# A CONTROL FRAMEWORK FOR A REMOTELY OPERATED VEHICLE

João B. Sousa[+]
Sérgio L. Fraga[+]
Alfredo Martins[**]
Fernando L. Pereira[+]

[+]{jtasso,slfraga,flp}@fe.up.pt
Laboratório de Sistemas e Tecnologias Subaquática
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal

[**]amartins@dee.isep.ipp.pt
Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, s/n 4200-072 Porto, Portugal

## Abstract

A control framework enabling the automated maneuvering of a Remotely Operate Vehicle (ROV) is presented. The control architecture is structured according to the principle of composition of vehicle motions from a minimal set of elemental maneuvers that are designed and verified independently. The principled approach is based on distributed hybrid systems techniques, and spans integrated design, simulation and implementation as the same model is used throughout. Hybrid systems control techniques are used to synthesize the elemental maneuvers and to design protocols, which coordinate the execution of elemental maneuvers within a complex maneuver.
This work is part of the Inspection of Underwater Structures (IES) project whose main objective is the implementation of a ROV-based system for the inspection of underwater structures.

Keywords: Remotely Operated Vehicle, Automated Maneuver, Underwater Inspection

## 1. Introduction

Remotely Operated Vehicles (ROV) are small, tethered submersibles. There are numerous applications for ROVs, such as oceanographic surveys, operations in hazardous environments, underwater structure inspection, and military applications. Although ROV control presents many difficult challenges, recent advances in navigation, power and communication systems offer the appropriate technology to enable the use ROV for data gathering in the coastal and open ocean.

In this context, we will present the development a control framework for a ROV enabling automated maneuvering. By a control framework, we mean the organization of a number of problems enabling the ROV to perform a given set of maneuvers and maneuver switching. From our experience, the design of the control and software framework is one of the most critical phases in automated vehicle development. The architecture helps us understand what the system does and how it works. Furthermore, it allows the modular development of the system, extend it, and reuse its parts to build another system.

Two levels of automation can be considered: one in which “high-level” maneuver commands are sent by the operator and another one enabling complete automation. The literature is abundant in low-level control techniques for ROV and/or ship applications. These techniques are mainly tailored to solve low level control problems (the level of control that directly interfaces with actuators such as auto-pilots, etc...) that are formulated in the framework of continuous time and differential equations. However, high-level maneuver automation involves the realm of logic, and of discrete event models interacting with differential equations modelling the ROV dynamics. To reach this level of automation, it is necessary to consider hybrid systems and new control techniques enabling the interaction between continuous time and discrete events.

ROV motions are organised in terms of maneuvers. From the application requirements, a complete set of maneuvers, of which some are simpler and others more complex, is defined. One of the design thrusts consists in specifying a basic set of "elemental" maneuvers from which all others can be derived. Each one of these elemental maneuvers is designed is verified for safety. Then, the ROV complex maneuvers are composed by using the elemental ones as basic building blocks. This ensures the correctness of all maneuvers, that is, that they meet the given specified requirements, which may include safety, ensured results and performance levels, even in the presence of disturbances. The novelties of the framework consist in the introduction of concepts and theories from distributed hybrid systems, and the use of systems engineering principles for architectural design [1]. We illustrate this approach with a case study from the project **I**nspection of Und**e**rwater **S**tructures (IES). This project concerns the design and implementation of an advanced low cost system for the inspection of underwater structures based on a ROV.

This paper is organized as follows. In section 2, the IES project is described. Section 3 presents our principled approach to design and implementation. Section 4 contains an overview of the control framework. Finally, in section 5, some concluding remarks and future work are outlined.

## 2. The IES project

The project **I**nspection of Und**e**rwater **S**tructures (IES) concerns the design and implementation of an advanced low cost system for the inspection of underwater structures based on a Remotely Operated Vehicle (ROV). The project started in 1999, has a total duration of 3 years, and is funded by PROGRAMA PRAXIS XXI - MEDIDA 3.1B, Portugal. IES is a collaborative project that involves the Associação dos Portos de Douro e Leixões (APDL)[1], Faculdade de Engenharia do Porto and Instituto de Sistemas e Robótica – Pólo do Porto. A summary of the inspection requirements for the IES system is presented in table 1. A typical operational scenario is depicted in Fig. 1.

| **Main requirements** | | |
|---|---|---|
| *Inspection objective* | *Features* | *Main difficulties* |
| Evaluation of the corrosion state of submerged steel plates | Metal plates are part of the pier structure.<br>Critical corrosion spots: links between consecutive plates | Low visibility due to pollution and marine growth.<br>Wave induced motions.<br>Unavailability of good models of thruster-wall interactions.<br>Sensing fissures and corrosion.<br>Size of the area to be inspected. |
| Evaluation of the conservation state of underwater structures | Fissures on the metal surface.<br>Misalignment of consecutive metallic curtains. | |
| Concrete pillars maintenance | Fissures on concrete pillars. | Wave-induced motions.<br>Currents produced by tide variation and Leça river stream. |
| **Additional requirements** | | |
| Verifying the state of the tetra pods that protect the harbour. | The locations of the tetra-pods change due to the action of the sea. | High waves.<br>Currents.<br>Vessel motion. |
| State of ship hulls. | Inspection should be made during loading and unloading operations. | Low visibility due to pollution and low luminosity. |

Table 1 - inspection requirements

The main innovations of the IES system with respect to commercially available ROV solutions are:

- On-board power and computer systems. This physical configuration minimizes the number of wires in the tether cable thus minimizing drag and improving performance.
- Two modes of operation: tele-operation and tele-programming. The tele-operation mode is a standard feature in ROV systems. The tele-programming mode enables the operator to program automated operations, such as trajectory or path tracking.

[1] Harbor Authority

- Integrated navigation. The IES navigation system integrates data provided by an external acoustic system and by internal sensors for better control performance and position accuracy.
- PC-based control. Easy to use, COTS technology, low development and maintenance costs.
- Advanced control systems. The ROV control system includes advanced automated operation modes that relief the operator from tedious tasks, and do not require extensive training.
- Open system. The project uses standard software development tools and principles. The software architecture allows for the easy integration of code developed by third parties.

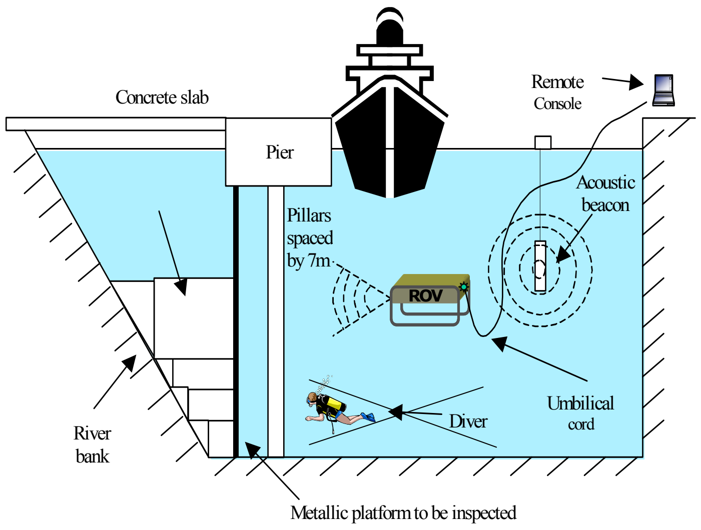

Fig 1 – Typical operational scenario

The IES system for the inspection of underwater structures integrates the following innovative technologies and systems developed at the Laboratório de Sistemas e Tecnologias Submarinas (LSTS) of Porto University:

- Acoustic navigation system (see [2,3]).
- Advanced control systems.
- Power and motor control.

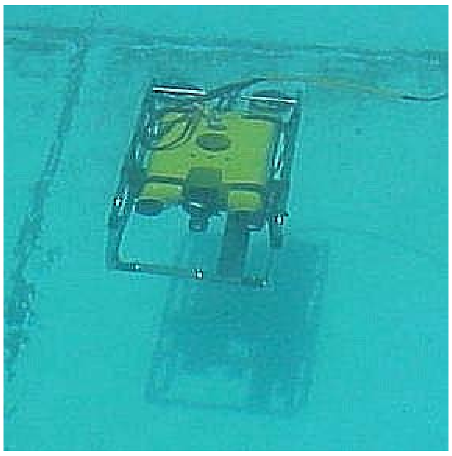

Fig. 2. The IES remotely operated vehicle

Except for the ROV frame, hull and thrusters, all the other components and systems were designed and implemented at LSTS. The ROV frame, hull and thrusters are a customized version of the Deep Ocean 500 S model from Deep Ocean Engineering (Fig. 2). The main difference with respect to the standard model is an additional cylinder that houses electronics and sensors.

In the basic configuration the IES system comprises the following systems:

**Computer system.** Consists of a PC-104 stack running the real-time operating system QNX, and a Windows based PC connected through an Ethernet cable. The PC-104 stack is housed in the main cylinder of the ROV, and controls the ROV systems through a CAN bus. Additional sensors are interfaced through an A/D card on the PC-104 bus. The PC runs the operator console. The PC also runs a Web server providing Web-based access to data from operations, while ROV control is restricted to the operator console. The PC-104 computer system runs the command, control and navigation software. Basically, this computer accepts high-level commands from the console, and informs the console about the state of the system.

**Power system.** A portable generator provides electrical power to the system. The umbilical cable feeds the ROV power system with 120 V DC through two power wires to minimize its weight. This design option aimed at minimizing the effects of the tether on the ROV dynamics, one of the traditional difficulties associated with ROV operations.

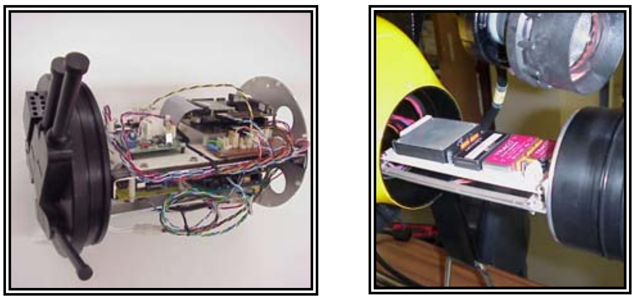

Fig 3 – PC104 stack and on-board power system

**Motor control system.** This system comprises two CAN nodes housed in the two upper vehicle cylinders. The controllers generate the reference signals to the four thruster power drives.

**Navigation system.** The suite of navigation sensors includes the on-board sensors, and an external acoustic navigation network (see [2,3]). The on-board sensors are: magnetic compass, inclinometers, inertial navigation unit, depth cell, altimeter and Doppler Velocity Log.

**Vision system.** In the basic configuration the vision system consists of a camera mounted in a pan-and-tilt unit and spot lights. The video image is converted to the digital format in the on-board frame-grabber, and sent to the console through the Ethernet connection.

The basic inspection configuration can be enhanced with a set of plug-and-play inspection and intervention tools. This set comprises a vision system, and an array of magnetic sensors. Other tools that are being considered for development include a tactile sensor array and a scraping device.

## 3. The approach

The general problem of vehicle control automation is not a trivial one. ROV control automation is not an exception. We envision two levels of automation: 1) automation of the *basic maneuvers* and, 2) full automation, from planning to operations.

From the control perspective, the main difficulties for automating ROV control come from the first level of automation, that of the elementary maneuvers. The problem is that traditional control techniques, that form the current practice in ROV control, are not adequate for this purpose. These techniques are mainly tailored to solve the low-level control problems. The problem of design for the second level of automation is not trivial, but it relies on the services provided by the first level. Let us look at what is involved in the design of a basic maneuver controller to understand why we need new techniques.

**Formal methods.** During manual control the operator uses some control logic to command the operation of the ROV. We need to be able to capture this logic, to express it mathematically, and to check it for consistency and correctness. This is why we need formal methods from computer science [4].

**Models.** The realm of the low-level control problems is that of continuous time and differential equations. Here, we enter the realm of logic, and of discrete event models interacting with differential equations that model ROV dynamics.

**Automation is not mimicking manual operation.** First, it may prove difficult to encode all the control logic in a compact representation. Second, even if this is done there is no assurance that there are no flaws in the control logic. Third, automation enables complex quantitative reasoning that can be used to optimise operations or to implement complex control routines that ensure safe operations, even in the presence of disturbances.

Obviously the realm of interactions between continuous time dynamic models and discrete events requires the consideration of advanced control techniques. Let us discuss what is required from control design.

*Logic based control*. The actions of each module follow some control logic. This control logic is complicated because it not only involves discrete event behaviour, but also complex continuous dynamics and interactions with other modules. The traditional practice of **if-then-else** programming is no longer adequate. It is not required to be an expert programmer to realize that the amalgamation of **if-then-else** statements is very difficult to verify, and often leads to unpredictable behaviour.

*Safety and predictability*. Two important requirements for automated operation are predictability and safety. The controller has to perform according to what it is expected to do while respecting safety constraints. Disturbances play against predictability and safety. The question is then how to design predictable and safe controllers. This entails designing not only control laws, but also regions for safe operation.

In our experience, a lot of control code is designed and implemented using techniques that are not particularly adapted to this level of automation. The problems are: 1) expressiveness and, 2) tool integration. We need compact representations to express the models and relations described above, and we need to interface the code with tools that facilitate design and verification.

It is now clear that control design at this level requires: 1) **formal models** that span the design and implementation process; 2) a framework where we can study the overall structure and properties of control design that are not appropriately addressed within the constituent modules -**control architecture**; 3) **a principled approach** to design and implementation; 4) **new control techniques**.

## 4. Control framework

This section describes the key elements of our approach. For the sake of clarity we opted to skip the mathematical details that can be found, for example, in [5].

### *A. Principles and techniques*

We organise the operations of the ROV as a sequence of maneuvers. First we define a basic set of "elemental maneuvers", from which all others can be derived. Then the complex ROV maneuvers can be defined by using the elemental ones as building blocks. From the operator's perspective this means having at his disposal a set of commands with which a complex mission can be planned and executed. The set of commands is designed to comply with the operational requirements while ensuring proper termination, or adequate fault handling. For the purpose of modularity tele-operation is defined as an elemental maneuver. The other maneuvers are tele-programming primitives.

The design of elemental maneuvers is incremental since each one is developed and verified independently. Then, the complex maneuvers are formed and tested. For each elemental maneuver, a least restrictive controller – one that provides us with "sets", or "windows" of possible actions – is first synthesised. When safety or final objectives are not at stake, one of the controls can be selected. This is convenient, since in control, precision is costly, yet the ocean environment cannot be quantified exactly. To handle safety and performance issues, we then complement the design with rules or controllers that incorporate some logic. Our controllers ensure that, if feasible, the maneuver's objective is indeed attained. Hence the control architecture is fault-tolerant since it uses correct elemental maneuvers.

The overall concept of operation is better explained with recourse to figure 5. The operator uses the console to command the execution of maneuvers. The ROV computer runs a supervisor that accepts (or rejects) commands from the console and controls their execution. Basically, there are three modes for the supervisor: executing command, idle and error. By default, and in the absence of any command sent by the operator, the supervisor is in the idle mode that instantiates a special maneuver - the safe maneuver. Upon receiving a command from the console, the supervisor either starts its execution, or does not accept the command. Command execution either terminates as programmed, or is aborted when an error occurs. In this case the transition to the error mode invokes error-handling procedures.

This control framework is based on recent developments in the theory of distributed hybrid systems. Informally, a distributed hybrid system is a collection of dynamic systems – each of which includes both continuous time activities and discrete-event features – that interact through the exchange of data and messages. Fig. 5 describes some of those interactions.

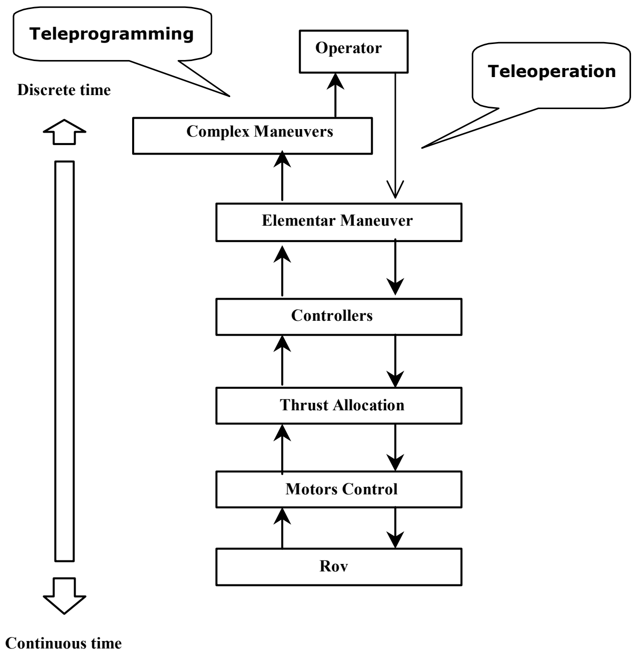

Fig. 5 – Control interactions

*B. Maneuver model*

**Definition** [Elemental maneuver]. Prototypical solution to a class of ROV motion problems that cannot be obtained from the composition of other maneuvers. It is characterized by:

- Objective.
- Hard and soft constraints.
- Information sets.
- Dynamic model and controllers.

A maneuver is specified according to the following table.

| Feature | Description |
|---|---|
| Inputs | Type of maneuver.<br>Maneuver parameters: acceleration profiles, time-outs, final position, etc.<br>User commands accepted during the execution. |
| Outputs | References to regulation layer.<br>Termination and error events. |
| Internal Variables | Used to evaluate termination conditions. |
| Termination Conditions | Specification of normal and error conditions for maneuver termination. |
| Controller | Algorithms for trajectory generation and control. |

Table 2 – Maneuver specification

*C. Architecture*

The control architecture is at the level of design that addresses the overall structure and the properties of control systems, hence providing a focus for certain aspects of design and development that are not appropriately addressed within the constituent modules (see [6] for a discussion on software architectures). This structure is formalized in terms of layers and interfaces [7]. We consider a tri-level control architecture.

*Regulation Layer* - the automated vehicles. The dynamical models of the vehicles are given in terms of nonlinear ordinary differential equations. This level deals with continuous signals, and interfaces directly with the vehicle hardware. Control laws are given as vehicle state or observation feedback policies for controlling the vehicle dynamics. Control laws at this level correspond to low level commands such as course keeping, turning, etc…

*Maneuver Layer* - control and observation subsystems responsible for safe execution of maneuvers - the first level of automation. The supervisory layer commands the execution of elemental according to some motion plan. Interactions with the regulation layer are mediated by the elemental maneuvers. Each elementary maneuver sends low-level commands to the regulation layer and receives events concerning their completion or failure. Elemental maneuver control is given in terms of hybrid automata. The current design uses protocols in the form of finite state machines.

*Supervisory Layer* – controls and coordinates the execution of maneuvers.

### *D. List of maneuvers*

The Maneuver class aggregates the common attributes and methods of all manoeuvres. Specific manoeuvres inherit all these attributes and methods and contain additional parameters or methods. A brief description of all of the elementary manoeuvres follows:

**Hovering**. Hover inside some prescribed boundaries while maintaining ROV orientation. Examples: inspection of a target with high precision.

**Plane**. Stabilize with respect to a vertical or a horizontal plane with bounded errors. Examples: inspect a wall while moving in a vertical plane parallel to it.

**Orientation**. Change ROV orientation in terms of its vertical axis. Examples: change ROV orientation in order to track some visual target.

**Trajectory.** Follow a prescribed trajectory, with bounded errors. Examples: inspect a wall in a yo-yo motion with a specific time parameterisation.

**Path.** Follow a prescribed path, with bounded errors. Examples: follow a prescribed path inspection pattern while allowing the operator to decide on the speed of the ROV.

**TrackTarget.** Track a target acquired with the visual system within bounded error. Examples: inspection a moving object.

**FollowWall.** Follow a prescribed wall, with bounded errors. Examples: inspection of a surface with unknown shape.

**ManualOperation.** Teleoperation. Assists the operator to drive the ROV from the GUI-based operator's console. This maneuver accepts operator commands during maneuver execution. The operator commands are filtered through a low-level control system that stabilizes the motion of platform. This fly-by-wire capability makes it possible for a novice operator with no special training to control the ROV with minimal effort. This feature is not available in commercial systems.

### *E. Control design*

Here we discuss control design for this architecture. For reasons explained before we will concentrate this discussion on control synthesis for the maneuver layer. The generic problem of elemental maneuver control synthesis can be described as follows: given a dynamic system, or a collection of interacting dynamic systems, synthesize a controller so that the system(s) satisfies(y) the maneuver specifications. The type of maneuver specification dictates the type of control formulation. Examples include: 1) the problem of invariance - staying inside some region; 2) the problem of attaining a given target set while the trajectories of the system remain inside some other set; 3) the problem of optimizing some criteria; 4) the problem of stabilizing a system. Inherent to most of these control formulations is the problem of reach set computation -- the set of all positions that the ship can reach from a given starting position. In fact, given the reach set, it is quite simple to solve most of these problems.

Reach set computation for hybrid systems has received considerable attention from the control systems community. For our developments we are interested in the approach proposed by Kurzhanskii and Varaiya [9] They use dynamic programming techniques to describe reach sets and related problems of forward and backward reachability. These problems are formulated as optimisation problems that are solved through the Hamilton-Jacobi-Bellman equations. The reach sets are the level sets of the value function solutions to these equations.

The control formulations that we use fall within the categories of differential games and viability theory. Viability can be described as follows: given a dynamical system, a region K, and a set of initial conditions that lie within K, synthesize a control law that ensures the state of the system never leaves K [10]. The setting of differential games is that of a dynamic optimization problem where the control inputs are partitioned into two classes: 1) those available for controlling the system, 2) those available to the adversary or the disturbance. This setting extends that of deterministic optimal control to the case where a stochastic characterization of the disturbances is not available, preventing the formulation of a stochastic optimal control problem.

Let us discuss the design of controllers for an elemental maneuver in the framework of hybrid systems. This is done in three phases: 1) translation of the maneuver specifications into restrictions on the system's reachable sets of states; 2) formulation of a differential game of the appropriate type (example reaching a target set), and derivation of Hamilton-Jacobi-Bellman equations whose solutions describe the boundaries of the reachable sets and of the safe set - the set of all positions from which there is a controller that ensures the target is always reached; 3) synthesis of the hybrid controller from these equations. The hybrid controller consists of: 1) a specification of the safe set; 2) a least restrictive controller that assumes the form of a feedback control law for the continuous and discrete variables which guarantees that the hybrid system remains in the "safe subset" of the reachable set; 3) a controller that selects the final control setting from the set of control options given by the least restrictive controller. The least restrictive controller solves the problem of correctness while leaving room for other control considerations when safety (constraint violation) or the attainment of objectives is not at stake. Prior to execution the controller performs a feasibility check for a maneuver specification. Upon this check, it ensures correctness of the maneuver. It does this by generating a non-deterministic output - the set of feasible controls - when safety is not at stake. The second controller selects the commanded control setting from this set. This two-level control design allows for the integration of qualitative design techniques that may account for qualitative human decision-making.

Results from differential games and viability theory have been used to synthesize maneuvers and controllers for aircrafts. A game theoretic approach to the design of safe collision-avoidance maneuvers for air traffic management systems was proposed in [11]. For a brief discussion on protocol design for complex maneuvers see [5].

## 5. Conclusions

This paper reports an automated maneuvering control framework for a ROV. The novelty of our approach stems from the consideration of systems engineering principles and of distributed hybrid systems techniques. The innovations are the introduction of safe elemental maneuvers in a three level coordination and control hierarchy, in order to manage complexity. We formalize the notion of elemental maneuver and synthesize safe elemental maneuvers using techniques from control theory, hybrid systems, differential games and viability theory.

Future work includes implementing a full set of safe elemental maneuvers and enriching the maneuver switching logic to accommodate faults and automated planning.

**Acknowledgements**

The authors thank Anouck Girard for stimulating discussions and valuable comments, insights and contributions.

## References

[1] IEEE (1999). “IEEE standard for application and management of the systems engineering process”.

[2] A. Matos, A. Martins, N. Cruz, F. Pereira, “Development and implementation of a low-cost LBL navigation system for an AUV”, MTS/IEEE Oceans'99, Seattle, U.S.A, Sept. 1999.

[3] N. Cruz,, L. Madureira, A. Matos and F. Pereira, “A versatile acoustic beacon for navigation and remote tracking of multiple underwater vehicles”, to appear in MTS/IEEE Oceans'01, Honolulu, HI, U.S.A, Nov. 2001.

[4] Jan van Leeuwen editor (1990). "Handbook of theoretical computer science". Elsevier.

[5] P. Varaiya and T. Simsek and J. Borges de Sousa (2001). "Communication and control in hybrid systems". Accepted as a tutorial session for ACC 2001.

[6] David Garlan. "Research directions in software architecture". ACM Computing Surveys (1995). Vol. 27, no 2, pp. 257-261.

[7] J. Borges de Sousa, F. Lobo Pereira and E. Pereira Silva (1995). “A dynamically configurable control architecture for autonomous mobile robots”. Procs of the 34th IEEE Conference on Decision and Control.

[8] G. Booch, J. Rumbaugh, I. Jacobson. “The Unified Modelling Language User Guide”. Addison-Wesley (1999).

[9] A. B. Kurzhanskii and P. Varaiya. “Ellipsoidal techniques for reachability analysis”. Computation and control. N. Lynch and B. Krogh. Lect. Notes in Computer Science. Springer-Verlag. (2000). pp 202-214.

[10] Jean-Pierre Aubin, "Viability theory", Birkhauser (1991).
[11] J. Lygeros and Datta N. Godbole and Shankar Sastry, "A game theoretic approach to hybrid system design", University of California, Berkeley. Electronics Research Laboratory", UCB/ERL M95/77" (1995).