

10.24425/123455

Archives of Control Sciences
Volume 28(LXIV), 2018
No. 2, pages 189–200

Hybrid motion control for multiple mobile robot systems

ELZBIETA ROSZKOWSKA

This article presents a hybrid control system for a group of mobile robots. The components of this system are the supervisory controller(s), employing a discrete, event-driven model of concurrent robot processes, and robot motion controllers, employing a continuous time model with event-switched modes. The missions of the robots are specified by a sequence of to-be visited points, and the developed methodology ensures in a formal way their correct accomplishment.

Key words: multi-robot system, hybrid control

1. Introduction

The use of a mobile robot team in place of one robot substantially increases the performance of many robotic applications, including those related to area searching, search and rescue, interplanetary exploration, extraction of minerals, agriculture, forestry, or transport. A key issue in the design of such systems is to coordinate the movement of a number of robots operating in the same workspace. Regardless of their tasks, the robots must be able to effectively share a common area in order to prevent the mutual disruption of traffic and effectively pursue their missions.

The prevailing approach to modeling such systems consists in the abstraction of each robot as *mobile agent*, whose dynamics are described in time, and considering the problem of the agents' coordination over time. Some representative results of this research line can be found, among others, in [1–7]. The main contribution of these works are methodologies of collision-free control of robot movement. The resulting algorithms have, however, two major drawbacks: a) due to high computational complexity, they are not scalable, and b) they do

E. Roszkowska is with Department of Cybernetics and Robotics, Faculty of Electronics, Wrocław University of Science and Technology, ul. Janiszewskiego 11/17, 50–372 Wrocław, Poland,
E-mail: elzbieta.roszkowska@pwr.edu.pl.

This work was supported by the research grant 2016/23/B/ST7/01441 funded by the National Science Center, Poland.

Received 26.01.2017.

not guarantee a formally correct operation of the system, i.e., the completion of the missions assigned to the robots.

The ineffectiveness of these models in providing the above properties is mainly due to the assumed representation of the robot system, whose operation is abstracted in continuous time. Devoid of these disadvantages is the approach proposed in [8–10], where the logic of robot coordination is derived using the DES (Discrete Event System) formalism. The proposed model ensures correct space sharing by a group of robots through imposing certain constraints on their motion, and can be applied to any mobile robots, accomplishing any arbitrarily assumed missions, and for both centralized and distributed supervisory-control architecture. To respect the requirements of a so-defined DES-based supervisory-control scheme, the robots need to modify their individual motion control, based on the CTS (Continuous Time System) abstraction.

The aim of this work is to develop a hybrid control model for multiple-mobile robot systems, that ensures the accomplishment of the mission of each particular robot, operating under its own CTS-based controller, while meeting the constraints enforced by the DES-based supervisory control. In Section 2, we present in more detail the objectives of this paper. In Sections 3–5, we systematically pursue the synthesis of the hybrid controller. Section 6 is devoted to the implementation of the concept and simulation tests. The last section contains the conclusions and prospects of further studies.

2. Problem statement

We will consider a group of autonomous mobile robots sharing a 2D space. Each robot performs a mission that requires the robot to travel along a specific path. The considered problem can be defined as follows.

Develop a control system for space-sharing mobile robots, ensuring that: 1) all the robots will accomplish their missions in a finite time interval $[0, T]$, 2) at any moment $t \in [0, T]$ the areas occupied by the particular robots are disjoint (which implies no collisions), and 3) the obtained solution is scalable, i.e., the time required for the computation of the control algorithms is bounded by a polynomial function of the number of robots.

Since requirement (3) eliminates the possibility of synchronous trajectory planning for all the robots, we assume that the robots operate asynchronously, based on the control algorithms that allow each particular robot to correctly accomplish its mission when alone in the stage. Fulfilling the assumptions (1–2) in the multiple-robot system requires from each robot a real-time reaction to the state of other robots, through a modification of its control algorithm. The approach proposed in this paper involves two control layers: the supervisory con-

trol – dealing with robots' coordination, and its subordinate – local robot motion control.

The concept underlying the supervisory-control layer comes from [8], and assumes a breakdown of the robot paths into sectors. The role of this layer is to coordinate the transition of the robots between path sectors, while the motion of the robots within the sectors is subject to the lower control layer. Each robot is abstracted by a disc large enough to cover the robot in any of its configuration. The avoidance of collisions among the robots is then based on the following concept. On the sector set, being the union of the sector sets of all the robots, we establish a conflict relation. Two sectors are in conflict if: i) each of them is associated with a different robot, and ii) the minimum distance between these sectors is less than the sum of the radiuses of the disks of the respective robots. The supervisor prevents the collision occurrence through forbidding a robot to enter zone x if another robot is travelling in zone y that is in conflict with x . To provide all the robots with the ability to complete their missions, it is necessary to further limit the robot motion concurrency in order to avoid potential deadlocks. Thus, the supervisor also prevents a robot from entering a sector if it results in a state of the system from which the final state is unreachable. An appropriate decision-making algorithm aimed at the deadlock avoidance in multiple mobile robot systems was developed in [8]. The model proposed therein relates the permissions for the robots to transit from one sector to another to the state of the system, and guarantees that under the proposed logics, the requirements (1–3) hold.

3. Specification of the multi-robot system

The mission of each particular robot is determined by a sequence of points in the plane, which the robot has to visit subsequently. It is possible to assume that these points also determine the breakdown of a path into sectors. It should be noted, however, that the sector lengths affect the efficiency of the system. If sector x of some robot is in conflict with sector y of another robot then as long as the former robot is traveling in x , the latter robot cannot enter y and is possibly idling. Therefore, for a pair of consecutive points spaced by distance D , $kd \leq D < (k+1)d$, we introduce additional points, partitioning the 'long' sector into k equal shorter sectors. The value of d is initially determined in an arbitrary manner and next experimentally optimized during the simulation studies. In order to describe the path passing through the specified n points by a continuous function $f(s) \in C$, we use the polynomial interpolation [11]. As a result, we obtain a polynomial function of $n-1$ order, example of which is shown in Fig. 1.

As discussed in Section 2, for the purpose of the robots coordination, we establish the relation of conflicts among sectors. To simplify the calculations, we employ an approximation of the path, the curve obtained by connecting each

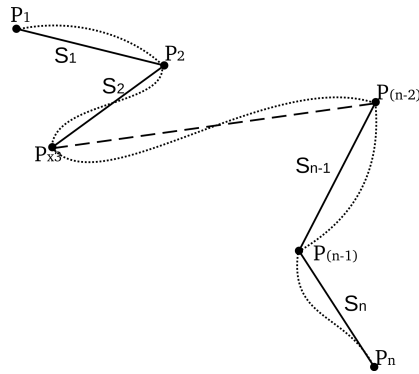


Figure 1: Example of the polynomial interpolation of a robot path specified by a sequence of points, and its approximation by a curve consisting of line segments

two consecutive points with a straight line, and later correct the error resulting from the difference between the polynomial path and the curve approximating it. Clearly, the sectors of such a path are the line segments of the curve. Assuming that the consecutive points specifying a path are described by the sequence

$$S = \{P_1, P_2, \dots, P_n\} \quad (1)$$

the successive segments of the curve, $S_k = \overline{P_k P_{k+1}}$, $k = 1, \dots, n-1$, can be described using a parameter s :

$$S_k \equiv \begin{cases} x(s) = a_1 s + b_1 \\ y(s) = a_2 s + b_2 \end{cases} \quad \text{where} \quad s \in \left\langle \frac{k}{n-1}, \frac{k+1}{n-1} \right\rangle. \quad (2)$$

In order to determine the conflict relation between the path sectors of a group of robots, for each pair of robots and for each pair of sectors S_k and S_l such that S_k belongs to the path of one robot, and S_l to the path of the other robot, we calculate their minimal distance. As can be demonstrated, it will be equal to zero if the sectors intersect, to the minimum of the distances between the ends of one sector and their orthogonal projections on the other sector, if at least one such a projection exists, or to the minimum of the distances between the ends of sectors S_k and S_l otherwise:

$$d(S_k, S_l) = \begin{cases} 0 & \text{if the path segments intersect} \\ \min \left(\|P_k, P_k^l\|, \|P_{k+1}, P_{k+1}^l\|, \|P_l, P_l^k\|, \|P_{l+1}, P_{l+1}^k\| \right) & \text{if there exists a relevant projection} \\ \min \left(\|P_k, P_l\|, \|P_k, P_{l+1}\|, \|P_{k+1}, P_l\|, \|P_{k+1}, P_{l+1}\| \right) & \text{otherwise} \end{cases} \quad (3)$$

Assuming that the two robots are represented by the disks of radiuses r and r' , sectors S_k and S_l are in conflict if

$$d(S_k, S_l) < r + r' + p, \quad (4)$$

where p is a constant parameter that extends the minimum distance between the sectors. It allows us to correct the fact that the value of the distance is calculated based on the approximated path, and not on the polynomial curve.

4. Robot control in collision-free system

In this section, we consider the control of the robot motion along a given path, based on a continuous time model that does not consider the presence of other robots on the stage. In the next section, we will use the obtained result in order to develop the hybrid model, describing the concurrent motion of multiple space-sharing robots.

For the purposes of this paper, we will consider mobile robots of class $(2, 0)$ that includes the simplest mobile platform, a so-called unicycle. We also assume that the robots roll on hard, flat, horizontal ground without skidding [12]. Such robots will be described at the kinematic level using the model of [13].

The state of a unicycle is described by three state variables, $q_m = (x, y, \theta)$, respectively specifying the positions of the central point relative to the x and y axis, and the angle describing the orientation of the platform in the global coordinate system (Fig. 2). The number of controls is two, $u = (v, \omega)$, where v is the linear velocity, and ω is the angular velocity of the platform. The kinematics of this robot are given by the following equation:

$$\dot{q}_m = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}. \quad (5)$$

To control the path tracking by the mobile platform, we employ the Frenet coordinates [14]. Then we use a modified Samson algorithm for the kinematic control of the unicycle [15], that not only provides for the robot to follow the path but also to move along the path at a constant speed. This algorithm is also characterized by a very good convergence, and there are no abrupt changes in speed. After the transformation of the robot coordinates with the Frenet parameterization (6), two variables, l and s , are obtained that describe, respectively, the distance between the coordinate system associated with the robot center and the tracked

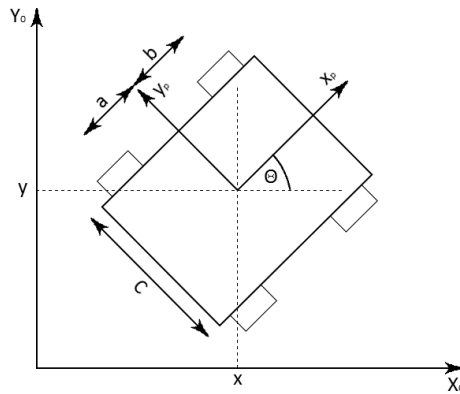


Figure 2: Graphical representation of the platform

path, and the displacement $s \in \langle 0, 1 \rangle$ of the robot on the path. Then

$$\begin{aligned} \dot{l} &= (-\sin \theta_r \quad \cos \theta_r) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \\ \dot{s} &= \frac{(\cos \theta_r \quad \sin \theta_r)}{1 - c(s)l} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \end{aligned} \quad (6)$$

where $c(s)$ is the curvature of the path, and θ_r is the reference orientation of the robot derived from the formula $\dot{\theta}_r = c(s)\dot{s}$. In this way, we can define the tracking error as:

$$\varepsilon^T = (l, \theta - \theta_r, s). \quad (7)$$

For a vehicle described by the kinematics (5), Frenet coordinates are as follows:

$$\begin{cases} \dot{l} = v \sin(\tilde{\theta}) \\ \dot{\tilde{\theta}} = \omega - v \cos \tilde{\theta} \frac{c(s)}{1 + c(s)l} \end{cases}. \quad (8)$$

For the system described by (8), we employ Samson controller [15], thus obtaining the following control formula.

$$\begin{cases} v_r = v(t) \\ \omega_r = -k_2 l v_r \frac{\sin \tilde{\theta}}{\tilde{\theta}} - k_3 \tilde{\theta} + v_r \cos \tilde{\theta} \frac{c(s)}{1 + c(s)l} \end{cases}, \quad (9)$$

where:

- k_2, k_3 – amplification of the algorithm, determined experimentally,

- $\tilde{\theta}$ – robot orientation tracking error, defined in (8),
- l – distance between the robot center and the path, defined in (8),
- $c(s)$ – the curvature of curve c , derived from the robot path.

5. Hybrid control of robot motion

As mentioned in Section 2, the assumed concept of the coordination of a multiple robot system consists in controlling the possibility of the transition of each particular robot to its next path sector based on the current state of the other robots. This viewpoint is implemented in the automaton depicted in the upper part of Fig. 3. In the initial state, denoted by (0), the robot is outside the system, in the contractual sector S_0 . The supervisor's decision to let the robot enter the next sector generates event g , which changes the state of the robot to state (0,1) corresponding to the robot transition from sector S_0 to sector S_1 . Having completed the transition, the robot generates event f . If the state of the system does not allow the transition to the subsequent sector, the robot remains in the current one until there occurs event g , which lets it to resume its motion and transit to sector S_3 . In the same manner, the robot continues its travel on the path until it leaves the system, i.e., enters the contractual sector S_{n+1} . The model underlying the supervisory control [8] ensures collision-free movement of the robots and a finite time of waiting by each particular robot for the opportunity to enter each subsequent sector, and thus the correct completion of all the robot missions.

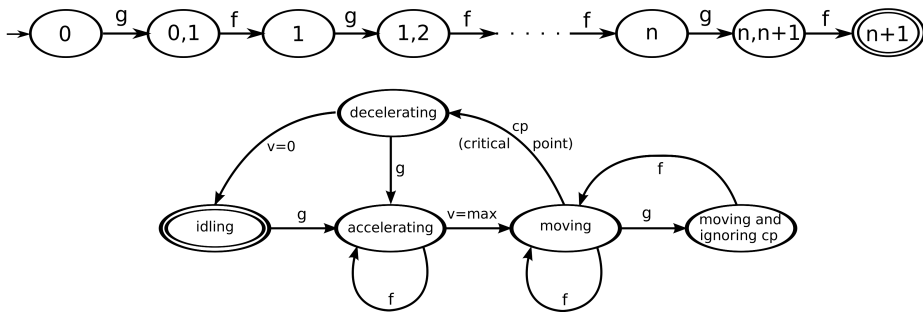


Figure 3: Complex automaton representing the robot movement. Upper automaton determines the transition between the path sectors, bottom automaton depicts the robot motion modes. Both automata are synchronized by common events g and f , corresponding respectively to allocation and release of sectors to robots

In order to make the robots subordinate to the decisions of the supervisor, it is necessary to modify their path tracking formula (9). To this end, we will

distinguish two characteristic points in each sector - the release point rl , lying at the distance of robot radius from the beginning of the sector, and the critical point cp , lying at a 'safe' distance to the end of the sector. rl is the point where robot's disk stops overlapping the previous sector, which can be now released. cp defines the point that allows the robot to come to a stop before the end of the sector, i.e., the point where the robot needs to start decelerating if it has not yet received the permission to transit to the next sector.

The travel of each robot in a sector is associated with the changes of its velocity profile according to the logic of the automaton depicted in the lower part of Fig. 3. In the initial or the final state and while waiting for the permission to enter the next sector, the robot is *idling*. Occurrence of event g allows it to start moving and results in the transition to state *accelerating*. The robot remains in this state until the velocity $v = v_{\max}$, and then turns to state *moving*. When passing point rl , which happens either in state *accelerating* or *moving*, the robot generates event f that informs the supervisor of the exemption of the previous sector. If event g occurs before reaching the critical point cp by the robot, it changes the state to *moving and ignoring cp* and proceeds to the next sector. Otherwise it passes to state *decelerating* and at achieving the velocity $v = 0$, the robot starts *idling*. The above discussed changes of the robot motion modes are achieved by varying the speed v_r according to the following formula (the meaning of the variable symbols as in (9)).

$$\left\{ \begin{array}{l} v_r = 1 \quad \text{if } mode = accelerating \\ v_r = v(t) \quad \text{if } mode = moving \\ v_r = 0 \quad \text{if } mode = decelerating \\ v_r = 0 \quad \text{if } mode = idling \\ \omega_r = -k_2 l v_r \frac{\sin \tilde{\theta}}{\tilde{\theta}} - k_3 \tilde{\theta} + v_r \cos \tilde{\theta} \frac{c(s)}{1 + c(s)l} \end{array} \right. \quad (10)$$

6. Simulation experiments

The proposed hybrid control model has been implemented in the Matlab/Simulink/ Stateflow environment for an example system of four robots of class (2,0) travelling on two circular paths depicted in Fig. 4. Each of the circles is broken down to four sectors numbered 1, 2, 3, 4 (left circle) and 5, 2', 4', 6 (right circle). The conflict relation contains two pairs of sectors: (2, 2') and (4, 4'), which impose constraints on the concurrent motion of the robots, requesting that for each pair no more than one sector be occupied at a time. The remaining sectors, 1, 3, 5, 6, are not in conflict with any other sector and allow the asynchronous traffic of the robots. Moreover, we also assume that each of them can be further

partitioned into two subsectors, and thus accommodate up to two robots at a time. It allows smoother robot motion and is sufficient to prevent deadlocks.

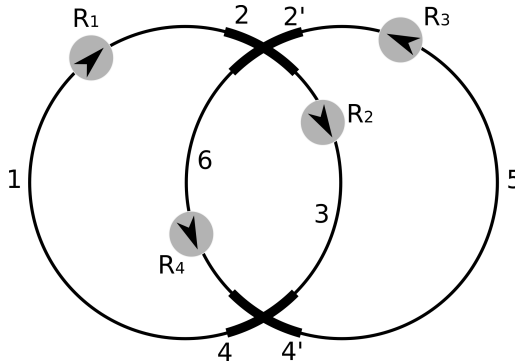


Figure 4: Simulation experiment: layout of the paths and their partitioning to sectors

The following figures present results of the simulation experiments that demonstrate the quality of the proposed control. Fig. 5 depicts the occupation of the sectors by the robots in time. As can be observed, the developed control enforces the mutually exclusive presence of the robots in the conflicting path sectors 2, 2' and 4, 4', which prevents collisions. Fig. 6(a) depicts the actual and the tracked path of robot R_1 , and Fig. 6(b) depicts the path tracking error measured by the distance $l = r - \sqrt{(x_p - x_{rob})^2 + (y_p - y_{rob})^2}$ between the realized and tracked path (value of l in the Fermat transform). One can notice a down-

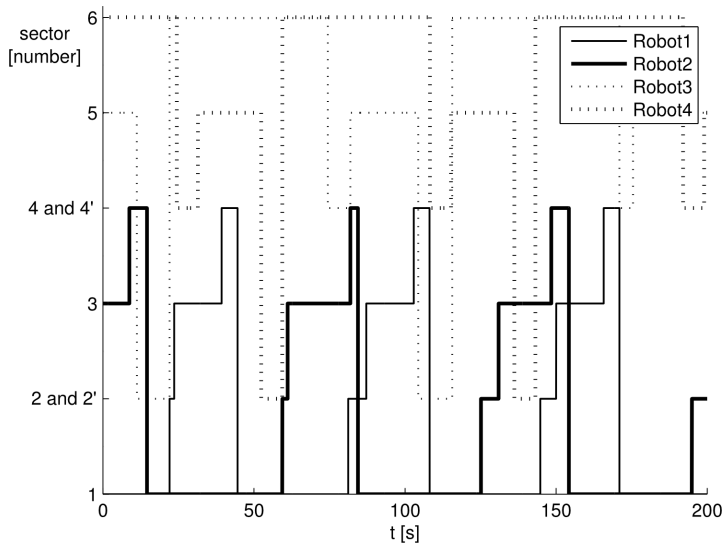


Figure 5: Simulation experiment. Occupation of the path sectors by the robots

ward trend of the control error, which suggests that the control system is stable, and even the asymptotic stability of the system can be achieved.

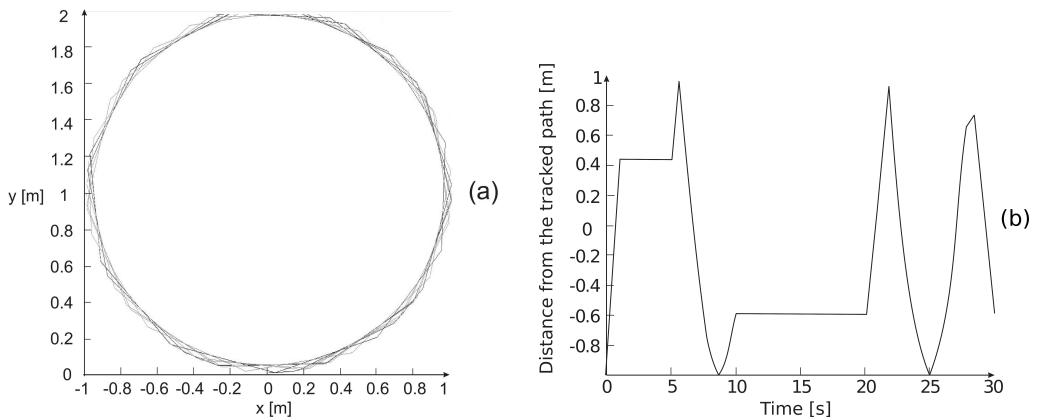


Figure 6: Simulation experiment: a) path tracking: realized path (gray) and tracked path (black), b) path tracking error

7. Conclusions

This article presents a hybrid methodology of the control synthesis for multiple mobile robot systems. The proposed concept consists in the discretization of the robots' paths through their breakdown to sectors, establishing the supervisory control for the coordination of the robots' transitions between the sectors, and developing an automaton model of robot motion modes and their changes, that allows each robot to obey the dynamically evolving traffic constraints of the coordination system. In this paper we considered robots of class $(2, 0)$, and the illustrative simulation experiment concerned a relatively simple system of four robots travelling on two circular paths. The proposed approach is, however, much more general. The supervisory control model captures paths of any shape and their arbitrary breakdown, and allows the consideration of any profile of the robot motion within the path sectors, while providing a formally correct execution of the missions of all robots.

The developed methodology can be implemented in both centralized and distributed way. In the first case, the master controller communicates with all the robots using the events g and f , respectively allowing them to move to the next sector and collecting information on the completion of the transition. These events change the state of the model used by the supervisor, based on which it develops the coordination decisions. In the second case, all the robots have the same local supervisors, whose state is updated as a result of communication be-

tween the robots, and based on it, they make their own decisions concerning the changes of the motion modes.

Due to the high flexibility of the approach, it provides for the optimization of the efficiency of the robotic system at many levels. This problem will be considered in our further work, both at the theoretical level and the practical level of the construction of suitable tools, allowing more efficient development of the coordination logic in the Matlab system.

References

- [1] L. PALLOTTINO, V. G. SCORDIO, A. BICCHI and E. FRAZZOLI: Decentralized cooperative policy for conflict resolution in multivehicle systems, *IEEE Transactions on Robotics*, **23**, (2007), 1170–1183.
- [2] C. TOMLIN, G. J. PAPPAS and S. SASTRY: Conflict resolution for air traffic management: a study in multiagent hybrid systems, *IEEE Transactions on Automatic Control*, **43**, (1998), 509–521.
- [3] J. LYGEROS, D. N. GODBOLE and S. SASTRY: Verified hybrid controllers for automated vehicles, *IEEE Transactions on Automatic Control*, **43**, (1998), 522–539.
- [4] G. INALHAN, D.M. STIPANOVIC and C.J. TOMLIN: Decentralized optimization, with application to multiple aircraft coordination, in *Proc. of CDC'02*, IEEE, (2002), 1147–1155.
- [5] S.M. LA VALLE and S.A. HUTCHINSON: Optimal motion planning for multiple robots having independent goals, *IEEE Transactions on Robotics & Automation*, **14**, pp. 912–925, 1998.
- [6] D.V. DIMAROGONAS, S.G. LOIZOU, K.J. KYRIAKOPOULOS and M.M. ZAVLANOS: A feedback stabilization and collision avoidance scheme for multiple independent non-point agents, *Automatica*, **42**, (2006), 229–243.
- [7] F. PECORA, M. CIRILLO and D. DIMITROV: On mission-dependent coordination of multiple vehicles under spatial and temporal constraints, in *Proc. IEEE/RAS Int. Conf. on Intelligent Robots and Systems*, (2012), 5262–5269.
- [8] E. ROSZKOWSKA: Provably correct closed-loop control for multiple mobile robot systems, in *Proc. of ICRA'05*, IEEE, (2005), 2810–2815.

-
- [9] S. REVELIOTIS and E. ROSZKOWSKA: Conflict resolution in free-ranging multivehicle systems: A resource allocation paradigm, *IEEE Transactions on Robotics*, vol. **27**(2), (2011), 283–296.
 - [10] E. ROSZKOWSKA and S. REVELIOTIS: A distributed protocol for motion coordination in free-range vehicular systems, *Automatica*, **49**(6), (2013), 1639–1653.
 - [11] P.J. DAVIS: Interpolation and approximation, Courier Corporation, 1975.
 - [12] A. MAZUR: Sterowanie oparte na modelu dla nieholonomicznych manipulatorów mobilnych, Oficyna Wydawnicza Politechniki Wrocławskiej, 2009.
 - [13] K. TCHOŃ, A. MAZUR, I. DULĘBA, R. HOSSA, and R. MUSZYŃSKI: Manipulatory i roboty mobilne, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.
 - [14] A. MAZUR: Hybrid adaptive control laws solving a path following problem for non-holonomic mobile manipulators, *International Journal of Control*, **77**(15), (2004), 1297–1306.
 - [15] A. MICAELLI and C. SAMSON: Path following and time-varying feedback stabilization of a wheeled robot, in *Proc. of ICARCV'92*, 1992.