A LEGO Mindstorms experimental setup for multi-agent systems

Daniele Benedettelli, Marco Casini, Andrea Garulli, Antonio Giannitrapani, Antonio Vicino

Abstract— **This paper presents an experimental setup based on the LEGO Mindstorms technology, suitable for the validation of cooperative control strategies for multiple vehicle systems. Despite its low cost, the proposed setup features good scalability and is versatile enough to be adopted for testing different solutions to a number of crucial problems arising in multi-agent systems, like formation control, motion coordination, coverage problems, etc. At the same time, it represents a challenging testbed, presenting several issues that have to be faced in a real-world scenario, such as communication constraints, control quantization, noisy measurements. As an application, the experimental validation of a recently proposed decentralized control law is presented, for the collective circular motion of a team of nonholonomic vehicles about a virtual reference beacon.**

I. Introduction

In recent years, multi-agent systems have received considerable attention for their potential in many different problems, like exploration of unknown environments, surveillance of large areas, distributed monitoring, to name just a few. This increasing interest is also supported by the large number of courses on this topic offered by academic institutions all over the world. By now, several motion coordination strategies have been proposed, and nice theoretical results have been obtained (e.g., see [1], [2], [3]). On the other hand, most of the proposed algorithms have been tested only in simulation and relatively few experimental results can be found in the literature [4], [5], [6]. This suggests the need for experimental setups where testing multi-agent system algorithms, for both educational and research activities. The LEGO Mindstorms technology [7] seems to fit particularly well to this purpose.

Although it was originally designed as an educational toy for children, LEGO Mindstorms is widely used in control systems and robotics education, thanks to its low-cost and simplicity [8]. Many educational systems have been built using this technology. For instance, a LEGO-based experiment for teaching fundamental concepts of control and digital signal processing is reported in [9], while a set of control experiences using LEGO are described in [10]. Regarding mobile robots built with LEGO, recent experiments can be found in [11], where a distributed network-based control of mobile robots is described, and in [12], where a human can collaborate in a natural way with a mobile robot.

The main contribution of the paper is to present an experimental setup, based on the LEGO Mindstorms technology, which can be adopted for the performance evaluation of different control strategies for multi-vehicle systems. It is made up of four unicycle-like vehicles and a webcam, and is controlled by a supervision system running on a desktop PC, under the MATLAB environment. Despite its low cost, the proposed setup features good scalability and versatility, while, at the same time, showing several issues that have to be faced in a real-world scenario, such as communication constraints, control quantization, noisy measurements. As an illustrative example of the proposed architecture, the results of the experimental validation of a decentralized control law are presented, for the collective circular motion of a group of agents [13].

The paper is structured as follows. In Section II the overall experimental setup is described. In Section III the collective circular motion problem is introduced and the decentralized control law under evaluation is recalled. Section IV presents some experimental results, while in Section V some conclusions are drawn.

II. Setup description

Driven by the demand of a teaching lab for robotics courses, the Control Group of the University of Siena has decided to develop an experimental setup for multiple vehicle systems. The purpose was to build an environment which allows the implementation and comparison of different algorithms for a variety of problems arising in multi-agent systems, like formation control, motion coordination, coverage problems. Special attention has been paid to versatility and easiness of use of the developed setup, in order to provide the students with a user-friendly environment for hands-on experiences on multi-agent control strategies.

After considering several possible alternatives, the final choice has fallen on the LEGO Mindstorms technology [7]. It is a cost-effective solution for rapidly building small-sized vehicles, which do not require large experimental areas. Moreover, the myriad of software tools available off-the-shelf makes robot programming easy.

The developed setup consists currently of four nonholonomic vehicles controlled by a Centralized Supervision System (CSS) running on a desktop PC.

A. Vehicles

All the vehicles are identical, except for a triangular marker placed on the top of each of them, whose purpose is to allow the CSS to detect the agent identities, and estimate their position and orientation. The robots have a differential drive kinematics and are driven by two motors, with an idle

The authors are with the Dipartimento di Ingegneria dell'Informazione, Università di Siena, Via Roma, 56, 53100 Siena, Italy. Email: danielebenedettelli@gmail.com, {casini,garulli,giannitrapani,vicino}@dii.unisi.it

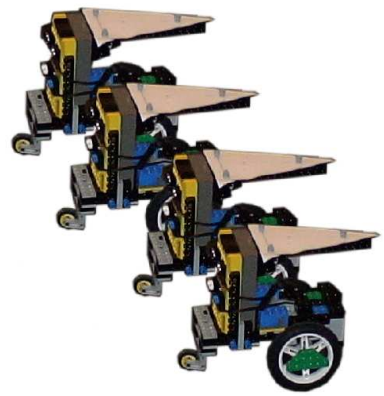


Fig. 1. LEGO Mindstorms mobile team.

wheel acting as third support (see Figure 1). Such vehicles can be modeled as unicycles

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i \\ \dot{y}_i &= v_i \sin\theta_i \qquad\qquad i = 1, \ldots, n \\ \dot{\theta}_i &= u_i, \end{aligned} \tag{1}$$

where $[x_i \ y_i \ \theta_i] \in \mathbb{R}^2 \times [-\pi, \pi)$ represents the i-th agent pose, and the linear speed v_i and the angular speed u_i are the control inputs. The motors drive the wheels with a 9:1 gear ratio, while the encoders are coupled to the motor shaft with a 1:5 gear ratio, thus ensuring enough torque and a satisfactory encoder resolution (720 ticks per wheel revolution).

The low-level control of each vehicle is in charge of a LEGO RCX programmable brick [14] equipped with a 16-bit 10Mhz H8/3292 Hitachi processor. The BrickOS real-time operating system [15] permits the execution of C/C++ programs to control the motors with 255 PWM levels, to read sensors and to communicate with the CSS via an IR serial protocol.

The encoder readings are used by the RCX for controlling wheel speeds. For each wheel, a two degrees of freedom controller is implemented to track the wheel speed references provided by the CSS. A PI feedback control is coupled with a feed-forward action, based on the estimated characteristic between the RCX PWM output and the wheel speed. Due to technological limitations (RCX numerical approximations, mechanical dead zones) vehicles cannot have a nonzero angular speed less than $0.05\,rad/s$, while the maximum achievable linear speed is about $0.07\,m/s$.

B. Centralized supervision system

The Centralized Supervision System is used to monitor and coordinate the robots during the experiments (see Figure 2). A camera placed on the lab ceiling is used to capture the motion of the vehicles and to simulate the presence of on-board sensors. Robot position, orientation and identity are detected thanks to white triangles attached on the top of each of them. The overall image processing is illustrated in Figure 3. The original image is captured at a resolution of

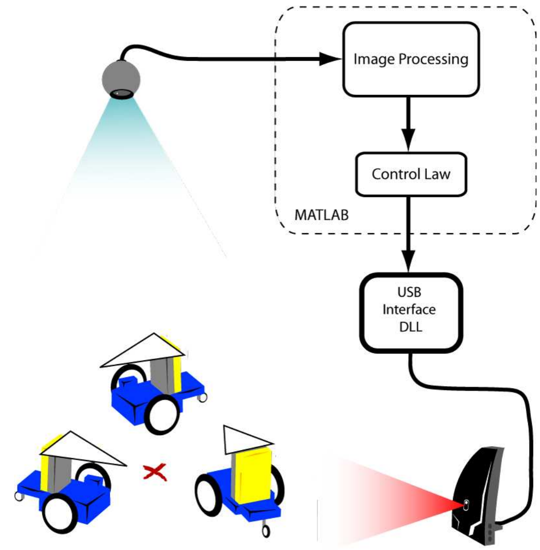
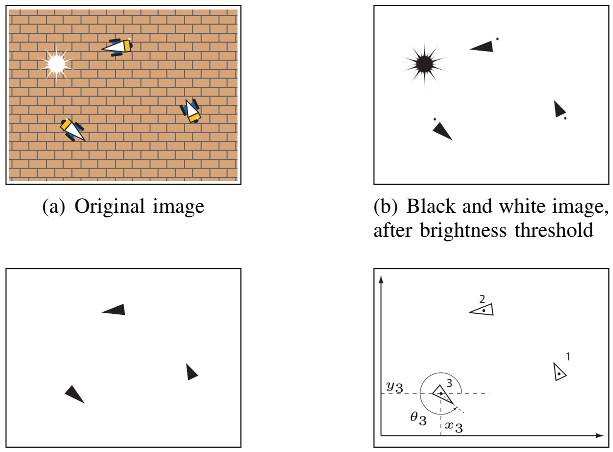


Fig. 2. Centralized Supervision System.

640×480 pixels (see Figure 3(a), where white spots represent disturbances due to ambient light). After discarding color information and applying a brightness threshold, a black and white image is obtained (Figure 3(b)). Then, the boundaries of the objects in the scene are extracted and filtered according to their shape and area. In this stage, artifacts due to light reflections are removed (Figure 3(c)). The position and orientation of a robot is estimated as the center of mass and the orientation of the corresponding triangle (Figure 3(d)). As a by-product, a unique identity is given to each robot on the basis of the area of the corresponding triangle (i.e., the first robot is the one associated with the smallest triangle and the last one is the vehicle labeled with the largest triangle). The extracted robot poses are then used to compute artificial measurements (e.g., range and/or bearing measurements among vehicles), thus simulating the presence of on-board sensors. In this phase, possible limitations of the sensors (e.g., limited field of view) can be easily taken into account. The advantage of using "virtual sensors" is twofold. Clearly, it contributes to limit the cost of each vehicle, removing the need for expensive hardware. Moreover, it increases the flexibility of the setup, making it easy experiencing with different kind of sensors or with sensors featuring different levels of accuracy. Finally, depending on the desired control law, the artificial measurements are used to compute the references of linear and angular speed to be sent to each vehicle.

Image grabbing and processing, as well as the computation of the control law, are carried out by a MATLAB function, which also sends speed references to the team via an infrared (IR) LEGO Tower, interfaced to MATLAB through an ad-hoc MEX DLL. Output commands are represented as floating point numbers, and need to be converted to 16-bit integers before being sent, in order to keep a good precision for on-robot integer arithmetic calculations. Commands for all the robots are packed together and broadcast to all vehicles at



(a) Original image

(b) Black and white image, after brightness threshold

(c) Triangle extraction

(d) Pose estimation

Fig. 3. Image processing.

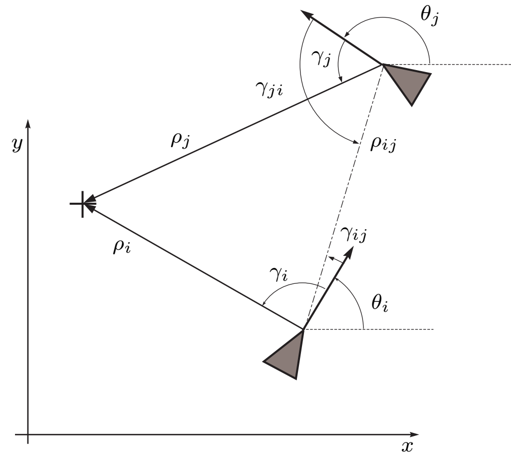


Fig. 4. Two vehicles (triangles) and a beacon (cross).

each sampling time. At the beginning of the experiment each vehicle is given an ID number according to its triangular marker, so that when a robot receives the packet, it is able to recognize which chunk contains its own data.

The centralized architecture described above has two main purposes. First, the computation of the control law can be done on a standard PC, without overloading the vehicle RCX, which is exclusively devoted to motor control. Second, the CSS provides the necessary flexibility to the setup, allowing one to test both centralized and decentralized control laws. In the latter case, it is sufficient to compute the input of each agent on the basis of the sole measurements the agent would have access to, if it was equipped with the chosen sensors. Additionally, the CSS provides also the ground truth of each vehicles, which allows one to reconstruct the collective motion of the team and to evaluate the performance of the tested algorithms.

III. Application to collective circular motion

As an example of experiment with the described setup, we consider the collective circular motion problem. In this section we will briefly formulate the addressed problem and recall the main features of the adopted control law.

Let us consider a group of n identical agents whose motion model is described by the equations (1). The forward speed $v_i = v$ is assumed to be constant and equal for all agents, while the angular speed u_i plays the role of control input for vehicle i. Each robot is supposed to be equipped with sensors providing range and bearing measurements with respect to: i) a virtual reference beacon, and ii) all its neighbors. Specifically, with reference to the i-th agent, $(\rho_i,\ \gamma_i)$ will denote the measurements with respect to the beacon, while $(\rho_{ij},\ \gamma_{ij})$ will denote the measurement with respect to the j-th agent (see Figure 4). In order to explicitly take into account sensor limitations, a *visibility region* $\mathcal{V}_i$ is defined for each agent, representing the region where it is assumed that the sensors of the i-th vehicle can perceive its neighbors. The visibility region is chosen as the union of two sets (see Figure 5):

- A circular sector of radius d_l and angular amplitude $2\alpha_v$, centered at the vehicle. It models the presence of a long range sensor with limited angular visibility (e.g., a laser range finder).
- A circular region around the vehicle of radius d_s, which models a proximity sensor (e.g., a ring of sonars) and plays the role of a “safety region” around the vehicle.

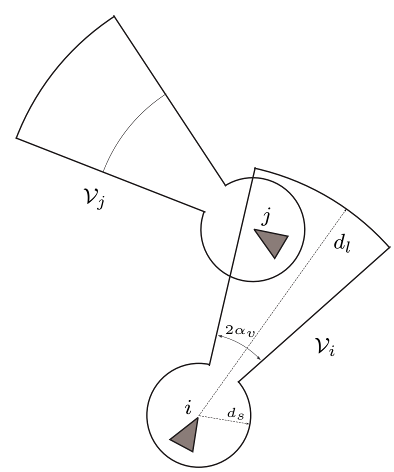


Fig. 5. Visibility region of i-th and j-th vehicle.

The objective is to design the control inputs u_i so that all the agents achieve circular motion around the beacon, with prescribed radius of rotation and distances between neighbors, while at the same time avoiding collisions.

In order to illustrate the considered control law, some definitions are in order. Let $\mathcal{N}_i$ be the set containing the indexes of the vehicles that lie inside the visibility region $\mathcal{V}_i$ of the i-th agent. Define the functions

$$g(\rho; c, \xi) = \ln\left(\frac{(c-1)\cdot\rho + \xi}{c\cdot\xi}\right) \qquad (2)$$

$$\alpha_d(\gamma;\psi) = \begin{cases} \gamma & \text{if } 0 \le \gamma \le \psi \\ \gamma - 2\pi & \text{if } \psi < \gamma < 2\pi. \end{cases} \quad (3)$$

$$\beta_d(\gamma_{ij}) = \begin{cases} \gamma_{ij} & \text{if } \ 0 \le \gamma_{ij} \le \pi \\ \gamma_{ij} - 2\pi & \text{if } \ \pi < \gamma_{ij} < 2\pi. \end{cases} \quad (4)$$

where c, ξ and $\psi \in (\frac{3}{2}\pi, 2\pi)$ are given constants. The considered control law, proposed in [13], computes the input $u_i(t)$ as

$$u_i(t) = f_{ib}(\rho_i, \gamma_i) + \sum_{j \in \mathcal{N}_i(t)} f_{ij}(\rho_{ij}, \gamma_{ij}). \quad (5)$$

where

$$f_{ib}(\rho_i, \gamma_i) = \begin{cases} k_b \cdot g(\rho_i; c_b, \rho_0) \cdot \alpha_d(\gamma_i; \psi) & \text{if } \ \rho_i > 0 \\ 0 & \text{if } \ \rho_i = 0, \end{cases} \quad (6)$$

and

$$f_{ij}(\rho_{ij}, \gamma_{ij}) = \begin{cases} k_v \cdot g(\rho_{ij}; c_v, d_0) \cdot \beta_d(\gamma_{ij}) & \text{if } \ \rho_{ij} > 0 \\ 0 & \text{if } \ \rho_{ij} = 0, \end{cases} \quad (7)$$

The constants $k_b > 0$, $c_b > 1$, $\rho_0 > 0$, $k_v > 0$, $c_v > 1$, $d_0 > 0$ are the controller parameters. In particular, d_0 is the desired distance between two consecutive vehicles when rotating about the beacon. The motivation for the control law (5)-(7) relies in the fact that each agent i is driven by the term $f_{ib}(\cdot)$ towards the counterclockwise circular motion about the beacon, while the terms $f_{ij}(\cdot)$ have a twofold aim: to enforce $\rho_{ij} = d_0$ for all the agents $j \in \mathcal{N}_i$ and, at the same time, to favor collision-free trajectories.

In the single-vehicle case, the control law $u_i = f_{ib}$ results in the counterclockwise rotation of the vehicle about the beacon, with a radius ρ_e depending on the controller parameters, for every initial configuration. For the multi-vehicle case, a sufficient condition has been derived which guarantees the local asymptotic stability of a family of team configurations corresponding to the collective circular motion about the beacon. For a thorough theoretical analysis of this control law, as well as a procedure for choosing the controller parameters, the interested reader is referred to [13].

IV. Experimental results

In this section, the results of experimental tests involving different number of vehicles are reported. The forward speed is set to $v = 0.06\ m/s$. To account for sensor limited field of view, a visibility region like that presented in Section III is assumed, with $d_l = 1\ m$ and $d_s = 0.3\ m$. The angular width α_v has been set to different values in order to simulate different sensors (see Figure 5).

During each experiment, range and bearing measurements ρ_i, γ_i, ρ_{ij} γ_{ij} are generated from the position and orientation data as described in Section II-B. Then, the input u_i of each vehicle is computed by the CSS according to the control law (5). It is worth remarking that although all the computations are carried out by the CSS, the tested control law is actually completely decentralized.

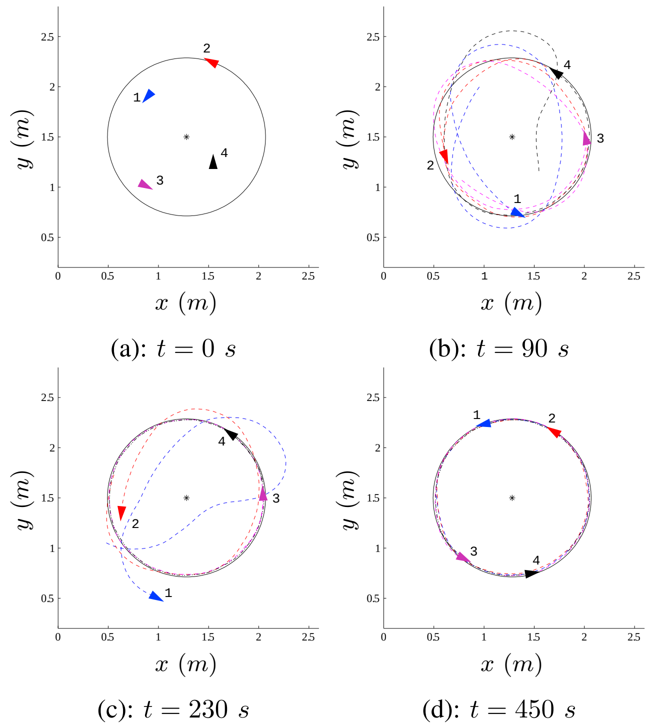


Fig. 6. Experiment A: Team configuration at different time instants. Dashed lines represent vehicle paths during the 90 seconds preceding each snapshot.

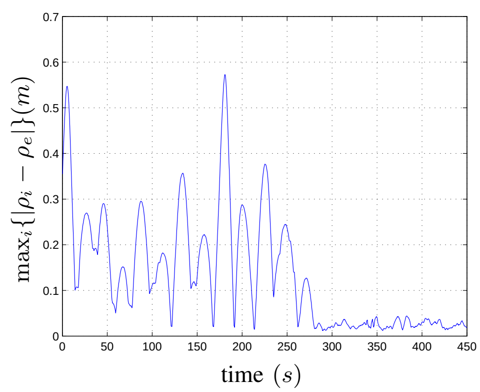


Fig. 7. Experiment A: Maximum deviation $|\rho_i - \rho_e|$ of vehicle distances to the beacon ρ_i, from the desired radius ρ_e.

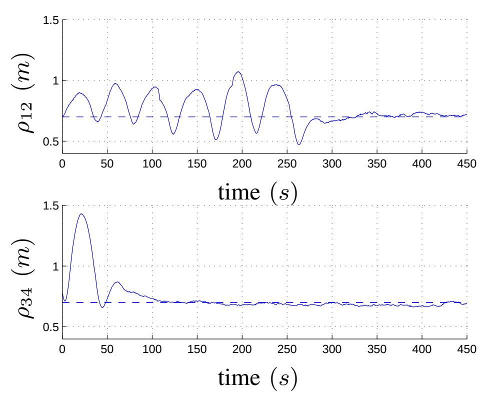


Fig. 8. Experiment A: Actual distances ρ_{12}, ρ_{34} between vehicles belonging to the same platoon (solid line), and desired distance $d_0 = 0.7$ (dashed line).

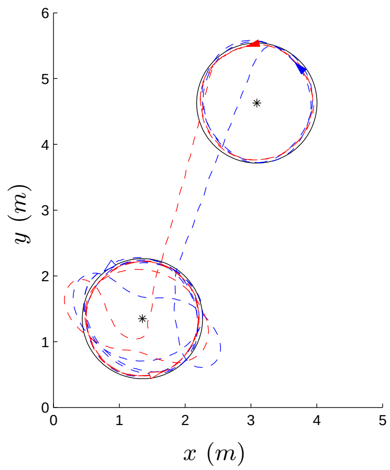


Fig. 9. Experiment B: Vehicle paths (dashed lines) and desired circular paths (solid line) about the switching beacon (asterisks). Filled triangles represent the vehicle initial poses, empty triangles represent the final vehicle poses.

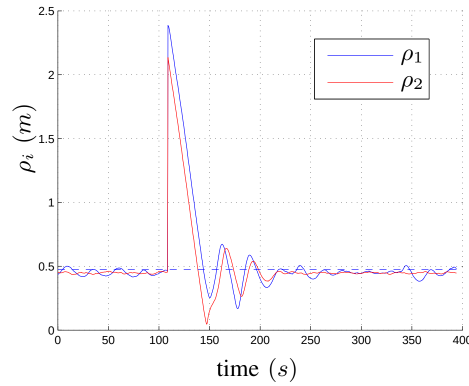


Fig. 10. Experiment B: Actual distances ρ_1, ρ_2 of the vehicles to the switching beacon (solid lines) and desired radius $\rho_e = 0.47$ (dashed line).

A. Static beacon

A first set of experiments has been carried out with four vehicles and $\alpha_v = \pi/4$ (Experiment A). The following controller parameters have been used: $\psi = 290°$, $k_b = 0.16$, $\rho_0 = 0.48\ m$, $c_b = 2$, $k_v = 0.3$, $d_0 = 0.7\ m$, $c_v = 2$. This choice of k_b and ρ_0 corresponds to a circular motion of radius $\rho_e = 0.79\ m$, while d_0 models a desired displacement between vehicles in circular motion of $0.7\ m$. Figure 6 shows four snapshots of the team evolution during a typical run. In this experiment, the vehicles end up in rotating around the beacon in two separate platoons, each one made of two agents. After about 300 seconds, the motion of all the agents stabilizes on the desired circle, as shown in Figure 7. Moreover, the separation between vehicles belonging to the same platoon eventually approaches the desired value d_0. In this case, agents 3 and 4 converge faster to the steady-state, than agents 1 and 2 (see Figure 8).

Several other experiments have been carried out over teams of 3 and 4 vehicles, with different visibility regions, controller parameters and initial vehicle poses. As expected, the final distribution of the robots in separate platoons depends on the initial configuration of the team, while the duration of the transient is mainly influenced by the number of robots in the team.

B. Moving beacon

A second experimental campaign has been carried out in order to evaluate the group behavior in case of moving beacon. To this purpose, two scenarios have been considered.

In the first one, the virtual beacon is allowed to instantaneously jump to a different position. The sequence of beacon locations can be thought of as a set of way points, about which the agents have to rotate at different time instants (e.g., a set of regions of interest or targets to be monitored). A similar scenario has been considered in [16], but employing two different control laws, one to enforce circular motion and one to track the new position of the beacon. Results of one of such tests are summarized in Figures 9-10 (Experiment B). The field of view of the simulated sensors and the controller parameters are the same as Experiment A, except for $d_0 = 0.4\ m$, $\rho_0 = 0.25\ m$, $k_b = 0.2$, resulting in a desired radius $\rho_e = 0.47\ m$. At the beginning of the experiment, the virtual reference beacon is located at position $(x,\ y) = (3.09,\ 4.63)$, and the two vehicles start from an initial configuration close to the equilibrium one. As long as the beacon does not move, the agents go on rotating on the desired circle (see upper-right circle in Figure 9). Suddenly, at $t = 109\ s$ the position of the virtual beacon switches to $(x,\ y) = (1.35,\ 1.35)$, thus pointing out that the target has changed. As a consequence, both vehicles leave the circular trajectory and point straight toward the new beacon location (see linear stretch of the trajectories in Figure 9). After a transient, both vehicles eventually settle on the circle of desired radius centered at the new beacon location (see left-bottom circle in Figure 9). This is confirmed by Figure 10, where the distance of each vehicle to the reference beacon is shown. It is worth remarking that the transition between circular and linear motion performed by the vehicles is achieved without making any changes in the control law.

In the second scenario (Experiment C), the beacon acts as a moving target which must be tracked by the team. In this experiment (carried out with the same controller parameters of Experiment B), the beacon is initially placed at $(1.74,\ 1.74)$, with the two vehicles rotating on the desired circle of radius $\rho_e = 0.47\ m$ (see left-bottom circle in Figure 11). After $53\ s$, the beacon starts moving straight at constant speed $(0.0055\ m/s)$, covering about $3.44\ m$ in $630\ s$. As a result, the agents keep on rotating about the beacon at roughly the desired distance ρ_e (see Figure 12, top), describing a circle translating according to the beacon motion. In this case, also the team cohesion is preserved, since the agents start in a single platoon and so remain during the whole experiment, with an actual inter-vehicle distance close to the desired one $d_0 = 0.4\ m$ (see Figure 12, bottom). Differently from the previous scenario (where the beacon jumped between different positions), in this case the agents

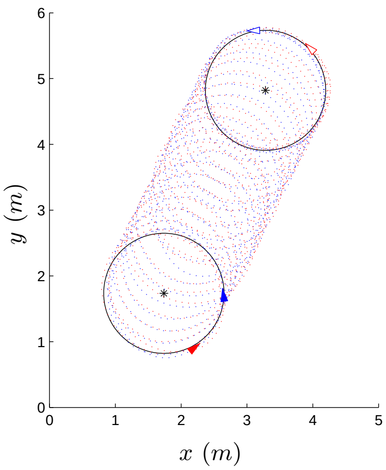


Fig. 11. Experiment C: Vehicle paths (dotted lines) and desired initial and final circular paths (solid line) about the moving beacon (asterisks). Filled triangles represent the vehicle initial poses, empty triangles represent the final vehicle poses.

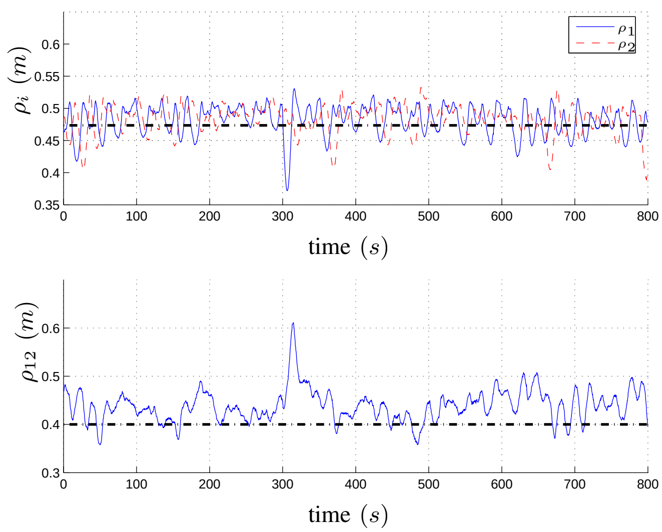


Fig. 12. Experiment C. Top: Actual distances ρ_1, ρ_2 of the vehicles to the moving beacon (solid and dashed lines), and desired radius $\rho_e = 0.47$ (dash-dotted line). Bottom: Actual distance ρ_{12} between the vehicles (solid line) and desired distance $d_0 = 0.4$ (dash-dotted).

are about 10 times faster than the beacon, thus allowing them to track it in circular motion.

The overall experimental validation has shown that the proposed setup can be actually adopted for testing control laws for multi-agent systems, despite its technological limitations (poorly accurate measurements, delays, nonlinear phenomena affecting actuators).

V. Conclusions

In this paper, a cost-effective experimental setup, based on the LEGO Mindstorms technology, has been presented. Currently, it is made up of four unicycle-like vehicles, controlled by a Centralized Supervision System. A downward-looking webcam is used to extract the pose of the agents, as well as to simulate the presence of on-board sensors. This information is processed by the supervisor to compute the control inputs which are then sent to each vehicle via an infrared tower. The proposed setup is pretty versatile and easy to use, making it suitable for testing different control strategies for multi-agent systems. As an example, the collective circular motion problem about a reference beacon has been considered and some results of the experimental validation of a recently proposed decentralized control law have been reported.

Currently, the experimental arena is going to be extended by adding a further wide-angle webcam, in order to enlarge the covered area. Experimental tests are being carried out for internal/external calibration of the cameras, as well as for correction of distortion. Next planned steps are the migration to the LEGO NXT technology and the integration of the developed experimental setup into the Automatic Control Telelab [17] to make it world-wide accessible via the Internet.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[2] Z. Lin, M. E. Broucke, and B. A. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 49(4):622–629, April 2004.

[3] R. Sepulchre, D. A. Paley, and N. E. Leonard. Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control*, 53(3):706–719, 2008.

[4] J. A. Marshall, T. Fung, M. E. Broucke, G. M. T. D'Eleuterio, and B. A. Francis. Experiments in multirobot coordination. *Robotics and Autonomous Systems*, 54(3):265–275, 2006.

[5] W. Ren and N. Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.

[6] S. Mastellone, D.M. Stipanovic, C.R. Graunke, K.A. Intlekofer, and M.W. Spong. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research*, 27(1):107–126, 2008.

[7] The LEGO Group. The Lego Mindstorms home page. [Online]. Available: http://mindstorms.lego.com.

[8] P.J. Gawthrop and E. McGookin. A LEGO-based control experiment. *IEEE Control Systems Magazine*, 24(5):43–56, 2004.

[9] B. S. Heck, N. S. Clements, and A. A. Ferri. A LEGO experiment for embedded control system design. *IEEE Control Systems Magazine*, 24(5):61–64, 2004.

[10] P.J. Gawthrop and E. McGookin. Using LEGO in control education. In *Preprints of 7th IFAC Symposium on Advances in Control Education*, Madrid, Spain, June 2006.

[11] A. Valera, M. Vallés, P. Albertos, R. Simarro, C I. Benítez, and Llácer. Embedded implementation of mobile robots control. In *Proceedings of the 17th IFAC World Congress*, pages 6821–6826, Seoul, South Korea, July 2008.

[12] S. A. Green, X. Chen, M. Billinghurst, and J. G. Chase. Collaborating with a mobile robot: An augmented reality multimodal interface. In *Proceedings of the 17th IFAC World Congress*, pages 15595–15600, Seoul, South Korea, July 2008.

[13] N. Ceccarelli, M. Di Marco, A. Garulli, and A. Giannitrapani. Collective circular motion of multi-vehicle systems. *Automatica*, 44(12):3025–3035, 2008.

[14] K. Proudfoot. RCX internals. [Online]. Available: http://graphics.stanford.edu/∼kekoa/rcx/.

[15] M. L. Noga. BrickOS. [Online]. Available: http://brickos.sourceforge.net.

[16] D. Paley, N. E. Leonard, and R. Sepulchre. Collective motion: bistability and trajectory tracking. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 1932–1936, Nassau, Bahamas, 2004.

[17] M. Casini, D. Prattichizzo, and A. Vicino. The Automatic Control Telelab: A web-based technology for distance learning. *IEEE Control Systems Magazine*, 24(3):36–44, 2004.