

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Using Model Predictive Control for Local Navigation of Mobile Robots

Lluís Pacheco, Xavier Cufí and Ningsu Luo
*University of Girona
Spain*

1. Introduction

Model predictive control, MPC, has many interesting features for its application to mobile robot control. It is a more effective advanced control technique, as compared to the standard PID control, and has made a significant impact on industrial process control (Maciejowski, 2002). MPC usually contains the following three ideas:

- The model of the process is used to predict the future outputs along a horizon time.
- An index of performance is optimized by a control sequence computation.
- It is used a receding horizon idea, so at each instant of time the horizon is moved towards the future. It involves the application of the first control signal of the sequence computed at each step.

The majority of the research developed using MPC techniques and their application to WMR (wheeled mobile robots) is based on the fact that the reference trajectory is known beforehand (Klancar & Skrjanc, 2007). The use of mobile robot kinematics to predict future system outputs has been proposed in most of the different research developed (Kühne et al., 2005; Gupta et al., 2005). The use of kinematics have to include velocity and acceleration constraints to prevent WMR of unfeasible trajectory-tracking objectives. MPC applicability to vehicle guidance has been mainly addressed at path-tracking using different on-field fixed trajectories and using kinematics models. However, when dynamic environments or obstacle avoidance policies are considered, the navigation path planning must be constrained to the robot neighborhood where reactive behaviors are expected (Fox et al., 1997; Ögren & Leonard, 2005). Due to the unknown environment uncertainties, short prediction horizons have been proposed (Pacheco et al., 2008). In this context, the use of dynamic input-output models is proposed as a way to include the dynamic constraints within the system model for controller design. In order to do this, a set of dynamic models obtained from experimental robot system identification are used for predicting the horizon of available coordinates. Knowledge of different models can provide information about the dynamics of the robot, and consequently about the reactive parameters, as well as the safe stop distances. This work extends the use of on-line MPC as a suitable local path-tracking methodology by using a set of linear time-varying descriptions of the system dynamics when short prediction horizons are used. In the approach presented, the trajectory is dynamically updated by giving a straight line to be tracked. In this way, the control law considers the local point to be achieved and the WMR coordinates. The cost function is formulated with parameters that involve the capacity of turning and going straight. In the

case considered, the Euclidean distance between the robot and the desired trajectory can be used as a potential function. Such functions are CLF (control Lyapunov function), and consequently asymptotic stability with respect to the desired trajectory can be achieved. On-line MPC is tested by using the available WMR. A set of trajectories is used for analyzing the path-tracking performance. In this context, the different parameter weights of the cost function are studied. The experiments are developed by considering five different kinds of trajectories. Therefore, straight, wide left turning, less left turning, wide right turning, and less right turning are tested. Experiments are conducted by using factorial design with two levels of quantitative factors (Box et al., 2005). Studies are used as a way of inferring the weight of the different parameters used in the cost function. Factor tuning is achieved by considering aspects, such as the time taken, or trajectory deviation, within different local trajectories. Factor tuning depicts that flexible cost function as an event of the path to be followed, can improve control performance when compared with fixed cost functions. It is proposed to use local artificial potential attraction field coordinates as a way to attract WMR towards a local desired goal. Experiments are conducted by using a monocular perception system and local MPC path-tracking. On-line MPC is reported as a suitable navigation strategy for dynamics environments.

This chapter is organized as follows: Section 1 gives a brief presentation about the aim of the present work. In the Section 2, the WMR dynamic models are presented. This section also describes the MPC formulation, algorithms and simulated results for achieving local path-tracking. Section 3 presents the MPC implemented strategies and the experimental results developed in order to adjust the cost function parameters. The use of visual data is presented as a horizon where trajectories can be planned by using MPC strategies. In this context local MPC is tested as a suitable navigation strategy. Finally, in Section 4 some conclusions are made.

2. The control system identification and the MPC formulation

This section introduces the necessary previous background used for obtaining the control laws that are tested in this work as a suitable methodology for performing local navigation. The WMR PRIM, available in our lab, has been used in order to test and orient the research (Pacheco et al., 2009). Fig. 1 shows the robot PRIM and sensorial and system blocs used in

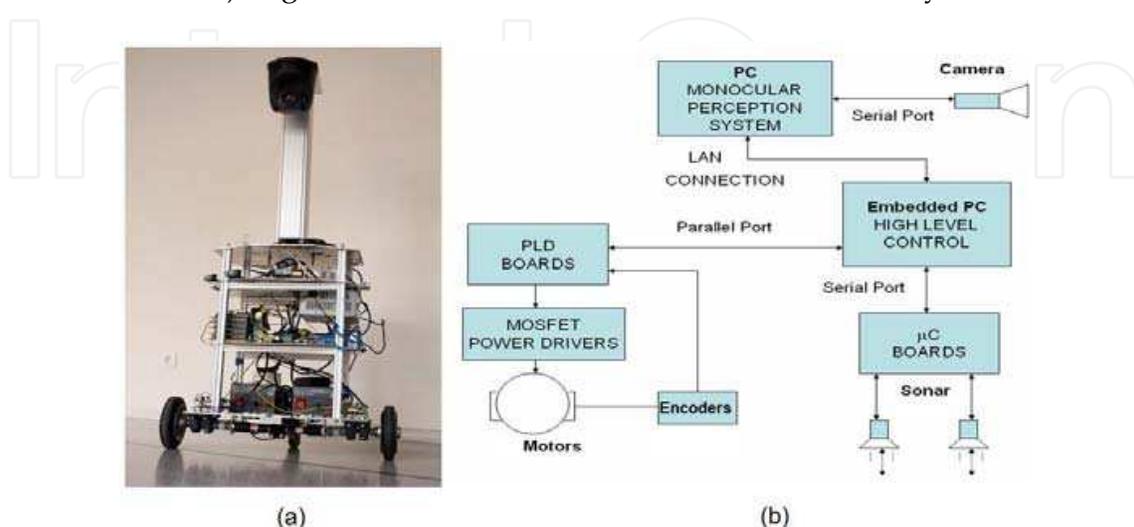


Fig. 1. (a) The robot PRIM used in this work; (b) The sensorial and electronic system blocs

the research work. The mobile robot consists of a differential driven one, with two independent wheels of 16cm diameters actuated by two DC motors. A third spherical omnidirectional wheel is used to guarantee the system stability. Next subsection deals with the problem of modeling the dynamics of the WMR system. Furthermore, dynamic MPC techniques for local trajectory tracking and some simulated results are introduced in the remaining subsections. A detailed explanation of the methods introduced in this section can be found in (Pacheco et al., 2008).

2.1 Experimental model and system identification

The model is obtained through the approach of a set of linear transfer functions that include the nonlinearities of the whole system. The parametric identification process is based on black box models (Norton, 1986; Ljung, 1989). The nonholonomic system dealt with in this work is considered initially to be a MIMO (multiple input multiple output) system, as shown in Fig. 2, due to the dynamic influence between two DC motors. This MIMO system is composed of a set of SISO (single input single output) subsystems with coupled connection.

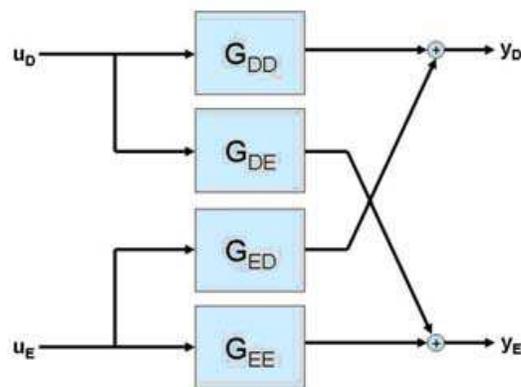


Fig. 2. The MIMO system structure

The parameter estimation is done by using a PRBS (Pseudo Random Binary Signal) such as excitation input signal. It guarantees the correct excitation of all dynamic sensible modes of the system along the whole spectral range and thus results in an accurate precision of parameter estimation. The experiments to be realized consist in exciting the two DC motors in different (low, medium and high) ranges of speed. The ARX (auto-regressive with external input) structure has been used to identify the parameters of the system. 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, the estimated output can be written as:

$$\hat{y} = \theta \varphi \quad (1)$$

with \hat{y} being the estimated output vector, θ the vector of estimated parameters and φ the vector of measured input and output variables. By using the coupled system structure, the transfer function of the robot can be expressed as follows:

$$\begin{pmatrix} Y_R \\ Y_L \end{pmatrix} = \begin{pmatrix} G_{RR} & G_{LR} \\ G_{RL} & G_{LL} \end{pmatrix} \begin{pmatrix} U_R \\ U_L \end{pmatrix} \quad (2)$$

where Y_R and Y_L represent the speeds of right and left wheels, and U_R and U_L the corresponding speed commands, respectively. In order to know the dynamics of robot system, the matrix of transfer function should be identified. In this way, speed responses to PBRS input signals are analyzed. The filtered data, which represent the average value of five different experiments with the same input signal, are used for identification. The system is identified by using the identification toolbox "ident" of Matlab for the second order models. Table 1 shows the continuous transfer functions obtained for the three different lineal speed models.

Linear Transfer Function	High velocities	Medium velocities	Low velocities
G_{DD}	$\frac{0.20s^2 - 3.15s + 9.42}{s^2 + 6.55s + 9.88}$	$\frac{0.20s^2 + 3.10s + 8.44}{s^2 + 6.17s + 9.14}$	$\frac{0.16s^2 + 2.26s + 5.42}{s^2 + 5.21s + 6.57}$
G_{ED}	$\frac{-0.04s^2 - 0.60s - 0.32}{s^2 + 6.55s + 9.88}$	$\frac{-0.02s^2 - 0.31s - 0.03}{s^2 + 6.17s + 9.14}$	$\frac{-0.02s^2 - 0.20s + 0.41}{s^2 + 5.21s + 6.57}$
G_{DE}	$\frac{-0.01s^2 - 0.08s - 0.36}{s^2 + 6.55s + 9.88}$	$\frac{0.01s^2 + 0.13s + 0.20}{s^2 + 6.17s + 9.14}$	$\frac{-0.01s^2 - 0.08s - 0.17}{s^2 + 5.21s + 6.57}$
G_{EE}	$\frac{0.31s^2 + 4.47s + 8.97}{s^2 + 6.55s + 9.88}$	$\frac{0.29s^2 + 4.11s + 8.40}{s^2 + 6.17s + 9.14}$	$\frac{0.25s^2 + 3.50s + 6.31}{s^2 + 5.21s + 6.57}$

Table 1. The second order WMR models

The coupling effects should be studied as a way of obtaining a reduced-order dynamic model. It can be seen from Table 1 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 neglect the coupling dynamics so as to obtain a simplified model. In order to verify the above facts from real results, a set of experiments have been done by sending a zero speed command to one motor and different non-zero speed commands to the other motor. The experimental result confirms that the coupled dynamics can be neglected. The existence of different gains in steady state is also verified experimentally. Finally, the order reduction of the system model is carried out through the analysis of pole positions by using the root locus method. It reveals the existence of a dominant pole and consequently the model order can be reduced from second order to first order. Table 2 shows the first order transfer functions obtained. Afterwards, the system models are validated through the experimental data by using the PBRS input signal.

Linear Transfer Function	High velocities	Medium velocities	Low velocities
G_{DD}	$\frac{0.95}{0.42s + 1}$	$\frac{0.92}{0.41s + 1}$	$\frac{0.82}{0.46s + 1}$
G_{EE}	$\frac{0.91}{0.24s + 1}$	$\frac{0.92}{0.27s + 1}$	$\frac{0.96}{0.33s + 1}$

Table 2. The reduced WMR model

2.2 Dynamic MPC techniques for local trajectory tracking

The minimization of path tracking error is considered to be a challenging subject in mobile robotics. In this subsection the LMPC (local model predictive control) techniques based on the dynamics models obtained in the previous subsection are presented. The use of dynamic models avoids the use of velocity and acceleration constraints used in other MPC research based on kinematic models. Moreover, contractive constraints are proposed as a way of guaranteeing convergence towards the desired coordinates. In addition, real-time implementations are easily implemented due to the fact that short prediction horizons are used. By using LMPC, the idea of a receding horizon can deal with local on-robot sensor information. The LMPC and contractive constraint formulations as well as the algorithms and simulations implemented are introduced in the next subsections.

2.2.1 The LMPC formulation

The main objective of highly precise motion tracking consists in minimizing the error between the robot and the desired path. Global path-planning becomes unfeasible since the sensorial system of some robots is just local. In this way, LMPC is proposed in order to use the available local perception data in the navigation strategies. Concretely, LMPC is based on minimizing a cost function related to the objectives for generating the optimal WMR inputs. Define the cost function as follows:

$$J(n, m) = \min_{\left\{ U(k+i|k) \right\}_{i=0}^{i=m-1}} \left\{ \begin{aligned} & \left[X(k+n|k) - X_{ld} \right]^T P \left[X(k+n|k) - X_{ld} \right] \\ & + \sum_{i=1}^{n-1} \left[X(k+i|k) - \overline{X_{ld} X_{l0}} \right]^T Q \left[X(k+i|k) - \overline{X_{ld} X_{l0}} \right] \\ & + \sum_{i=1}^{n-1} \left[\theta(k+i|k) - \theta_{ld} \right]^T R \left[\theta(k+i|k) - \theta_{ld} \right] \\ & + \sum_{i=0}^{m-1} U^T(k+i|k) S U(k+i|k) \end{aligned} \right\} \quad (3)$$

The first term of (3) refers to the attainment of the local desired coordinates, $X_{ld}=(x_d, y_d)$, where (x_d, y_d) denote the desired Cartesian coordinates. $X(k+n/k)$ represents the terminal value of the predicted output after the horizon of prediction n . The second one can be considered as an orientation term and is related to the distance between the predicted robot positions and the trajectory segment given by a straight line between the initial robot Cartesian coordinates $X_{l0}=(x_{l0}, y_{l0})$ from where the last perception was done and the desired local position, X_{ld} , to be achieved within the perceived field of view. This line orientation is denoted by θ_{ld} and denotes the desired orientation towards the local objective. $X(k+i/k)$ and $\theta(k+i/k)$ ($i=1, \dots, n-1$) represents the predicted Cartesian and orientation values within the prediction horizon. The third term is the predicted orientation error. The last one is related to the power signals assigned to each DC motor and are denoted as U . The parameters P , Q , R and S are weighting parameters that express the importance of each term. The control horizon is designed by the parameter m . The system constraints are also considered:

$$\left\{ \begin{aligned} & G_0 < |U(k)| \leq G_1 \quad \alpha \in (0, 1] \\ & |X(k+n/k) - X_{ld}| \leq \alpha |X(k) - X_{ld}| \\ & \text{or } |\theta(k+n/k) - \theta_{ld}| \leq \alpha |\theta(k) - \theta_{ld}| \end{aligned} \right\} \quad (4)$$

where $X(k)$ and $\theta(k)$ denote the current WMR coordinates and orientation, $X(k+n/k)$ and $\theta(k+n/k)$ denote the final predicted coordinates and orientation, respectively. The limitation of the input signal is taken into account in the first constraint, where G_0 and G_1 respectively denote the dead zone and saturation of the DC motors. The second and third terms are contractive constraints (Wang, 2007), which result in the convergence of coordinates or orientation to the objective, and should be accomplished at each control step.

2.2.2 The algorithms and simulated results

By using the basic ideas introduced in the previous subsection, the LMPC algorithms have the following steps:

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

The minimization of the cost function is a nonlinear problem in which the following equation should be verified:

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \quad (5)$$

The use of interior point methods can solve the above problem (Nesterov et al., 1994; Boyd & Vandenberghe, 2004). Gradient descent method and complete input search can be used for obtaining the optimal input. In order to reduce the set of possibilities, when optimal solution is searched for, some constraints over the DC motor 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 will result in the reduction of the computation time and the smooth behavior of the robot during the prediction horizon (Maciejowski, 2002). Thus, the set of available input is reduced to one value, as it is shown in Fig. 3.

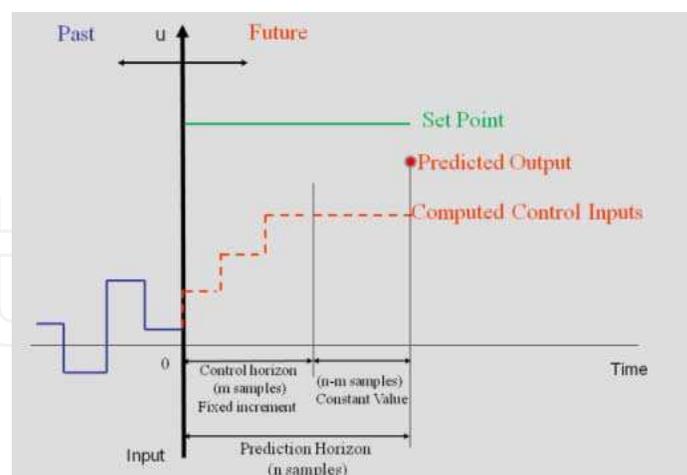


Fig. 3. LMPC strategy with fixed increment of the input during the control horizon and constant value for the remaining time

Both search methods perform accurate path-tracking. Optimal input search has better time performance and subinterval gradient descent method does not usually give the optimal solution. Due to these facts obtained from simulations, complete input search is selected for the on-robot experiences presented in the next section.

The evaluation of the LMPC performance is made by using different parametric values in the proposed cost function (3). In this way, when only the desired coordinates are considered, ($P=1, Q=0, R=0, S=0$), the trajectory-tracking is done with the inputs that can minimize the cost function by shifting the robot position to the left. The reason can be found in Table 2, where the right motor has more gain than the left one for high speeds. This problem can be solved, ($P=1, Q=1, R=0, S=0$) or ($P=1, Q=0, R=1, S=0$) by considering either the straight-line trajectory from the point where the last perception was done to the final desired point belonging to the local field of perception or the predicted orientations. Simulated results by testing both strategies provide similar satisfactory results. Thus, the straight line path or orientation should be considered in the LMPC cost function. Fig. 4 shows a simulated result of LMPC for WMR by using the orientation error, the trajectory distance and the final desired point for the cost function optimization ($P=1, Q=1, R=1, S=0$). Obtained results show the need of R parameter when meaningful orientation errors are produced.

The prediction horizon magnitude is also analyzed. The possible coordinates available for prediction when the horizon is larger ($n=10, m=5$), depict a less dense possibility of coordinates when compared with shorter horizons of prediction. Short prediction horizon strategy is more time effective and performs path-tracking with better accuracy. For these reasons, a short horizon strategy ($n=5, m=3$) is proposed for implementing experimental results.

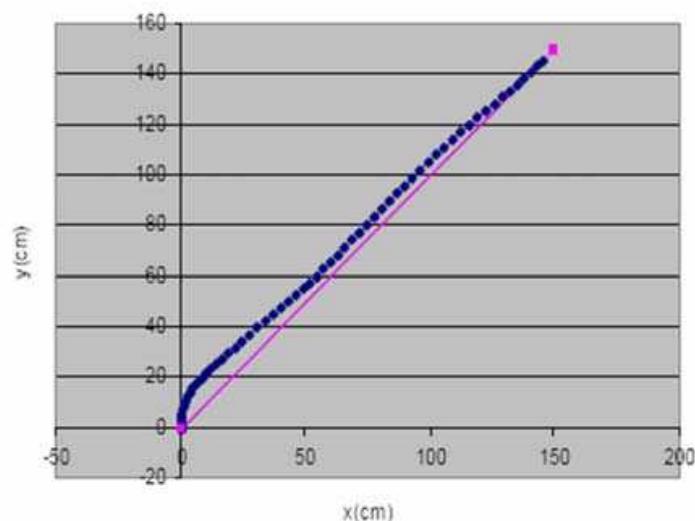


Fig. 4. Trajectory tracking simulated result by using the orientation error, trajectory distance and the final desired point for the optimization.

The sampling time for each LMPC step was set to 100ms. Simulation time performance of complete input search and gradient descent methods is computed. For short prediction horizon ($n=5, m=3$), the simulation processing time is less than 3ms for the complete input search strategy and less than 1ms for the gradient descent method when algorithms are running in a standard 2.7 GHz PC. Real on-robot algorithm time performance is also compared for different prediction horizons by using the embedded 700 Mhz PC and additional hardware system. Table 3 shows the LMPC processing time for different horizons of prediction when complete optimal values search or the gradient descent method are used. Surprisingly, when the horizon is increased the computing time is decreased. It is due to the fact that the control horizon is also incremented, and consequently less range of signal increments are possible because the signal increment is kept fixed within the control horizon. Thus, the maximum input value possibilities decrease with larger horizons. Hence for $n=5$ there are 1764 possibilities (42×42), and for $n=10$ there are 625 (25×25).

Horizon of prediction (n)	Complete search method	Gradient descent method
n=5	45ms	16ms
n=8	34ms	10ms
n=10	25ms	7ms

Table 3. LMPC processing times

3. Tuning the control law parameters by using path-tracking experimental results

In this section, path-tracking problem and the cost function parameter weights are analyzed, within a constrained field of perception provided by the on-robot sensor system. The main objective is to obtain further control law analysis by experimenting different kind of trajectories. The importance of the cost function parameter weights is analyzed by developing the factorial design of experiments for a representative set of local trajectories. Statistical results are compared and control law performance is analyzed as a function of the path to be followed. Experimental LMPC results are conducted by considering a constrained horizon of perception provided by a monocular camera where artificial potential fields are used in order to obtain the desired coordinates within the field of view of the robot.

3.1 The local field of perception

In order to test the LMPC by using constrained local perception, the field of view obtained by a monocular camera has been used. Ground available scene coordinates appear as an image, in which the camera setup and pose knowledge are used, and projective perspective is assumed to make each pixel coordinate correspond to a 3D scene coordinate (Horn, 1998). Fig. 5 shows a local map provided by the camera, which corresponds to a field of view with a horizontal angle of 48° , a vertical angle of 37° , H set to 109cm and a tilt angle of 32° .

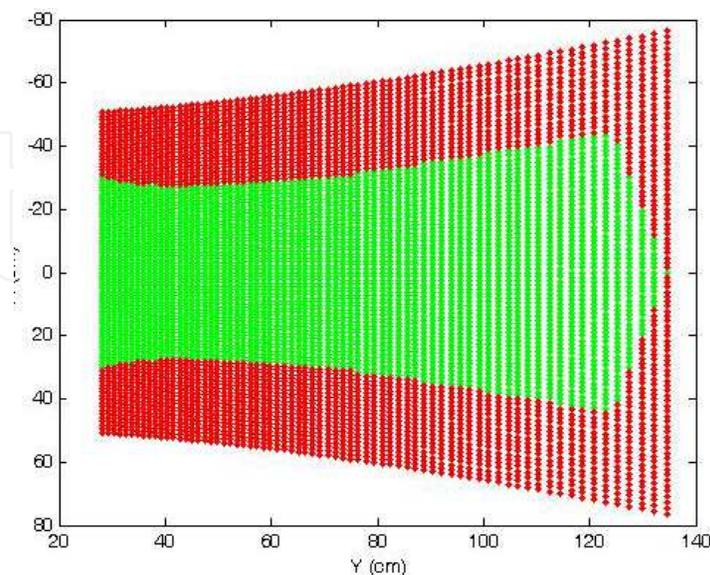


Fig. 5. Available local map coordinates (in green), the necessary coordinates free of obstacles and the necessary wide-path (in red).

It is pointed out that the available floor coordinates are reduced due to the WP (wide-path) of the robot (Schilling, 1990). It should also be noted that for each column position corresponding to scene coordinates Y_j , there are R row coordinates X_i . Once perception is introduced, the problem is formulated as finding the optimal cell that brings the WMR close to the desired coordinates (X_d, Y_d) by searching for the closest local desired coordinates (X_{ld}, Y_{ld}) within the available local coordinates (X_i, Y_j) . In this sense, perception is considered to be a local receding horizon on which the trajectory is planned. The local desired cell is obtained by minimizing a cost function J that should act as a potential field corridor. Thus, the cost function is minimized by attracting the robot to the desired objective through the free available local cell coordinates. It is noted that from local perception analysis and attraction potential fields a local on field path can be obtained. The subsequent subsections infer control law parameter analysis by considering a set of path possibilities obtained within the perception field mentioned in this section.

3.2 The path-tracking experimental approach by using LMPC methods

The path tracking performance is improved by the adequate choice of a cost function that is derived from (3) and consists of a quadratic expression containing some of the following four parameters to be minimized:

- The squared Euclidean *approaching point distance* (APD) between the local desired coordinates, provided by the on-robot perception system, and the actual robot position. It corresponds with the parameter “ P ” of the LMPC cost function given by (3).
- The squared *trajectory deviation distance* (TDD) 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. It corresponds with the parameter “ Q ” of the cost function shown by (3).
- The third parameter consists of the squared *orientation deviation* (OD); it is expressed by the difference between the robot desired and real orientations. It corresponds with the parameter “ R ” of the LMPC cost function depicted by (3).
- The last parameter refers to changes allowed to the input signal. It corresponds with the parameter “ S ” of the LMPC cost function given by (3).

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 magnitude of the deviation distance is normally in the order of cm, which becomes effective only when the robot is approaching the final desired point. Hence, when reducing the deviation distance further to less than 1cm is attempted, an increase, in the weight value for the deviation distance in the cost function, is proposed.

The subsequent subsections use statistical knowledge for inferring APD (P) and TDD (Q) or APD (P) and OD (R) factor performances as a function of the kind of paths to be tracked. Other cost function parameters are assumed to be equal to zero.

3.3 Experimental tuning of APD and TDD factors

This subsection presents the results achieved by using factorial design in order to study the LMPC cost function tuning when APD and TDD factors are used. Path-tracking performance is analyzed by the mean of the different factor weights. The experiments are developed by considering five different kinds of trajectories within the reduced field of view as shown in Fig. 5. Therefore, straight, wide left turning, less left turning, wide right turning

and less right turning trajectories are tested. Experiments are conducted by using factorial design with two levels of quantitative factors (Box et al, 2005). Referred to the cost function, let us assume that high value (H) is equal to "1" and low value (L) is equal to "0.5". For each combination of factors three different runs are experimented. The averaged value of the three runs allows statistical analysis for each factor combination. From these standard deviations, the importance of the factor effects can be determined by using a rough rule that considers the effects when the value differences are similar or greater than 2 or 3 times their standard deviations. In this context, the main effects and lateral effects, related to APD and TDD, are analyzed. Fig. 6 shows the four factor combinations (APD, TDD) obtained by both factors with two level values.

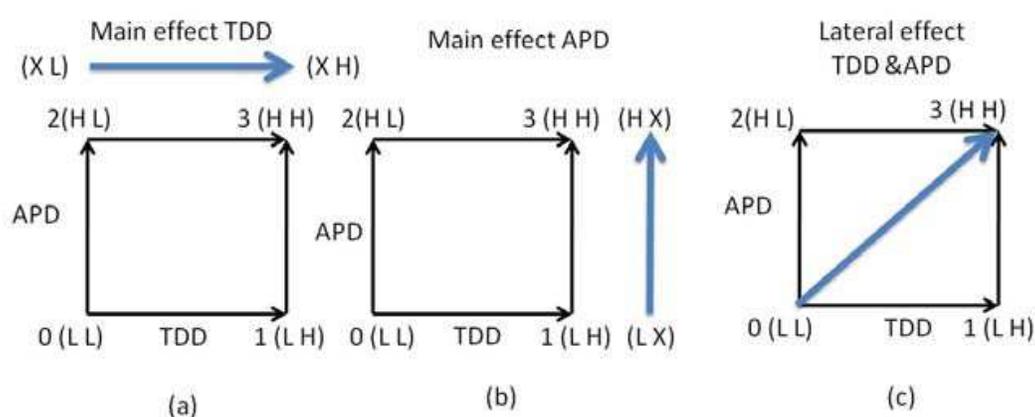


Fig. 6. The different factor combinations and the influence directions, in which the performances should be analyzed.

The combinations used for detecting lateral and main effect combinations are highlighted by blue arrows. Thus, the main effect of APD factor, ME_{APD} , can be computed by the following expression:

$$ME_{APD} = \frac{Y_3 + Y_2}{2} - \frac{Y_1 + Y_0}{2} \quad (6)$$

Path-tracking statistical performances to be analyzed in this research are represented by Y . The subscripts depict the different factor combinations. The main effect for TDD factor, ME_{TDD} , is computed by:

$$ME_{TDD} = \frac{Y_3 + Y_1}{2} - \frac{Y_2 + Y_0}{2} \quad (7)$$

The lateral effects are computed by using the following expression:

$$LE_{APD_TDD} = Y_3 - Y_0 \quad (8)$$

The detailed measured statistics with parameters such as time (T), trajectory error (TE), and averaged speeds (AS) are presented in (Pacheco & Luo, 2011). The results were tested for straight trajectories, wide and less left turnings, and wide and less right turnings. The main and lateral effects are represented in Table 4.

The performance is analyzed for the different trajectories:

- The factorial analysis for straight line trajectories, ($\sigma_T = 0.16s$, $\sigma_{TE} = 0.13cm$, $\sigma_{AS} = 2.15cm/s$), depicts a main time APD effect of $-0.45s$, and an important lateral effect of $-0.6s$ and $-0.32cm$. Speed lateral effect of only $1.9cm/s$ is not considered as meaningful. Considering lateral effects that improve time and accuracy, high values (APD, TDD) are proposed for both factors.
- The analysis for wide left turning trajectories, ($\sigma_T = 0.26s$, $\sigma_{TE} = 0.09cm$, $\sigma_{AS} = 0.54cm/s$) show negative APD main effect of $0.53s$, and $0.15cm$. However, the TDD factor tends to decrease the time and trajectory deviation. The $0.3cm/s$ speed TDD main factor is irrelevant. In this case, low value for APD factor and high value for the TDD factor is proposed.
- The factor analysis for less left turning, ($\sigma_T = 0.29s$, $\sigma_{TE} = 0.36cm$, $\sigma_{AS} = 0.84cm/s$), depicts a considerable lateral effect of $-0.46s$ and $-0.31cm$. Speed $-0.2cm/s$ lateral effect is not important. In this sense high values are proposed for APD and TDD factors.
- The analysis for wide right turning, ($\sigma_T = 0.18s$, $\sigma_{TE} = 0.15cm$, $\sigma_{AS} = 1.04cm/s$) does not provide relevant clues, but small time improvement seems to appear when TDD factor is set to a low value. Low values are proposed for APD and TDD factors.
- Finally, the factorial analysis for less right turning trajectories, ($\sigma_T = 0.12s$, $\sigma_{TE} = 0.18cm$, $\sigma_{AS} = 1.94cm/s$), depicts APD and lateral effects that increase the trajectory time with $0.32s$ and $0.44s$. Main or lateral effects related to the speed have not been detected. Low values are proposed for APD and TDD factors.

Straight line trajectory			
Parameter Performance	Main Effect TDD factor	Main Effect APD factor	Lateral Effect TDD & APD factors
Time	$-0.05s$	$-0.45s$	$-0.6s$
Trajectory accuracy	$-0.18cm$	$-0.14cm$	$-0.32cm$
Averaged speed	$1.25cm/s$	$0.6cm/s$	$1.9cm/s$
Wide left turn trajectory			
Time	$-0.34s$	$0.53s$	$0.16s$
Trajectory accuracy	$-0.17cm$	$0.15cm$	$-0.01cm$
Averaged speed	$0.3cm/s$	$0.4cm/s$	$0.7cm/s$
Slight left turn trajectory			
Time	$-0.24s$	$0.02s$	$-0.46s$
Trajectory accuracy	$-0.14cm$	$-0.17cm$	$-0.31cm$
Averaged speed	$0.8cm/s$	$-1cm/s$	$-0.2cm/s$
Wide right turn trajectory			
Time	$0.27s$	$-0.10s$	$0.17s$
Trajectory accuracy	$-0.22cm$	$0.1cm$	$-0.12cm$
Averaged speed	$0.7cm/s$	$0.2cm/s$	$0.9cm/s$
Slight right turn trajectory			
Time	$0.12s$	$0.32s$	$0.44s$
Trajectory accuracy	$-0.18cm$	$-0.06cm$	$-0.25cm$
Averaged speed	$-1.3cm/s$	$2.8cm/s$	$1.5cm/s$

Table 4. Main and lateral effects

3.4 Experimental performance by using fixed or flexible APD & TDD factors

Once factorial analysis is carried out, this subsection compares path-tracking performance by using different control strategies. The experiments developed consist in analyzing the performance when a fixed factor cost function or a flexible factor cost function is used. The trajectories to be analyzed are formed by straight lines, less right or left turnings, and wide right or left turnings. The fixed factor cost function maintains the high values for APD and TDD factors, while the flexible factor cost function is tested as function of the path to be tracked.

Different experiments are done; see (Pacheco & Luo, 2011). As instance one experiment consists in tracking a trajectory that is composed of four points $((0, 0), (-25, 40), (-25, 120), (0, 160))$ given as (x, y) coordinates in cm. It consists of wide left turning, straight line and wide right turning trajectories. The results obtained by using fixed and flexible factor cost function are depicted in Table 5. Three runs are obtained for each control strategy and consequently path-tracking performance analysis can be done.

Results show that flexible factor strategy improves an 8% the total time performance of the fixed factor strategy. The turning trajectories are done near 50% of the path performed. Remaining path consists of a straight line trajectory that is performed with same cost

Trajectory coordinates (cm): $(0, 0), (-25, 40), (-25, 120), (0, 160)$

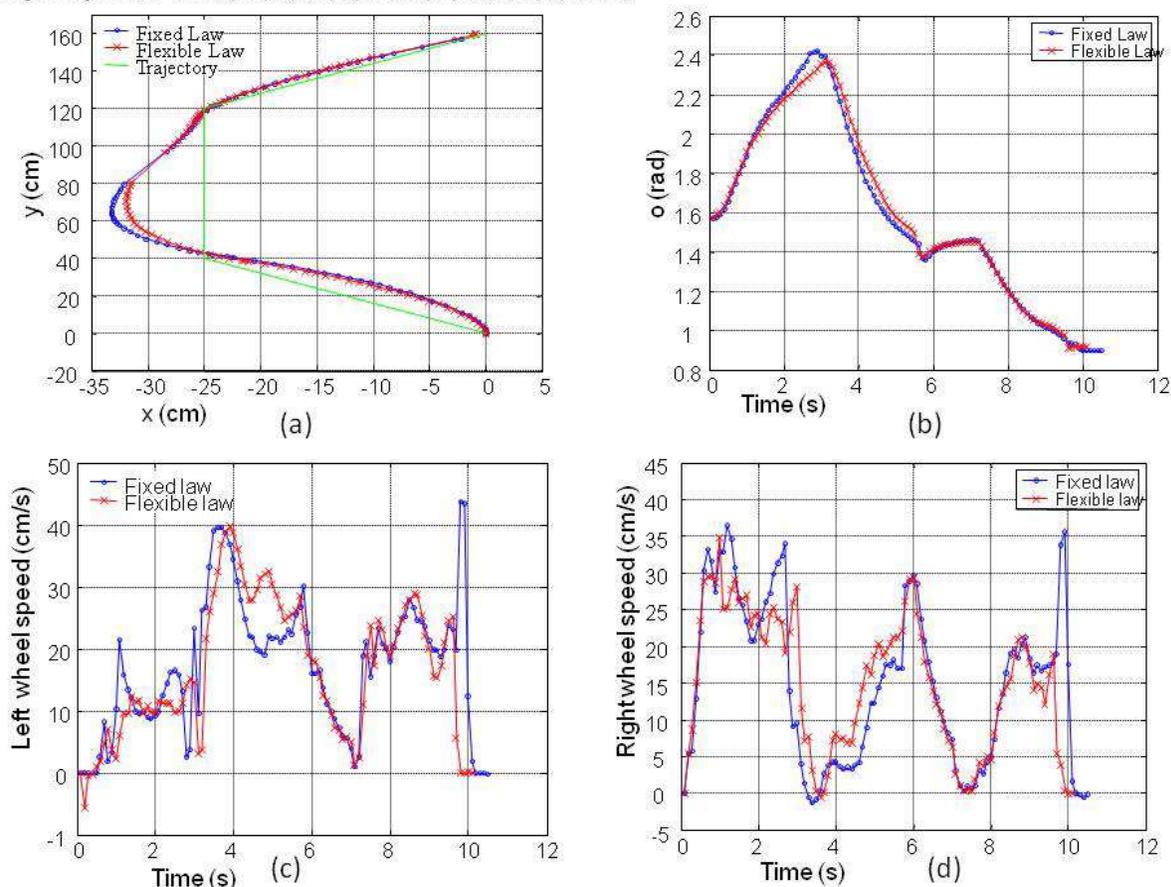


Fig. 7. (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

function values for fixed and flexible control laws. It is during the turning actions, where the two control laws have differences, when time improvement is nearly 16%. Fig. 7 shows an example of some results achieved. Path-tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies are shown.

It can be seen that flexible cost function, when wide left turning is performed approximately during the first three seconds, produces less maximum speed values when compared with fixed one. However, a major number of local maximum and minimum are obtained. It results in less trajectory deviation when straight line trajectory is commanded. In general flexible cost function produces less trajectory error with less orientation changes and improves time performance.

	Trajectory points: (0,0), (-25,40), (-25,120), (0,160) ((x,y) in cm)					
	Time (s)		Trajectory error (cm)		Averaged Speed (cm/s)	
Experiment	Fixed Law	Flexible Law	Fixed Law	Flexible Law	Fixed Law	Flexible Law
Run 1	10,5	10,3	3,243	3,653	18,209	16,140
Run 2	10,9	9,8	3,194	2,838	16,770	16,632
Mean	10,70	10,05	3,219	3,245	17,489	16,386
Variance	0,0800	0,1250	0,0012	0,3322	1,0354	0,1210
Standart deviation	0,2828	0,3536	0,0346	0,5764	1,0175	0,3479

Table 5. Results obtained by using fixed or flexible cost function

Developed experiences with our WMR platform show that flexible LMPC cost function related with the path to be tracked can improve the control system performance.

3.5 Experimental tuning using APD and OD factors

In a similar way APD and OD factors can be used. This subsection compares path-tracking performance by using different control strategies. The experiments developed consist in analyzing the performance when a fixed factor cost function or a flexible factor cost function is used. The trajectories to be analyzed are formed by straight lines, less right or left turnings, and wide right or left turnings. The fixed factor cost function maintains the high values for APD and OD factors, while the flexible factor cost function is tested as function of the path to be tracked. The experiments developed show the measured performance statistics, time, trajectory accuracy, and averaged speeds, for straight trajectories, wide and less left turnings, and wide and less right turnings. The standard deviation obtained as well as the main and lateral effects are represented in Table 6. The time, trajectory error and averaged speed standard deviations are respectively denoted by σ_T , σ_{TE} , and σ_{AS} . Table 6 represents the experimental statistic results obtained for the set of proposed trajectories. The standard deviations computed for each kind of trajectory by testing the different factor weights under different runs are also depicted.

The main and lateral effects were calculated by using (6), (7), (8), and the mean values obtained for the different factor combinations. Therefore, in Table 6 are highlighted the

significant results achieved using experimental factorial analysis. The inferred results obtained can be tested using different trajectories.

Straight trajectory			
Parameters	OD	APD	APD & OD
Time (s) $\sigma_T = 0.06s$	0,02	-0,13	-0,10
Trajectory error (cm) $\sigma_{TE} = 0.69cm$	-0,24	1,34	1,10
Speed (cm/s) $\sigma_{AS} = 0.88cm/s$	0,87	0,70	1,57
Wide left turning			
Parameters	OD	APD	APD & OD
Time (s) $\sigma_T = 0.06s$	-0,10	0,20	0,10
Trajectory error (cm) $\sigma_{TE} = 0.18cm$	0,36	0,38	0,02
Speed (cm/s) $\sigma_{AS} = 0.59cm/s$	0,36	-0,87	-0,52
Less left turning			
Parameters	OD	APD	APD & OD
Time (s) $\sigma_T = 0.09s$	-0,12	0,07	-0,05
Trajectory error (cm) $\sigma_{TE} = 0.11cm$	0,58	1,08	0,50
Speed (cm/s) $\sigma_{AS} = 0.92cm/s$	0,60	-0,13	0,47
Wide right turning			
Parameters	OD	APD	APD & OD
Time (s) $\sigma_T = 0.11s$	0,10	0,35	0,45
Trajectory error (cm) $\sigma_{TE} = 0.08cm$	0,44	0,45	0,01
Speed (cm/s) $\sigma_{AS} = 0.67cm/s$	-0,58	-1,67	-2,25
Less right turning			
Parameters	OD	APD	APD & OD
Time (s) $\sigma_T = 0.26s$	-0,07	0,07	0,00
Trajectory error (cm) $\sigma_{TE} = 0.20cm$	1,38	0,65	-0,73
Speed (cm/s) $\sigma_{AS} = 0.13cm/s$	-0,33	-0,14	-0,48

Table 6. Main and lateral effects

The experiments developed consist in analyzing the time performance when a fixed factor cost function or a flexible factor cost function is used. The trajectories to be analyzed are formed by straight lines, less right or left turnings, and wide right or left turnings. The fixed factor cost function maintains the high values for APD and OD factors, while the flexible factor cost function is tested as function of the trajectory to be tracked. The experiments presented consist in tracking a trajectory that is composed of three points $((0, 0), (-25, 40), (-25, 120))$ given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost function are depicted in Table 7.

Trajectory (x,y) in cm: (0,0), (-25,40), (-25,120)						
Features	Time (s)		Error (cm)		Aver. speed (cm/s)	
	Fixed	Flexible	Fixed	Flexible	Fixed	Flexible
Run 1	7,2	7,0	3,8	3,0	19,4	17,5
Run 2	7,4	6,6	2,2	3,5	16,5	20,1
Mean	7,3	6,8	3,0	3,2	18,0	18,8
Variance	0,02	0,1	1,3	0,1	4,2	3,4
Stand. dev.	0,14	0,3	1,1	0,3	2,0	1,9

Table 7. Experimental performances

Two runs are obtained for each strategy and consequently time performance analysis can be done. The averaged standard deviation between the two cost function systems is of 0.22s, and the difference of means are 0.5s. Thus, flexible factor strategy improves a 6.85% the time performance of the fixed factor strategy. However, left turning is done only a 33% of the trajectory. Thus, time improvement during the left turning is of near 20%. Fig. 8 shows an example of some results achieved. Path-tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies are shown. Trajectory error and averaged speed statistical results are not significant, due to the fact that the differences of means between fixed and flexible laws are less than two times the standard deviations.

4. Conclusion

This research can be used on dynamic environments in the neighborhood of the robot. On-line LMPC is a suitable solution for low level path-tracking. LMPC is more time expensive when compared with traditional PID controllers. However, instead of PID speed control approaches, LMPC is based on a horizon of available coordinates within short prediction horizons that act as a reactive horizon. Therefore, path planning and convergence to coordinates can be more easily implemented by using LMPC methods. In this way, contractive constraints are used for guaranteeing the convergence towards the desired coordinates. The use of different dynamic models avoids the need of kinematical constraints that are inherent to other MPC techniques applied to WMR. In this context the control law is based on the consideration of two factors that consist of going straight or turning. Therefore, orientation deviation or trajectory deviation distance can be used as turning factors. The methodology used for performing the experiments is shown. From on-robot depicted experiences, the use of flexible cost functions with relationships to the path to be tracked can be considered as an important result. Thus, control system performance can be improved by considering different factor weights as a function of path to be followed.

The necessary horizon of perception is constrained to just few seconds of trajectory planning. The short horizons allow real time implementations and accuracy trajectory tracking. The experimental LMPC processing time was 45ms, ($m=3$, $n=5$), running in the WMR embedded PC of 700MHz. The algorithms simplicity is another relevant result obtained. The factorial design, with two levels of quantitative factors, is presented as an easy way to infer experimental statistical data that allow testing feature performances as function

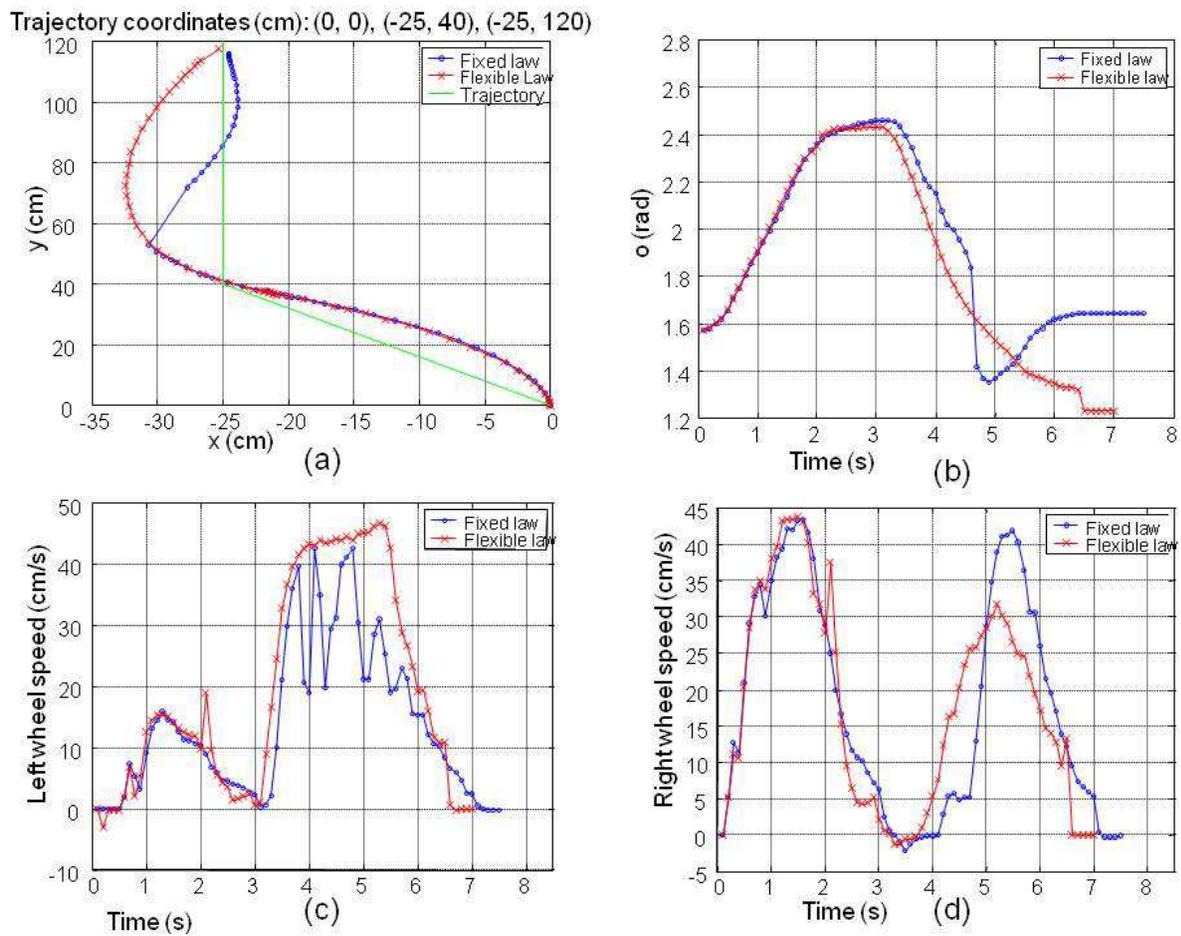


Fig. 8. (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

of the different factor combinations. Further studies on LMPC should be done in order to analyze its relative performance with respect to other control laws or to test the cost function performance when other factors are used. The influence of the motor dead zones is also an interesting aspect that should make further efforts to deal with it.

5. Acknowledgement

This work has been partially funded by the Commission of Science and Technology of Spain (CICYT) through the coordinated projects DPI2007-66796-C03-02 and DPI 2008-06699-C02-01.

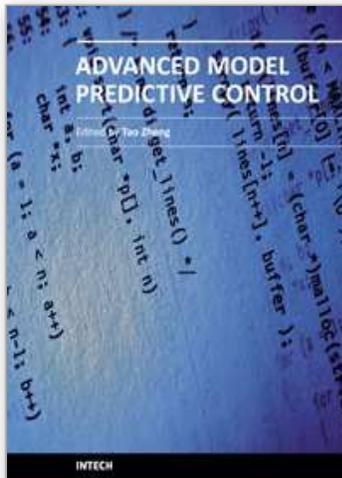
6. References

- Box, G. E. P., Hunter, J. S., Hunter W. G. (2005). *Statistics for Experimenters*, Ed. John Wiley & Sons, ISBN 0-471-71813-0, New Jersey (USA).
- Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*, Cambridge University Press, ISBN-13: 9780521833783, New York (USA).
- Fox, D.; Burgard, W. & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, Vol. 4, No. 1, (Mar 1997) 23-33, ISSN 1070-9932
- Gupta, G.S.; Messom, C.H. & Demidenko, S. (2005). Real-time identification and predictive control of fast mobile robots using global vision sensor. *IEEE Trans. On Instr. and Measurement*, Vol. 54, No. 1, (February 2005) 200-214, ISSN 1557-9662
- Horn, B. K. P. (1998). *Robot Vision*. MIT press, Ed. McGraw-Hill, ISBN 0-262-08159-8, London (England)
- Klancar, G & Skrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time, *Robotics and Autonomous Systems*, Vol. 55, No. 6, (June 2007), pp. 460-469, ISSN: 0921-8890.
- Kühne, F.; Lages, W. F. & Gomes da Silva Jr., J. M., (2005). Model predictive control of a mobile robot using input-output linearization, *Proceedings of Mechatronics and Robotics*, ISBN 0-7803-9044-X, Niagara Falls Canada, July 2005
- Ljung, L. (1991). Issues in System Identification. *IEEE Control Systems Magazine*, Vol. 11, No.1, (January 1991) 25-29, ISSN 0272-1708
- Maciejowski, J.M. (2002). *Predictive Control with Constraints*, Ed. Prentice Hall, ISBN 0-201-39823-0, Essex (England)
- Nesterov, I. E.; Nemirovskii, A. & Nesterov, Y. (1994). *Interior Point Polynomial Methods in Convex Programming*. Siam Studies in Applied Mathematics, Vol 13, Publications, ISBN 0898713196
- Norton, J. (1986). *An Introduction to Identification*. Academic Press, ISBN 0125217307, London and New York, 1986
- Ögren, P. & Leonard, N. E. (2005). A convergent dynamic window approach to obstacle avoidance. *IEEE Transaction on Robotics*, Vol. 21, No. 2., (April 2005) 188-195, ISSN: 1552-3098
- Pacheco, L., Luo, N., Cufí, X. (2008). Predictive Control with Local Visual Data, In: *Robotics, Automation and Control*, Percherková, P., Flídr, M., Duník, J., pp. 289-306, Publisher I-TECH, ISBN 978-953-7619-18-4, Printed in Croatia.
- Pacheco, L., Luo, N.; Ferrer, I. and Cufí, X. (2009). Interdisciplinary Knowledge Integration Through an Applied Mobile Robotics Course, *The International Journal of Engineering Education*, Vol. 25, No. 4, (July, 2009), pp. 830-840, ISSN: 0949-149X
- Pacheco, L., Luo, N. (2011) Mobile robot local trajectory tracking with dynamic model predictive control techniques, *International Journal of Innovative Computing, Information and Control*, Vol.7, No.6, (June 2011), in press, ISSN 1349-4198
- Schilling, R.J. (1990). *Fundamental of Robotics*. Prentice-Hall (Ed.), New Jersey (USA) 1990, ISBN 0-13-334376-6

Wan, J. (2007) Computational reliable approaches of contractive MPC for discrete-time systems, PhD Thesis, University of Girona.

IntechOpen

IntechOpen



Advanced Model Predictive Control

Edited by Dr. Tao ZHENG

ISBN 978-953-307-298-2

Hard cover, 418 pages

Publisher InTech

Published online 24, June, 2011

Published in print edition June, 2011

Model Predictive Control (MPC) refers to a class of control algorithms in which a dynamic process model is used to predict and optimize process performance. From lower request of modeling accuracy and robustness to complicated process plants, MPC has been widely accepted in many practical fields. As the guide for researchers and engineers all over the world concerned with the latest developments of MPC, the purpose of "Advanced Model Predictive Control" is to show the readers the recent achievements in this area. The first part of this exciting book will help you comprehend the frontiers in theoretical research of MPC, such as Fast MPC, Nonlinear MPC, Distributed MPC, Multi-Dimensional MPC and Fuzzy-Neural MPC. In the second part, several excellent applications of MPC in modern industry are proposed and efficient commercial software for MPC is introduced. Because of its special industrial origin, we believe that MPC will remain energetic in the future.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Lluís Pacheco, Xavier Cufí and Ningsu Luo (2011). Using Model Predictive Control for Local Navigation of Mobile Robots, Advanced Model Predictive Control, Dr. Tao ZHENG (Ed.), ISBN: 978-953-307-298-2, InTech, Available from: <http://www.intechopen.com/books/advanced-model-predictive-control/using-model-predictive-control-for-local-navigation-of-mobile-robots>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen