].

F. Real-Time Issues

Resource allocation problems arise most acutely when the software is separated into “pieces” (also called tasks or processes) that run concurrently. This is a very standard way to organize software. By breaking up a larger program into smaller pieces, the resulting source code is much easier to understand and maintain. All of the processes are working together to solve a larger problem, and must occasionally communicate with one another to share data or synchronize their activities. However, several difficulties are created by the separation into concurrent processes, and must be dealt with. The first problem is that of determinism: how can you tell exactly which process will run next? The second is that of resource management: given several processes, running concurrently, how do you ensure that the data is communicated safely and reliably, and not accessed simultaneously by more than one process? There are several standard approaches to the resource management problem, including shared memory, publish and subscribe type mechanisms, etc… Examples of each can be found in [8, 25, 26]. Scheduling algorithms exist that solve the determinism problem, and are available in the literature, for example in [26-27].
In multi-vehicle systems, typically each vehicle has its own clock, which may be different from that of the other vehicles. The starting time will almost certainly be different, yet it is also possible that one second on vehicle 1 is not exactly equivalent to one second on vehicle 2. If each

vehicle logs its own data, the data relevant to the group of multiple vehicles may not be coherent. Techniques to solve this problem are addressed in [15].

V. LESSONS LEARNED, CONCLUDING REMARKS AND OPEN PROBLEMS

The future will see many more applications of coordinated control of networked multi-vehicle systems. The communications network will by necessity be a wireless one due to the mobility requirement. There are many research issues that remain to be solved before a robust, fault-tolerant system can be achieved. These include:

- Control architecture
- Mesh stable control algorithms
- Fault detection/tolerant management
- Communication architecture.

This paper has concentrated on the control and communications issues. There are many other issues that were not addressed in this paper that are also critical to the successful implementation of multiple-vehicle coordinated missions including mission planning, obstacle detection and sensor fusion.

Acknowledgements: The material is based upon work supported by DARPA (MoBIES) under grant number F33615-00-C-1698, and the Link Foundation. The authors would like to thank Prof. Varaiya for helpful discussions providing insight into this problem.

References

[1] G. Dudek, M.R. Jenkins, E. Milios, D. Wilkes, “A Taxonomy for Multi-Agent Robotics”, Autonomous Robots Vol.3 No.4, Dec. 1996, p. 375-97.
[2] R. Horowitz and P. Varaiya, “Control Design of an Automated Highway System”, Proceedings of the IEEE, Vol. 88, No. 7, July, 2000, pp. 913-925.
[3] A. Girard, J. Borges de Sousa and J.K. Hedrick, “A Hierarchical Control Architecture for Mobile Offshore Bases”, Marine Structures Journal, Special Issue on Very Large Floating Structures, Vol. 13 (2000), pp. 459-476.
[4]T.J. Koo, “Hierarchical System Architecture for Multi-Agent, Multi-Modal Systems”, Submitted to the 2001 IEEE Conference on Decision and Control under paper number CDC0-INV3106.
[5] A. J. Healey, D.B. Marco and R. B. McGhee, “Autonomous Underwater Vehicle Control Coordination Using A Tri-Level Hybrid Software Architecture”, Proc. of 1996 IEEE Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp 2149-2159.
[6] A. Girard and J.K. Hedrick, “Dynamic Positioning of Ships using Nonlinear Dynamic Surface Control”, Accepted for publication in the Proceedings of the Fifth IFAC Symposium on Nonlinear Control Systems, St. Petersburg, Russia, July 2001.
[7] P. Varaiya, "Smart Cars on Smart Roads: Problems of Control", IEEE Trans. on Automatic Control, Vol. 38, No. 2, February 1993.
[8] A. Girard, "A Convenient State Machine Formalism for High-Level Control of Autonomous Underwater Vehicles", Master’s Thesis, Florida Atlantic University, Boca Raton, FL, May 1998.
[9] D. Garlan, “Research Directions in Software Architecture”, ACM Computing Surveys, Vol. 27, No. 2, pp. 257-261, 1995.
[10] T. Curtin, J. Bellingham, J. Catipovic and D. Webb, “Autonomous Ocean Sampling Networks”, Oceanography, Vol. 6, No. 3, pp. 86-94, 1993.
[11] P. Varaiya, “A Question About Hierarchical Systems”, Personal Communication, 1999.
[12] D. A. Schoenwald, “AUVs: In Space, Air, Water, and on the Ground”, Guest Editorial, IEEE Control Systems Magazine, December 2000.
[13] P.R. Kumar, “New Technological Vistas for Systems and Control: the Example of Wireless Networks”, IEEE Control Systems Magazine, Vol.21, No.1, IEEE, Feb. 2001, p.24-37.
[14] F.L. Lian, J.R. Moyne, and D.M. Tilbury, “Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet”, IEEE Control Systems Magazine, Vol.21, No.1, IEEE, Feb. 2001, p. 66-83.
[15] M. G. Rodd, “Engineering Real-Time Systems”, Computing and Control Engineering Journal, October 1995, pp 233-240.
[16] D. N. Godbole, J. Lygeros and S. Sastry, “Hierarchical Hybrid Control: An IVHS Case Study”, in Hybrid Systems II (P. Antsaklis, A. Nerode and S. Sastry, eds.), no 999 in LNCS, Springer Verlag, 1995.
[17] http://robotics.eecs.berkeley.edu/bear/
[18] S. Sheikholeslam and C.A. Desoer, “Control of Interconnected Nonlinear Dynamical Systems: The platoon problem”, IEEE Trans. on Automatic Control, Vol. 37, No. 6, pp. 806-810, June 1992.
[19] J.K. Hedrick and D. Swaroop, “Dynamic Coupling in Vehicles under Automatic Control”, 13th IAVSD Symposium, August 1993.
[20] D. Swaroop and J.K. Hedrick, “String Stability of Interconnected Systems”, IEEE Trans. on Automatic Control, Vol. 41, No. 3, pp. 349-357, March 1996.
[21] A. Pant, P. Seiler, M. Broucke, T.J. Koo and J.K. Hedrick, “Coordination and Control of Autonomous Vehicles”, submitted to IEEE Transactions in Robotics.
[22] P. Seiler, A. Pant, and J.K. Hedrick, “Preliminary Investigation of Mesh Stability For Linear Systems”, IMECE99/DSC-7B-1, November 1999.
[23] J.G. Bellingham, T.R. Consi, R.M. Beaton and W. Hall, “Keeping Layered Control Simple”, Proc. of the Symposium on AUV Technology – AUV 1990, pp. 3-8, June 1990.
[24] J. Borges de Sousa, F. Lobo Pereira and E. Pereira da Silva, “A Dynamically Configurable Architecture for the Control of AUVs”, Proceedings of the IEEE Oceans Conference, pp. 131-136, October 1994.
[25] S. Tripakis and S. Yovine, “Analysis of Timed Systems Based on Time-Abstracting Bisimulations”, in Proc. of the 8th International Conference on Computer Aided Verification, CAV '96, New Brunswick, NJ, USA, Aug. 1996, pp.232-43.
[26] S. Vasan and R. Rajkumar, “A Methodology for Predictable Real-Time System Design”, in Proc. of the 17th Digital Avionics Systems Conference, Bellevue, WA, USA, Nov. 1998, pp. C26/1-8.
[27] http://www.teja.com
[28] http://mob.nfesc.navy.mil/