



HAL
open science

Control of an intelligent robot system over a wireless network

Amine Mechraoui, Jean-Marc Thiriet, Sylviane Gentil

► **To cite this version:**

Amine Mechraoui, Jean-Marc Thiriet, Sylviane Gentil. Control of an intelligent robot system over a wireless network. IAR 2008 - 23rd IAR Workshop on Advanced Control and Diagnosis, Nov 2008, Coventry, United Kingdom. pp.201. hal-00343384

HAL Id: hal-00343384

<https://hal.science/hal-00343384>

Submitted on 1 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONTROL OF AN INTELLIGENT ROBOT SYSTEM OVER A WIRELESS NETWORK

A.Mechraoui *, J.M.Thiriet *, S.Gentil *

* *GIPSA-Lab, Control department - BP 46
38402 Saint Martin d'Hères, France*

Email: {amine.mechraoui,jean-marc.thiriet,sylviane.gentil}@gipsa-lab.inpg.fr

Abstract: This paper emphasizes the control-communication co-design problem of a mobile robot with varying network load. A Khepera mobile robot driven by two dc motor-powered wheels is used as an example. Each motor is controlled directly by a PI controller whereas other types of controllers are used for the control of linear and angular velocities of the robot. A Zigbee network is implemented between the controller and the plant and between the sensors and the controller. Difficulties due to wireless network such as delays and packet dropout are studied. The controlled system and the communication network are simulated respectively with Matlab/Simulink and TrueTime.

Keywords: Network Control Systems, Wireless Network, Co-design problem, True Time simulation, 802.15.4, Unicycle robot.

1. INTRODUCTION

Nowadays, network control systems are more and more used, because they are easy to design and maintain. However, introducing a network in the loops presents some inconvenience such as band limited channels, sampling delays and packet dropouts (Hespanha et al., 2007).

Network influence on the control performance has been studied for a long time, especially the influence of the delay introduced by the network (Zhang et al., 2001). The diagnosis of networked control systems has been studied in (Berbra et al., 2008) and the influence of different parameters of the network have been examined too.

This paper presents the study of a unicycle robot, a Khepera robot (Lambercy and Caprari, 2007). Algorithms for tracking trajectory and obstacle avoidance are implemented. ZigBee wireless network will be integrated in the control loop to establish the communication between the con-

trollers and the robot and between the sensors and the controllers.

The paper is organized as follow. The second section presents a brief description of the model of the khepera robot, notably the kinematic model and the dynamic one. In the third section, different controllers are studied. Two algorithms will be presented, the first is the tracking trajectory one and the second is the obstacle avoidance one. Simulations are shown in the same section. Section 4 presents the control over ZigBee wireless network and the influence of the integration of this network is studied. Finally, the last section presents the conclusion and perspectives.

2. ROBOT MODEL

Consider a unicycle robot (Khepera) as shown in Fig.2. Let x , y and φ be the state variables where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are the Cartesian coordinates,

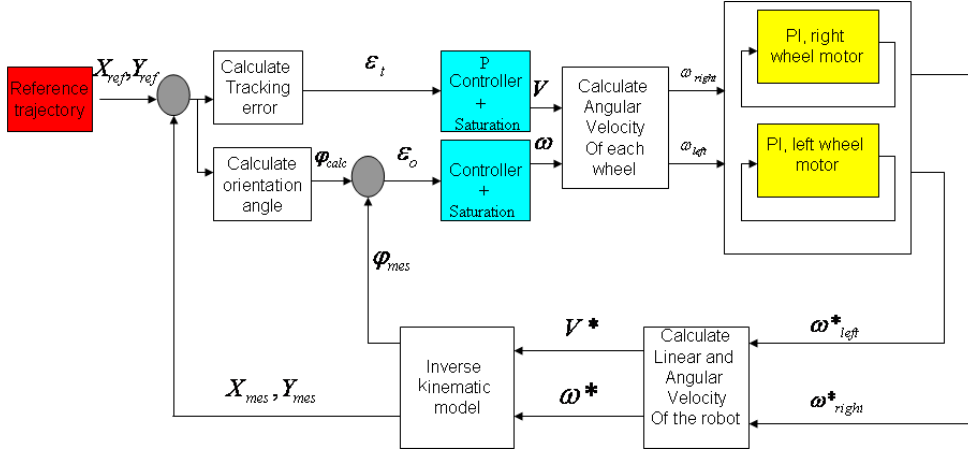


Fig. 1. General block of control

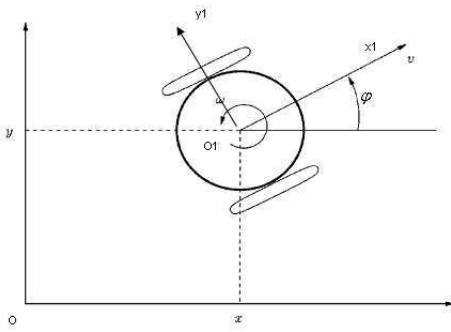


Fig. 2. Robot model.

$\varphi \in [0, 2\pi[$ is the robot's orientation with respect to the X-axis. We consider v and ω respectively as the linear and the angular velocities of the robot. The kinematic equations of this robot are:

$$\dot{x} = v \cos \varphi \quad \dot{y} = v \sin \varphi \quad \dot{\varphi} = \omega. \quad (1)$$

The kinematic model of the mobile robot has two control inputs ω_{left} and ω_{right} i.e. the left and right wheels velocities. These are related to the linear velocity v and the angular velocity ω of the robot according to the following equations:

$$\omega_{right} = \frac{v}{R} + \omega \quad \omega_{left} = \frac{v}{R} - \omega. \quad (2)$$

where R is half the distance between the two robot's wheels.

The dynamic model of the robot wheels is characterized by the equations of the DC motors driving the wheels. They are represented by a first order model:

$$\frac{\omega^*(left, right)}{U} = \frac{K}{\tau s + 1}. \quad (3)$$

where U is the voltage applied to the motor and $\omega^*(left, right)$ are the angular velocities generated by each motor.

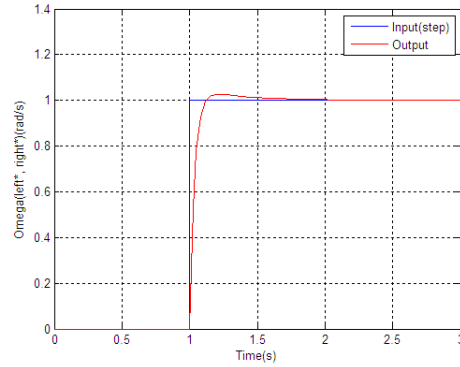


Fig. 3. Closed loop response

3. ROBOT CONTROL WITHOUT NETWORK

One of the objective of this work is to define a multi-robot cooperative architecture. To achieve this objective, the first step consists in controlling one robot, with tracking trajectory and obstacle avoidance algorithms.

To control this robot, different controllers are proposed and compared in (Kim and Tsiotras, 2002) and (Samson and Ait-Abderrahim, 1991). Two different controllers are required. A PI controller is used to control the angular velocity of each motor, and two controllers with a saturation function are used to control the linear and angular velocities of the robot, as shown in Fig.1.

The coefficients of the PI controller are chosen in order to get a closed loop time response equal to 100ms.

The closed-loop transfer function is:

$$H_{closed-loop}(s) = \frac{H_{open-loop}(s)}{1 + H_{open-loop}(s)}. \quad (4)$$

$$H_{open-loop}(s) = \frac{K_p T_i s + 1}{T_i s} \frac{K}{\tau s + 1}. \quad (5)$$

The corresponding coefficients are: $K_p = 3.5$ and $T_i = 10$. The closed loop response is shown in Fig.3.

3.1 Target chasing

In this section, a simulation is presented corresponding to the choice of the shortest path to reach a defined target. To accomplish this task, the distance between the actual position of the robot and the target is calculated according to equation 6.

$$\varepsilon_t = \sqrt{e_x^2 + e_y^2}. \quad (6)$$

$$\text{with } e_x = X_{ref} - X^* \text{ and } e_y = Y_{ref} - Y^*. \quad (7)$$

This distance is used as a tracking error to generate a linear velocity using a P controller according to equation (8) with the saturation function. The orientation error (9) is also required to generate the angular velocity using the controller according to equation (17).

$$v = K_v \varepsilon_t. \quad (8)$$

$$\varepsilon_\varphi = \arctan\left(\frac{e_x}{e_y}\right). \quad (9)$$

$$\omega = K_\omega v \varepsilon_\varphi. \quad (10)$$

Simulation results are shown in Fig.4. In this simulation, references are $X_{ref} = 0.4m$ and $Y_{ref} = 0.6m$ and the initial position of the robot is $(X_0, Y_0, \varphi_0) = (0.3, 0.1, 0)$.

3.2 Trajectory tracking

The objective of this simulation is to track any trajectory with a minimal error. Tracking a reference trajectory means that ε_t and ε_φ must go to 0. Many methods are presented in the literature (Micaelli and Samson, 1993). A set of points is required to define the trajectory. The reference trajectory used in this simulation is shown in Fig.5.

The initial condition of the robot is: $X_0 = 0.3m$, $Y_0 = 0.1m$ and $\varphi_0 = 0rad$. The simulation is shown in Fig.5.

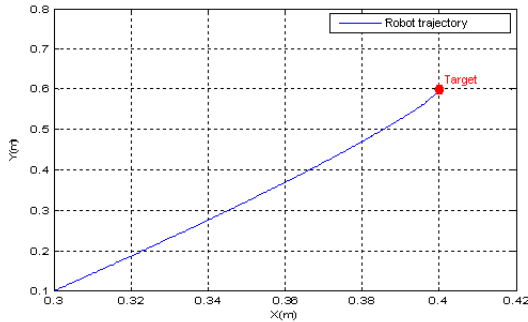


Fig. 4. Position response with reference $(X, Y) = (0.4, 0.6)(m)$.

3.3 Obstacle avoidance

The limit cycle navigation algorithm proposed in (Kim and Kim, 2003) is briefly introduced. Let's consider that the robot must reach a target which coordinates are (X_{tar}, Y_{tar}) while avoiding a circular target whose center coordinates are (X_{obs}, Y_{obs}) and whose radius is R_{obs} with:

$$R_{obs} = R_{circle} + R_{safe} \quad (11)$$

R_{circle} is the real radius of the obstacle and R_{safe} is the distance including the robot size and a safety margin as shown in Fig.6.

According to the robot ultrasonic sensor information, the decision to find the convenient trajectory is made according to Fig.7. The decision is made as follows. If $D_{robot-obstacle} \leq R_{obs}$ then the robot follows the trajectory generated by the avoidance algorithm. Else, the trajectory remains the initial one.

The steps of the limit-cycle method are as follows:

- (1) The line l is described by equation (12).

$$ax + by + c = 0. \quad (12)$$

- (2) The distance d from the center of the obstacle to the line l is calculated using equation 13.

$$d = \frac{a * X_{obs} + b * Y_{obs} + c}{\sqrt{a^2 + b^2}}. \quad (13)$$

- (3) The desired direction of the unicycle robot is calculated at each position by equations (14) and (15).

$$\dot{x} = \frac{d}{|d|} y + x(R_{obs}^2 - x^2 - y^2) \quad (14)$$

$$\dot{y} = -\frac{d}{|d|} x + y(R_{obs}^2 - x^2 - y^2). \quad (15)$$

where x and y are the relative values of the robot position according to the obstacle. Indeed, if d

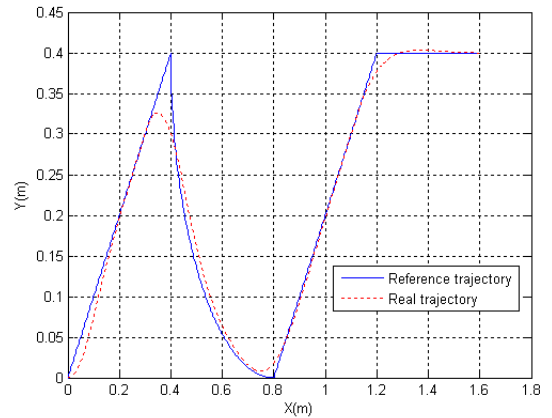


Fig. 5. Trajectory tracking with $(X_0, Y_0) = (0.3, 0.1)$

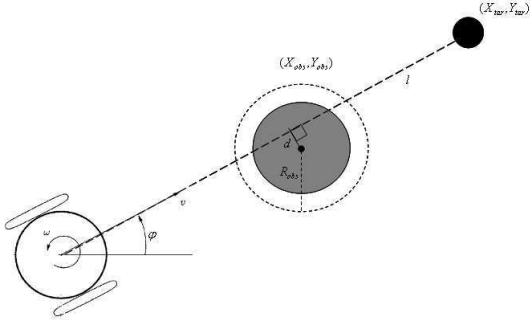


Fig. 6. Limit cycle navigation method

is positive, then the robot avoids the obstacle clockwise. Else, it avoids it in a counter-clockwise direction.

The simulation is presented in Fig.8, where the target is in $(X_{tar}, Y_{tar}) = (1, 1)$ and the initial position of the robot is $(X_0, Y_0, \varphi_0) = (0, 0, \frac{\pi}{4})$.

4. ROBOT CONTROL WITH ZIGBEE NETWORK

4.1 Zigbee 802.15.4

ZigBee is a specification for small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). Zigbee can also be defined as a low-cost, low-power, wireless mesh networking standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

The main reasons for adopting this wireless network in our application is its low power consumption together with its ad-hoc networking capabilities.

4.2 Control over wireless network

The controllers of the linear and the angular velocities presented in the previous section are discretized according to equation 16. The saturation is still used.

$$v_k = K_{v_k} \varepsilon_{t_k} \quad (16)$$

$$\omega_k = K_{\omega_k} v_k \varepsilon_{\varphi_k} \quad (17)$$

The control feedback loops are closed through a real-time network as shown in Fig.9.

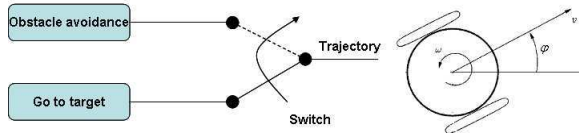


Fig. 7. Obstacle avoidance architecture

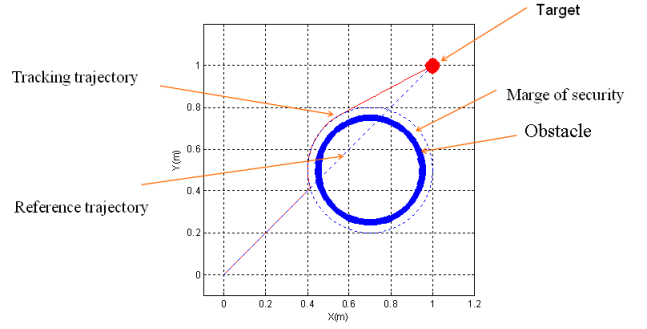


Fig. 8. Simulations of obstacle avoidance.

The aim of this section is to compare the results of the control “without a network” with the control “over the wireless network” in order to show the influence of delays generated by the integration of the wireless network. The maximum value of the random delay in IEEE 802.15.4 protocol is $\tau = 30ms$.

TrueTime simulator (Anderson et al., 2007) is used to simulate Zigbee Network. Two tasks are programmed, the first is the controller task that generates the controller flow and the second is the sensor task that generates the sensors flow. The controller task is event-triggered, which means that the controller calculates and sends the control signals V_k and ω_k only when it has received all the measures X_{mes_k}, Y_{mes_k} and φ_{mes_k} and finished computations. The sensor task is time-triggered (periodic).

4.3 Results comparison

In this section, the results obtained without the network (see section 3) and the results obtained with the wireless network are compared. Fig.10 shows the different trajectories of the robot. Three trajectories have been presented, the reference, the real trajectory of the robot without the network and the real trajectory of the robot with the wireless network. The points t_{ref}, t_r and t_{net} represent respectively the coordinates of the robot at different times (each second). In this figure, the trajectories are not really different except during the period from $t = 0$ to $t = 3s$. In this period, we note that the angular velocities of the two wheels corresponding to the real trajectory with network are higher than those corresponding to the trajectory without network. This difference is the consequence of the delay produced by the wireless network (delay can be estimated using a TrueTime function (Anderson et al., 2007)). We should precise that the wheel’s angular velocities obtained with the control over wireless network are not always greater than those obtained with the control without network ones: it depends on the position of the robot with respect to the reference.

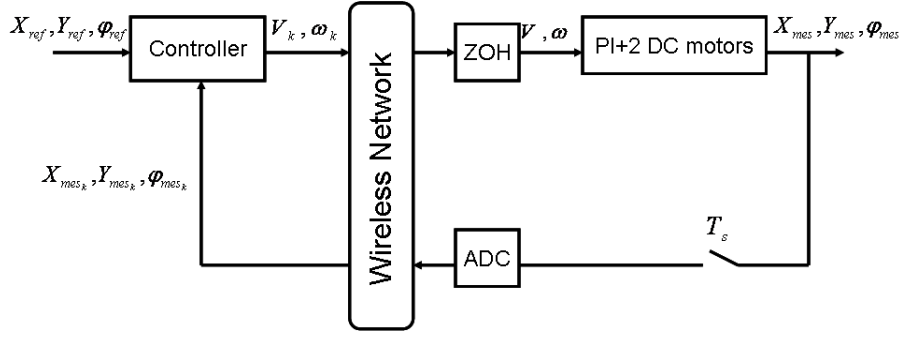


Fig. 9. Communication over wireless network

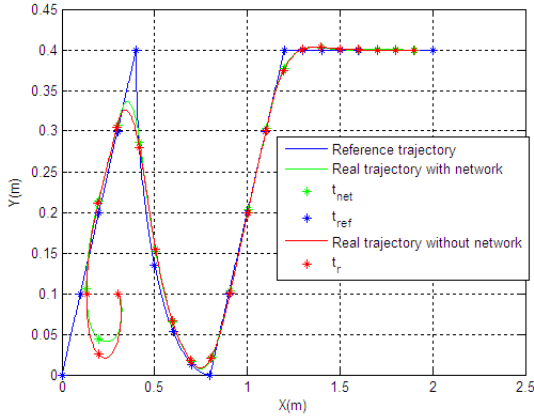


Fig. 10. Robot trajectory

4.4 Robot control with loaded network

In this section, two scenarios are proposed, the first one considers that the network is used by another robot simulated by one task called the external flow. In the second scenario, the network is used by a set of robots. The following conditions are considered:

- Use of the Zigbee Network.
- A bit rate of 250 kb/s in the physical layer.
- The sensor flow and the controller flow use $\Gamma = 512\text{bits}$. The sensor flow uses 20 % and the controller one uses 15 % of the network capacity with a sampling period T_s equal to 30 ms.
- The external task sends a flow of $\Gamma_c = 1016\text{bits}$ (the largest frame in Zigbee is 127 bytes). The use of the network capacity will depend on its period T_c .
- The Use Request Factor (URF) which represents the network load, is calculated as follows: $URF = (\frac{\Gamma}{T} + \frac{\Gamma}{T} + \frac{\Gamma_c}{T_c})$.
- With the TrueTime Simulation of ZigBee Network, there is only the CAP period (Contention Access Period which uses CSMA/CA protocol), therefore there is not a priority mode contrary to CFP (Contention Free Period) which allows Guaranteed time Slots (GTS) (see (van den Bossche et al., 2007) for more information about 802.15.4 protocol).

URF	T_c
50%	25ms
70%	11ms
100%	6ms
120%	4.7ms

Table 1. Period of the external task for each URF value

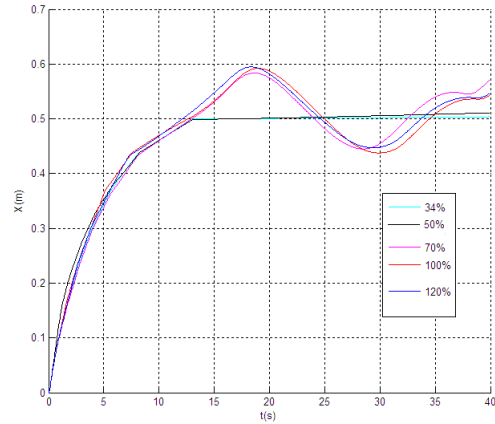


Fig. 11. Various simulations of different percentages of URF.

4.5 Results

4.5.1. Scenario 1 This subsection describes the control performance which is evaluated as a function of the Use Request Factor. Table 1 presents the different percentages of URF varying with the sampling period of the external task.

Results are shown in Fig.11. In this simulation, references are $X = 0.5\text{m}$ and $Y = 0.5\text{m}$ and the initial position is $(X_0, Y_0, \varphi_0) = (0, 0, 0)$. Only the curve describing the coordinate X is given because the results are the same for Y . It can be concluded that the control performance is degraded from 70 % of the network load. Even with more than 100%, the probability to send the control value is not zero. The degradation is explained by the delay in the control loop.

4.5.2. Scenario 2 In this scenario, a set of robots is considered, which use the same network,

Number	T_c
1	25ms
2	12.8ms
3	8.5ms

Table 2. Period of each number of external flow at 50 % of URF

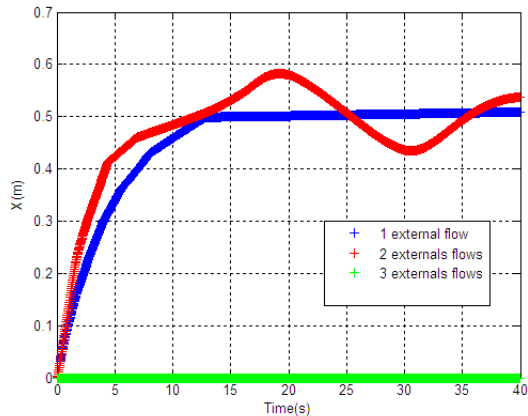


Fig. 12. Various simulations with different number of external flow nodes.

and the control performance is evaluated in function of the number of tasks using the network. In all cases, the sampling period of the external tasks is chosen such us a fixed URF of 50 % is obtained. It can be concluded that with two external tasks, the control performance is degraded and with three external tasks, the controller suffers so many data losses that the robot is not able to move at all (see Fig.12). The main reason of this degradation is that each time two tasks decide to send a message at the same time, ZigBee assigns a random time period called *Backoff* (see (van den Bossche et al., 2007) for more information) to avoid collisions. Therefore, the tasks need more time to be sent. The probability to send information for three externals flows tends to zero.

5. CONCLUSION

In this paper, the model of a unicycle robot has been studied. Algorithms of tracking trajectory and obstacle avoidance have been implemented. Finally, the study of the influence of the integration of a wireless network, ZigBee, (using Matlab/TrueTime simulator) has been studied. To study the influence of the wireless network on the system, two different scenarios have been examined. It can be concluded that delays and packet losses have a non negligible effect on the robot performance especially without GTS. Comparing with CAN network which is the most used in networked control systems, Zigbee without GTS has worse performance, more delay and aleatory transmission due to the *backoff*. The *backoff* can also influence the quality of the control over wire-

less network even if the network is not overloaded. But the stronghold of Zigbee network is long battery life and low cost, which are the main reasons for using it. The main objective of our work is to propose a communication architecture for the distributed diagnosis of a set of mobile robots. The next step is to define a communication strategy to avoid the possible overload of the network if there are many robots communicating with each others using the same network.

REFERENCES

- M. Anderson, D. Henriksson, and A. Cervin. True-time 1.5 reference manual. Technical report, Department of Automatic Control, Lund Institute of Technology, Sweden, 2007.
- C. Berbra, S. Gentil, S. Lesecq, and J.M. Thiriet. Co-design for a safe network control dc motor. In *17th IFAC World Congress*, 2008.
- J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *proceeding of IEEE*, volume 95, pages 138–162, 2007.
- B. Kim and P. Tsiotras. Controllers for unicycle-type wheeled robots: Theoretical results and experimental validation. In *proceeding of IEEE transactions on robotics and automation*, volume 18, pages 294–307, 2002.
- D. Kim and J. Kim. A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. In *Robotics and Autonomous Systems*, volume 42, pages 17–30, 2003.
- F. Lamercy and G. Caprari. Khepera *iii* user manual version 2.2. Technical report, K-Team, Switzerland, 2007.
- A. Micaelli and C. Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. INRIA report 2097, INRIA Sophia-Antipolis, 1993.
- C. Samson and K. Ait-Abderrahim. Mobile robot control. part 1 : Feed-back control of a non holonomic mobile robots. INRIA report 1281, INRIA Sophia-Antipolis, 1991.
- A. van den Bossche, T. Val, and E. Campo. Prototyping and performance analysis of a qos mac layer for industrial wireless network. In *7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, 2007.
- W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. In *IEEE Control Systems Magazine*, 2001.