Mobile Robot Local Predictive Control under Perception Constraints

LLUIS PACHECO, NINGSU LUO, JORDI FERRER
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló s/n, SPAIN
lluis.pacheco@udg.edu, ningsu.luo@udg.edu, jferrerp@eia.udg.es

Abstract: **This research extends a previously developed work concerning about the use of local model predictive control in mobile robots. Hence, experimental results are presented as a way to improve the methodology by considering aspects as trajectory accuracy and time performance. In this sense, the cost function and the prediction horizon are important aspects to be considered. The platform used is a differential driven robot with a free rotating wheel. The aim of the present work is to test the control method by measuring trajectory tracking accuracy and time performance. Moreover, strategies for the integration with perception system and path planning are also introduced. In this sense, monocular image data provide an occupancy grid where safety trajectories are computed by using goal attraction potential fields.**

Keywords: **autonomous mobile robot, computer vision control, system identification, model based control, predictive control, trajectory planning, obstacle avoidance, robot vision**.

1. INTRODUCTION

The minimization of path tracking error is considered as a challenging subject in mobile robotics. The main objective of highly precise motion tracking consists in minimizing the error between the robot and the desired path. Other aspects as vehicle speed or even acceleration profile can also be analyzed as important issues to be considered in the path tracking strategies [1]. Hence, the dynamics of the robot becomes an important issue in planning accurate and safe trajectories, in which on-robot sensors provide enough environmental knowledge to avoid colliding with obstacles and to reach the final desired coordinates. The scientific community has carried out several studies in this field. The use of a dynamic window approach, with available robot speeds, the reactive and safety stopping distances, derived from robot motion dynamics, allow WMRs (wheeled mobile robots) reactively avoiding obstacles [2]. Rimon presented methodologies for exact motion planning and control, based on artificial potential fields where complete information about the free space and goal are encoded [3]. The main drawback of the potential fields is the local minimal failures. However, the flexibility is reported as an advantage when small or moving obstacles are met. Probabilistic Roadmap Method (PRM) [4-5] is a methodology not suffered from local minimal failures, but with drawbacks as unattractive path generation and lack of flexibility. Path optimization can be improved using local motions controlled by local potential fields [6-7]. Some approaches to mobile robots propose using potential fields, which satisfy the stability in a Lyapunov sense on a short prediction horizon [8].

In the present work, model predictive control (MPC) is presented as an accurate nonlinear methodology that allows an exact trajectory tracking by minimizing the cost function. In this sense, simulated and experimental results are presented as an extension of previously developed work [9]. Hence, several studies concerning about the trajectory tracking accuracy are reported. Moreover, a trajectory planning framework consisted of a narrow and dense field of view and a goal attracting potential field point are used to test the MPC performance. From a monocular perception system, a local narrow occupancy grid is acquired, which constrains the possible WMR trajectories. Thus, the selected free cell should avoid the obstacle collisions and approach the robot to the final desired configuration through the attraction objective field. Once the local desired cell is obtained a straight line between the cell and the robot is generated. These kinds of trajectories are used also to test the performance of robot trajectory tracking. The control action consists in steering and straight line following, such actions are common to a wide range of nonholonomic vehicles [10]. Hence, at each perception step, a tracking straight line is commanded as a new trajectory to be followed by the robot. Thus, experimental results reported assume different local grid occupancy frames in order to test realistic control trajectories. The visual data are the meaningful source of information in order to accomplish with the tasks of obstacle detection. However, other data provided by the encoder-based odometer system are also considered.

This paper is organized as follows. Section 1 briefly presents the aim of the present work. Section 2 introduces the MPC methodology and algorithms. The WMR PRIM, consisting of a differential driven robot with a free rotating wheel, is presented as an available platform in order to orient the results [9]. Once the robot models are known several simulations are depicted. In the Section 3, the experimental results attaining the trajectory tracking performance are presented. Hence, horizon of prediction, cost function, and time performance are studied. Moreover, the experimental results are developed within a context of path planning strategy. Thus, local monocular visual horizon provides an occupancy grid, in which trajectories with safe obstacle avoidance can approach the robot to the goal. Consequently, it can be used to test the control performance. Finally, in Section 4, some conclusions are drawn and future research is outlined.

2. LOCAL MODEL PREDICTIVE CONTROL

Model predictive control, MPC, has many interesting aspects that can be applied to mobile robot control. In comparison to the standard PID control, MPC is the most effective advanced control technique with significant impact on industrial process control [11]. Real-time implementation of MPC in the mobile robotics has been developed using global vision sensing [12]. In [13], MPC based optimal control was studied, which is useful for cases when nonlinear mobile robots are used under several constraints. In general, real-time implementation is possible when a short prediction horizon is used. By using MPC, the idea of receding horizon can deal with the local sensor information. MPC is based on minimizing a cost function related to the objectives for generating the optimal inputs. Experimental trajectory tracking using MPC has been also reported. Hence, agricultural robot applications, using similar prediction horizons to those in this work, are reported in [14], in which the trajectory was previously human-assigned and the GPS was used as main sensor. In general, global trajectory planning becomes unfeasible due to that the sensorial system of some robots is only local. Consequently, the local model predictive control (LMPC) is proposed to use the available visual data in the navigation strategies for the goal achievement. Define the cost function as follows:

$$J(n,m)=$$

$$\min_{\left\{U(k+i|k)_{j=0}^{j=m-1}\right\}} \left\{ \begin{array}{l} \left[X(k+n|k)-X_{ld}\right]^T P\left[X(k+n|k)-X_{ld}\right] \\ +\sum_{i=1}^{n-1}\left[X(k+i|k)-X_{l,do}\right]^T Q\left[X(k+i|k)-X_{l,do}\right] \\ +\sum_{j=0}^{m-1} U^T(k+i|k) R U(k+i|k) \end{array} \right\} \quad (2.1)$$

The first term of (2.1) refers to reaching the desired coordinates, $X_d=(x_d,y_d,q_d)$. The second one is related to the distance between the predicted robot positions and the trajectory segment $X_{l,do}$ given by a straight line between the robot coordinates $X_{l0}=(X_{l0},Y_{l0},q_{l0})$ from where the local perception frame was acquired, and the desired local position $X_{ld}=(X_{ld},Y_{ld},q_{ld})$ belonging to the local perception grid. The last one is related to the input signals. The parameters P, Q and R are weighting parameters that express the importance of each term. The system constraints are also considered:

$$\left\{ \begin{array}{ll} \left|U(k+i|k)\right| \le G_1 & a \in [0,1) \\ \left\|\left[x_{k+i}, y_{k+i}\right]-\left[x_o, y_o\right]\right\| \ge G_2 & \\ \left\|\left[x_{k+n}, y_{k+n}\right]-\left[x_d, y_d\right]\right\| \le a\left\|\left[x_k, y_k\right]-\left[x_d, y_d\right]\right\| & \end{array} \right\} \quad (2.2)$$

The limitation of the input signal is taken into account in the first constraint. The second constraint is related to the obstacle points where the robot should avoid the collision. The last one is simply a convergence criterion. The MPC algorithm is based on the model knowledge. Hence in this section the experimental modelling methodology is firstly introduced. The odometer system formulation is also given as the necessary data for controlling the trajectory tracking. Finally, once the models of the robot are known the proposed algorithms and simulation results are reported.

2.1 Experimental Model Identification and Odometer System Formulation

In this work the parameter identification is based on the black box approaches [15]-[17]. The transfer functions are related to a set of polynomials that allow the use of analytic methods to deal with the problem of controller design. The nonholonomic system dealt with in this work was initially considered as a MIMO (multiple input multiple output) system, which is composed of a set of SISO subsystems with coupled dynamic influence between two DC motors. The approach of the multiple transfer function consists in carrying out the experiments with different speeds. In order to find a reduced-order model, several studies and experiments were done through the system identification and model simplification. The parameters were estimated using a

PRBS (Pseudo Random Binary Signal) as the excitation input signal. The experiments done consist in exciting two DC motors in different (low, medium and high) speed ranges. The problem consists in finding a model that minimizes the error between the real and estimated data by expressing the ARX equation as a lineal regression. Fig. 2.1 shows the speed response of the left wheel corresponding to a left PBRS input signal.

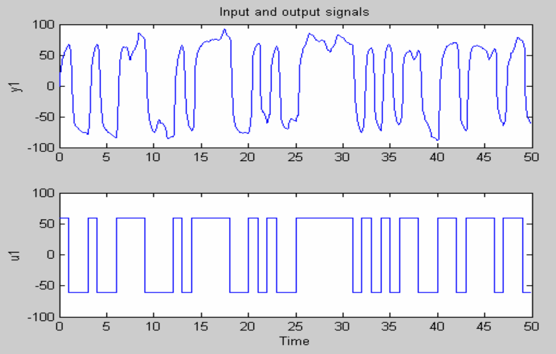

Fig. 2.1: Speed output for a left PRBS input signal.

Experimental data is treated before the parameters are estimated. This includes data filtering, using the average value of five different experiments with the same input signal, frequency filtering and tendency suppression. The system is identified by using the identification toolbox “ident” of Matlab for the second order models. The simulation results show that the obtained model fits the experimental data well. However, it is observed that the dynamics of two DC motors are different and the steady gains of coupling terms are relatively small (less than 20% of the gains of main diagonal terms). Thus, it is reasonable to omit the coupling dynamics so as to obtain a simplified model. Finally, the order reduction of the system model is carried out by analysing pole positions with the root locus method. Afterwards, the system models were validated through the experimental data and using the PBRS input signal [9].

2.2 Odometer System Expression

Denote (*x, y, θ*) as the coordinates of position and orientation, respectively. Figure 2.2 describes the positioning of the robot as a function of the radius of the left and right wheels (R$_e$, R$_d$), and the angular incremental positioning (θ$_e$, θ$_d$), with *E* being the distance between the two wheels and dS the incremental displacement of the robot. The position and angular incremental displacements are expressed as:

$$dS = \frac{R_d dq_d + R_e dq_e}{2} \qquad dq = \frac{R_d dq_d - R_e dq_e}{E} \tag{2.3}$$

The coordinates (x, y, θ) are expressed as:

$$\begin{aligned} x_n &= x_{n-1} + dS\cos(q_{n-1} + dq) \\ y_n &= y_{n-1} + dS\sin(q_{n-1} + dq) \\ q_n &= q_{n-1} + dq \end{aligned} \tag{2.4}$$

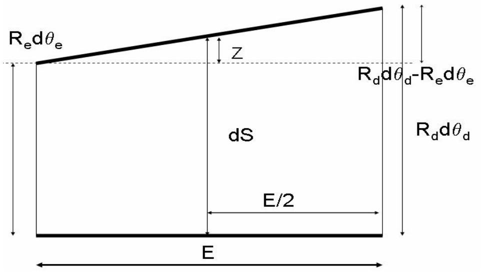

Fig. 2.2.: Positioning of the robot as functions of the angular movement of each wheel.

Thus, the incremental position of the robot can be obtained through the odometer system with the available encoder information from (2.3) and (2.4).

2.3 LMPC Algorithms and Simulation Results

By using the basic ideas previously introduced, it is presented here the LMPC algorithms that can be run in the following steps:

- Read the actual position
- Minimize the cost function and obtain a series of optimal input signals
- Choose the first obtained optimal input signal as the command signal.
- Go back to step 1 in the next sampling period.

Minimizing the cost function is a nonlinear problem in which the following equation should be verified:

$$f(a\ x + b\ y) \le a\ f(x) + b\ f(y) \tag{2.5}$$

It is a convex optimization problem caused by the trigonometric functions used in (2.4), [18]. The use of interior point methods can solve the above problem [19]. Among many algorithms of optimization, the gradient descent algorithm [20] was implemented in this work.

In order to obtain the optimal solution, some constraints on the inputs are taken into account:

- The signal increment is kept fixed within the prediction horizon.
- The input signals remain constant during the remaining interval of time.

The above considerations can reduce the computation time and result in the smooth behaviour of along the prediction horizon. Thus, the set of available inputs is reduced to one value. In order to reduce the search for an optimal signal value that minimizes the LMPC cost function, the possible input sets are considered as a bi-dimensional array, as shown in Fig. 2.3.

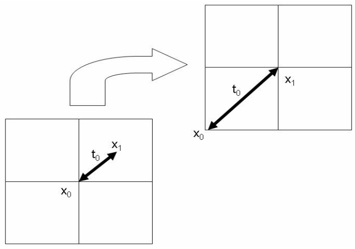

Fig. 2.3: Interval search

The array is decomposed into four zones. The search consists in analyzing the central points of each zone. It is considered only the region that offers the better optimization, where the algorithm is repeated for each sub-zone, until no sub-interval can be found. The results were obtained by testing all possible inputs and the subinterval search algorithm, which were compared by simulating a 2m straight line tracking, as shown in Fig. 2.4.

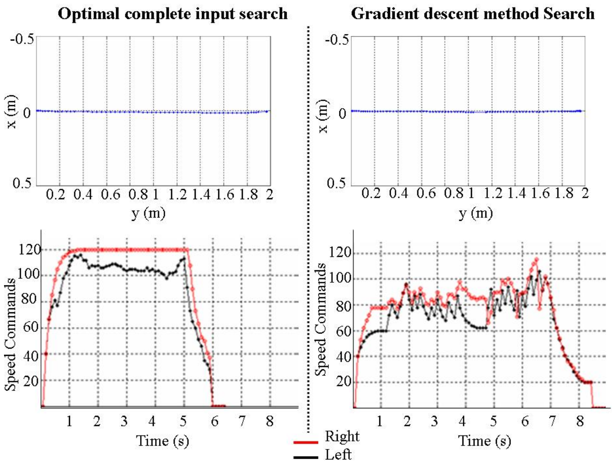

Fig. 2.4: Trajectory tracking and command speed profiles during 2 m straight line tracking simulation, by computing the cost function for all the input combinations or by using the gradient descent method.

The results show the discrepancy between the two methods in which the subinterval gradient descent method usually does not give the optimal solution. However, when acceleration is produced both results are similar.

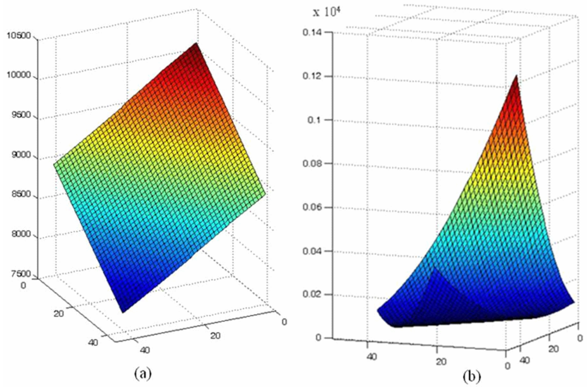

Figure 2.5

Fig. 2.5.a illustrates this case by showing the results of cost function obtained by all possible inputs, in which a unique minimum is found. However, when the acceleration process is finished and other criteria as final point and desired trajectory distances are considered, only suboptimal solutions are obtained. As shown in Fig. 2.5.b, local minimum can be obtained instead of being globally optimal. The gradient descent results can be considered as nearly optimal. Once the algorithm was proposed, several simulations were carried out to verify its effectiveness, and then to make improvement. Thus, when only the desired coordinates are considered, the robot could not arrive at the final point.

Fig. 2.6 shows that the inputs can minimize the cost function by shifting the robot position to the left due to that the left motor has more gain than the right. This problem can be easily solved by considering a straight-line trajectory from the current point of the robot to the final desired point. Thus, the trajectory should be included in the LMPC cost function.

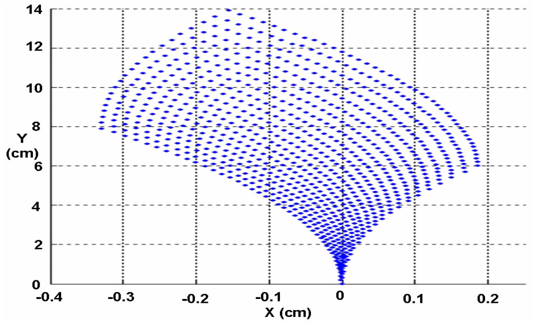

Fig. 2.6: Predicted coordinates from speed zero, n=5, m=3. The left deviation is due to the bigger left gain of the robot.

Fig. 2.7 shows a simulated result of LMPC for the WMR by first using the orientation error as the cost function and then the local trajectory distance and the final desired point in the optimization.

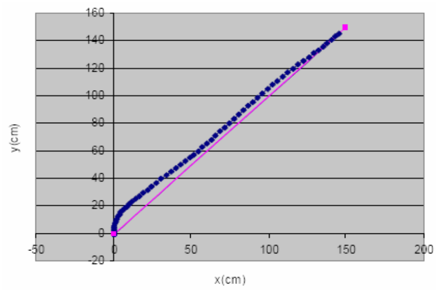

Fig. 2.7: LMPC simulated results following a 45º trajectory. It is used firstly squared orientation difference minimization and then squared trajectory and final distances minimization.

The prediction horizons between 0.5s and 1s were proposed and the computation time for each LMPC step was set to less than 100ms, running in an embedded PC of 700MHz. The simulated results, using a prediction horizon (n=10) and a control horizon (m=5), are shown in Fig. 2.8. The computing time for the complete search of an optimal input is between 13 and 14 ms when m=3 and n=5. In the case of using gradient descent method, the computing time is set to less than 1ms.

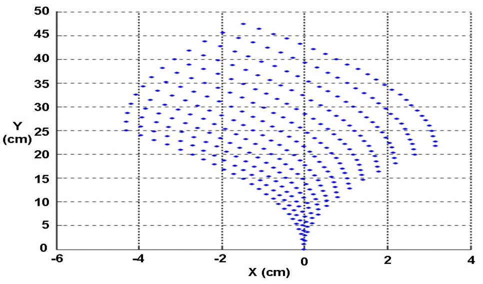

Fig. 2.8: Predicted coordinates from speed zero, n=10, m=5.

The possible coordinates available for prediction, as shown in Fig. 2.6, depict a denser horizon. Trajectory tracking and final point reaching are other interesting aspects to be analyzed. Fig. 2.9 shows the results obtained in tracking a straight line of 2m using two different prediction horizons.

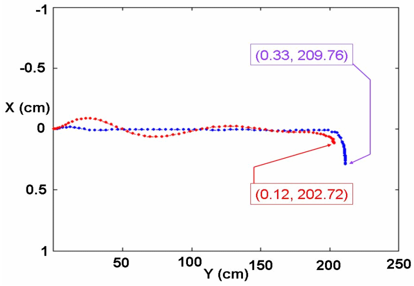

Fig. 2.9: Trajectory tracking in red (n=10, m=5) and in blue (n=5, m=3). The larger prediction horizon shows a closer final point achievement and worse trajectory tracking.

Fig. 2.10 shows the velocities of both wheels using the above strategies. The wide prediction strategy shows a softer behaviour due to the larger control horizon.

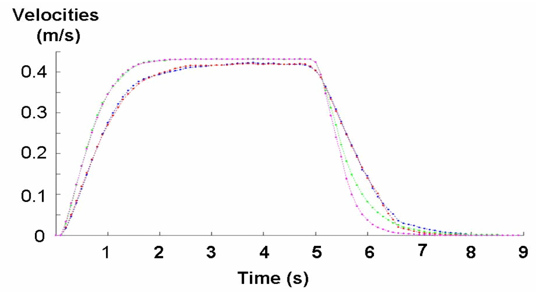

Fig. 2.10: Wheel speeds during the 2m straight line tracking. The red and blue dots show the right and left speeds respectively, with n=10 and m=5. The magenta and green dot lines depict the right and left speeds with n=5 and m=3.

3. Experimental Results

In this section, two important aspects are analyzed: the trajectory tracking and the velocity estimation.

The trajectory tracking is improved by the adequate choice of a cost function that is derived from (2.1) and consists of a quadratic function containing some of the following four parameters to be minimized:

- The squared Euclidean *approaching distance* between the local desired coordinates, provided by the on-robot perception system, and the actual robot position.
- The squared trajectory *deviation distance* between the actual robot coordinate and a straight line that goes from the robot coordinates, when the local frame perception was acquired, and the local desired coordinates belonging to the referred frame of perception.
- The third parameter refers to changes allowed to the input signal. However, some constraints over the input are considered in section 2.3.
- The last parameter consists in the squared orientation difference that is used only when the desired orientation is greater than a selected threshold. In this case, other parameters of the cost function are not used.

One consideration that should be taken into account is the different distance magnitudes. In general, the approaching distance could be more than one meter. However, the deviation distance has its magnitude normally in the order of cm, which becomes effective only when the robot is approaching to the final desired point. Hence, when it is tried to reduce further the deviation distance to less than 1cm, it is proposed to increase the weight value for the deviation distance in the cost function.

In the MPC research, Kalman filter has been employed for improving the system performance [21]. In the present work, the technique of Kalman filter is used for the velocity estimation. It is implemented as output prediction estimation by using both inputs and outputs, and the knowledge of the estimated outputs obtained by the system model [22]. The implemented filter is given by the following equations:

$$\tilde{x}(k+1|k)=G\tilde{x}(k|k)+Hu(k) \tag{3.1}$$

$$P(k+1|k)=GP(k|k)G^{T}+R_{1} \tag{3.2}$$

where *G, H, C* are given by the system model, $\tilde{x}$ is the estimated system output, *y* the measured output, and *P* represents the estimation error variance at the instant *k*. The initial value P(0/0) can be set to zero. Thus, recursive output estimation can be done, as function of the estimated outputs, the inputs, and the last measured output:

$$K(k+1)=P(k+1|k)C(R_{2}+CP(k+1|k))^{-1} \tag{3.3}$$

$$P(k+1|k+1)=P(k+1|k)-K(k+1)CP(k+1|k) \tag{3.4}$$

$$\tilde{x}(k+1|k+1)=\tilde{x}(k+1|k)+K(k+1)(y(k+1)-C\tilde{x}(k+1|k)) \tag{3.5}$$

The parameters R_1 and R_2 are used to model the noise variance empirically done in this work. Fig. 3.1 shows the acquired and estimated outputs.

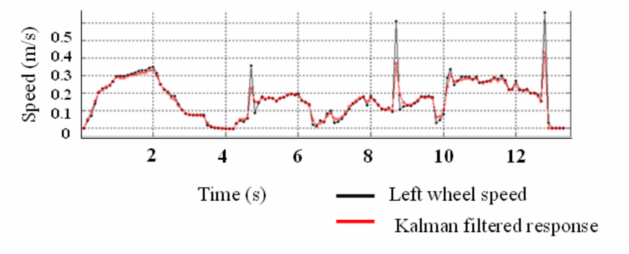

Fig. 3.1: Experimental results of the Kalman filter

3.1 Local Occupancy Grid perception and Trajectory Planning by Using Monocular Data

This section introduces first the proposed path planning strategy that arises from the local perception knowledge and potential attraction field provided by a global desired robot configuration. Then, the experimental results of trajectory tracking are explained. The occupancy grid framework provides a robust and unified approach to a variety of problems in spatial robot perception and navigation [23]. In this sense, computer vision research, developed to obtain scene information, has solved many interesting topics by using different methods based on the understanding of biological systems. Hence, machine vision systems have used some features of eyes as stereopsis, optical flow or accommodation as meaningful clues [24]-[26]. In this paper it is supposed that the occupancy grid is given by developed methods [30]. Hence, the local visual data provided by the camera are used to plan a feasible trajectory and to avoid obstacle collision [9]. Fig. 3.2 shows the on-robot camera configuration studied in this work

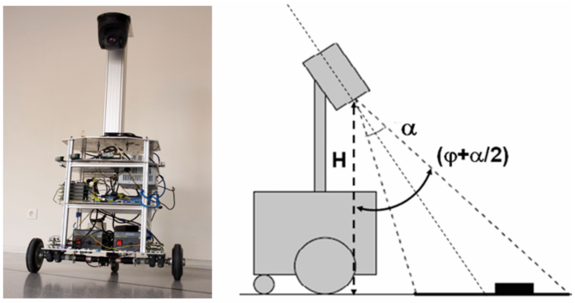

Fig. 3.2: Fixed camera configuration

with its analysis focused on particular indoor environments with flat floor surfaces. In Fig. 3.2, α, β and φ are angles of the vertical and horizontal field of view and the tilt camera pose, respectively. The vertical coordinate of the camera is represented by H.

The scene coordinates can be computed using the trigonometric relationships [26]. Figure 3.3 shows a local map provided by the camera, which corresponds to a viewing field with a horizontal angle of 48º, a vertical angle of 37º, *H* set to 112 cm and a tilt angle of 32º. Scene grids of low resolution are used to speed up the computing process. Hence, image processing methods, such as pyramidal resolution or down sampling methods, can be used to compress images [27]. The results of coordinate maps can be improved by using calibration techniques that allow, for instance, radial distortion removal [28].

In this research, the wide-path in experimental sense should be big enough so as to allow the WMR passing without any risk of collision. Thus, this parameter should allow a trajectory following deviation as well as an enough space in order that the robot can turn around itself. When the concept of the wide-path is considered in the local visual map, the number of possible local map coordinates that can be attained by the robot is reduced [29].

Fig. 3.3 shows a feasible trajectory, including the wide-path of WMR, within the local visual map, where the allowed trajectory should consider other aspects related with robot dynamics such as the safety stop distance, the reactive criteria and the visual dead zone.

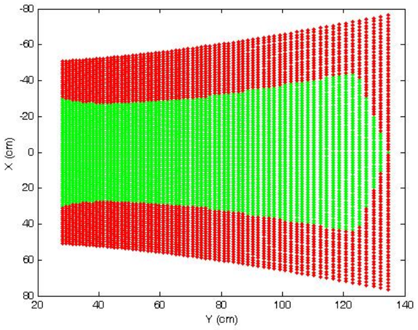

Fig. 3.3: Available local map coordinates (in green), the necessary coordinates free of obstacles and the necessary wide-path (in red)

The available information provided by the camera is considered as a local receding horizon where a trajectory without obstacles is planned. Hence, a local map with feasible coordinates is provided. Optimal free coordinate points are obtained by minimizing a cost function *J*, consisting in the Euclidean distance between the global desired coordinates and the available local scene coordinates:

$$J(i,j) = \min\left(\left(x_{i,j} - x_d\right)^2 + \left(y_j - y_d\right)^2\right)^{1/2} \qquad (2.3)$$
$$\forall\ 1 \le \mathrm{i} \le R \quad 1 \le j \le C \Leftrightarrow \text{IMAGE}\,(\mathrm{i},\,\mathrm{j}) \ne 1$$

where (x_d,y_d) are the global desired coordinates, and *IMAGE(i,j)=1* means that the visual coordinate is occupied by an obstacle. Thus, a local free cell that minimizes the cost function objective is selected as the local desired coordinates to be achieved by the robot [30].

3.2 Experimental Results on Trajectory Tracking

The trajectory tracking accuracy and time performance are two important aspects to be considered. The tested trajectories are obtained from the available set of local map coordinates as shown in Fig. 3.3. The LMPC results are analyzed when different trajectories tracking are commanded, as it is depicted in Fig. 3.4.

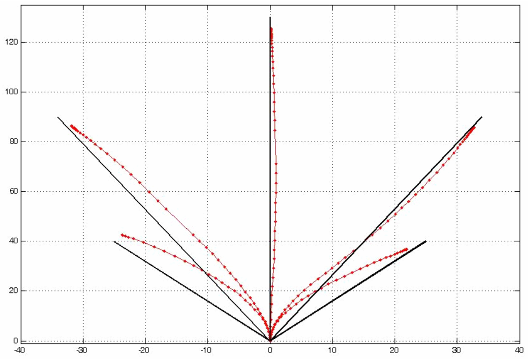

Fig. 3.4: Trajectory tracking tested from point to point by using the available local map coordinates provided by the monocular perception system.

Denote E_1 as the average final error, E_2 the maximal average tracking error, E_3 the average tracking error, E_4 the standard deviation of average tracking error. Table 3.1 presents the statistics concerning about the error obtained in cm testing the trajectories shown in Fig. 3.4.

Table 3.1: Point to point trajectory tracking statistics

Trajectory	E_1	E_2	E_3	E_4
From (0,0) to (0,130)	4.4cm	0.9cm	0.54cm	0.068
From (0,0) to (34,90)	3.8cm	3.9cm	2.3cm	0.82
From (0,0) To (25,40)	4.5cm	5.3cm	3.9cm	1.96

It can be seen that the accuracy of trajectory tracking, when straight line is commanded, has a deviation error of 0.54cm. However, when a turning action is performed, the error in straight line tracking is bigger as consequence of the robot dynamics when it is moving forward. Hence, the forward movement consists in usually a steering action. Fig. 3.4 gives a clue about what is happening. Thus, the major turning angle will produce the major deviation distance. Usually, it is very difficult to reduce the approaching distance to zero, due to the control difficulty of dead zone for the WMR and to the fact that the final target is considered in the present work as being reached by the robot when the Euclidean approaching distance is less than 5cm.

Other interesting results consist in testing the LMPC performance when the trajectory is composed of a set of points to be tracked. In this sense, when it is regarded to the kind of robot used, a pure rotation is possible by commanding the same speed with different sense to each wheel motor. Hence, when a trajectory is composed of many points, two possibilities exist: *continuous movement* in advancing sense, or *discontinuous movement* in which the robot makes the trajectory orientation changes by turning around itself at the beginning of the new straight segment.

Fig. 3.5 shows the tracking performance of the robot by tuning around itself, when the robot follows a trajectory composed of a set of points (0,0), (-25,50), (-25,100), (0, 150) and (0,200). The reported trajectory deviations are less than 5cm. However, the tracking time may reach up to 25s.

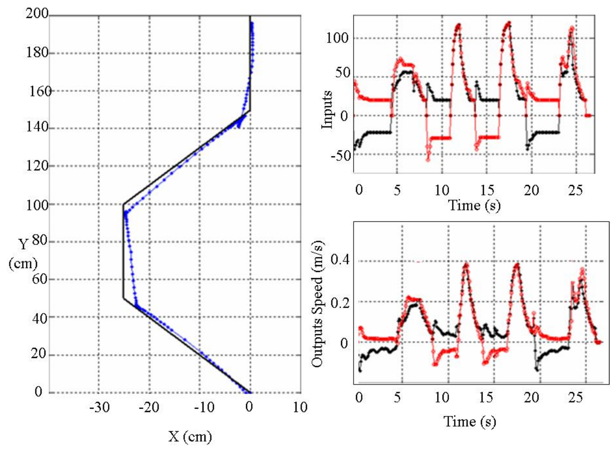

Fig. 3.5: Trajectory tracking with discontinuous movement.

The trajectory tracking strategy with continuous movement , for a set of points (0,0), (25,50), (25,100), (0,150) and (0,200), is represented in Fig. 3.6. In this case, it is reported a bigger trajectory deviation, due to the WMR's mechanical dynamics. The trajectory tracking is performed much faster (≤15s). Hence, in the continuous moving case, it needs a turning action with a minimum radius, once the direction is attained the robot deviation is very small. Thus, trajectories following straight lines have reported errors less than 1cm. When time performance is analyzed the continuous movement presents a better behaviour.

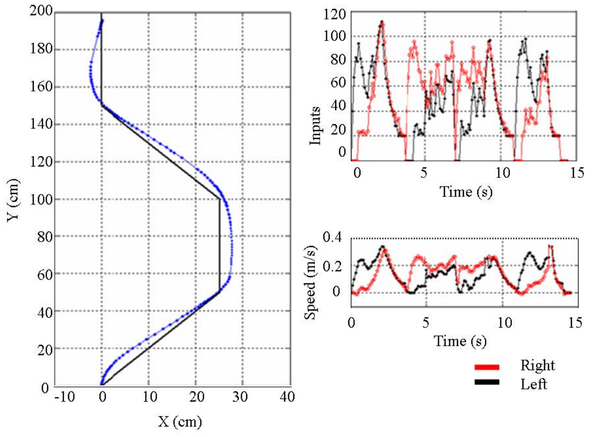

Fig. 3.6: Trajectory tracking with continuous movement.

4. Conclusions and Future Work

In this paper, LMPC strategies have been used to track the trajectory and to reach the local desired coordinates, which result in the improved accuracy and time performance. The desired local points and optimal local trajectory are computed by using machine vision data and robot constraint information. Hence, the odometer system is not just confined to controlling the velocity of each wheel. Thus, the absolute robot coordinates were used for planning a trajectory to the desired global or local objectives. Local trajectory planning was done using the relative robot coordinates that correspond to the instant when the frame was acquired.

Research is currently focused on implementing the presented methods by developing flexible software tools that allow vision methods to be tested and locally readable, and virtual obstacle maps to be created taking into account the robot's position and the selected camera configuration. The use of virtual visual information can be useful for testing the robot in synthetic environments and for simulating different camera configurations. Further studies on LMPC should be done in order to analyze improvements such as changing the tracking set-point when the WMR is not close to the desired point or its relative performance with respect to other control laws. The influence of the motor dead zones is also an interesting aspect that should make further efforts.

Acknowledgement

This work has been partially funded by the European Union (European Regional Development Fund) and the Commission of Science and Technology of Spain (CICYT) through the coordinated project DPI-2005-08668-C03 and by the Government of Catalonia through SGR00296.

References

[1] S.V. Vougioukas, "*Reactive Trajectory Tracking for mobile Robots based on Non Linear Model Predictive Control*" (IEEE Int. Conf. on Robotics and Autom., 2007), pp. 3074-3079,.

[2] D. Fox, W. Burgard, S. Thun "*The dynamic window approach to collision avoidance*" (IEEE Robot. Autom. Mag. Vol. 4, 1997), pp. 23-33.

[3] E. Rimont, D. Koditschet *"Exact robot navigation using artificial potential function"* (IEEE Trans. Robot Autom., vol. 8, no5, 1992), pp. 501-518.

[4] J. Barraquand, L. Kavraki, J. C. Latombe, T. Y. Li, R. Motwani, and P. Raghavan, *"A random sampling scheme for path planning"* (Inter. Journal of Robotics Research, vol. 16 1997), pp. 759-744.

[5] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, *"Probabilistic roadmaps for path planning in high-dimensional configuration spaces"* (IEEE Trans. Robotics and Automation, vol. 12, no. 4 1996), pp. 566-580.

[6] R. Katz, S. Hutchinson, *"Efficiently PRMs with Passage Potentials"* (IEEE Int. Conf. on Robotics and Autom., 2006), pp. 889-894.

[7] R. Geraerts, M. Overmars, "*The Corridor Map Method: Real-Time High-Quality Path Planning"*, (IEEE Int. Conf. on Robotics and Autom. 2007), pp. 1023-1028.

[8] P. Ögren, N. Leonard "*A convergent dynamic window approach to obstacle avoidance"* (IEEE Trans. Robotics, Vol. 21, no 2, 2005).

[9] L. Pacheco, N. Luo, "*Mobile Robot Local Predictive Control Using a Visual Perception Horizon"* (Int. Journal of Factory Autom., Robotics and Soft Comp., I. 2, April 2007), pp. 73-81.

[10] J. A. Reeds, L. A. Shepp, "*Optimal path for a car that goes forwards and backwards"*(Pacific Journal of Mathematics, vol. 145, 1990).

[11] J. M., Maciejowski, "*Predictive Control with Constraints"* (Ed. Prentice Hall 2002).

[12] G. S. Gupta, C. H., Messom,, S., Demidenko, "*Real-time identification and predictive control of fast mobile robots using global vision sensor*" (IEEE Trans. On Instr. and Measurement, Vol. 54, No. 1, 2005).

[13] F., Küne, W., Lages, W., J., Silva, "*Point stabilization of mobile robots with nonlinear model predictive control"* (Proc. IEEE Int. Conf. On Mech. and Aut., 2005), pp. 1163-1168.

[14] R. Lenain, B. Thuilot, C. Cariou, P. Martinet, "*Model Predictive Control for Vehicle Guidance in Presence of Sliding: Application to Farm Vehicles Path Tracking"* (Int. Conf. on Robotics and Autom., 2005), pp. 885-890.

[15] L. Lju, "*System Identification: Theory for the User*" (Ed. Prentice Hall, 1989).

[16] J. P. Norton, "*An Introduction to Identification"*, (Ed. Academic Press, New York, 1986).

[17] P. Van Overschee, B. Moor "*Subspace Identification for Linear Systems: Theory, Implementation, Applications*", (Ed., Kluwer Academic, 1996).

[18] S., Boyd, L., Vandenberghe, "*Convex Optimization"* (Cambridge University Press 2004).

[19] Y., Nesterov, A., Nemirovskii, "*Interior_Point Polynomial Methods in Convex Programming" (*SIAM Publications 1994).

[20] J. M., Ortega, W. C., Rheinboldt, W.C., "*Iterative Solution of Nonlinear Equations in Several Variables*" (Society for Industrial and Applied Mathematics 2000).

[21] E. F. Camacho, C. Bordons, "*Model Predictive Control*" (Springer-Verlag London 2002).

[22] K. Astrom, B. Wittenmark, "*Computed Controlled Systems: Theory and Design*" (Prentice-Hall Publ. 1988).

[23] A. Elfes, "*Using occupancy grids for mobile robot perception and navigation*" (IEEE Computer, Vol. 22, 1989), pp. 46-57.

[24] B. K. P. Horn, "*Robot Vision*" (McGraw-Hill Book Company, MIT Press Edition, 12th printing 1998).

[25] J. Campbell, R. Sukthankar, I. Noubakhsh "*Techniques for Evaluating Optical Flow in Extreme terrain*," (Proc. of IROS 2004).

[26] I. Nourbakhsh, D. Andre, C. Tomasi, M. R. Genesereth, "*Mobile Robot Obstacle Avoidance Via Depth From Focus"* (Robotics and Aut. Systems, Vol. 22, June, 1997), pp. 151-58

[27] R. C. Gonzalez, R. E. Woods, "*Digital Image Processing*," (Prentice Hall Int. Ed., 2th Ed., 2002).

[28] E. Elsayed Hemayed, "*A Survey of Camera Self-Calibration,"* (IEEE Conference on Adv. Video and Signal Based Surveillance 2003).

[29] R. J. Schilling, *Fundamental of Robotics*. (Prentice-Hall International 1990)

[30] L. Pacheco, N. Luo, "*Trajectory Planning with Control Horizon Based on Narrow Local Occupancy Grid Perception*" (Robot Motion and Control, Springer-Verlag, Kozlowski (Ed.), 2007) pp. 99-106.