Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Leader-Follower String Formation using Cascade Control for Mobile Robots

Stefana M. Cristescu, Clara M. Ionescu, *Member, IEEE,* Bart Wyns, Robin De Keyser, and Ioan Nascu

***Abstract*— This paper proposes a simple yet effective control algorithm for a platoon of "car-like" robots. The formation used is a line, leader-follower formation, i. e. each robot is a follower for the previous robot and a leader for the next robot. The formation is implemented using speed measurements from optical encoders and distance measurements from image processing, but there is *no communication between robots*. String stability analysis on the resulting formation is performed. The results of the proposed method are verified by both simulations in Matlab(R) as well as measured data from the actual mobile robots.**

I. Introduction

ONE direction of research extended from mobile robots in the last ten years is multi-robot systems. One of the first implementations of such a system was done in the Swarm-bots project and resembled to an insect colony [1]. The multi-robot system consists of more than two robots in a formation. Each robot in the formation is an autonomous system, unmanned, and can be specialized for one or more particular tasks. The system formed by the group of robots can be designed to achieve different goals and can be adapted in order to accomplish different specifications.

Once the multi-robot system started to develop, the idea of *formation control* appeared. The leader-follower formation can have different forms, starting from a line, where each robot is a follower of the robot in front and a leader for the next robot. Alternatively, it can also consist of one single leader followed by all the other robots. Combinations between these two types of formation can lead to various geometric shapes and can morph in order to avoid obstacles or to pass through narrow spaces [2]. A more complex approach of robot formations consists of replacing the leader with a master system to supervise the robots and to give them tasks [3].

The most common practical application of leader follower formation is represented by a convoy of autonomous cars on a highway. Nowadays, the accidents on the highways caused by the traffic jam are daily reported. The cause of collisions, traffic jams, and stop-and-go waves, making traffic prone to slinky-type instabilities, is the time delay introduced by the human drivers as presented in [4]. This paper addresses the problem of designing a leader-follower framework with sensor-limited robots that will serve as a proof of concept at a microscopic level for such a real formation on a highway.

An implementation for a convoy of autonomous robots is presented in [5] in which the distance between the robots is approximated using real-time image processing by applying the rank transform. The control algorithm runs on-board implemented in an FPGA. In [6], the authors used a group of Pioneer 3AT mobile robots to validate their platooning control strategy where the vehicles have information only about the distance and orientation of the preceding one. A global method for formation control using a panoramic camera is compared to local formation control where the distance and orientation between the robots is measured using a laser range finder. The conclusion is that the followers are able to track the leader much better in the global implementation since information about all the robots is available at a central point.

This research work has also an educational purpose. The advantage of this implementation is that the robots are communicating with a server computer. Moreover, all the techniques used are the basic algorithms taught in any control engineering course. Therefore, the project can be easily incorporated into a remote laboratory [8].

The paper is organized as follows. Section II presents the mobile robots used for this work with their hardware architecture in order to explain the information we have to control the system. In section III, we describe the control algorithm, the step response identification and the design of the controllers. String stability analysis and remarks are included in Section IV, followed by simulations and results. Finally, in Section V we discuss the conclusions and future research topics.

II. Mobile robots

A. System protocol

The system depicted in Fig. 1 considers one leader, and one follower; however, this can be extended with more robots in line formation. From the general kinematics of any "car like" robot we can divide the motion in lateral (X axis) and longitudinal motion (Z axis), [7]. One controller is used for

Manuscript received January 9, 2012.

Stefana Cristescu is with theEeSA - department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 913, 9052 Gent, Belgium (e-mail: StefanaMiruna.Cristescu@UGent.be).

Clara Ionescu is with the EeSA - department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 913, 9052 Gent, Belgium(e-mail: ClaraMihaela.Ionescu@UGent.be).

Bart Wyns is with the EeSA - department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 913, 9052 Gent, Belgium (e-mail: Bart.Wyns@UGent.be).

Robin de Keyser is with the EeSA - department of Electrical energy, Systems and Automation, Ghent University, Technologiepark 913, 9052 Gent, Belgium (e-mail: Robain.DeKeyser@UGent.be).

Ioan Nascu is with the department of Automation and Computer Scince, Technical University of Cluj Napoca, G. Baritiu 26-28, 400027, Cluj-Napoca, Romania (e-mail: Ioan.Nascu@aut.utcluj.ro)

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

each axis, although there is a strong coupling between them. For the leader we use only lateral controller because we do not measure the movement on the X axis. The leader has the camera oriented to the floor and it follows a colored line, which in our test bed it is a green line. The followers track the colored marker attached to the back of the previous robot, i.e. a red rectangle. The data for both lateral and longitudinal controllers are extracted using a camera and image processing techniques. The robots are affected by a manifold of disturbances, making the control more challenging.

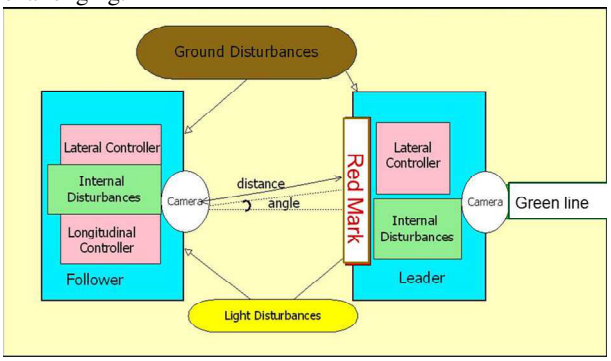

Fig. 1. Schematic representation of the leader-follower formation control system

B. Hardware architecture

The type of robot used is Surveyor SRV-1 produced by the Surveyor Corporation[1]. Surveyor SRV-1 is a differential tracked mobile robot with tank-like treads. The treads are rotated by 4 DC motors, two on each side. Between the shaft of the motor and the wheel that moves the tread there is a gearbox with reduction ratio of 100:1.

A major drawback of this robot is the lack of wheel encoders. Different characteristics of the motors will cause the left track and the right track to rotate at different speeds in open loop, resulting in curved directions. For this reason we decided to mount on the robot optical encoders, to enable speed measurement and to control the lateral direction. Since the main processor is responsible for the image processing task, adding to it the interruptions from an additional sensor would increase the computation time and would challenge the requirements for the sampling period. Additionally, one processor is limited to only one thread. To overcome these issues, we propose a simple and cost-effective solution: adding another microprocessor on the robot in order to implement multi-threading and compute in parallel at a lower sampling period for an inner loop.

The communication between the two microprocessors is done via the serial port. The main processor is responsible for maintaining the formation using image processing and for sending information (speed reference) to the slave processor for the speed control. The second processor is used for measuring the speed using the signals form the wheel encoders and for all the computations needed to control the velocity of the mobile robot. The slave processor commands the motors using a controller that generates PWM inputs.

The range of the speed of the robot is between 10 cm/s – 60 cm/s and has a nonlinear characteristic, with a dead-zone in the beginning due to the Coulomb friction. This poses additional challenges to the control objective.

C. Image processing

The information (distance and angle) for the outer feedback loop is extracted via image processing as proposed in [8]. The principle of image processing is depicted in Fig. 2. The first step in the detection algorithm is to recognize the blob having the specified color. The second step consists of extracting the upper left (x_1, y_1) and lower right (x_2, y_2) corners of this blob on a fully black background.

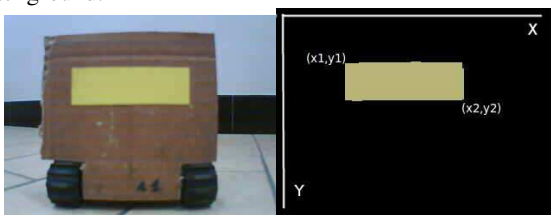

Fig. 2.Data extraction from image processing: on left side, the raw image taken by the camera and on right side the image after color segmentation, with the points of interest extracted.

For the *lateral control*, we want the robots to be aligned, i.e. the angle formed between two robots to be equal to 0^0. In practice, we consider that this goal is achieved if the average of x_1 and x_2 is equal to the x_c, where (x_c, y_c) is the center of the entire image seen by the camera (Fig. 2 on the left).

For the *longitudinal control* (i.e. specified distance between the robots), we consider the height of the blob, which is the difference between y_1 and y_2. Based on a lookup table we determine the distance to the robot in front. The relationship between distance and pixels was established experimentally, by moving the robot to different positions relative to the colored mark. The resulting characteristic is depicted in Fig. 3. If the distance exceeds 45 cm, the measurements are no more reliable because one pixel can correspond to more than 5 cm.

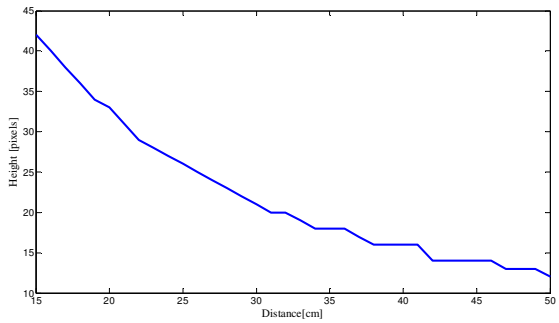

Fig. 3. Schematic representation of the leader-follower formation control system

[1] http://www.surveyor.com

III. String Formation Control

A. Control Structure

The overall control can be split in a two levels hierarchical controller.

On the upper level, the robot decides what action to take according to the information extracted from image processing. Adding the action of the lateral controller to one side of the robot and subtracting it from the other side, is a common approach to decouple the lateral and longitudinal movement [8]. In order to compute an accurate controller for the longitudinal movement, a model is required. Hence, for the purpose of system identification, we must have a clear input signal that should not be affected by the signal given by the lateral controller. Therefore this motivates our decision to add the compensation for the lateral movement only on one side, as shown in Fig. 4.

Low-level closed loop control is responsible for the speed of rotation of the tracks. This control compensates for the bias that causes the robot to deviate from a straight path. It also provides information about the actual linear speed and angular velocity of the robot, integrating these in the higher

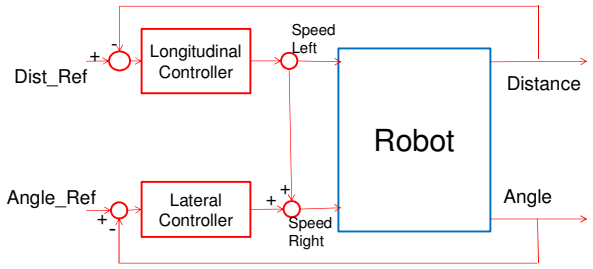

Fig. 4 Schematic representation of the lateral and longitudinal control.

tracking control level.

The speed control for the left side is depicted in Fig. 5 and similarly for the right side. We compute the velocity of the car as being the average between the left and right side treads velocities. The angular velocity is then the difference between these two, divided by l_A, which is the distance between the left tread and the right tread.

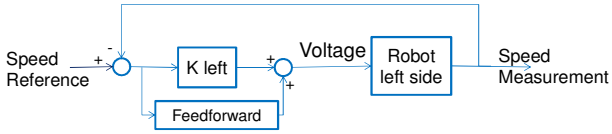

Fig. 5. Speed control loop.

Once the overall speed is obtained, its integral represents the position in absolute coordinates. From the camera we read the relative distance to the robot in front, which is:

$$D=P_2-P_1 \quad (1)$$

where P_2 is the absolute position of the robot in front and P_1 is the absolute position of the follower, as shown in Fig. 6. Hence, the position of the robot in front is a disturbance for the consequent robot.

Integrating all the elements presented above we obtain the complete control schematic as described in Fig. 7.

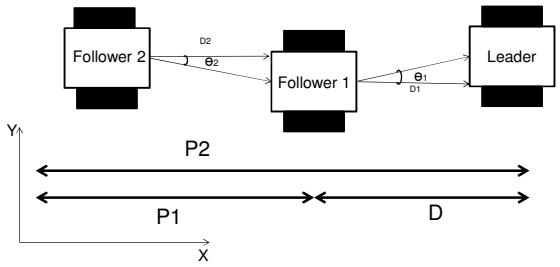

Fig. 6. Relative distance computation.

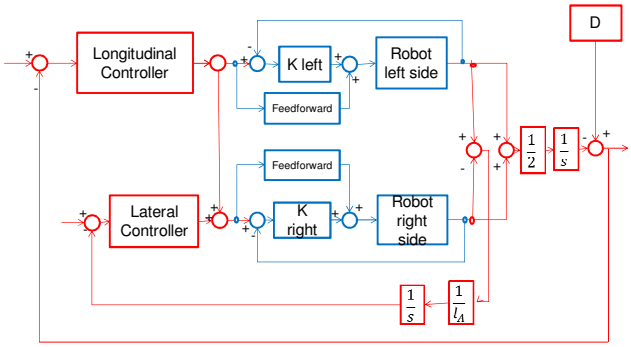

Fig. 7. The complete control algorithm.

B. Speed control

As explained in the previous subsection, the speed control is an inner loop (Fig. 5) in the cascade control schematic (Fig. 7). The inner loop is based both on feedback and on feedforward. The control output is the actual commands to the motor, which are based on a PWM signal with the duty cycle from 0 to 1023. As explained in section 2, the feedback information is supplied by the optical encoders.

It is necessary to apply data filtering, since the signals from the encoders are very noisy. The chosen filter is a moving average, because of its efficiency and low computational cost.

After filtering the signals, the next step is to add the feedforward, which makes the overall response much faster and prevents the command to be negative. In order to compute the feedforward command, the static characteristic from voltage to speed has to be determined (Fig. 8).

One of the requirements of the cascade control is that the inner loop should have much faster dynamics than the outer loop. In our test-bed, for the outer loop we have the sampling period 30 ms (20 ms, the time needed for image processing + 10 ms delays from the communication between the two microcontrollers). Therefore, the sampling period for the inner loop should be smaller than 8 ms. However, for lower speed, around 10 cm/s, we have an impulse triggered from the encoder only every 10 ms. The solution is to vary the sampling time for the inner loop between 2 ms and 10 ms, i.e. every time the encoder reads a new value. However, in most cases this value is around 3

ms, thus quasi-constant. This approach would satisfy the condition to implement cascade control.

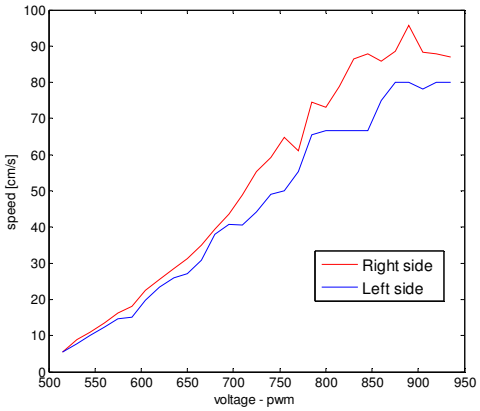

Fig. 8.Static characteristics for the motors.

The final step in controlling the speed is to implement a P controller with gain=13 on the PIC microprocessor. The advantage is that the major disturbances from this system are rejected in the inner loop, without affecting the dynamics of the outer loop. There is of course a steady state error, in this case of 10%, introduced by the lack of an integrative term. This steady state error is corrected by the outer loop.

C. *Lateral control*

With the speed control at hand, the next step consists of making the mobile robot drive straight. The schematic of the lateral control loop is presented in Fig. 9. The gain is tuned by means of trial-and-error with a final value of 1.5. The experiments for tuning the gain consist in applying a step signal on one side and compensate for the error of the lateral movement on the other side, with respect to the center of the image on X axis (80 [pixels] for resolution of 160x120 [pixels]).An illustrative example of this gain tuning is given in Fig. 10 and it is similarly determined for all the robots in the formation. Notice that the present paper is focused more on the longitudinal control. The lateral control is used just to adjust the formation therefore a simple P controller is sufficient.

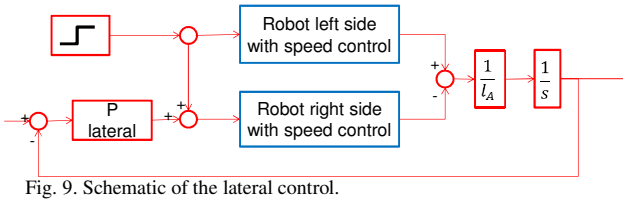

Fig. 9. Schematic of the lateral control.

D. *Step response identification*

Since it is a simple process, based on DC motors and only P controllers, one can represent it by a first order transfer function. In this hypothesis, we apply a step signal to the input and based on its response the identification is performed, [10]. The result obtained in this way is not accurate and introduces a difference between simulations and real robots. Nevertheless, it gives an idea about the process and makes the controller design more reliable. It is also important to notice that the identified transfer function is from reference speed to output speed, including the lateral control.

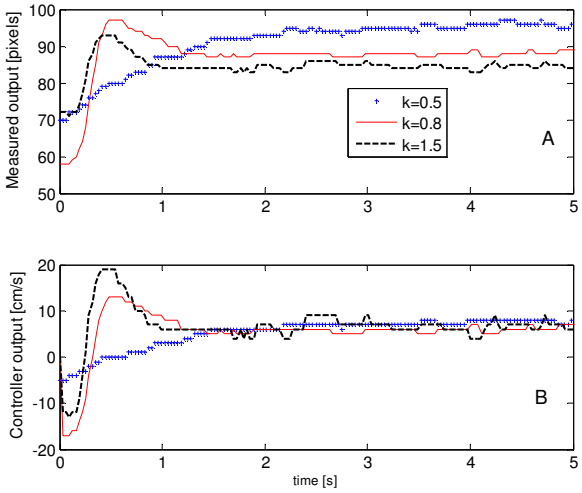

Fig. 10. Gain tuning for lateral control: (A) Distance from the center of the image [pixels], and (B) Controller output – speed [cm/s]

Fig. 11 shows the block scheme of this experiment. The part in the green contour is considered a black box. The speed is controlled internally and computed based on the kinematics formulas. Fig. 12 depicts the result of the identification.

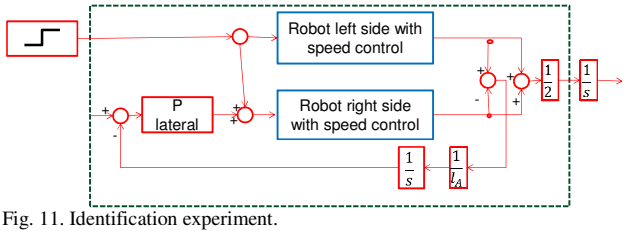

Fig. 11. Identification experiment.

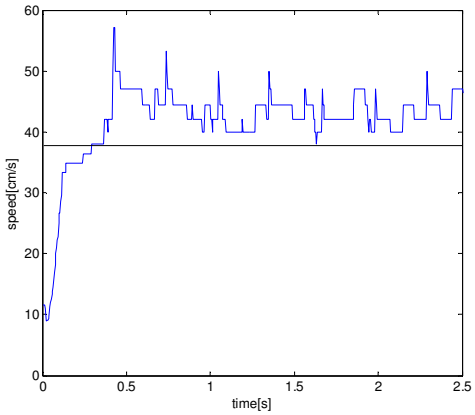

Fig. 12. Output response for step input

After the identification we have the following transfer function:

$$H(s) = \frac{1.24}{0.1s+1} \quad (2)$$

to which we need to add the integrator from the position output.

E. Longitudinal control

For the longitudinal movement of the mobile robot we use the transfer function from (2). The full control scheme is depicted in Fig. 7. The reference is the desired distance between the mobile robots and the control effort is the speed. The feedback is done via the camera, as presented in the subsection A, equation (1). In this case, notice that the distance D can be controlled only by controlling the position P_1 since there is no communication between the robots. Therefore, the position P_2 is considered as output disturbance. Since it represents a position in time, the disturbance will be a ramp function.

Obviously, a P controller in the closed loop has a steady state error because of the ramp disturbance, which requires a double integral action on the forward path in order to be rejected. One integrator exists already in the process, the other one is added in the controller.Because the system has both input and output constraints we use a PI controller with anti-windup strategy. The tuning of the PI controller is done using computer-aided design, namely the Frequency Response Toolbox (FRTool) [11]. A snapshot from this tool is depicted in Fig. 13, which represents the frequency response, i.e. Nichols plot. The blue line corresponds to the frequency response of the controller and the process in open loop, designed to have the specified performances, i.e overshoot, settling time, robustness. The design is validated against step disturbances. However, in our case we must deal with ramp disturbances.

Controller transfer function is:

$$Hc(s) = \frac{3(s+0.5)}{s} \quad (3)$$

IV. SIMULATIONS AND RESULTS

The stability of the system is a major aspect to be analyzed whenever we deal with control problems. In a platoon of cars, the notion is extended to the string and the principal phenomena are the error propagation and the disturbances attenuation along the string formation. The system discussed in this paper is connected as depicted in Fig. 14.

The transfer functions are different, and the gain of the first 3 robots is larger than 1, and for the last 3 it is smaller than 1. Also, the time constant is varying from one robot to another, and as we increase the number of the robots in the formation, the behavior of the last robots is similar to that of a high order transfer function (i.e. serial connection Fig. 14).

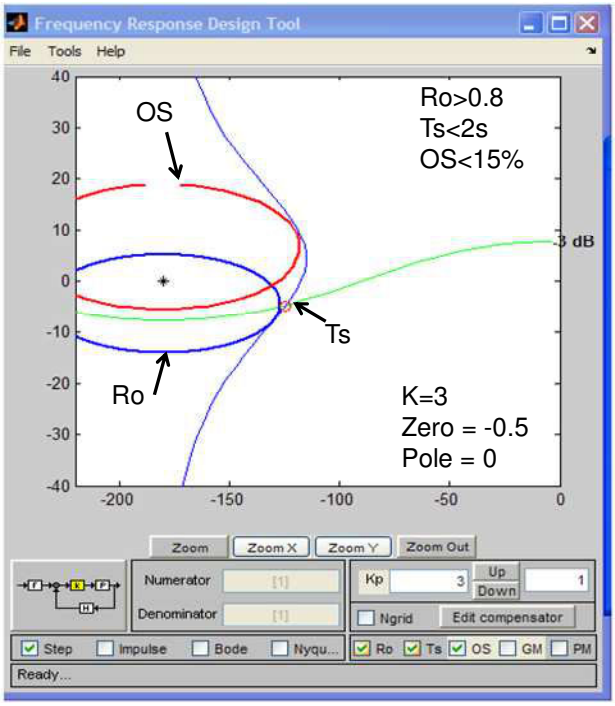

Fig. 13. Illustrative example extracted from FRTool, showing the frequency response of a controller and process based on the Nichols chart. Ro – robustness Ts – settling time, OS – overshoot.

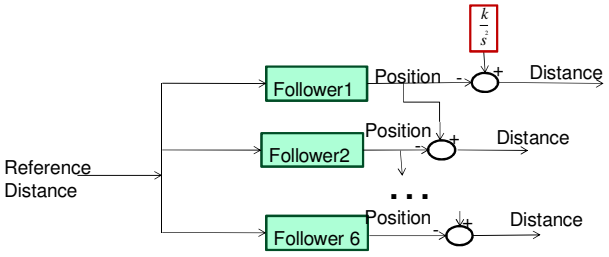

Fig. 14. Formation principle

It was proven in [12] and in [13], that without communication between vehicle in an infinite string with identical vehicles, and with constant desired distance, one can observe "string instability". Since our test bed does not have communication between robots and the distance reference is constant, we analyze the stability for a string with the 7 robots available via simulations and experimental results.

The originality of this study consists in taking into account the realistic situation, when there are no identical models for the cars. Therefore, the parameters of the transfer function will vary from one robot to another. The controller was designed only for one, but as shown in the previous section, it is robust enough to be applied for all the robots. Fig. 15 presents the results of the mobile robots formation simulated in Simulink(R), Matlab(R). The simulations are based on the linear transfer functions presented in this paper, and do not take into account the nonlinearities of the actual robots, i.e. friction, inertia etc.

This, along with quantization errors, leads to a difference between simulations and real life experiments. The overshoot is much bigger than 15% as depicted by the FRTool, because the system is affected by a ramp disturbance, instead of step disturbance. Another challenge of the closed loop is the input and output constraints. As explained in section II.C, the distance is read via the camera. The operating range for estimating correctly the distance is 0 -45 cm. These saturations are visible in Fig.15, as well as in Fig. 16-17.

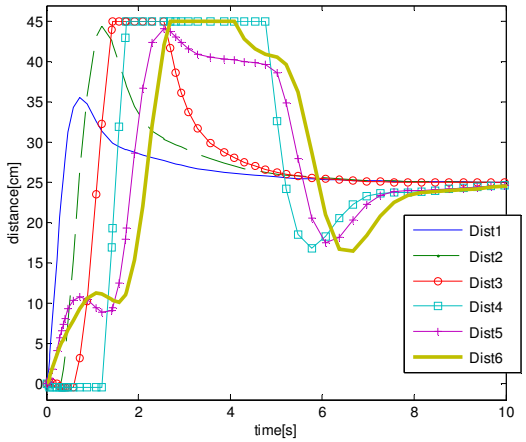

Fig. 15. Simulations results from Simulink model

Fig. 16 presents the experimental tests on the real robots. In this figure, it is important to notice that the controller output remains on zero until the leader robot gets to a certain distance where the blob can be seen i.e. the color is not distinguished well if the camera is too close. Consequently, this fact leads to non-zero initial conditions.

Fig. 17 depicts the results on the real robots when a step disturbance is applied at time 50 s. The step disturbance applied has the meaning of a change in the speed of the leader robot. All controllers reject the disturbance effects and the robots maintain the reference distance. Notice the change in the steady state values on the control output. These values are now closer to each other which suggest we are in a different operating point. At lower speed (second operating point) the gains are also closer to each other. Therefore, robustness design and analysis are more challenging at higher speed.

V. CONCLUSIONS

This paper presents a control strategy for the implementation of formation control applied to mobile robots. The system is maintained in string formation with a robust cascade PI-P controller, despite significant disturbances and modeling errors. Additionally, in this paper we apply an original solution for the compensation of the lateral movement.

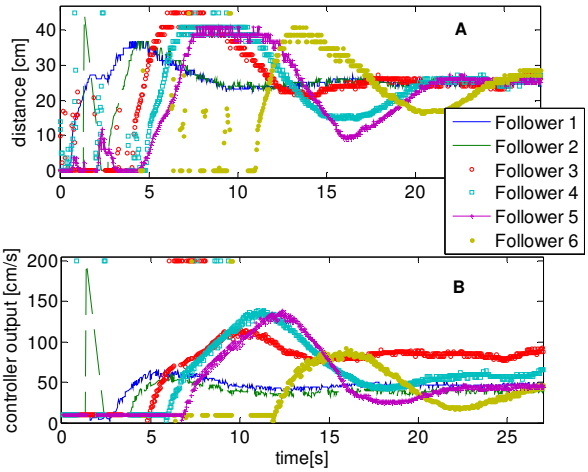

Fig. 16. Initial start up: (A) Measured output and (B), controller output for 25 cm distance reference

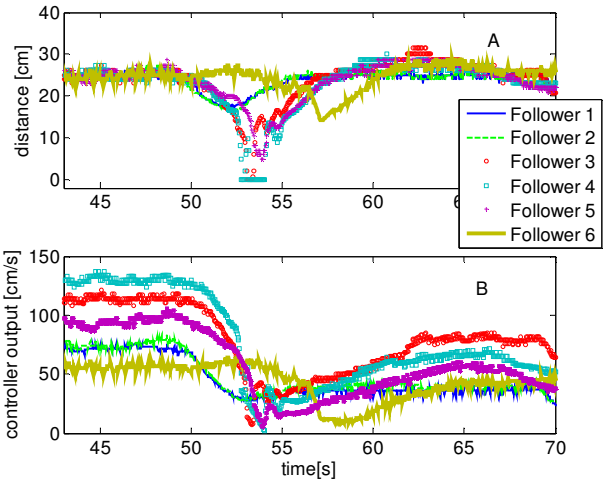

Fig. 17. Disturbance rejection: (A) Measured output and (B), controller effort for 25 cm distance reference

REFERENCES

[1] E. Bonabeau, M. Dorigo, G. Theraulaz,"Swarm Intelligence: From Natural to Artificial Systems". Oxford University Press, New York, NY (1999)

[2] D. Cruz, J. McClintock, B. Perteet, O.Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control - a multivehicle platform for research in networked embedded systems," *Control Systems Magazine, IEEE*, vol. 27, pp. 58 -78, 2007.

[3] M. Henke, M. Tichy, T. Schneider, J. Bocker, and W. Schafer, "Organization and control of autonomous railway convoys," *9th International Symposium on Advanced Vehicle Control*, pp. 1-6, 2008.

[4] R. Sopahi, S.I. Niculescu, C.T. Abdallah, W. Michiels, K. Gu, "Stability and stabilization of systems with time delay", *IEEEControl System Magazine*, pp. 38-65, February 2011.

[5] Y. Han, and H. Hahn, "Visual tracking of a moving target using active contour based SSD algorithm," *Robotics and Autonomous Systems*, vol. 53, pp. 265 -281, 2005.

[6] G. Klanccar, D. Matko, and S. Blazic, "Wheeled mobile robots control in a linear platoon," *J Intell Robot Syst,* vol. 54, pp. 709 – 731 ,2009.
[7] M. Nitulescu, "Theoretical Aspects in Wheeled Mobile Robot Control*", IEEE International Conference on Automation, Quality and Testing, Robotics, pp. 331 – 336, 2008.*
[8] D. V. Neamtu, E. Fabregas, B. Wyns, R. De Keyser, S. Dormido, and C. M. Ionescu, "A Remote Laboratory for Mobile RobotApplications", 18th IFAC World Congress (IFAC), pp. 7280-7285, 2011
[9] N. S. Nise, *Control System Engineering*, 5th ed, 2007.
[10] L. Ljung*: System Identification - Theory For the User,* 2nd ed*, PTR Prentice Hall, Upper Saddle River, N.J., 1999*
[11] R. De Keyser, C. M. Ionescu, "FRtool: A frequency response tool for CACSD in Matlab",*IEEE International Symposium on Computer Aided Control Systems Design*,pp. 2275–2280, Munich, 2006.
[12] C. Y. Liang, and H. Peng, "String stability analysis of adaptive cruise controlled vehicles",*JSME International Journal Series C* (2000) , Volume: 43, Issue: 3, Publisher: Citeseer, Pages: 671–677
[13] L. Peppard, "String stability of relative-motion PID vehicle control systems," *IEEE Trans. on Automatic Control*, vol. 19, no. 5, pp. 579–581, 1974.