MODELING ROBOT BEHAVIORS WITH HYBRID AUTOMATA

Nelson Gonçalves [*] **João Sequeira** [*]

[*] *Instituto Superior Técnico,*
Institute for Systems and Robotics, Portugal
email: {ngoncalves,jseq}@isr.ist.utl.pt

Abstract: The problem of designing behaviours for mobile robots employed in an active surveillance system is considered. The mobile robots are modelled using differential inclusions and the behaviors are synthesised from viability sets. A basic hybrid automaton is proposed for the control of all designed behaviors. A composition rule is presented for the mixing of different behaviors. The main advantage of the presented approach is that behaviors are obtained by specifying regions in the environment. This is useful in surveillance, where the location of intruders is not known precisely. Another advantage is that the same framework is used to control each of the available behaviors. Consequently, the behaviors can be used as basic building blocks in general robot control architectures. A set of experimental simulations where conducted to evaluate the proposed design approach and the results are also presented.

Keywords: Hybrid automata, viability, differential inclusions

1. MOTIVATION

In active surveillance setups, mobile robots are required to exhibit different behaviors such as patrolling, intruder interception and also moving in formation. The purpose of this last behavior is to increase the area covered during the patrol behavior and also to reduce the escape routes of the detected intruders.

The approach proposed in this paper for behaviors synthesis is based in constraining the robot state to remain inside a bounded, closed set of the space state. For example, a patrolling behavior can be obtained by constraining the mobile robot position to a closed and narrow region containing the locations to visit. The width of this region is obtained from the tolerance on the distance that the robot must pass from the locations. The lower the tolerance, the narrower the region containing the locations.

The position constraint, *per se*, is not sufficient for the mobile robot to generate a useful trajectory. Consider again the patrolling behavior and assume that initially the mobile robot is positioned inside the narrow region. In this case, the state constraints are trivially satisfied if the robot does not move. In order to avoid similar situations, the desired robot velocity is introduced. Thus the behaviors are designed by specifying state constraints in the form of bounded, closed sets and the desired velocity for when the state constraints are verified.

An hybrid automaton, with two discrete states, is used to control the behavior execution. The discrete states account for the two possible cases in which the state constraints are either verified or not. If the constraints are verified, the control inputs are selected so as to output the desired velocity without violating the constraints. But if the state constraints are not verified, the control inputs are selected to guide the robot towards the region designed for the behavior.

The combination of different behaviors is made possible by the introduction of a composition rule for the hybrid automata. The resulting automaton has the same structure as the two input automata. Therefore, the control architecture is only required to handle one type of hybrid automata.

The remainder of this document is as follows. In Section 3 is presented the mobile robot dynamic model. In Section 4 the behavior synthesis approach is presented. In Section 5 behavior design examples are provided. Finally, in Section 6 conclusions and the future work are presented.

2. RELATED WORK

A wide variety of behavior-based control architectures have been proposed and an exhaustive survey is beyond the scope of this paper. In general, different of methods are used to design behaviors such as fuzzy logic in (Innocenti *et al.*, 2006), neural networks in (Wang *et al.*, 2007) and Lyapunov control in (Freire *et al.*, 2004). In these control architectures, the mobile robot behavior is an weighted average of the output of each individual behavior. The mixing weights are determined as a function of the sensory data and the robot goals.

In contrast with these architectures, no weights are used to combine behaviors in the presented approach. As a result, no tuning of parameters or learning phases are required previous to using the mobile robots. The behavior design approach used is also different, where state constraints are used instead of set-point regulation controllers. Thus, the mobile robots have some degree of freedom to adapt their state to the local surroundings.

The composition of hybrid systems has been previously proposed in (Tabuada *et al.*, 2001), for example. These composition procedures are aimed at building complex systems from simpler hybrid dynamic models. The composition procedure presented in this paper is used to obtain a new behavior from the available mobile robot behaviors. Thus, the same hybrid automaton structure is obtained after the composition procedure.

3. MOBILE ROBOTS MODEL

The mobile robots considered are those for which their kinematic constraints can be modeled by the differential equation

$$\dot{q} = f(q, u) \tag{1}$$

where $q \in \mathbb{R}^n$ is the robot state, $u \in \mathbb{R}^m$ the control input and f is a Lipschitz continuous function. Without loss of generality, in this paper the robot state is identified with the robot body pose in the world frame. Also, it is assumed that the robot is able to stop

$$\forall q,\ \exists u^* \Rightarrow f(q, u^*) = 0\ \wedge \|u^*\| < k \tag{2}$$

where k is a finite, positive constant. This model can be used to model common wheeled robots, such as the unicycle, and vehicles with limited minimum turning radius.

In practice, the robot state evolution is constrained by the limited control inputs available. A differential inclusion (DI), can be used to model the availability of limited control inputs

$$\dot{q} \in F(q, U) = \{p \in \mathbb{R}^n, u \in U \mid p = f(q, u)\} \tag{3}$$

where F is a set-valued map and U is a compact subset of $\mathbb{R}^m$. Since f is a Lipschitz continuous function, from Proposition 2.4 in (Smirnov, 2001) it can be asserted that F is a Lipschitz continuous set-valued map. Furthermore, a strictly positive Lipschitz constant is assumed and the output values of F are convex, closed and contained in a ball centered at the origin. The differential inclusions with such set-valued maps are denoted *Lipschitzian*.

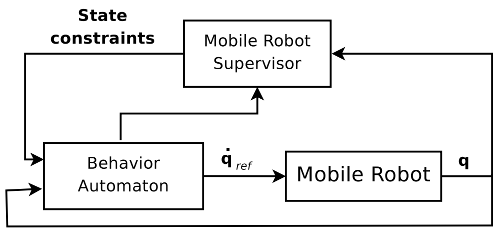

Fig. 1. Robot control architecture

In practice, the Lipschitzian assumptions may not hold for some mobile robots kinematic configurations. For these robots, the control inputs can be further constrained to a subset of U in order to verify the Lipschitzian assumptions on the set-valued map F.

Definition 1. (DI solution set, (Smirnov, 2001)). Consider the Lipschitzian differential inclusion

$$\dot{x}(t) \in H(x(t)), \quad t \in [0, T] \tag{4}$$

with initial value $x(0) = x_0$. The solution to the differential inclusion is the set of absolutely continuous functions $y(t)$ that verify the inclusion (4) and $y(0) = x_0$. This set of functions is represented by $\mathcal{S}_{[0,T]}(H, x_0)$.

The solution to the mobile robot differential inclusion (3) can be interpreted as the set of trajectories it can execute, given the constraints on the control inputs.

The mobile robot control architecture used is presented in Figure 1. The supervisor is responsible for computing the state constraints for the robot, along with a desired velocity. The behavior automaton is used to generate velocity reference signal for the mobile robot controller. The reference is the closest possible, in euclidean distance, to the desired velocity received from the supervisor, without violating the state constraints.

4. BEHAVIOR SYNTHESIS

A mobile robot behavior is identified, in this paper, with a subset $B \subset \mathcal{S}_{[0,T]}(F, q_0)$. The trajectories in the subset B of each behavior are those that verify a viability condition.

Definition 2. (Viable trajectories). Consider the piecewise constant, set-valued map $C : [0, \infty) \rightarrow \mathbb{R}^n$, with closed, convex output values. The output set during the time interval $\tau = [t_0, t_1]$, is represented by C_τ. A trajectory $x(t) \in \mathcal{S}_{[t_0,t_1]}(H, x_0)$ is said to be viable in C_τ *iff*

$$x(t) \in C_\tau, \quad \forall t \in [t_0, t_1]$$

Proposition 1. If initially robot state is viable, $q(t_0) \in C_\tau$, then the behavior subset B is not empty.

Proof Assume that initially, the robot state is in the interior of the viable set, $q(t_0) \in int\, C_\tau$. Consider a ball with radius ϵ, centered on the robot state, $O_\epsilon(q(t_0))$, completely contained in the viable set. The first order Taylor expansion of any solution $q(t) \in \mathcal{S}_{[t_0,t_1]}(F, q_0)$ at instant $t = t_0 + \delta$ is

$$q(t_0 + \delta) \simeq q(t_0) + f(x(t_0), u(t_0))\delta + O(n^2)\ldots$$

Consider the control input $u(t)$ in the form

$$u(t) = \begin{cases} u_0 \in U & t = t_0 \\ u^* \in U & t > t_0 \end{cases}$$

where u_0 is any bounded control input such that $f(q(t_0), u_0)$ is not null. Then using the triangle inequality, select a small enough δ such that the robot state is still viable:

$$\|q(t_0 + \delta) - q(t_0)\| < \epsilon$$

$$\Leftrightarrow \delta < \frac{\epsilon}{\|f(q(t_0), u_0)\|}$$

For case when the initial robot state is on the border of the viable set, $q(t_0) \in bdC_\tau$, then at least the trivial solution is viable, $q(t) = q(t_0)$ for all $t \in [t_0,\, t_1]$.

□

The viability condition is thus used to specify the robot state constraints. In Section 5 are presented viable sets for the obstacle avoidance and moving in formation behaviors. An important limitation of this approach is that only the trivial trajectory may be viable for a given set C_τ. A possible solution is to have the supervisor in the control architecture compute a new viable set C'_τ, such that $C_\tau \cap C'_\tau \neq \emptyset$ and $q(t) \in intC'_\tau$. In this manner more viable trajectories are available to the robot.

Although a robot behavior is a class of trajectories, in general some individual trajectories may be preferred over others. For instance, in the patrol behavior the mobile robot should visit the locations within a limited time period. Thus viable, patrol trajectories with higher velocity modulus are preferred in this behavior. This preference over trajectories is expressed with the introduction of a velocity reference signal, $\dot{q}_{ref}$, represented by

$$\dot{q}_{ref}(t) = \pi(p(t), F(q(t), U)) \quad (5)$$

where $p(t)$ is a Lipschitz continuous function and $\pi(.)$ is the euclidean projection of a point on a set. Because of the different kinematic and control input constraints, the same function $p(t)$ may result in different velocity reference signals. The function $p(t)$ is denoted a *velocity profile*.

It is clear that exist initial viable states and velocity profiles such that the mobile robot state will exit the viable set in finite time. And also that for any given set C_τ and time interval $[t_0,\, t_1]$, initially $q(t_0)$ can only be either viable or not. If it is viable, then the control architecture should constrain the robot state to remain inside C_τ. This may be at the cost of modifying the velocity reference signal. But if $q(t_0)$ is not viable, then the controller should be such that the robot state tends towards C_τ. Thus, two different controllers are required to produce a desired behavior. These are denoted as the (i) *behavior-active* and (ii) *behavior-inactive* controllers, respectively.

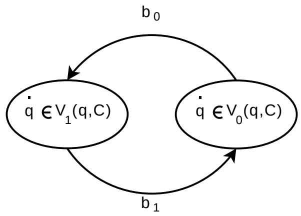

Fig. 2. Behavior hybrid automaton

The *behavior-active* controller is used when the robot state is initially inside the viable set, $q(t_0) \in C_\tau$. In this situation, the set of possible directions of motion for the robot should be constrained by $V_0(q,\, C_\tau)$:

$$\dot{q} \in V_0(q, C_\tau) \subseteq F(q, U) \Rightarrow q(t) \in C_\tau,\ t \in [t_1,\, t_2] \quad (6)$$

This controller can be interpreted as altering the robot set of possible control inputs in a manner such that the state is always viable. As a result, original the velocity reference signal in (5) is modified accordingly.

When the robot state is not initially viable, then it should move towards the viable set. Therefore, the *behavior-inactive* controller must constrain the robot state trajectories to those that make q enter C_τ. For this case, the robot directions of motion should be confined by the set-valued map $V_1(q,\, C_\tau)$:

$$\dot{q} \in V_1(q, C_\tau) \subseteq F(q, U) \Rightarrow \exists t' > t_0 : q(t') \in C_\tau \quad (7)$$

Thus, with the *behavior-inactive* controller the robot state must reach the viable set C_τ in finite time.

Because two different control decisions are required to produce a behavior, the robot controller is modeled with the hybrid automaton

$$\Gamma(C_\tau, p) = (C_\tau, p, \{V_0, V_1\}, q, \{z_0, z_1\}, \{b_0, b_1\}) \quad (8)$$

where the discrete state z_0 is used to indicate that q is viable, while z_1 signals that q is outside the viable set. This automaton is denoted the *behavior automaton* and is represented in Figure 2.

Definition 3. (Mobile robot behavior). It is the trajectory generated by the robot for the viable set C_τ, velocity profile $p(t)$ and the behavior automaton Γ. A mobile robot behavior is represented by the tuple $\langle C_\tau,\, p(t),\, \Gamma \rangle$.

4.1 *Behavior-Active Controller*

If initially $q(t_0) \in C_\tau$, then the motion of the robot can be confined to remain inside C_τ. The idea is to restrict $\dot{q}$ to the motion directions such that $q(t)$ will remain inside C_τ. These directions are computed using the notion of tangent cone.

Definition 4. (Tangent cone) Consider the point $x \in A$, where A is a closed, convex subset of $\mathbb{R}^n$. Let $d(w, A)$

be the euclidean distance between point w and set A. The tangent cone of A at point x is the set defined as

$$T(x, A) = \{y \in \mathbb{R}^n | \lim_{\theta \to 0^+} \frac{d(x + y\theta, A)}{\theta} = 0\}$$

Intuitively, the tangent cone $T(x, A)$ is the set of all directions $y \in \mathbb{R}^n$ such that starting at x and moving along y an infinitesimal, one is guaranteed to remain inside A. If x is an interior point of A then all motion directions are possible. But if x is a point in the border of A only the directions that point inward the set A are part of the tangent cone. Thus, the tangent cone can be used to implement the viability condition on the state trajectories.

The proposed set-valued map for the behavior-active controller is

$$V_0(q, C_\tau) = T(\tilde{q}, C_\tau) \cap F(\tilde{q}, U) \tag{9}$$

where $\tilde{q} = \pi(q, C_\tau)$.

Proposition 2. If initially the mobile robot state is viable, $q(t_0) \in C_\tau$, then all solutions of the differential inclusion $\dot{q} \in V_0(q, C_\tau)$ are viable.

Proof Consider the set valued map $W : C_\tau \to \mathbb{R}^n$ defined as $W(q) = T(q, C_\tau) \cap F(q, U)$, By construction, this set valued-map has closed convex output values and is upper semi-continuous. Also, we have that $W(q) \subset T(q, C_\tau)$. Set $V_0(q, C_\tau) = W(\tilde{q})$, where $\tilde{q} = \pi(q, C_\tau)$. The result follows from the application of the Theorem 5.7 in (Smirnov, 2001).

□

4.2 *Behavior-Inactive Controller*

If $q(t_0)$ is not initially inside $C(t)$ then, the robot state $q(t)$ should move towards C_τ. The synthesis of the behavior-inactive controller is formulated as a posture stabilization problem. It is assumed that the only constraint to the robot motion is the kinematic configuration. Thus, the behavior-inactive set-valued map is

$$V_1(q, C_\tau) = F(q, U) \tag{10}$$

Definition 5. (Weak stability, (Smirnov, 2001)). Consider the differential inclusion $\dot{x} \in H(x)$. The equilibrium point $x = 0$ is weakly stable if

$$\forall x \in \mathcal{O}_\eta(0),\ \exists l \in H(x)\ :\ D^- L(x)(l) \leq 0$$

where $\mathcal{O}_\eta(0)$ is a ball centered on the origin with radius η, L is a Lyapunov function candidate and D^- the lower Dini derivative. The equilibrium is weakly, asymptotically stable if the inequality is strict.

Thus, the stabilization problem is to determine a suitable direction of motion such that the robot state asymptotically converges to a point inside the viable set. The solution to the stabilization problem is, in general, specific to the kinematic configuration of each robot. For the unicycle robot, the direction of motion can be determined using a switching control law.

Let the unicycle robot position and orientation be, respectively, $\overline{q} = [q_x\ q_y]'$ and q_θ. The unicycle kinematics can be described, for a suitable choice of coordinate frames, by the differential inclusion

$$\dot{q} \in [cos(q_\theta)U_v\ sin(q_\theta)U_v\ U_w]' \tag{11}$$

where the real-valued intervals U_v and U_w form the control input set, $U = U_v \times U_w$. These intervals are in the form

$$U_v = [0,\ V_{max}],\ U_w = \frac{1}{2}[-1,\ 1]W_{max} \tag{12}$$

with V_{max} and W_{max} the maximum linear and angular velocities of the robot. These values are determined such that the motor control inputs of the robot do not saturate. Let the center of the convex set C_τ be the point $\overline{c}_\tau$, $\rho = \overline{c}_\tau - \overline{q}$ be the position error vector, β the orientation of the error vector with the world x axis and $\alpha = \beta - q_\theta$. Consider the Lyapunov candidate function

$$L(q) = \frac{1}{2}\left[\alpha^2 + \rho' K \rho\right] \tag{13}$$

with K a positive definite matrix, and the direction vector

$$l = \begin{cases} W_{max}\, tanh(\alpha)[0\ 0\ 1]', & if\ |\alpha| \neq 0 \\ V_{max}\, tanh(\|\rho\|)[\frac{\rho}{\|\rho\|}\ 0]', & otherwise \end{cases} \tag{14}$$

It is clear that for any initial, non viable robot state $q(t_0)$, the condition $l \in V_1(q(t_0), U)$ is verified. The Dini derivative of the Lyapunov function along this direction is

$$D^- L(q)(l) = \begin{cases} -\alpha\, W_{max}\, tanh(\alpha),\ |\alpha| \neq 0 \\ -\frac{V_{max}\, tanh(\|\rho\|))}{\|\rho\|}\rho' K \rho,\ otherwise \end{cases} \tag{15}$$

Thus, the direction vector l is first used to orientate the robot towards the center point of the viable set. When it is pointed, the robot proceeds along the direction given by the position error vector until reaching the center point $\overline{c}_\tau$.

4.3 *Composition of Behavior Automata*

The behavior of a robot is designed, in this work, using a velocity profile and a viable set. But in practical situations, the specifications that produce a desired behavior may be hard to determine. Also, the design of new behaviors should take advantage of previously designed and tested behaviors. In these cases, the composition of behavior automata can be used for the design of new behaviors. Finally, during the robot operation different behaviors can be combined to improve the robot interaction with the environment.

Definition 6. (Composition rule). Consider two behavior automata, $\Gamma^1(C^1_\tau, p^1)$ and $\Gamma^2(C^2_\tau, p^2)$. The composition defined as

$$\Gamma^3(C^3_{\tau^3}, p^3) = \Gamma^1(C^1_{\tau^1}, p^1) \odot_g \Gamma^2(C^2_{\tau^2}, p^2)$$

is the behavior automaton $\Gamma^3(C^3_{\tau^3}, p^3)$:

$$C^3_{\tau^3} = C^1_{\tau^1} \cap C^2_{\tau^2}$$

$$p^3 = g(p^1, p^2)$$

$$\tau^3 = [\max\{t^1_0,\, t^2_0\},\, \min\{t^1_1,\, t^2_1\}]$$

where $g(.)$ is a Lipschitz continuous function. The composition rule can only be applied if $C^3_{\tau^3} \neq \emptyset$.

The result of the proposed composition rule is an hybrid behavior automaton with the same structure of the input automata. The viable set is a convex, closed subset of the original viable sets. And the velocity profile is a function of the input velocity profiles. Thus the composite behavior can be interpreted as a refinement of the input behaviors.

Because the result of the composition is also a behavior automaton, the overall control architecture complexity is not increased. This is useful for architectures designed in a bottom-up approach.

The main disadvantage of the proposed composition rule is that it the viable sets of the input behaviors must intersect. As a result, the control architecture must be able to reason with the available behavior automata to determine when the composition is valid. Also, it can be required to plan a sequence of compositions such that a desired behavior is obtained.

5. BEHAVIOR DESIGN EXAMPLES

Assume the surveillance system has available a set of identical mobile robots, with unicycle kinematic drives. These robots are modeled by the differential inclusion (11). For the composition rule, Definition 6, the sum of the velocity profiles is used

$$p^3 = g(p^1, p^2) = p^1 + p^2 \tag{16}$$

A commonly required behavior in mobile robots is the obstacle avoidance. This behavior can be obtained by constraining the robot position to an obstacle-free region, C_{oa}. The viable set C_{oa} is determined such that the robot position it always inside and a safety distance is kept from the obstacles. In Figure 3 an example of a constant width C_{oa} is presented, where the small filled circle is the robot position.

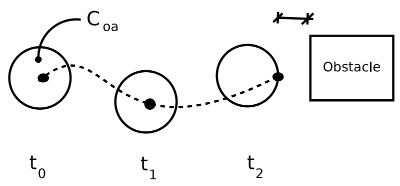

Fig. 3. Example of the obstacle avoidance viable set

Because the viable set is, by construction, free of obstacles then any viable state trajectory is permitted. And since the sum of velocity profiles is used to compose behaviors, a null velocity profile is used, $p_{oa} = 0$.

A set of maneuvering behaviors, e.g turn left or go to location, can also be synthesised with the proposed methodology. These behaviors are obtained by specifying the appropriate velocity profiles and using an arbitrary viable set, C_m, for the robot position. The viable set can be identified with particular regions of the environment, such as rooms or corridors. Let $R(q)$ be the rotation matrix from the robot frame to the world frame. The velocity profiles

$$p_1 = R(q) \cdot [1\,0\,0]', \;\; p_2 = R(q) \cdot [0\,0\,1]' \tag{17}$$

produce respectively, the *go forward* and *turn left* behaviors. In order to prevent collisions with the environment, the maneuver and obstacle avoidance obstacles can be composed.

The *leader-follower* approach is used to synthesise the move in formation behavior. The role of leader is assigned arbitrarily to one of the robots, while the others are assigned the role of followers. The motion of the leader robot is independent of the others. The position of each follower robot is constrained to a viable set, C^i_f, defined with respect to the position of the leader robot. In this manner, the relative positions in the formation for each follower are specified. The velocity profile of the follower robots is copied from the leader robot. Thus, a communication link among the robots is required for this behavior.

5.1 *Simulation Results*

A set of three simulations where conducted to test the proposed behaviors and also the composition rule. The simulations where executed in the Octave [1] environment. The viable sets used are convex polyhedrons, circumscribed within circles.

In the first simulation, the dynamics of the behavior automaton where examined. The robot desired behavior, denoted *elliptic*, is the composition of the *go forward* and the *turn left* behaviors. Both use the same viable set, a fifteen facets polyhedron, centered at the origin and with a radius of 0.3. Initially the robot used the behavior-inactive controller to approach and enter the viable set. Then, the robot moved with a elliptic trajectory, until hitting the set border. The robot motion was constrained by the viable set, and it proceeded along the set border. Eventually the trajectory was again in the interior of the viable set and the robot moved away from the set border. The robot motion and the viable set for this simulation are presented in Figure 4.

The obstacle avoidance behavior was tested in the second simulation. An obstacle is placed $(0, 0)$, the required safety distance is 0.05 and the width of the obstacle avoidance viable set is 0.1. The robot behavior is the composition of the previous elliptic maneuver behavior with the obstacle avoidance. Because initially there are no obstacles in the vicinity of the robot, it exhibited the elliptic trajectory.

[1] http://www.gnu.org/software/octave/

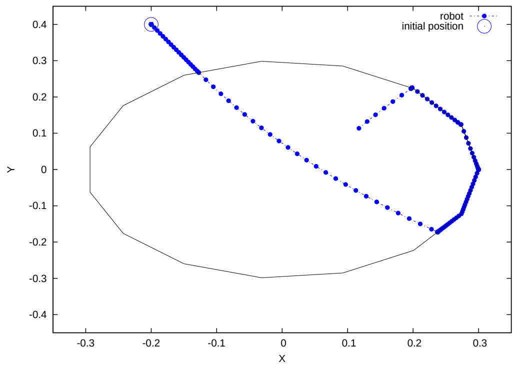

Fig. 4. Active and inactive behaviors

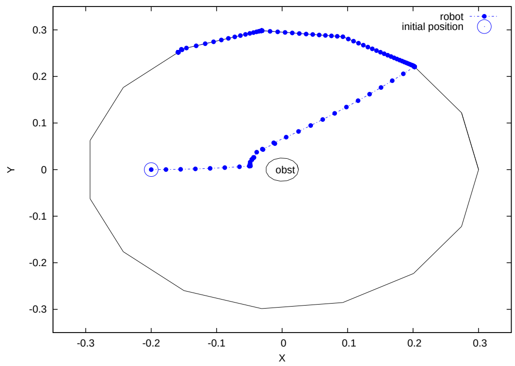

Fig. 5. Obstacle avoidance behavior

When it became within the safety distance of the obstacle, the robot moved along the obstacle avoidance viable set border. Eventually it was sufficiently distant from the obstacle, and continued with the elliptic behavior. The robot motion in this simulation is presented in Figure 5.

The coordinated motion of the robots was tested in the third simulation, with two follower robots and one leader. The behavior of the leader robot is the elliptic maneuver behavior, but within a larger viable set. For the follower robots, the desired behavior is to follow parallel to the leader. Their viable sets are two parallel polyhedrons of radius 0.05, each set a distance of 0.1 from the leader robot. Initially, the behavior-inactive controller is used by the follower robots to approach the leader. After the follower robots reached their viable set, the leader constant elliptic velocity profile was used by them. The motion of the three robots is presented in Figure 6. The trajectory of the leader is represented by a thick line, while the followers are marked with "+" and "*" symbols. The robots initial position is marked with a circle. The relative distance from the leader to the followers at sampled time instants is represented by thin lines.

6. CONCLUSIONS AND FUTURE WORK

A behavior synthesis method was proposed for mobile robots in surveillance applications. The behaviors are defined through a viable state set and a velocity profile. Although only unicycle wheeled robots where considered, the proposed approach can be applied to other types of

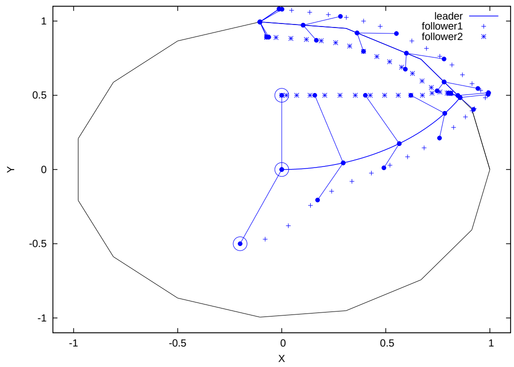

Fig. 6. Coordinated motion

kinematic drives. The methodology was exemplified on the synthesis of a diverse range of commonly required mobile robot behaviors.

A composition rule was also presented, for the combination of different behaviors. With the use of this rule, new behaviors can be synthesized from the available ones. In this manner, behaviors can be used as basic building blocks of a general control architecture.

Future work includes (i) extending the synthesis approach to other kinematic solutions, (ii) considering the robot state estimation error and (iii) further development of the control architecture.

Acknowledgments

This work was supported by ISR/IST plurianual funding through the POS_Conhecimento Program that includes FEDER funds. Nelson Gonçalves is working under grant SFRH/BD/23804/2005, from Fundação para a Ciência e a Tecnologia.

7. REFERENCES

Freire, E., T. Bastos-Filho, M. Sarcinelli-Filho and R. Carelli (2004). A new mobile robot control approach via fusion of control signals. *Systems, Man and Cybernetics, Part B, IEEE Transactions on* **34**(1), 419–429.

Innocenti, Bianca, Beatriz López and Joaquim Salvi (2006). A multi-agent collaborative control architecture with fuzzy adjustment for a mobile robot. In: *ICINCO-RA*. pp. 523–526.

Smirnov, Georgi (2001). *Introduction to the Theory of Differential Inclusions (Graduate Studies in Mathematics)*. 1st ed.. American Mathematical Society.

Tabuada, P., G. J. Pappas and P. Lima (2001). Compositional abstractions of hybrid control systems. In: *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*. Vol. 1. pp. 352–357 vol.1.

Wang, Xiuqing, Zeng-Guang Hou, Anmin Zou, Min Tan and Long Cheng (2007). A behavior controller based on spiking neural networks for mobile robots. *Neurocomputing*.