Decentralized Formation Control of Autonomous Mobile Robots

Edwin Carvalho, Miguel Pedro Silva and Carlos Cardeira
IDMEC/IST, Technical University of Lisbon
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
edwin.carvalho@gmail.com, {miguel.silva, cardeira}@dem.ist.utl.pt

Abstract

Control of vehicle formations is an area of great interest that requires knowhow from several scientific areas such as control theory, vision, communication system, etc. In this work, several approaches to maintain a formation of vehicles are explained and tested.

In this paper, the approach towards formation control is achieved through Position Based Visual Servoing (PBVS). In PBVS, a cartesian space controller is used for target tracking. Two 2D pose estimation methods are described and two local controllers developed (Discrete Linear Quadratic Regulator and an explicit Model Predictive Controller).

To maintain the mobile robots in formation, a Neighbor Referenced (NR) control scheme is used. NR-based control allows decentralized formation control schemes since each member of the formation only has information of its neighbors and not of the entire robots in the formation.

Finally, simulation and experimental results of the developed work are presented.

1. Introduction

Mobile robotics has been achieving newer and higher levels of importance in recent years and with it, new, complex applications in which mobile robots may aid human beings have been imagined. These new challenging applications demand more complex control strategies, advanced sensors and actuators, more computational power and, in some situations, advanced cooperation and communication schemes between mobile robots.

One of the most renown challenges regarding mobile robots is to make these work in a cooperative fashion with the objective of maintaining the robots in a predetermined formation. The importance of formations is quite visible in nature. For millions of years nature has favored animals that work in groups and collaborate between themselves. An example of this are geese who fly in V shaped formations[1] to decrease the effects of air resistance, allowing the members of this specie to save energy while flying.

Human beings also have to gain from formations. By controlling various robots in order to maintain a formation is of vital importance in Automated Highway Systems (AHS) [14], satellite clustering systems [15], cooperative flight control [8], [1] and others. Cooperative flight control has recently been considered by NASA and the United states Air force as a key technological milestone for the twenty first century [11].

Mobile robot formations, their creation and maintenance, have been gaining widespread attention from the scientific community in recent years [13], [11], [5], and several control strategies have been proposed with the intent of increasing knowledge in this area. Not only has the control problems in multi-agent systems been tackled, but improvements in neighbor areas such as communication systems for consensus problems and global positioning systems for mobile robots, has also been occurring.

[1]V shaped formations are commonly referred to as wedge formations

2. Kinematic Equations

This section intends to present the kinematic equations of the mobile robot and the target tracking problem.

The classic kinematic model of a robot is obtained when dynamic effects such as, for instance, the robot's weight and inertia as well as slip between the robot's wheels and the floor are not taken into account [17] [9] [3]. In this model, the robot is considered a solid weightless body which is able to move freely on the horizontal plane and able to achieve infinite accelerations though subjected to non-holonomic constraints. In Figure 1, the geometric parameters as well as the global W and mobile M coordinate systems are presented.

In Figure 1, s represents the distance in the x axis between the center of mass of the robot and the motor axis, b represents the distance in the y axis between the center of the robot and the wheels, G represents the center of mass of the mobile robot and the origin of the mobile coordinate system M, C is the center of the AMRs motor axis. In this model, the inputs of the system are the angular velocity at which each wheel should rotate. The linear velocity of each wheel can be calculated by using equations (1) and (2), where the subscripts *l* and *r* refer to left and right wheels,respectively.

$$v_l = r\Omega_l \quad (1)$$

$$v_r = r\Omega_r \quad (2)$$

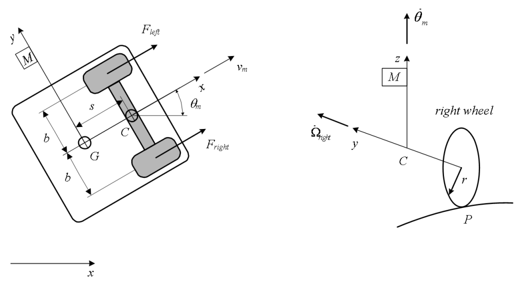

Figure 1. Geometric parameters of the mobile robot

The mobile robot's linear, angular and lateral velocities in the mobile robot's coordinate system can be calculated using the following equations:

$$v_m = \frac{r}{2}\left(\Omega_r + \Omega_l\right) \quad (3)$$

$$\dot{\theta}_m = \frac{r}{2b}\left(\Omega_r - \Omega_l\right) \quad (4)$$

$$\zeta_m = \frac{sr}{2b}\left(\Omega_r - \Omega_l\right) \quad (5)$$

where v_m represents the mobile robots linear velocity, $\dot{\theta}$ represents the robots angular velocity and ζ_m represents the carts lateral velocity. r is the radius of the robot's wheels, s as can be seen in (1) is the distance between the center of mass of the robot and center of the differential drive system and b is the distance between the wheels and the center of the differential drive.

Throughout this work, it is assumed, as usual in the literature, that $s = 0$, hence the lateral velocity $\zeta_m = 0$.

Lets now look at the target following problem. In this situation, the desired errors are those that relate the errors in the mobile frame of the pursuer with the orientation and both angular and linear velocities of the robots.

Considering an autonomous mobile robot moving in the fixed global frame as a target robot that must be followed. It's kinematic equations are presented in (6)

$$\begin{bmatrix}\dot{x}_t\\ \dot{y}_t\\ \dot{\theta}_t\end{bmatrix} = \begin{bmatrix}v_t\cos\theta_t\\ v_t\sin\theta_t\\ \omega_t\end{bmatrix} \quad (6)$$

where v_t represents the AMR's linear velocity, θ_t is the robots orientation and w_t is its angular velocity.

Since both target and pursuer robots are considered similar, their kinematic equations are considered equal. In the following equations, v_p and θ_p represent the pursuer robot's linear speed and orientation in the global coordinate system.

$$\begin{bmatrix}\dot{x}_p\\ \dot{y}_p\\ \dot{\theta}_p\end{bmatrix} = \begin{bmatrix}v_p\cos\theta_p\\ v_p\sin\theta_p\\ \omega_p\end{bmatrix} \quad (7)$$

By calculating and deriving the error coordinates relative to time, the following kinematic equations are obtained.

$$\begin{bmatrix}\dot{x}_e\\ \dot{y}_e\\ \dot{\theta}_e\end{bmatrix} = \begin{bmatrix}\dot{\theta}_p y_e - v_p + v_t\cos\theta_e\\ -\dot{\theta}_p x_e + v_t\sin\theta_e\\ \dot{\theta}_t - \dot{\theta}_p\end{bmatrix} \quad (8)$$

With x_e, y_e and θ_e presented below.

$$x_e = (x_t - x_p)\cos\theta_p + (y_t - y_p)\sin\theta_p \quad (9)$$

$$y_e = (y_t - y_p)\cos\theta_p - (x_t - x_p)\sin\theta_p \quad (10)$$

$$\theta_e = \theta_t - \theta_p \quad (11)$$

3. Position Based Visual Servoing

PBVS is a form of visual servoing in which, the pose of the object to be tracked is obtained and used together with cartesian control laws.

In this work, two position based controllers were developed. A Discrete Linear Quadratic Regulator (DLQR) and an explicit Model Predictive Controller (explicit MPC).

3.1 Discrete Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) is an optimal controller characterized by a quadratic cost function,also known as the performance index, and a plant with linear dynamics. The LQR attempts to find a control vector trajectory that minimizes the quadratic cost function.

Since the developed controller for the AMR will be implemented through a digital computer, lets consider the discrete time LQR. Assume that:

$$J = \frac{x_N^T S_N x_N}{2} + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) \quad (12)$$

subject to the following dynamic restrictions

$$x_{k+1} = Ax_k + Bu_k, \quad (13)$$

where J is the cost function, $x \in \mathcal{R}^n$ is the state vector of the plant,$u \in \mathcal{R}^m$ is the input vector, S_N is the terminal cost matrix, Q is a positive, symmetrical semi-definite matrix that weighs the state, R is a positive definite symmetrical matrix that weighs the input.

A possible solution consists of resolving a sub-optimal LQR problem in which a unique K matrix is computed, such that:

$$u_x = -Kx_k \quad (14)$$

If S_N is a positive symmetric semi-defined matrix, then the stationary solution S_∞ to the Riccati equation can be computed, yielding the sub-optimal Kalman gain:

$$K = (B^T S_\infty B + R)^{-1} B^T S_\infty A \quad (15)$$

3.2 Explicit Model Predictive Control

Model Predictive control (MPC), also known as Receding Horizon Control (RHC), is one of the most popular forms of advanced process control with a large history of implementation in the chemical and oil industries which has been receiving attention from control theorists and practitioners due to the several unique characteristics it presents [12]. Lets start by summarizing some of the main advantages of MPC:

- Can take into account state, input and output constraints.
- Allows the system to work very near to the constraints without reaching them, which, in turn, may increase the system's performance.
- Can deal with unstable systems.
- Can deal with non minimum phase systems.
- Can deal with Multiple Input, Multiple Output (MIMO) systems.

MPC is a control strategy in which the current control action is obtained by solving, at each sampling instant, a constrained finite horizon open-loop optimal control problem, such that, the current state of the system is used to compute the optimal input and state trajectories. The solution of the optimization problem is an optimal control sequence and, according to the MPC strategy, only the first control action in the sequence is applied to the system.

Next, a brief theoretical description of the MPC framework based on the work of [7] is presented. Consider the following discrete-time system:

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k)) \end{cases} \tag{16}$$

where $x(k+1)$ and $x(k)$ are the values of the states at instants $k+1$ and k respectively, $u(k)$ are the inputs at instant k and $y(k)$ is the system's outputs at instant k. The objective of the control scheme is to steer the states to an equilibrium state x_r for which the output $y_r = r$ where r is the constant reference.

The generic cost function can be defined as:

$$V(x_t, \mathbf{u_t}) \triangleq \varphi(x(N|t)) + \sum_{k=0}^{N-1} L(x(k+t|t), u(k+t|t)) \tag{17}$$

Where x_t represents the value of the states at instant t and $\mathbf{u_t}$ is the sequence of control actions to optimize, also at instant t, L is the stage cost function and φ is the terminal cost function associated with the optimization problem. Several conditions must be verified for the sequence of control actions and states to be valid. First, $u(k+t|t) \in \mathcal{U}$ with $\mathcal{U}$ a convex compact subset of $\mathcal{R}^n$, $x(k+t|t) \in \mathcal{X}$ and $\mathcal{X}$ is a closed and convex subset of $\mathcal{R}^n$

By finding the solution of the cost function presented in equation (17), a sequence of optimum control actions $\mathbf{u_t^*}$ can be found. In terms of stability, several problems emerge from MPC theory due to the finite horizon of the strategy. Nonetheless, several solutions have been suggested such as, for instance, forcing the terminal state $x(N|t) = 0$ proposed by [6] which suffers from the limitation that the terminal state must be achieved in exactly N steps. [10] also suggested using a dual mode controller to guarantee stability. In this dual-mode approach a terminal set χ_f which contains the origin is considered. The MPC controller would then only have to send the states to the terminal set χ_f in which a second controller would guarantee stability.

Due to the high sampling rate of the robotic system, online computation of the optimal control actions is not currently feasible, considering the current computational power of today's CPUs. A solution proposed in [2], [16] and [4] consists on using multiparametric programming to calculate explicit control laws for the above problem, which allows the control actions to be calculated offline, hence the original problem which consisted on finding the optimal control actions by reading the system's current states and optimizing the cost function presented in equation 17, becomes a problem of reading the system's current states and verifying in which region of the space state the system is currently in and using the explicit law for that region of the space state.

The model predictive controller obtained in this work was developed using Bemporad's hybrid toolbox which uses multiparametric programming to obtain the explicit control laws for the state space subjected to the previously described restrictions. To guarantee stability, Bemporad adds a terminal cost function and uses a linear quadratic regulator to obtain adequate gains for the terminal cost function.

4 Communication Architecture and Formation Control Strategy

In both decentralized and centralized (networked) control systems, communication between agents is fundamental to increase stability and performance of the multi agent system [3].

Several approaches have been proposed in recent years that attempt to promote fast, reliable seemingness communication between devices with the intent of allowing realtime information exchange between agents.

However the speed and reliability of the communications are not the only focal point in information exchange protocols. The ease of integration of new agents that want to enter the system and the flexibility of using the communication protocol with any programming language or operating system is equally important and has deserved attention from investigators from diverse fields.

In this work, the approach chosen to transmit and exchange information between the several agents (robots) is the User Datagram Protocol (UDP).

4.1 UDP

A UDP information exchange scheme presents several interesting characteristics that makes it a classic choice in such systems. The main qualities of UDP communications responsible for its popularity in noncritical realtime information exchange systems are resumed below:

- UDP does not require connections to be created or destroyed between the origin and the destination of the message, hence the required to send information is largely reduced.
- In question and answer applications, UDP sends one fifth of the packages that TCP sends due to the way information is encapsulated.
- No verification of package arrival is performed which speeds up transmission of data and lowers network latency.
- Packages sent as datagrams. Packets are sent individually and are guaranteed to be whole if they arrive. Packets have definite bounds and no split or merge into data streams may exist.
- Supports multicasting and broadcasting.

5 Formation Control Strategy

Lets start by recalling some of the basic concepts regarding formation design and position.

The formation communication strategy is one of the key factors in controlling a formation since it defines the relations between agents. In this article, a Neighbor-referenced formation communication strategy will be tackled.

Besides the formation communication strategy, the shape of the formations is another key factor in the definition of relations between agents. Formations with different shapes require different information flows between agents to remain stable and to obtain an adequate performance.

In neighbor-referenced formations, the information flows between adjacent agents of the system, in other words, the system works in a decentralized fashion since each agent only has local information of the system [3].

Lets consider the simplest possible neighbor-referenced formation. That would be a platoon formation in which, each agent only receives information from the agent in front (if any) and the agent behind (if any).

One question that usually arises is what information should actually be exchanged between agents? The normal choice is the state and input information (x_e, y_e, θ_e, v_l, ω_e) of the neighbor robots.

The information of the neighbor robots inputs are used as feedforward terms while the neighbors state information is generally used to calculate the state information to the controller.

Lets start by looking into the longitudinal error problem. This problem is quite serious since an incorrect error calculation can lead to incorrect control actions which in turn can lead to collisions between system agents. The Longitudinal error calculation problem can be seen as a weighting problem where two different objectives have to be weighed. The first objective is to maintain the robot exactly in the middle of the adjacent robots. This is essential since it minimizes the probability of collisions between agents and can be realized by forcing to zero, the following error equation:

$$x_{e_j} = (x_{j-1} - x_j) - (x_j - x_{j+1}), j \in [2, ..., n-1] \quad (18)$$

where j is the j^{th} robot of a formation with a total of n robots.

However the above state equation is not enough to guarantee that the robots will not collide or spread apart to the point where the distanced cannot be measured, besides, the above state equation is not valid for the last mobile robot in the formation.

The second objective is for the robot to maintain a predefined distance from the robot in front or the robot behind. One need to only set that the longitudinal distance to the AMR in front is the predefined distance since the condition presented in (18) will guarantee that the distance to the robot behind will also be the same predefined distance.

The predefined distance error equation can be represented by the following expression:

$$x_{e_j} = -(x_{j-1} - x_j) + d, j \in [2, ..., n] \quad (19)$$

Notice that this equation can also be applied the final robot of the formation.

By weighing the two equations, the following formula arises.

$$x_{e_j} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} (x_{j-1} - x_j) - (x_j - x_{j+1}) \\ -(x_{j-1} - x_j) + d \end{bmatrix}, \quad j \in [2, n-1] \quad (20)$$

where, K_1 and K_2 represent the weights of each error, hence:

$$K_1 + K_2 = 1 \quad (21)$$

In the case of the last mobile robot, the equation pans down to equation (19). Therefore, by mathematical manipulation of equations (18) - (21) the longitudinal state equation of agent $n-1$ is:

$$x_{e_{n-1}} = x_{e_n} + K_1(x_{n-2} - x_{n-1} - d) \quad (22)$$

With:

$$x_{e_n} = (x_{n-1} - x_n + d) \quad (23)$$

If the weights (K_1, K_2) are constant for all robots, the error equation reduces to the following equation:

$$\begin{aligned} x_{e_{j-2}} &= x_{e_{j-1}} - (x_{n-1} - x_n) - (x_{j-2} - x_{j-1}) + b \\ b &= K_1((x_{j-3} - x_{j-2}) - (x_{j-2} - x_{j-1})) \end{aligned} \quad (24)$$

If $K_1 = 1$, the AMRs in the middle of the platoon will focus on keeping themselves in the middle of the two neighboring robots while the last robot in the formation will only focus on maintaining a predefined distance to the second last robot. Since the second last robot will try to keep the distance between the adjacent robots the same and the distance to the robot behind it will remain at the predefined distance, the distance to the robot in front will become the predefined distance after some time instances. This will propagate through the formation until all robots are exactly at the predefined distance from their neighbor robots. In this case, the error equation is given by (25)

$$x_{e_{j-2}} = x_{e_{j-1}} - (x_{n-1} - x_n) - 2(x_{j-2} - x_{j-1}) + \\ +(x_{j-3} - x_{j-2}) \quad (25)$$

The transversal error calculation is simpler. Although in some situations keeping in the center of the neighbor robots can be benefic (e.g., in some wedge and diamond formations, the correct transversal position is when the AMR is at the transversal center of its neighbors), simply trying to keep the transversal error in relation to the previous robot equal to a predefined constant is acceptable.

$$y_{e_j} = -(y_{j-1} - y_j) + d, j \in [2, ..., n] \quad (26)$$

The orientation error can be considered in the same way as the transversal error hence equation (27) arises.

$$\theta_{e_j} = -(\theta_{j-1} - \theta_j) + d, j \in [2, ..., n] \quad (27)$$

The system input information from neighbor robots can be used in a similar form as the state information, however the input information works as feedforward terms for the controller. As with the transversal and orientation errors, the angular velocity of agent $j-1$ should be used as a feedforward term while the average of the linear velocities of adjacent robots should be used as a feedforward term to help maximize the space between both neighbor vehicles.

6 Simulation and Experimental Results

6.1 Simulation Results - DLQR Controller

A DLQR controller with the following Q and R weighting matrices was used as a local controller.

$$Q = \begin{bmatrix} 2000 & 0 & 0 \\ 0 & 19300 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad (28)$$

$$R = \begin{bmatrix} 2 \times 10^{+03} & 0 \\ 0 & 2 \times 10^{+03} \end{bmatrix} \quad (29)$$

The mean, maximum and mean square error of the controller in simulation are presented in the following tables.

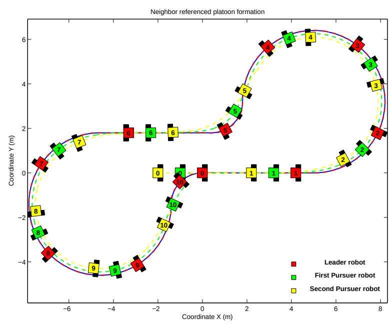

Figure 2. Platoon formation in a Neighbor Referenced formation position

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	-0.0011	-0.0319
Maximum absolute error	0.2760	0.0534
Mean square error	0.0016	0.0015

Table 1. Longitudinal error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	-0.0337	-0.0414
Maximum absolute error	0.2346	0.1580
Mean square error	0.0056	0.0056

Table 2. Transversal error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	0.1657	0.1685
Maximum absolute error	0.5658	0.3690
Mean square error	0.0819	0.0656

Table 3. Orientation error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	0.0494	0.0393
Maximum absolute error	0.2048	0.0928
Mean square error	0.0031	0.0017

Table 4. Longitudinal Error of the Neighbor Referenced platoon formation

6.2 Simulation Results - Explicit MPC Controller

The state and input weighing matrices used to create the explicit MPC are presented below

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	0.0262	0.0026
Maximum absolute error	0.3604	0.3583
Mean square error	0.0095	0.0061

Table 5. Transversal Error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	0.1615	0.1819
Maximum absolute error	0.7427	0.3914
Mean square error	0.0908	0.0793

Table 6. Orientation Error of the Neighbor Referenced platoon formation

$$Q = \begin{bmatrix} 2000 & 0 & 0 \\ 0 & 19300 & 0 \\ 0 & 0 & 1,0 \times 10^{-01} \end{bmatrix} \quad (30)$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \quad (31)$$

6.3 Experimental Results

The following graphs represent the evolution of the states during experimental tests. The referencing scheme used was neighbor-based and the local controller chosen was the explicit MPC controller.

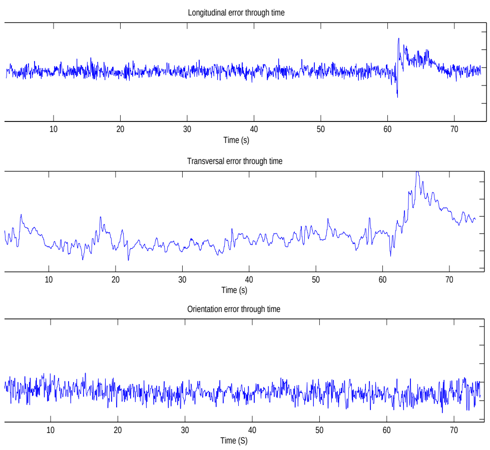

Figure 3. Errors of the first pursuer robot in a platoon formation, in a Neighbor Referenced formation and using a local explicit MPC controller.

6.4 Discussion of the Results

By analyzing the above graphs and tables several conclusions regarding the efficiency of both control schemes can be withdrawn. In simulation, both control schemes

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	-0.0087	0.0059
Maximum absolute error	0.1212	0.1227
Mean square error	4.3913×10^{-04}	9.3130×10^{-04}

Table 7. Longitudinal Error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	-0.0011	0.0055
Maximum absolute error	0.0737	0.1107
Mean square error	2.9314×10^{-04}	7.6481×10^{-04}

Table 8. Transversal Error of the Neighbor Referenced platoon formation

	Longitudinal error (m)	
	First pursuer	Second pursuer
Average error	-0.1069	-0.2507
Maximum absolute error	0.3308	0.4220
Mean square error	0.0172	0.0655

Table 9. Orientation Error of the Neighbor Referenced platoon formation

(DLQR and the explicit MPC) present average longitudinal and transversal errors fewer than five centimeters and maximum errors under forty centimeters. These errors are well below the predefined distance of one meter between mobile robots hence, should be enough to avoid collisions between AMRs and to guarantee an adequate control over the formation. The orientation error in simulation was considerably higher than the longitudinal and transversal errors. This is largely due to the very low weights applied to the orientation state in the Q state weighing matrix. The experimental results confirmed what was previously verified in the simulations. The average longitudinal and transversal errors of the AMRs are, in average, almost null while the orientation error is substantially higher. It is almost of importance to note that the experimental tests were performed in an unstructured environment which complicates the control of the formation.

7 CONCLUSIONS

The main objective of this paper was to study the problems associated with the maintenance of formations of mobile robots and develop control solutions able to maintain a formation of robots as well as implementing a system able to localize robots in an indoor environment.

Although is possible to develop the controllers based on the dynamic model of the robot, using the kinematic equations is advantageous since it greatly reduces the complexity of the problem and simultaneously, the complexity of the controllers. For the system to be approximately kinematic, a controller with a low sampling time (compared to the sampling time of the target following controller) had to be programmed into a microprocessor and encoders installed on the AMR.

Using pose estimation algorithms which allowed the error functions to be obtained in cartesian coordinates, several position based control strategies were implemented. The strategies ranged from a discrete LQR, in which the control law is unique for the entire state space, to an explicit MPC controller, due to the very low sampling time of the system. By using multi-parametric programming, an explicit control law dependant on the current states of the system was obtained, hence reducing the online implementation of the problem from the resolution of an optimization problem to a lookup table problem. To develop the controllers, the kinematic equations of the robot were linearized at the origin.

The above controllers were tested and proved to work. From the tests presented, one can easily conclude that the importance of the orientation error, in this particular formation shape, is very small when compared to that of the longitudinal or transversal errors.

8 Acknowledgments

This work was partly funded by the FCT through program POCI 2010 - 10-46 - IDMEC, subsidized by FEDER.

References

[1] R. Beard, J. Lawton, and F. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, 2001.

[2] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[3] E. Carvalho. Localization and cooperation of mobile robots applied to formation control. Master's thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, Avenida Rovisco Pais, n.º1, Lisbon, Portugal, October 2008.

[4] V. Dua, N. Bozinis, and E. Pistikopoulos. A multiparametric programming approach for mixed-integer quadratic engineering problems. *Computers and Chemical Engineering*, 26(4-5):715–733, 2002.

[5] J. Fax. *Optimal and Cooperative Control of Vehicle Formations*. PhD thesis, California Institute of Technology, 2001.

[6] D. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[7] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[8] M. Mesbahi and F. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *Journal of Guidance, Control, and Dynamics*, 24(2):369–377, 2001.

[9] N. Mestre. Controlo preditivo híbrido de robos moveis terrestres para seguimento de pista. Master's thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, Lisbon, Portugal, 2008. In Portuguese.

[10] H. Michalska and D. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[11] D. Naffin. *Multi-robot formations*. PhD thesis, University of Southern California, 2006.

[12] M. Nikolaou. Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs. *Advances in Chemical Engineering Series*, 2001.

[13] O. Orqueda. *Vision-based control of multi-agent systems*. PhD thesis, Oklahoma State University, 202 Engineering South, Stillwater, OK 74078, December 2006.

[14] D. Swaroop and J. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[15] H. Tanner, G. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, 2004.

[16] P. Třndel, T. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3):489–497, 2003.

[17] A. A. Wahyudi. Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, pages 149–154, 2004.