International Journal of Systems Science
Vol. 37, No. 6, 15 May 2006, 361–376

A semiotic approach to the control of semi-autonomous robots

JOÃO SEQUEIRA* and ISABEL RIBEIRO

Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal

(*Received 26 November 2004*)

This paper describes the application of basis concepts in semiotics and nonsmooth systems to robot control. The resulting framework yields a vocabulary of movements tailored to the control of semi-autonomous robots. Semiotics provides a set of classes of objects to model the interactions among humans. The paper proposes the use of these semiotic objects to model human–robot interactions, accounting for features such as semantics and ambiguity. Mathematical models for the objects are presented including objects to generate paths for the robot to follow and operators to act on such objects. Experiments with single and multiple robots illustrating the main aspects of the proposed framework are presented.

Keywords: Human–robot interaction; Semiotics; Semi-autonomous robots

1. Introduction

The focus of this paper is on the development of a semi-autonomous robot (SAR) control framework accounting for the semantics and ambiguity in the interactions between humans and robots. Basis semiotic objects are mapped into a set of mathematical objects that forms the core of the proposed human–robot interaction (HRI) paradigm, this correspondence being the novelty of the paper.

Semi-autonomous robotics has been gaining importance in the last few years, mainly due to the slow developments in its fully autonomous robotics counterpart. The study of the interactions between humans and robots dates back to the early days of robotics and has been considered either as a formal language design problem, a communication protocol design problem or as a natural language recognition problem.

Traditionally, semi-autonomous robots have been considered under a single robot perspective. The fast development of networking technologies fosters the spreading of applications of teams of robots. The interactions among robots and between robots and humans are then of key importance for a cooperative operation of teams of robots and humans. The added complexity of the networked robotics control problem further pushes the study of alternative models for the interactions among robots and between humans and robots. Humans handle their interactions, namely in what concerns semantics and ambiguity, with a remarkable degree of success. These are the key features as they often allow the reduction of complexity in the interactions. Therefore, it is natural to adapt human interaction models, such as those provided by semiotics, to teams of robots and teams composed by both robots and humans, as in SAR teams. This is the main goal of this work, that is, establishing a mapping between semiotics and mathematical objects. The work presented in this paper builds upon previous work on conceptual robot control architectures described in Sequeira and Ribeiro (2000).

Often, HRI and SAR control have been considered independently of each other. Most of the architectures for robot control presented in the literature have specific forms of representing a mission which, in some sense, amounts to a simplified form of HRI. In general, the

---

*Corresponding author. Email: jseq@isr.ist.utl.pt

ISSN 0020–7721 print/ISSN 1464–5319 online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720500438522

full autonomy characteristic, common to most of them, does not restrain the adaptation to SAR control. Furthermore, semiotic principles can be identified in several approaches. The subsumption architecture (Brooks 1986), introduced the concept of behavior. Though not directly related with SAR, the idea underlying the use of behaviors is to decrease the complexity in architecture design and mission specification. This fosters the idea of communication among heterogeneous agents through behaviors, which has tight relations with semiotics. Human factors, such as the anthropomorphic characteristics of a robot, are a key subject in HRI as humans tend to interact better with robots with human characteristics (Kiesler and Hinds 2004). A framework based on stable dynamic systems was developed in Bicho and Schöner (1996). By identifying mission goals with equilibrium points in dynamic systems described by mission dependent vector fields it is possible to guarantee the successful execution of a mission. The RoboCup events have been providing benchmark scenarios for problems in fully autonomous cooperative robotics. In Utz *et al.* (2004) the robots interact with each other by exchanging state information, including the robot configuration, information on uncertainty and symbolic information to handle ambiguous situations. A behavioral approach to the reactive control of formations of robots was presented in Balch and Arkin (1999), with the robots exchanging position information among them.

Among the architectures tailored to SAR applications, (Georgia Tech 1984) considers a reactive behavioral layer and additional layers to input mission specifications. The CAMPOUT proposal (Schenker *et al.* 2001, Huntsberger *et al.* 2003) is supported on a hierarchy of behaviors. The MACTA (Aylett and Barnes 1998) is also behavior based, with the HRI handled by a mission organizer subsystem. The MAUV architecture (Albus 1987), implements a sense-process-act loop supported on Artificial Intelligence techniques. A hybrid, deliberative/reactive architecture is presented in Kortenkamp *et al.* (1999), supported on a functional hierarchy with planning, sequencing and skill managing layers. In Nicolescu and Matarić (2001) the robots are equipped with behaviors that convey information on their intentions to the outside environment. These behaviors allow a form of implicit communication between agents.

Context dependent functional languages have been proposed to simulate robot systems and also as a means to interact with them (Hudak 1998, Hager and Peterson 1999). Web technologies have also been used to support HRI paradigms (Makatchev and Tso 2000).

The paper details the SAR control framework both at conceptual and practical implementation levels and is divided into five additional sections. Section 2 introduces the proposed paradigm from a semiotics perspective, describing the main concepts used in HRI modeling. Section 3 details the mathematical objects in the proposed paradigm, accounting for the relevant semiotic principles. The semiotics concepts are morphed into motion strategies and operators acting on them and hence have direct consequences on the controllability of the robot. Section 4 extends the controllability concept of dynamic systems theory to the proposed paradigm. In the context of this paper, controllability is a mission independent concept. However, the success of a mission depends not only on the available motion strategies but also on how they are chosen, this being dependent on the assigned mission. Section 5 complements the paradigm by defining a hybrid architecture using the blocks previously defined. Section 6 presents experiments using common robot models (simulated and real). These experiments demonstrate the control of a team of robots operating autonomously as a formation and the control of a single SAR using a basic interface to highlight the semiotics aspects of the paradigm. Section 7 presents the conclusions and points directions for future research.

2. HRI and semiotics

In general, robots and humans work at very different levels of abstraction. Humans work primarily at high levels of abstraction whereas robots are commonly programmed to follow trajectories, hence operating at a low abstraction level. Thus, the HRI maps different abstraction levels. Mapping different abstraction levels in robot control has been done by current architecture paradigms such as hierarchical (Saridis 1996) and subsumption (Brooks 1986).

Semiotics is a branch of Philosophy which studies the interactions among humans, such as the linguistic ones (see for instance Eco (1984) for an introduction to semiotics). Over the last decade semiotics has been brought to intelligent control and then it naturally spread to robotics (see for instance Meystel and Albus (2002)). Different paradigms have been presented to model such interactions. See, for instance Albus (1996) and Meystel (1996) related to intelligent systems, Malcolm and Goguen (1998) in algebraic semiotics and its use in interface design, Neumüller (2000) in the hypertext theory applied to World Wide Web, or Codognet (1996) in machine–machine and human–human interactions over electronic media (such as the Web).

The idea underlying semiotics is that humans communicate among each other through signs. Signs can be of three categories† (Codognet 1996, Malcolm and Goguen 1998): (i) symbols, expressing arbitrary relationships, such as conventions, (ii) icons, such as images, and (iii) indices, as indicators of facts or conditions.

The study of signs is made according to three different perspectives: semantics, pragmatics and syntactics (Neumüller 2000). Semantics deals with the general relations among the symbols. Pragmatics handles the hidden processes and meanings that require the agents to perform some inference on the symbols‡. Syntactics is related to the structural rules to be observed by the agents when composing signs into strings and messages.

The HRI paradigm presented in this paper is concerned only with (i) semantics and how it constrains the success of a mission, and (ii) the mathematical representation of the signs used in HRI. For the sake of simplicity, the HRI model considered uses only symbols as basic units for inter-agent communication. Icons are often used by humans in their interactions, (e.g., in art to convey an idea) and are also often found in robotics; for example, topological features can be extracted from an image aiming at self-localization. Indices are also often used among humans, e.g., in literary texts and when inferring a fact from some sentence as in "the robot has no batteries" can be inferred from the observation "the robot is not moving", or as in "the robot is moving away from the programmed path" and hence there must be an obstacle in the path.

Symbols have been extensively used in HRI without bearing any explicit relation to semiotics. For example, robots have been made to interact within teams using a variety of languages, each having specific sets of symbols, namely keywords and function names. Besides the languages/protocols that exchange raw (numeric) state information (Utz *et al.* 2004), the exchange of symbolic information has also been considered in multiple works (Kortenkamp *et al.* 1999, Jung and Zelinsky 2000). Discrete event systems provide multiple examples of the use of symbols, e.g., as labels for states and events used to model a mission (Milutinovic and Lima 2002). Commonly, symbols establish a close relation between a label and its meaning, almost as a one-to-one map. If the humans are assumed to have enough knowledge on the robots and the environment these tight relations may be easy to establish and standard computer languages can be used for HRI. Multiple computer languages have been used in robotics. Imperative languages, such as C++ and declarative languages like Haskell (Peterson *et al.* 1999) and FROB (Hager and Peterson 1999), have been used for robot control. RoboML (Makatchev and Tso 2000), supported on XML technology, is an example of a language explicitly designed for HRI, accounting for low complexity programming, communications and knowledge representation.

The ambiguities common in human–human interactions amount to say that different language constructs (labeled by symbols) can be interpreted equivalently, that is, as synonyms and hence symbols may share a weak relation with their corresponding meanings. Therefore, a key feature of an HRI language must be the ability to cope with semantics so that language differences can be smoothed out before commands are sent to the robot motion controller. Standard computer languages tackle this issue using several syntactic constructs to define general equivalence classes among symbols.

The paradigm presented in the paper includes three classes of objects: motion primitives, operators on the set of motion primitives, and decision making systems (also in the set of motion primitives). Each of these objects establishes a weak relation with the meaning that is assigned to it by the human operator while interacting with the robots. This set of objects forms a semiotic system of signs (see for instance Malcolm and Goguen (1998) for a definition of semiotic system) aiming at accounting, at least to a limited extent, for semantic relations in the set of symbols exchanged in HRI.

3. An architecture for SAR

A conceptual control architecture is mostly independent of the autonomy level of the robot, i.e., both SAR and fully autonomous robots (FAR) share most of the architectural building blocks. The major difference lies in the increased level of sophistication in the HRI that is required by SAR control. Some of the objects used as semiotic symbols in this section have been introduced in Sequeira and Ribeiro (2003) in the context of fully autonomous cooperative robotics§.

From a purposeful robotics perspective, the HRI vocabulary must include symbols related to motion, namely motion commands, and to the robot state querying. These form the core of the language used by

---

†This classification goes back to the works of C.S. Pierce. For the sake of simplicity, only recent works are cited on semiotics.

‡Given two mathematical objects a, b, the relationship, $a \Rightarrow b$ means that b is inferred from a.

§For the sake of readability, slight changes on the terminology used in Sequeira and Ribeiro (2000) have been made.

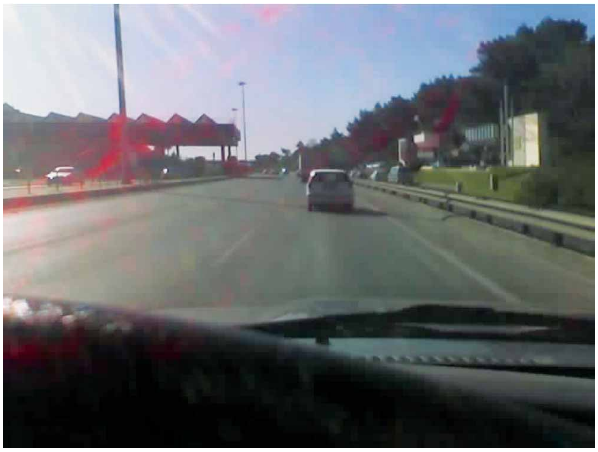


Figure 1. Typical scenarios in car overtaking.

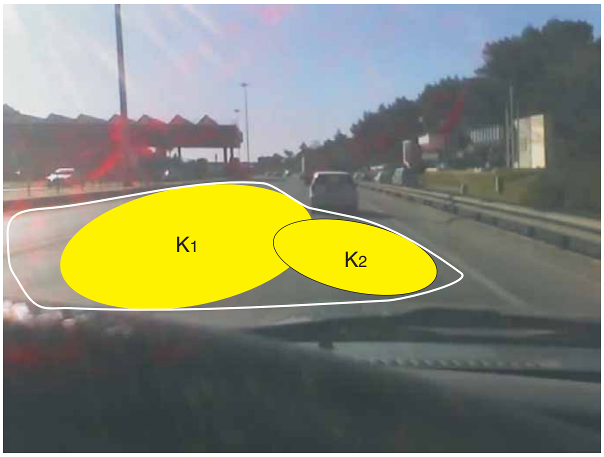


Figure 2. Semiotic symbol definition in a car overtaking scenario.

humans and robots to interact with each other. Figure 1 illustrates a semi-autonomous control scenario where the cars are identified with the robots and the humans operating the SARs are identified with the drivers of the cars. Multiple approaches to automatic driving have been proposed, most of them relying in scene interpretation through image feature extraction (Bertozzi *et al*. 2000, Gray 2000).

Driving a car is an example of a control activity which can be specified in terms of a finite number of symbols (at least for most situations of safe driving). These can be, for example, 'sharp/medium/soft turn', 'move straight' or 'overtake by left/right'. In addition, qualifiers such as 'low/high speed' may be used as complements to the main symbols. Each of these symbols is intrinsically related with a region in the road where the car must stay for a safe drive, i.e., where the trajectories of the robot are allowed to stay. Furthermore, the driving activity implicitly assumes that each of these regions contains a goal region and thus the car is driven inside the allowed region towards this goal region. Driving is thus an example of a task for which it is easy to identify a number of situations which have associated specific meanings. By labeling these situations one obtains a set of semiotic symbols that can be used to control the car.

Following the previous example of car driving, for a wide class of missions, the symbols used by the human operator for SAR control can thus be summarized into three classes: (i) motion strategies, (ii) regions bounding the admissible paths to be followed by the robot, and (iii) goal regions towards which the robot must be driven.

Figure 2 illustrates a possible definition of semiotic symbols in the scenario of figure 1. In the context of this mission, a driver (the human component in this SAR system) perceives the interior of the region bounded by the white line as a free region that the car can use in the overtaking maneuver. The figure illustrates two possible convex subregions, entirely contained inside the free region that can be used as mission goal regions. Therefore, two classes of objects have been identified in the image: (i) a compact region where the trajectories of the robot can evolve, i.e., a bounding region for the trajectories, and (ii) compact goal regions to attract the robot through the bounding region previously defined. These objects are semiotic signs used by the driver to control the car. It is noteworthy that even though the shapes of the free and goal regions are time varying, the meaning associated with the symbols stays constant. This is an example of the aforementioned weak relation between a label and a meaning. The successful execution of this mission (overtaking of a car by another car) depends on the ability of the driver to extract the semiotic symbols from the perceived image, process them and send to the car the semiotic symbols related to the adequate motion strategies.

The remainder of the paper formally defines a two-layer control architecture using the ideas previously outlined. The lower layer contains the objects that handle the motion of the robot and a set of operators on this set of objects. The upper layer contains the decision mechanism and the HRI interface. This interface is simply a parser for a language supported on the semiotic objects. These objects do not require sophisticated identification techniques and hence are appealing to the development of an HRI language to cope with both knowleadgeable and non-knowledgeable agents controlling an SAR. For example, for the scenario in figure 2, one can distinguish a two stage perception: (i) the extraction of a free region (the region bounded by the white line) whereto the driver can steer the car, and (ii) the focus of the driver's attention on a part of the region to define a single goal region where to steer the car, e.g., the K_i sets (shown as oval shapes).

Within the proposed paradigm, indicating a goal region, as opposed to indicating a specific trajectory, implicitly specifies an intention of motion. This intention is mapped into controls by other objects without the intervention of the human operating the robot. Multiple architectures have similar ideas implicit. For example, the subsumption architecture, Brooks (1986), and related paradigms aimed also at simplifying robot control by providing a user friendly programming paradigm, though not directly targeting HRI.

3.1 *Basis objects in the architecture*

The architecture is first developed in conceptual terms by defining the main building blocks as free objects, that is, only implementation independent properties are defined. Next, the objects implementing each building block are defined.

The first object defined in the conceptual architecture is related to the motion of the robot. This object aims at generating paths for the robot to follow and execute its mission.

Definition 1 (Free action)**:** Let k be a time index, q_0 the configuration of a robot where the action starts to be applied and $a(q_0)|_k$ the configuration at time k of a path generated by action a.

A free action is defined by a triple (q_0, a, B_a) where B_a is a compact set and the initial condition of the action, q_0, verifies,

$$q_0 \in B_a, \tag{1}$$

$$a(q_0)|_0 = q_0, \tag{2}$$

$$\exists_{\epsilon>\epsilon_{\min}}: \mathcal{B}(q_0, \epsilon) \subseteq B_a, \tag{3}$$

with $\mathcal{B}(q_0, \epsilon)$ a ball of radius ϵ centered at q_0, and

$$\forall_{k\geq 0} a(q_0)|_k \in B_a. \tag{4}$$

Definition 1 creates an object, the action, able to enclose different paths with similar (in a wide sense) objectives. Paths that can be considered semantically equivalent, for instance, because they lead to a successful execution of a mission, may be enclosed within a single action. When referring to an action during the interaction with teammates, an agent simultaneously refers to all the paths enclosed therein as they are, in a sense to be defined ahead, equivalent. This tends to simplify the interactions among the agents.

Often, the shape of the bounding region B_a is independent of the configuration the robot is in. In figure 2 the light areas K_1 and K_2 are, roughly, independent of the specific position the cars occupy in the road. If, at some configuration q_0, the driver of the car with the inboard camera decides to start an overtaking maneuver the first step is to extract a free region, aiming at keeping the trajectory of the vehicle inside. The vehicle is then driven towards a goal region inside the chosen free space.

Bounding regions of the type shown in the image can be obtained either through rough image processing techniques, e.g., extracting regions in a specific color range followed by a smoothing of the boundaries or by using sophisticated techniques such as snake fitting. The degree of sophistication depends on the requirements of the application.

Semantics, the semiotic perspective considered in this paper, encompasses the meaning extraction from a symbol and matching against some reference. Depending on the agents, identical symbols may yield different meanings and different symbols may yield identical meanings. In both situations, once the agent acquires a symbol, it performs either explicitly or implicitly an equivalence test, for example trying to check whether or not the symbol belongs to the known vocabulary. A key concept to model such semantic relationships between symbols, namely actions, is thus equivalence.

The equivalence concept in the space of actions is defined to account for the motion generated, i.e., equivalent actions lead to similar executions for the same mission. Given the properties in Definition 1, equivalent actions generate trajectories (i) contained inside the same bounding sets and (ii) evolving according to a temporal similarity.

Definition 2 (Free action equality)**:** Two actions (a_1, B_{a_1}, q_{0_1}) and (a_2, B_{a_2}, q_{0_2}) are equal, the relation being represented by $a_1(q_{0_1}) = a_2(q_{0_2})$, if and only if the following conditions hold

$$a_1(q_{0_1}), a_2(q_{0_2}) \subset B_{a_1} \cap B_{a_2} \tag{5}$$

$$\forall_{k_2\geq 0}, \exists_{k_1\geq 0}, \exists_{\epsilon}: a_1(q_{0_1})|_{k_1} \in \mathcal{B}(a_2(q_{0_2})|_{k_2}, \epsilon) \subset B_{a_1} \cap B_{a_2} \tag{6}$$

Condition (5) indicates that any path generated by each of the actions will be contained entirely inside the region common to both actions. This corresponds to a spatial similarity in the sense that both actions span the same region of space. Condition (6) indicates that any two paths generated by each of the actions must exhibit a temporal similarity in the sense that they start in a close neighborhood of each other and that each point of one of the paths lies in a close neighborhood of some point in the other path†.

The equality relation in Definition 2 can be converted into an operator that performs a basic form of inference to be used when extracting the meaning of a symbol. For example, if one agent sends an action (i.e., the semiotic symbol representing the action) to another agent, e.g., indicating that it is executing the action, then the receiving agent may try to extract its meaning by testing the equality with other known symbols such that it can decide on its own motion strategy. In this sense, two actions are semantically equal if and only if they are equal in the sense of Definition 2, meaning that although they are defined differently they both lead to a similar execution of the same mission.

In this paper the realization of a free action verifying Definition 1 is given by the following proposition.

Proposition 1 (Action): *Let $a(q_0)$ be a free action. The paths generated by $a(q_0)$ are solutions of a system in the following form,*

$$\dot{q} \in F_a(q) \tag{7}$$

where F_a is a Lipschitzian set-valued map (see Appendix 1) with closed convex values verifying,

$$F_a(q) \subseteq T_{B_a}(q) \tag{8}$$

where $T_{B_a}(q)$ is the contingent cone to B_a at q (see Appendix 2 for the definition of this cone).

The demonstration of this proposition is just a re-statement, in the context of this paper, of Theorem 5.6 in Smirnov (2002) on the existence of invariant sets for the inclusion (7).

The convexity of the values of the F_a map must be accounted for when specifying an action. The Lipschitz condition imposes bounds on the growing of the values of the F_a map. In practical applications, this assumption can always be verified by a proper choice of the map. This condition is related to the existence of solutions to (7), namely as it implies upper semi-continuity (see Smirnov (2002), Proposition 2.4).

Figures 3 and 4 illustrate the paths generated by two examples of robot actions, of intuitive meaning, in a 2D configuration space. These are labeled 'move to goal' and 'turn right', respectively.

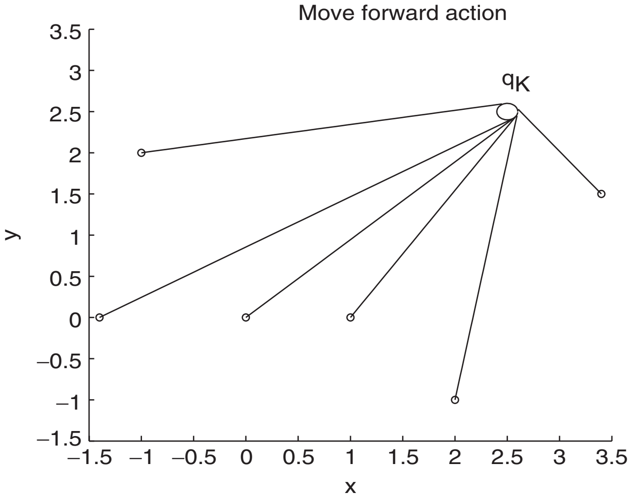


Figure 3. Trajectory generated by the 'move to goal' action.

The first action, 'move to goal' generates straight line paths inside the cone defined by the initial action condition, the current configuration q, and a goal region defined by a circle of radius $\epsilon = 0.1$ centered in point q_K. This action is defined by the inclusion,

$$\dot{q} \in \mathrm{Rot}_\theta\left(\frac{q_K - q}{\|q_K - q\|}\right),$$

with θ a parameter given by

$$0 \leq \theta \leq \sin^{-1}\left(\frac{\epsilon}{\|q_K - q\|}\right), \tag{9}$$

where $\mathrm{Rot}_\theta(x)$ stands for a rotation of θ rad applied to point x.

For this simple bounding region the contingent cone to B at q is given by

$$T_B(q) = \begin{cases} Q & \text{if } q \in \mathrm{Interior}(B), \\ \mathrm{Rot}_\theta\left(\frac{q_K - q}{\|q_K - q\|}\right) & \text{if } q \text{ is the vertex}, \\ \mathrm{Rot}_\theta\left(\pm\frac{q_K - q}{\|q_K - q\|}\right) & \text{if } q \text{ is in the boundary} \\ & \text{of the cone, with} \\ & \theta = \pm\sin^{-1}\left(\frac{\epsilon}{\|q_K - q\|}\right), \\ \text{undefined} & \text{otherwise} \end{cases} \tag{10}$$

and hence the right-hand side of (9) verifies the condition (8). The computation of this cone is made directly

---

†For the sake of simplicity actions will often be denoted without explicit reference to their corresponding bounding sets.

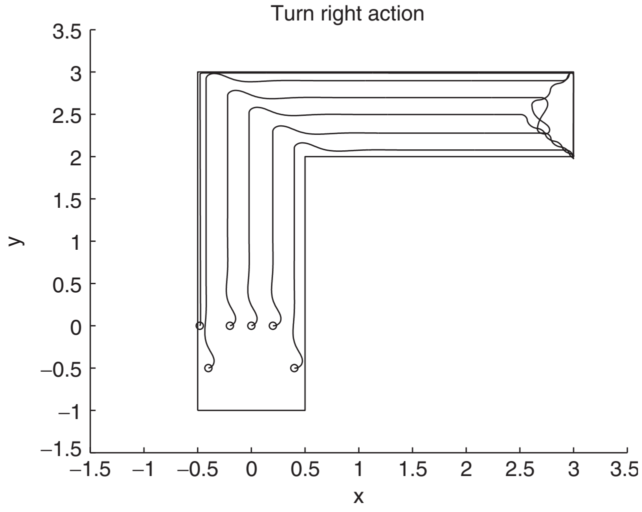


Figure 4. Trajectory generated by the 'turn right' action.

after the definition of contigent cone in Appendix 2. Points in the boundary of the bounding region yield motion directions that keep the trajectory over the boundary. Points in the interior of the bounding region yield the entire space of motion directions as any of them will keep the robot inside. Points in the outside of the bounding region do not yield any motion direction as none will keep the trajectory inside the bounding region.

Figure 3 clearly shows the straight line paths generated for different initial positions (marked with the symbol ○). Note that each path coincides with one of the segments of the cone boundary.

This simple action is an example where the initial configuration is not kept fixed at the configuration the action started being executed. Instead, the bounding region is computed at each instant. However, note that the map $F(q)$ on the right-hand side of (9) preserves the convexity and Lipschitz properties required by Proposition 1.

The choice of the specific motion direction verifying (8) can be made either explicitly, as in (9) where a goal region (the ball centered at q_K) is explicitly incorporated in the inclusion, or implicitly, assuming *a posteriori* choice of the specific motion direction the generated path will take.

The second action, 'turn right', generates paths lying inside the compact polygonal (L-shaped) region shown in figure 4. The motion strategy used by the action is given by the inclusion,

$$\dot{q} \in \begin{bmatrix} \cos(\theta)v \\ \sin(\theta)v \end{bmatrix} \tag{11}$$

where v stands for a positive constant and θ for a parameter given below. The specific motion direction in the right-hand side of (11) is chosen to give a smooth convergence between the robot current velocity vector and the orientation of the tangent to the boundary of the bounding region, taken at the closest point to q, denoted by $\pi_{Q\backslash B}(q)$, (when the robot is inside the bounding region). Defining the error,

$$e(q) = \min\big(\arg\big(T_B(\pi_{Q\backslash B}(q))\big) - \arg(\dot{q})\big), \tag{12}$$

the parameter θ is given by

$$\theta = \arg(e(q)) \quad \text{if } q \in \text{Boundary}(B) \tag{13}$$

otherwise, it is computed from

$$\dot{\theta} = K_p e(q) + K_i \int_{t_0}^{t} e(q) \quad \text{with } \theta(0) = 0, \tag{14}$$

where K_p, K_i are positive constants and $\pi_{Q\backslash B}(q)$ stands for the best approximation projection of q onto the boundary of B (see Aubin and Cellina 1984). The integral term sums the errors along the time interval the action is being executed. Expression (13) forces the orientation of the motion direction to be such that the trajectory is kept on the boundary of the bounding region. Expression (14) smoothly changes the orientation until it aligns with the tangent to the boundary of B.

As in the previous example, the contingent cone to B at q can be computed easily using the physical insight (see expression (27) in Appendix 2), yielding

$$T_B(q) = \begin{cases} Q & \text{if } q \in \text{Interior}(B) \\ (\pm 1, 0) & \text{if } \begin{cases} y = 3 & x \in]-0.5, 3[\\ y = 2 & x \in]0.5, 3[\\ y = -1 & x \in]-0.5, 0.5[\end{cases} \\ (0, \pm 1) & \text{if } \begin{cases} x = -0.5 & y \in]-1, 3[\\ x = 0.5 & y \in]-1, 2[\\ x = 3 & y \in]2, 3[\end{cases} \\ \{(1,0),(0,1)\} & \text{if } x = -0.5 \;\; y = -1 \\ \{(1,0),(0,-1)\} & \text{if } x = -0.5 \;\; y = 3 \\ \{(-1,0),(0,-1)\} & \text{if } x = 3 \;\; y = 3 \\ \{(-1,0),(0,1)\} & \text{if } x = 3 \;\; y = 2 \\ \{(1,0),(0,-1)\} & \text{if } x = 0.5 \;\; y = 2 \\ \{(-1,0),(0,1)\} & \text{if } x = 0.5 \;\; y = -1 \\ \text{undefined} & \text{otherwise} \end{cases} \tag{15}$$

If $q \in \text{Interior(B)}$ then (11) clearly verifies (8), independently of the θ and v parameters. Given the limiting constraint (13), condition (8) is always verified. The smoothness of the curve is obtained by using the PI-like dynamics given by (14). If q_0 is far enough from the boundary, the error $e(q)$ converges to 0 as q approaches the boundary of B.

Figure 4 illustrates examples of paths obtained with the "turn right" action for several initial configurations (represented by $\circ$). Each of the curves, generated after (14), aims uniquely at staying inside the bounding region. No goal region is implicitly defined (as done in the "move to goal" action). As a result of the discrete integration step used in the simulations, the boundary of B can be crossed. This crossing represents an example of the events that must be detected by the system. In such a case, the simulation was stopped.

In general, the analytic computation of the contingent cones is either non-trivial or results in cumbersome expressions like (15). Despite this apparent complexity, the interpretation for the contingent cone in Appendix 2 yields a simple algorithm when polygonal lines are used to bound 2D regions.

3.2 *Basis operators*

The objects defined in section 3.1, namely the action, define a new space where missions take place. This new space of actions represents an abstraction of a mission.

In the actions space, a mission is represented by a sequence of actions, each chosen at some event and robot configuration by a supervisor controller. Creating a sequence of actions to successfully execute a mission requires a set of operators to act on the space of actions.

The equality relation of Definition 2 implicitly defines an identity operator. The following proposition supports the implementation of this operator for actions defined as in Proposition 1.

Proposition 2 (Action identity): *Two actions* a_1 *and* a_2, *implemented as in Proposition* 1, *are said equal if*,

$$B_{a_1} = B_{a_2} \tag{16}$$

$$\exists_{k_0}\colon \forall_{k>k_0},\ F_{a_1}(q(k)) = F_{a_2}(q(k)) \tag{17}$$

The demonstration follows from direct verification of the properties in Definition 2.

By assumption, both actions verify the conditions in Proposition 1 and hence their generated paths are contained inside $B_{a_1} \cap B_{a_2}$ which implies that (5) is verified.

Condition (17) states that there are always motion directions that are common to both actions. For example, if any of the actions a_1, a_2 generates paths restricted to $F_{a_1} \cap F_{a_2}$ then condition (6) is verified. When any of the actions generates paths using motion directions outside $F_{a_1} \cap F_{a_2}$ then condition (17) indicates that after time k_0 they will be generated after the same set of motion directions. Both actions generate paths contained inside their common bounding region and hence the generated paths verify (6).

Another important operator is the action composition. The need for this operator arises naturally from the fact that a mission is a sequence of actions.

Definition 3 (Free action composition): Let $a_i(q_{0_i})$ and $a_j(q_{0_j})$ be two free actions. Given a compact set M, the composition $a_{j\circ i}(q_{0_i}) = a_j(q_{0_j}) \circ a_i(q_{0_i})$ verifies,

$$\text{if } B_{a_i} \cap B_{a_j} \neq \emptyset$$

$$a_{j\circ i}(q_{0_i}) \subset B_{a_i} \cup B_{a_j} \tag{18}$$

$$B_{a_i} \cap B_{a_j} \supseteq M \tag{19}$$

otherwise, the composition is undefined.

A composed path contains portions of the paths of each of the two actions joined by a link path. Condition (19) indicates that a minimum amount of overlapping is necessary for a composition to be meaningful. In fact, M is chosen such that it contains sufficient space for the robot to perform any maneuvering. The rationale behind this condition is that it is necessary to ensure a minimal amount of space to account for link paths that may require maneuvering.

The action composition operator of Definition 3 maps actions into actions and hence it can be thought as the operator that generates motion in the space of actions. Within this framework it is possible to define an inverse operator and an identity action such that the space of actions is given the structure of a group of transformations (see Sequeira and Ribeiro 2000 for details). Inverse actions allow for motion reversal, e.g., a corrective action. The identity (or null) action allows the robot to maneuver in a close neighborhood of its current configuration.

When the actions are defined using the differential inclusion model of Proposition 1 the composition operator results from the following proposition.

Proposition 3 (Action composition): *Let* a_i *and* a_j *be two actions defined by the inclusions*

$$\dot{q}_i \in F_i(q_i) \quad \textit{and} \quad \dot{q}_j \in F_j(q_j)$$

with initial conditions q_{0_i} *and* q_{0_j}, *respectively. The action* $a_{j\circ i}(q_{0_i})$ *is generated by* $\dot{q} \in F_{j\circ i}(q)$, *with the* *map $F_{j\circ i}$ given by*

$$F_{j\circ i} = \begin{cases} F_i(q_i) & \text{if } q \ni B_i \backslash M \quad \text{(20a)} \\ F_i(q_i) \cap F_j(q_j) & \text{if } q \in M \quad \text{(20b)} \\ F_j(q_j) & \text{if } q \in B_j \backslash M \quad \text{(20c)} \\ \emptyset & \text{if } B_i \cap B_j = \emptyset \quad \text{(20d)} \end{cases}$$

for some $M \subset B_j \cap B_i$.

Outside M the values of F_i and F_j verify the conditions in Proposition 1. *Whenever $q \in M$ then $F_i(q_i) \cap F_j(q_j) \subset T_{B_j}(q)$.*

The first trunk of the resulting path, given by (20a), corresponds to the path generated by action $a_i(q_{0_i})$ prior to the event that determines the composition. The second trunk, given by (20b), links the paths generated by each of the actions. Note that by imposing that $F_i(q_i) \cap F_j(q_j) \subset T_{B_j}(q_j)$ the link paths can move out of the M region. The third trunk, given by (20c), corresponds to the path generated by action $a_j(q_{0_j})$.

By Proposition 1, each of the trunks is guaranteed to generate a path inside the respective bounding region and hence the overall path verifies (18).

The action composition in Proposition 3 generates actions that resemble each individual action outside the overlapping region. Inside the overlapping area, the link path is built from motion directions common to both actions being composed. The set M defines the events marking the transition between the trunks.

Whenever $F_i(q_i) \cap F_j(q_j) = \emptyset$ it is still possible to generate a link path, provided that M has enough space for maneuvering. The basic idea is to locally enlarge either $F_i(q_i)$ or $F_j(q_j)$. Iterative procedures can be used for this purpose (see Sequeira and Ribeiro (2004b) for details).

The composition of actions requires the overlapping of the bounding regions of the two actions. From Definition 1, a minimum amount of overlapping must be imposed, e.g., to avoid situations in which (20b) is empty. Whenever the overlapping is small an additional operator can be used to expand the overlapping area.

Definition 4 (Free action expansion)**:** Let $a_i(q_{0_i})$ and $a_j(q_{0_j})$ be two actions with initial conditions at q_{0_i} and q_{0_j} respectively. The expansion of action a_i by action a_j, denoted by $a_j(q_{0_j}) \boxtimes a_i(q_{0_i})$, verifies the following properties,

$$B_{j\boxtimes i} = B_j \cup M \cup B_i, \quad \text{with } M \supseteq B_i \cap B_j \tag{21}$$

where M is a compact set representing the expansion area and such that the following property holds

$$\exists_{q_{0_k} \in B_j}\colon a_i(q_{0_i}) = a_j(q_{0_k}) \tag{22}$$

meaning that after having reached a neighborhood of q_k, $a_i(q_i)$ behaves like $a_j(q_j)$.

The expansion operator of Definition 4 can be implemented as follows.

Proposition 4 (Action expansion)**:** *Let a_i and a_j be two actions defined after the inclusions*

$$\dot{q}_i \in F_i(q_i) \quad \textit{and} \quad \dot{q}_j \in F_j(q_j)$$

The expansion $a_{j\boxtimes i}(q_{0_i})$ verifies the following properties

$$\begin{aligned} F_{i\boxtimes j} = F_i \quad & \text{if } q \ni B_i \backslash M && \text{(23a)} \\ F_j \cup F_i \quad & \text{if } q \in B_i \cap B_j \cup M && \text{(23b)} \end{aligned}$$

where $M \supseteq B_i \cap B_j$ is the expansion set chosen large enough such that $F_j \cup F_i$ verifies (8).

Condition (23a) generates paths corresponding to the action $a_i(q_{0_i})$. These paths last until an event, triggered by the crossing of the boundary of M, is detected. This crossing determines an event that expands the overall bounding region by M and the set of paths, by F_j, as expressed by (23b).

Assuming that $F_j \cup F_i \subset T_{B_i \cap B_j \cup M}$, that is, it verifies (8), the complete path is entirely contained inside the expanded bounding region.

M can be defined as the minimum amount of space that is required for the robot to perform any maneuver and hence the ambiguity in Definition 4 and Proposition 4 is removed. Instead of computing *a priori* this minimal M, the expansion operator can be defined as a process by which action a_i converges to action a_j in the sense that $F_i(q_i) \rightarrow F_j(q_j)$ and M is the space spanned by this process.

Additional operators may be defined in the space of actions. For the purpose of the paper, i.e., defining the properties of a set of actions sufficient to design successful missions, action composition and expansion are the necessary and sufficient operators, as demonstrated in the next section.

4. Controllability in the space of actions

A purposeful framework for robot control must preserve the controllability of the system. Otherwise some missions may not be successful. In terms of the framework developed in section 3 this means that a given set of actions, chosen in the adequate sequence, must be enough for the robot to reach a given goal set.

Controllability has been extensively studied in control theory, with well-known results for linear and nonlinear systems. In the last decades, robotics has incorporated the knowledge on nonlinear systems, namely on the controllability of nonholonomic robotics, e.g., the controllability rank condition (see for instance Hermann and Krener 1977). The approach to controllability presented in this paper weakens the usual point-to-point accessibility concept† to a region-to-region accessibility concept, suitable to characterize the motion of a robot in specific conditions of the proposed paradigm.

Within the SAR context, the notion of a successful mission is linked to the notion of controllability. In natural language, an SAR is controllable if, for any assigned mission, there is a sequence of actions, either issued by the human operator or autonomously computed by the robot, such that the mission is successfully executed.

The human element in an SAR system does not introduce additional motion strategies. Instead, it introduces additional supervision (i.e., decision making) capabilities, e.g., choosing adequate motion strategies in contingency scenarios not *a priori* accounted for in the robot autonomous programming.

The SAR controllability can thus be studied regardless of the human in the loop. The supervising role of the human is then to ensure that the adequate sequence of actions is executed and leads to a successful mission.

Controllability is a global property of the system and thus it must be supported on a point-to-point property such as accessibility.

Definition 5 (point-to-point accessiblity)**:** If there exists a path joining two configurations q_1 and q_2 then q_2 is said to be "accessible from" q_1, this relationship being represented as $q_2 A q_1$.

In general, *accessibility* is not an equivalence relation. The reflexive property and transitivity are always verified. However, symmetry often does not hold (Hermann and Krener 1977), thus making accessibility a weak property.

Definition 5 must be adapted to the paradigm developed in the previous sections by replacing the point by a goal region.

Definition 6 (Region-to-region accessibility)**:** Given two configurations q_1, q_2, if there exists a path joining at least a point in a neighborhood $\mathcal{B}(q_1, \epsilon_1)$, for some ϵ_1, and a point in a neighborhood $\mathcal{B}(q_2, \epsilon_2)$, for some ϵ_2, then q_2 is said to be "region accessible from" q_1, the relationship being represented by $q_2 A_{\epsilon_2, \epsilon_1} q_1$.

The width of the neighborhoods in Definition 6, defined by ϵ_1 and ϵ_2, is determined by mission requirements. When these widths tend to zero, this relation tends to the relation used to define the standard controllability of a dynamic system.

Controllability in the context of this paper is thus obtained by extending Definition 6 to cover the whole space. This controllability concept is defined up to the region widths ϵ_1, ϵ_2.

Definition 7 (Controllability)**:** A robot is said to be controllable if $\forall_{q_1, q_2},\ q_2 A_{\epsilon_2, \epsilon_1} q_1$.

The following proposition establishes the relation between actions and controllability.

Proposition 5 (SAR controllability)**:** *Consider the SAR equipped with a set of actions* $a_1, \ldots, a_n$ (*the initial conditions left unspecified*) *forming a group of transformations acting on the space of actions with the group operation given by the action composition of Definition* 3.

Then the SAR is controllable if and only if, for any configuration, there is a wide enough neighborhood, $\mathcal{B}(q, \epsilon)$, *with a covering verifying,*

$$\begin{aligned} &\forall_q, \exists_{\epsilon > \epsilon_{\min}}: \mathcal{B}(q, \epsilon) \subset \cup_{i=1}^{n} B_i \\ &\quad \wedge \forall_{i,j}, \exists_{q_i \in B_i, q_j \in B_j}: q_j A_{\epsilon_j, \epsilon_i} q_i \end{aligned} \tag{24}$$

where $\epsilon_i, \epsilon_j < \epsilon_{\min}$.

Proving that (24) holds for a large enough neighborhood, $\mathcal{B}(q, \epsilon)$, of an arbitrary configuration amounts to prove controllability as the whole configuration space, Q, can be covered by $Q = \cup_{i=1}^{m} \mathcal{B}(q_i, \epsilon_i)$, for q_i selected such that $\mathcal{B}(q_i, \epsilon_i) \cap \mathcal{B}(q_{i+1}, \epsilon_{i+1}) \neq \emptyset,\ i = 1, \ldots, m-1$.

The sufficiency of the proposition follows from Definition 7, as the controllability implies the existence of a covering for the whole space.

As for the necessity, the demonstration follows from checking that, given a group of transformations in the space of actions, it is always possible to find a sequence of actions such that any element in the covering is region accessible from any other element.

Consider a generic covering for the neighborhood $\mathcal{B}(q, \epsilon)$, that is, $\mathcal{B}(q, \epsilon) \subset \cup_{i=1}^{n} B_i$, and assume that each element B_i in this covering is the bounding region of the action a_i.

Furthermore, assume that there is a permutation $\{p_1, \ldots, p_n\}$ of the action indices $i = 1, \ldots, n$ such that

$$B_{p_i} \cap B_{p_{i+1}} \supseteq M, \quad i = 1, \ldots, n-1$$

where M is the minimal set in Definition 3. This does not imply any lack of generality as each B_{p_i} can always be expanded with enough space and still be a covering

---

†Roughly, controllability amounts to have each point of the state space accessible from any other point.

element for $\mathcal{B}(q, \epsilon)$. Therefore, the composition $a_{p_j} \circ \cdots \circ a_{p_i}$ makes any region $\mathcal{B}(q_{p_j}, \epsilon_{p_j}) \supset B_{p_j}$ accessible from a region $\mathcal{B}(q_{p_i}, \epsilon_{p_i}) \supset B_{p_i}$, for some $\epsilon_{p_i}, \epsilon_{p_j} < \epsilon_{\min}$. It remains to prove the symmetry of the relationship.

It is straightforward to define the inverse of an action and a null action, such that a set of actions with the composition in Definition 3 forms a group of transformations (Sequeira and Ribeiro 2000).

Therefore, a region $\mathcal{B}(q_{p_i}, \epsilon_{p_i})$ can be accessed from a region $\mathcal{B}(q_{p_j}, \epsilon_{p_j})$ just by using the inverse transformation. The composition $a_{p_i}^{-1} \circ \cdots \circ a_{p_j}^{-1} \circ a_{p_j} \circ \cdots \circ a_{p_i}(q_0)$ yields the null element of the group of transformations and hence there is a path joining $\mathcal{B}(q_{p_j}, \epsilon_{p_j})$ and $\mathcal{B}(q_{p_i}, \epsilon_{p_i})$.

Inverse actions allow the looping back to any region thus providing the symmetry to the accessibility relationship. Therefore, under the above group assumption, testing the controllability amounts to prove that, for any initial condition q_0, a given set of actions yields a covering of a wide enough neighborhood around q.

For most practical cases, namely 2D robotics, the use of Proposition 5 is straightforward†. For example, a set of actions of the type defined by (9) can easily be defined to yield a controllable SAR. In the case of turn actions, such as the one defined by (11), a dual 'turn left' action must be included in the set (otherwise only the right half of the space is spanned).

The extension of this controllability concept to teams of robots is straightforward. A team is controllable if and only if each of the robot is controllable in the sense of Proposition 5.

5. Supervision – the overall architecture

The supervisory control addressed in this paper amounts to the discrete decision making involved in the choice of adequate motion strategies. Assuming the controllability as defined in section 4, the success of a specific mission depends on the ability of the supervisor controller to generate the adequate sequence of motion strategies. This process is inherently discrete and hence the overall system is of hybrid nature.

5.1 *Controllability of a supervised robot*

In the paradigm developed in the previous sections, the objects are combined, at specific events, through operators. Given a set of actions such that the SAR is controllable, the success of a mission is thus dependent on the supervision scheme, which must enable/disable the adequate events, e.g., the crossing of bounding regions or the detection of some relevant feature on an image. Controllability conditions for discrete event systems are well known (see for instance Cassandras and Lafortune 1999, p. 150) and basically state that any sequence of uncontrolled events cannot disturb the progression of the robot towards its goal. Despite its importance, this result is of a limited use for the purpose of this paper as it requires that a mission be specified prior to verifying the controllability of the supervised robot.

Controllability conditions for multiple classes of hybrid systems has been considered in the literature with some of them focusing on the controllability as a concept independent of the assigned mission. For example, an extension of Chow's theorem on the controllability of analytic nonlinear systems to multiple model dynamic systems (a class of hybrid systems) is presented in Murphey and Burdick (2002). The basic ideas are the modeling of a system through differential inclusions, the extension of the Lie brackets to set-valued maps and the definition of a distribution of vector fields based of this Lie bracket. Analogously to the single model analytic case (see for instance Hermann and Krener 1977), Chow's theorem is then used to test controllability, i.e., if the vector field distribution spans a space of identical dimension as the state space then the system is controllable. For implementation of actions supported in differential inclusions, as in Proposition 1, the aforementioned results on the controllability of a single robot can be used as an alternative to the results in section 4.

5.2 *A conceptual perspective of supervision*

Hybrid architectures have been presented by multiple researchers, Alur *et al.* (1999) and Milutinovic and Lima (2002), with complex behaviors being obtained from the interaction of multiple primitive behaviors supervised using finite state automata and Petri nets as discrete decision mechanism. Figure 5 illustrates the block architecture including the actions, a supervisor block as the discrete decision mechanism and a set of three operators to act on the space of actions.

The supervisor block includes an HRI interface and an additional layer of abstractions built over the objects developed in the previous sections. These abstractions are either actions built upon different configurations of the basic locomotion capabilities, or predefined compositions of actions. For instance, a "move forward fast" is

---

†Recall that, in the framework of systems described by ordinary differential equations, testing controllability is a non-decidable problem (Laumond 1993).

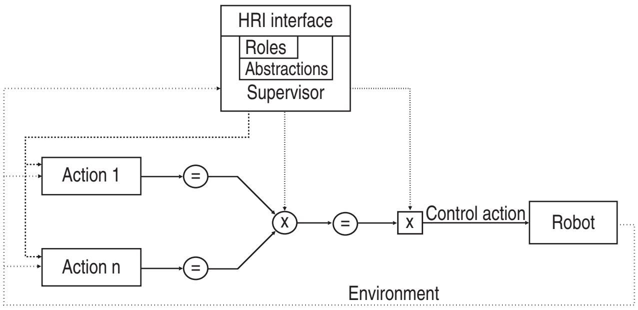


Figure 5. Conceptual model of a semi-autonomous robot in a team.

an abstraction for "move forward with a specific velocity". A set of "move … fast" commands represents a "fast moving robot" abstraction. An "avoid obstacle" action may be an abstraction for "avoid obstacle moving around the right side". Such abstractions tend to simplify both the robot–robot and the human–robot interactions as less information is exchanged between teammates.

A relevant subclass of this type of abstractions is given by roles. A human may interact with a robot under a variety of roles for instance, as a "supervisor", "operator", "mechanic", "peer" or "bystander" Scholtz (2003). Each of these roles grants specific priviledges to the human and configures the robot to react accordingly. In the sense of the equality relation of Definition 2, each role assigns a particular semantic to each of the actions. For instance, a "move forward" action may have different limit velocities whether it is executed under the "operator" or the "mechanic" role.

The HRI implicitly assumes that some form of language is used for the interfacing. Such language can be defined using the semiotic symbols in the architecture: operators and actions. For example, strings such as $a_n \circ \cdots \circ a_1(q_{0_1})$ have a precise meaning within the proposed paradigm and hence can be used as sentences of an HRI language.

Graphical interfaces have been extensively studied in the context of SAR and, more generally, in the context of human–web interfacing (Codognet 1996 and Malcolm and Goguen 1998), as the importance of providing easy interaction through web browsers has been recognized. The HRI language can be simplified up to the exchange of goal regions for the robots to reach, leaving to the supervisor the task of finding the adequate actions that steer the SAR between the current configuration and the goal region specified by the human. This does not imply any lack of generality, as having a user specifying action bounding regions of directly forcing an action composition has the same operational effect as letting the supervisor autonomously take such decisions.

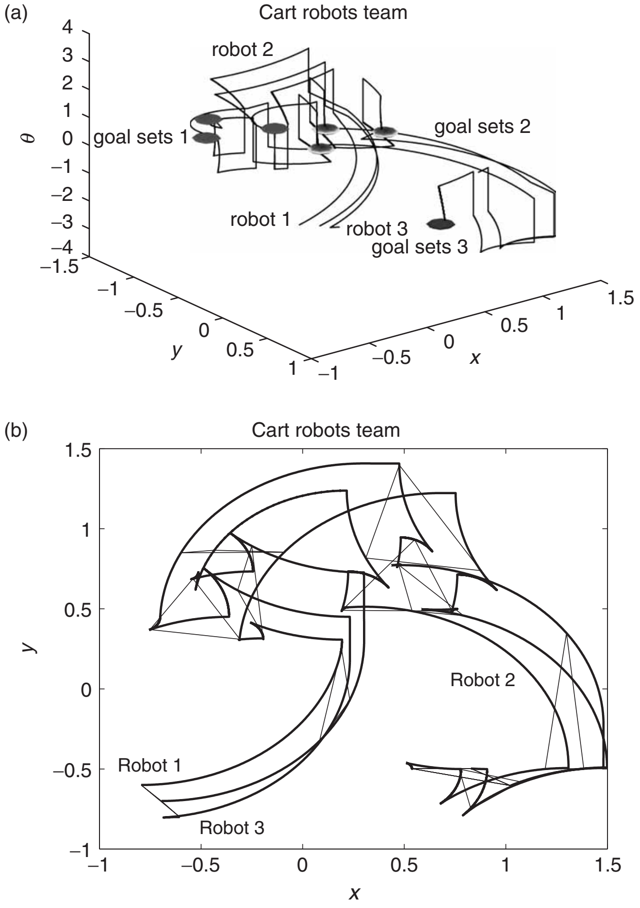


Figure 6. Team of 3 unicycle robots. (a) Trajectories in the configuration space. (b) Trajectories in the xy plane.

6. Practical implementation and results

The framework developed in this paper has been applied to the control of unicycle and car-like robots executing a variety of missions both in simulation and using real robots. The experiments presented are very simple, yet they demonstrate how the agents in a SAR team may interact. Experiments of similar complexity (human following) have been proposed in Nicolescu and Matarić (2001) to demonstrate HRI capabilities.

Proposition (1) implicitly assumes that $F_a(q) \neq \emptyset$, meaning that the generated paths can be followed by the robots with the trajectories staying inside the action bounding region. When considering the dynamics model of the robot this is often not true, e.g., kinematic constraints may render impossible some of the following motion directions specified by an action. If the bounding region has enough space to accommodate the trajectory of the robot during the time when it is converging to the

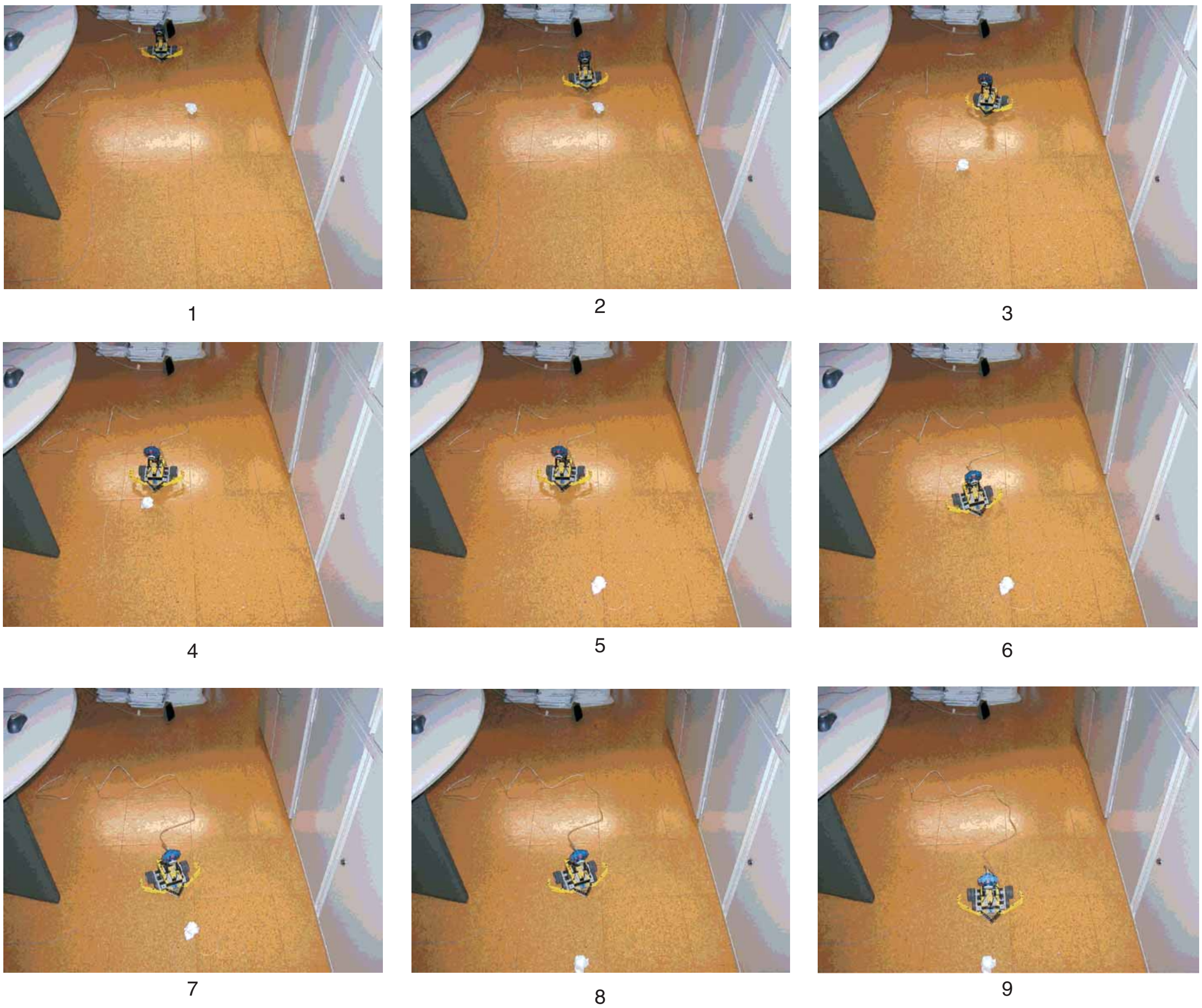


Figure 7. Snapshot sequence of the visual servoing problem.

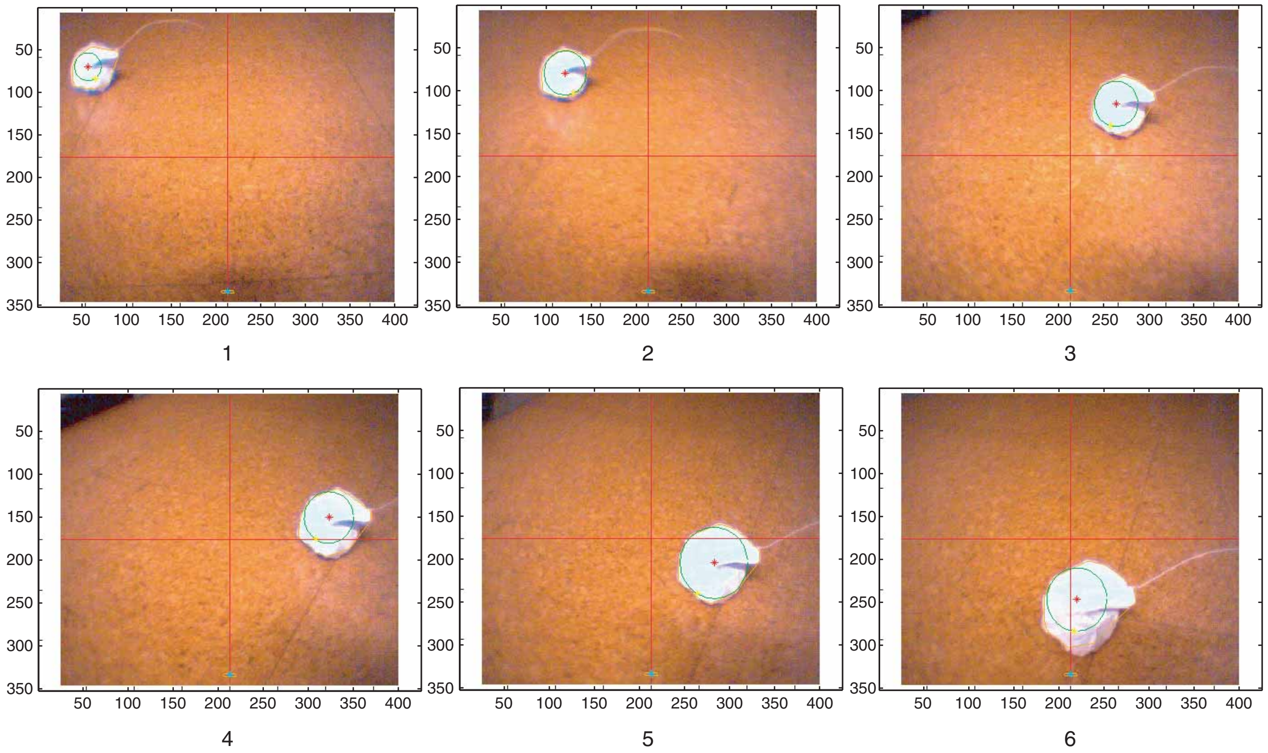
path generated by the action then the framework remains valid. Whenever the space is not enough, it suffices to apply the expansion operator to augment the bounding region. Therefore, the assessment of the framework can be carried out assuming that the bounding regions are always wide enough to accomodate any maneuvering that may be necessary.

In both the experiments presented, the robots have two basis actions, namely *move to goal* and *stop moving*. The practical implementation of the proposed paradigm has been described in several papers. Action expansion is implemented through the convergence process referred in section 3.2. Additional details on this convergence process are given in Sequeira and Ribeiro (2003a, 2003b, 2004a).

The example in figure 6 illustrates the behavior of a team of unicycle robots, each of them trying to reach a sequence of goal sets (shown as the dark regions in figure 6(a) while moving in a loose formation. This formation aims at keeping a minimal distance between the robots while they are moving towards their respective goals. Each supervisor is basically a two-state finite state automation that detects a number of events after the relative positions of the teammates using them to control the state transitions.

The intense maneuvering that results from the interaction between the robots is clearly visible in figure 6(b). The robots do not explicitly communicate among themselves. Instead, they use their motion to implicitly express their intentions. Each supervisor detects events related to the motion of the teammates that can be considered as symbols, with a meaning local to each robot (in some sense this form of communication is similar to that of the car driving example of section 2).

Figure 7 illustrates the behavior of a single robot in a visual servoing application. Vision is a key sensor in SAR missions as images often allow humans a quick



1 2 3 4 5 6

Figure 8. Onboard video images for part of the mission.

perception of the environment the robot is operating in. Furthermore, images provide interesting supports where to extract semiotic symbols such as those used by the proposed paradigm, namely goal regions and bounding regions, and hence provide a good insight on the performance of the paradigm in an SAR context. Using a *move to goal* action similar to (9), the robot chases a white paper ball that is being moved by a human at sparse instants of time. The goal regions are computed directly by the robot from images acquired by an onboard camera using basic color segmentation and filtering procedures. The motion of the paper ball by the human mimics the HRI when the mission goals are specified by pointing to a region in an image displayed in a graphical interface.

Figure 8 shows the snapshot sequence of images used by the robot to extract the goal regions. The goal regions extracted from the images are shown as circles superimposed over the white paper ball. The sequence clearly shows the goal region converging to the bottom center of the image, which corresponds to the position of the robot in the image plane.

7. Conclusions

The paper presented a robot control paradigm based on semiotic concepts. This paradigm is tailored to SAR control as it defines a number of semiotic signs close to those used in human interactions. Nevertheless, the paradigm can also be applied to FAR control.

The paradigm develops into a hybrid architecture including actions, operators on the actions, and a supervisor as building blocks. The set of available actions defines the admissible motion strategies. The supervisor block (i) has decision capabilities on what actions to choose for execution and on the application of the operators, and (ii) interfaces the robot and the human through a language built upon the semiotic symbols defined.

Basic experiments illustrate the behavior of unicycle robots interacting within a team and operating isolatedly only interacting with an external human agent.

Further work includes the study of supervisor design for various degrees of autonomy and its effect on the success of a mission.

Acknowledgments

This work was supported by the FCT project SACOR – POSI/SRI/40999/2001 and Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III.

Appendix 1. Lipschitz set-value maps

A set-valued map F is said to be Lipschitz if it verifies.

$$\exists_{\epsilon\geq 0}\colon \forall_{x_1, x_2 \in X},\ F(x_1) \subset F(x_2) + \epsilon |x_1 - x_2|_X \mathcal{B}_Y \tag{25}$$

where

$$\mathcal{B}_Y = \left\{ y \in Y\colon |y| \leq 1 \right\} \tag{26}$$

where $|\cdot|_X$ stands for a norm in X.

Appendix 2. Contingent cones

Nonsmooth analysis uses tangency concepts for which a variety of contingent cones is defined (see for instance Simirnov (2002)).

The contingent cone used in the paper is defined as

$$T_B(q) = \left\{ v\colon \lim_{h\to 0+} \inf \frac{d_B(q+hv)}{h} = 0 \right\} \tag{27}$$

where

$$d_B(q) = \inf_{p\in B} |p-q|_Q \tag{28}$$

References

J. Albus, "A control system architecture for intelligent systems", in *Procs. of the 1987 IEEE Intl. Conf. on Systems, Man and Cybernetics*, Alexandria, VA, October, 1987, pp. 20–23.

J.S. Albus, "The engineering of mind", in *4th International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 4*, Cope Code, MA, USA, September 1996.

R. Alur, J. Esposito, M. Kim, V. Kumar and I. Lee, "Formal modeling and analysis of hybrid systems: a case study in multirobot coordination", in *FM 99: Procs. of the World Congress on Formal Methods, FM'99*, Springer, LNCS 1708, 1999, pp. 212–232.

J.P. Aubin and A. Cellina, *Differential Inclusions*, Berlin: Springer-Verlag, 1984.

R. Aylett and D. Barnes, "A multi-robot architecture for planetary rovers", in *Procs. of the 5th European Space Agency Workshop on Advanced Space Technologies for Robot. & Autom.*, ESTEC, The Netherlands, December 1998.

T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams", *IEEE Trans. on Robotics and Automation*, 14, pp. 926–939, December 1999.

M. Bertozzi, A. Broggi and A. Fasciolli, "Development and test of an intelligent vehicle prototype", in *Procs. 7th World Congress on Intelligent Transportation Systems*, November 6–9, Torino, Italy, 2000.

E. Bicho and G. Schöner, "The dynamic approach to autonomous robotics demonstrated on a low-level platform", in *Procs. of the 4th Symposium on Intelligent Robotic Systems, SIRS'96*, Lisbon, Portugal, July 1996.

R. Brooks, "A robust layered control system for a mobile robot", *IEEE J. Robot. Autom.*, RA-2(1), pp. 14–23, 1986.

C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, International Series on Discrete Event Systems, Boston: Kluwer Academic, 1999.

P. Codognet, "The semiotics of the web", in *Procs. 2nd Int. Summer School on Semiotics*, September 1996.

P. Codognet, "The semiotics of the web", *Leonardo*, Vol. 35, Cambridge: The MIT Press, 2002.

U. Eco, *Semiotics and the Philosophy of Language*, Bloomington: Indiana University Press, 1984.

Georgia Tech, "Real-time cooperative behavior for tactical mobile robot teams – subsystems specification", Technical Report A002, Georgia Tech College of Computing and Georgia Tech Research Institute, October 1984.

R. Gray, "Risky driving behavior: a consequence of motion adaptation for visually guided motor action", *J. Experiment. Psy.: Human Percep. Perform.*, 26(6), pp. 1721–1732, December 2000.

G.D. Hager and J. Peterson, "Frob: a transformational approach to the design of robot software", in *Procs. of the 9th Int. Symp. of Robotics Research, ISRR'99*, October 9–12, Snowbird, Utah, USA, 1999.

R. Hermann and A. Krener, "Nonlinear controllability and observability", *IEEE Trans. Autom. Cont.*, AC-22(5), pp. 728–740, October 1977.

P. Hudak, "Modular domain specific languages and tools", in *Proceedings of Fifth International Conference on Software Reuse*, June 1998, pp. 134–142.

T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H.D. Nayar, H. Aghazarian, A. Ganino, M. Garrett, S.S. Joshi and P.S. Schenker, "Campout: a control architecture for tightly coupled coordination of multi-robot systems for planetary surface exploration", *IEEE Trans. Systems, Man & Cybernetics, Part A: Systems and Humans*, 33(5), pp. 550–559, 2003, Special Issue on Collective Intelligence.

D. Jung and A. Zelinsky, "Grounded symbolic communication between heterogeneous cooperating agents", *Autonom. Robots J.*, 8(3), pp. 269–292, June 2000, Special issue on Heterogeneous Multi-robot Systems.

S. Kiesler and P. Hinds, "Introduction to the special issue on humanrobot interaction", *Human-Computer Interaction*, 19(1–2), pp. 1–8, 2004.

D. Kortenkamp, R. Burridge, R.P. Bonasso, D. Schrekenghost and M.B. Hudson, "An intelligent software architecture for semiautonomous robot control", in *Procs. 3rd Int. Conf. on Autonomous Agents – Workshop on Autonomy Control Software*, May 1999.

J.P. Laumond, "Singularities and topological aspects in nonholonomic motion planning", in *Non-Holonomic Motion Planning*, Z. Li and J. Canny, Eds, Boston: Kluwer Academic, 1993, pp. 149–199.

M. Makatchev and S.K. Tso, "Human-robot interface using agents communicating in an xml-based markup language", in *Procs. of the IEEE Int. Workshop on Robot and Human Interactive Communication*, September 27–29, Osaka, Japan, 2000.

G. Malcolm and J.A. Goguen, "Signs and representations: semiotics for user interface design", in *Procs. Workshop in Computing*, Springer, Liverpool, UK, 1998, pp. 163–172.

A. Meystel, "Intelligent systems: a semiotic perspective", *International Journal of Intelligent Control and Systems*, 1(1), pp. 31–57, 1996.

A. Meystel and J. Albus, *Intelligent Systems: Architecture, Design, and Control*, New York: Wiley Series on Intelligent Systems. J. Wiley and Sons, 2002.

D. Milutinovic and P. Lima, "Petri net models of robotic tasks", in *Procs. of IEEE Int. Conf. on Robotics and Automation*, Washington D.C., 2002.

T. Murphey and J. Burdick, "A local controllability test for nonlinear multiple model systems", in *Procs. American Control Conference, ACC 2002*, 2002.

M. Neumüller, "Applying computer semiotics to hypertext theory and the world wide web", in *Procs. of the 6th Int. Workshop and the 6th Int. Workshop on Open Hypertext Systems and Structural Computing*, S. Reich and K.M. Anderson, Eds, Lecture Notes in Computer Science, New York: Springer-Verlag, 2000, pp. 57–65.

M. Nicolescu and M. Matarić, "Learning and interacting in humanrobot domains", *IEEE Trans. Sys., Man, and Cybernetics, Part A: Systems and Humans*, 31(5), pp. 419–430, September 2001.

J. Peterson, P. Hudak and C. Elliott, "Lambda in motion: controlling robots with haskell", in *First International Workshop on Practical Aspects of Declarative Languages (PADL)*, January 1999.

G.N Saridis, "On the theory of intelligent machines: a compre- hensive analysis", *Inter. J. Intelligent Cont. Sys.*, 1(1), pp. 3–14, 1996.

P.S. Schenker, T.L Huntsberger, P. Pirjanian and G.T. McKee, "Robotic autonomy for space: cooperative and reconfigurable mobile surface systems", in *Procs. of the 6th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'01)*, June 18–21, Montreal, Canada, 2001.

J. Scholtz, "Theory and evaluation of human robot interactions", in *Procs. of the 36th Hawaii Int. Conf. on System Sciences, HICSS'03*, January 5–8, Hawaii, USA, 2003.

J. Sequeira and M.I. Ribeiro, "Behaviour-based cooperation between two mobile manipulators", *J. Integ. Comp.-Aided Eng.*, 7(3), pp. 193–215, 2000.

J. Sequeira and M.I. Ribeiro, "A geometric approach to single and multiple robot control", in *Preprints of the 7th International IFAC Symposium on Robot Control – Syroco 2003*, September 1–3, Wroclaw, Poland, 2003a, pp. 137–142.

J. Sequeira and M.I. Ribeiro, "Robot team control: a geometric approach", in *Proceedings of the 11th International Conference on Advanced Robotics, ICAR 2003*, Universidade de Coimbra, Portugal, June 30–July 3, 2003b, pp. 1502–1507.

J. Sequeira and M.I. Ribeiro, "Behavior-based control for semiautonomous robots", in *Preprints of the 5th IFAC Symposium on Intelligent Autonomous Vehicles, IAV2004*, July 5–7, Lisbon, Portugal, 2004a.

J. Sequeira and M.I. Ribeiro, "Hybrid control of semi-autonomous robots", in *Procs. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, September 28–October 2, Sendai, Japan, 2004b.

G. Smirnov, *Introduction to the Theory of Differential Inclusions,* volume 41 *of Graduate Studies in Mathematics*, Providence, RI: American Mathematical Society, 2002.

H. Utz, F. Stulp and A. Mühlenfeld, "Sharing belief in teams of heterogeneous robots", in *Procs. RoboCup 2004 International Symposium*, July 4–5, Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, Springer-Verlag, Lisbon, Portugal, 2004.