ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

Motion Control of Wheeled Mobile Robots with Model Predictive Techniques

L. Pacheco. N. Luo
Institute of Computer Vision and Robotics, UdG, E-17071 SPAIN
Institute of Informatics and Applications, UdG, E-17071 SPAIN

Abstract*—This work outlines MPC (model predictive control) studies for mobile robots when local motion is considered. Different kinematic mobile robot configurations are introduced. Experiences are conducted by using a differential driven WMR (wheeled mobile robot) as an available vehicle that can represent nonholonomic kinematics constraints. Motion control is done using dynamic models, and selecting orientation and local desired-point distances as control law parameters. MPC is implemented by computing the horizon of suitable coordinates. Command speeds corresponding to the desired point are obtained by minimizing a cost function in which the population of available coordinates is taken into account. The research reported in this paper includes a set of experimental results obtained by testing different trajectories. Parameters such as time, accuracy, speed, and travelled distance are compared from statistics for different control laws.

***Index Terms*—Local motion control, autonomous mobile robots, kinematic and dynamic models, model predictive control.**

I. INTRODUCTION

WMR (wheeled mobile robots) have locomotion disadvantages in harsh environments when are compared with legged robots. However, when flat surfaces are considered, WMR can perform specific tasks faster and are more energy efficient. In this context, WMR are increasing their industrial applicability. Moreover, other research fields as service robotics, where flexible robot capabilities are required, are continuously growing and opening new challenges [1]. WMR navigation is a necessary robot ability in which safe conditions are preserved while obstacle avoidance and goal approaching policies are performed. Robot navigation involves robot ability in order to determine its own position and to plan a safe path towards goal [2]. Path-planning is a necessary task that consists of finding a feasible path from current robot position to the desired environment coordinates [3]. Local path-planning, in which only a segment of global path is considered, sometimes is also known as motion-planning because WMR kinematics and dynamics are considered. Therefore, avoid obstacles and local goal achievements should be performed dealing with motion planning that means to select and control a set of dynamic and kinematic parameters such as: heading angles, appropriate speeds, safety stops, etc [4]. In this context, local path-following is related with the ability of the robot to accurately follow the path by performing a motion control strategy that generates system outputs as positions and velocities, which are close to the desired output profiles [5]. Classical control theory is normally used by implementing low level closed loop solutions based on PID controllers [6]. This work deals with low level motion control of a nonholonomic WMR. In this context, local path-planning and motion control are strongly related with WMR kinematics and dynamics. Such issues are presented from a general point of view that considers WMR kinematics and dynamics for different mobility configurations. Therefore, WMR are classified as holonomic or nonholonomic. Nonholonomic vehicles perform motions which are constrained by non integrable constraints [7]. Once WMR kinematics are known, motion control strategies consist of performing local path-tracking by using low level controllers and constraining the set of velocities and speeds. Despite PID controllers are commonly used as low level control laws, MPC (model predictive Control) can be also an effective control technique [8]-[9]. In this work, on-line LMPC (local model predictive Control) is tested by using the available WMR. This article is organized as follows: Section I gives a brief presentation about the aim of the present work. In Section II, the set of WMR kinematic configurations are presented. Nonholonomic systems are analyzed and formulated as a set of kinematic models that allow motion planning when the speeds and accelerations of wheels are controlled. In Section III, local path-planning is introduced by using kinematic models. Furthermore, experimental dynamic models are obtained as useful data for implementing low level control methods. MPC formulation, algorithms and simulated results for achieving local path-tracking are described in Section IV. Section V presents LMPC implemented strategies and the experimental results developed in order to adjust the control law parameters. Finally, in Section VI conclusions are made.

II. KINEMATIC MODELS OF WMR

Local path-planning not only consists of finding free of collisions trajectories that approach WMR to the goal. Feasible path-tracking is performed when the kinematic configuration of the robot is considered as well as the constraints related with robot mobility. Usually, WMR are constrained to horizontal plane frameworks in which robot configuration is given by (X, Y, θ). X and Y denote de ground-plane vehicle coordinates and θ the heading orientation. WMR are classified according to their kinematic properties that have a great influence when local path-planning is performed. Nonholononic WMR are

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

presented as an important segment of vehicles. In this way, developed experiences are addressed to this kind of vehicles.

A. Kinematic Classification of WMR

Wheels are analyzed by considering their rotation constraints; we can make a classification of the basic different types: fixed standard wheels, centered orientable wheels, off-centered orientable wheels, and Swedish wheels. Figure 1 show these wheels, it is noted that types (a), (b), (c) use universal wheels while type (d) can be considered as a special kind of wheels that allow lateral rotation. Fixed standard wheels can rotate only along a horizontal axis; the angle between the WMR structure and the wheel axis is fixed. Centered orientable wheels have an additional degree of freedom that allows rotation around a vertical axis. Off-centered orientable wheels are similar to the centered orientable wheels but the vertical axis of rotation does not pass through the ground contact point, these wheels are also known as castor wheels. Similar to the off-centered orientable wheels, the Swedish wheels allow omnidirectional movements through the ground plane. However, Swedish wheels have not a vertical axis of rotation; they are configured as a universal wheel that has attached rollers to the wheel circumference. The axes of rotation of the rollers are perpendicular to the horizontal main axis of the universal wheel. Considering the different kind of wheels and the mobility restrictions, a kinematic classification of five different types of WMR is obtained [10]-[11]. Class 1 can be understood by a model that uses three Swedish wheels actuated by 3 motors that provide the rotating speed to each wheel, see Figure 2. Slide sideways is accomplished by using omnidirectional wheels that allow instantaneously movements in any direction. Such vehicles are called omnidirectional or holonomic WMR because they can reach any direction without the need of reorientation.

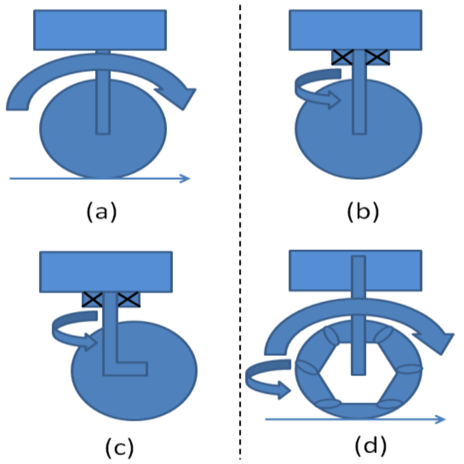


Fig. 1. (a) Fixed wheel (b) Centered orientable wheel (c) Off-centered orientable wheel also known as castor wheel (d) Swedish wheel.

Class 2 can be represented by a kinematic model that uses 2 fixed wheels; a third omnidirectional or off-centered orientable wheel is used to give stability to the robot. They are usually known as differential driven WMR because forward turning is achieved by commanding different speed to each fixed wheel. Figure 2 depicts an example of this type of robots that are considered as nonholonomic vehicles. Class 3 does not have any fixed wheels but has at least one conventional centered orientable wheel. Figure 3 depicts an example of such model that uses a centered orientable wheel and two off-centered orientable wheels which give stability to the system allowing movements in any direction. Vertical rotation axis is used in order to modify the heading orientation of the robot while horizontal rotation axes are used to perform displacements by using a determined heading direction.

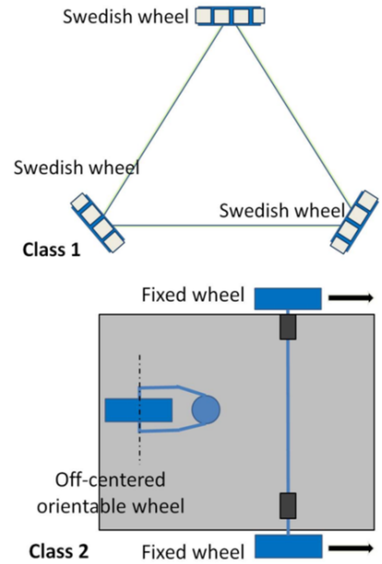


Fig. 2. Kinematic class 1, holonomic WMR with 3 Swedish wheels. Class 2 depicts a nonholonomic WMR with two fixed wheels and one castor wheel.

Class 4 is represented by a model that has two fixed wheels on the same axle and a centered steerable wheel, see Figure 3. It is similar to tricycles of kids or cars where the orientation is selected by a steering wheel and the angular speed of the wheels is shared by using a common axle. This WMR can be driven by two motors that are used for selecting the desired heading orientation and the rotating axle speed of the fixed wheels. Steering angles of such vehicles are constrained to a reduced set of values as happens in real tricycles and cars. This type of WMR is considered as a nonholonomic vehicle. Class 5 vehicles can be depicted by a model that have two centered orientable wheels. The vehicle has not fixed wheels, vehicle stability is provided by a third off-centered orientable wheel, see Figure 3. As in class 1, and class 3 types, WMR can achieve any configuration. Consequently, these vehicles are called omnidirectional or holonomic WMR.

B. Kinematics of a Nonholonomic WMR

Despite the fact that holonomic vehicles have advantages when manoeuvrability is considered, nonholonomic vehicles are commonly used. Time efficiency and vehicle cost are advantages of a nonholonomic WMR. In this subsection

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

kinematic models of Class 4 and Class 2 are introduced with more level of detail.

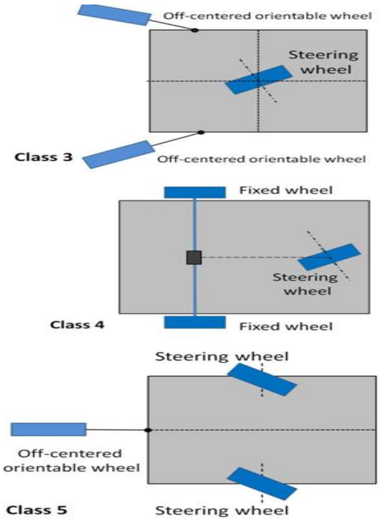


Fig. 3. Kinematic class 3, holonomic WMR with 1 centered orientable wheel and two off-centered orientable wheels. Kinematic class 4, two fixed wheels on the same axle and a steerable wheel. Class 5 shows a kinematic model with two orientable wheels and one onmidirectional castor wheel.

Class 4 is widely used because represent the car-like a robot case, see Figure 4. It is noted that nonholonomic constraints arise due to the fact that a car cannot move sideways because the back wheels may produce wheel slippage instead of pure rolling. The vehicle is considered as a rigid body. WMR configuration is denoted by (x, y, θ) coordinates that are referenced to the center of rear axle as is shown in Figure 4. It is noted that L represents the distance between front and rear axles. If the steering angle has a fixed value, φ see Figure 4, then circular motion with rotation radius ρ is produced. The radius is determined by intersection of rear axle and front wheel axis as is shown in Figure 4. If we denote by *v* the velocity of the vehicle then car motions can be described by the following functions:

$$\begin{aligned} \dot{x} &= f_1(x, y, \theta, v, \varphi) \\ \dot{y} &= f_2(x, y, \theta, v, \varphi) \qquad (1) \\ \dot{\theta} &= f_3(x, y, \theta, v, \varphi) \end{aligned}$$

Where the dot denotes the differentials referred to time. When small intervals of time are analyzed (Δt→0), see Figure 4, the rear wheels direction is considered as the point towards the vehicle is moving and the following approximation is obtained:

$$\tan\theta = \frac{\Delta y}{\Delta x} = \frac{\sin\Delta\theta}{\cos\Delta\theta} \qquad (2)$$

Considering $\Delta x = \cos\Delta\theta$ and $\Delta y = \sin\Delta\theta$, we scale the equations using the vehicle displacement, and we can compute the differential equations referred to the time. Then, the translation equations are obtained:

$$\dot{x} = v\cos\Delta\theta \qquad (3)$$

$$\dot{y} = v\sin\Delta\theta \qquad (4)$$

Heading orientation is also analyzed in order to derive an equation for θ. D denotes the vehicle displacement and ρ the radius of a circle centered on rear axle center. When the steering angle remains fixed then a circular trajectory is expected. Considering $\Delta D = \rho\Delta\theta$ as an infinitesimal approach and $\rho = L/\tan\varphi$ from trigonometry, the following equation is obtained:

$$\dot{\theta} = \frac{v}{L}\tan\varphi \qquad (5)$$

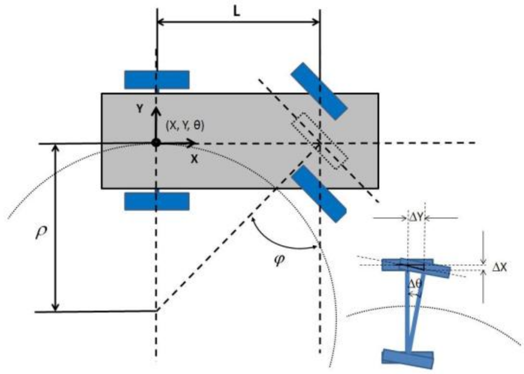


Fig. 4. The nonholonomic car-like robot case has three configuration coordinates (x, y, θ) and two control inputs. Down right side shows the small increments considered.

It is noted that L depicts the distance between front wheels and rear wheels axles and the speed *v* is the first time derivative of D. Equations (3) and (4) represent the Cartesian velocities that are obtained as function of the translational speed and the heading orientation. Equation (5) depicts the circular motion speed. Once the car-like robot case is introduced, Class 2 vehicles are analyzed. This kind of vehicles are commonly known as differential driven vehicles because have two main wheels driven by its own motor and a third omnidirectional or castor wheel that gives stability to vehicle. Figure 5-(a) shows the WMR scheme where *L'* denotes the distance between the two wheels and *R* depicts the wheel radius. The velocity vector $(v_R, v_L) = (R_R\omega_R, R_L\omega_L)$ is used to express the right and left wheel speeds. If both speeds are greater than zero then the WMR moves forward. It is the pure translation case as it is shown in Figure 5-(b). If the velocities of both wheels are equal but have different sign then pure rotation is produced because the wheels are turning in opposite directions. If $v_L = -v_R$ then clockwise rotation is done, see Figure 5-(c). Otherwise, anticlockwise rotation is achieved. If both wheels have the same sense, angular speed, and radius then pure translation is produced and the travelled distance can be obtained by multiplying the translational speed by the time. If both wheels have different speed but they rotate in the same sense then the heading orientation is modified and a turning action is done. From previous considerations the kinematic model for pure translation case, see Figure 5-(d), can be stated as follows:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

$$\dot{x} = \frac{1}{2}\,(v_R + v_L)\cos\theta \qquad (6)$$

$$\dot{y} = \frac{1}{2}\,(v_R + v_L)\sin\theta \qquad (7)$$

$$\dot{\theta} = \frac{1}{L'}\,(v_R - v_L) \qquad (8)$$

Equations (6) and (7) represent the translational part; it has *cosθ* and *sinθ* expressions as in the car-like a robot case. The WMR moves towards the heading direction. It is noted that translational velocity is computed as average of right and left wheel speeds. Considering Class 2 vehicles in the pure translation case, we outline the similarity with the kinematic models obtained for Class 4 vehicles. Both kinematic models have similar equations when translational terms are analyzed, equations (3) and (4) and equations (6) and (7). Circular motion speeds, $\dot{\theta}$, are equated as the product of the radius of rotation by the rotational speed. Class 4 and Class 2 vehicles have radius of rotations related with their physical dimensions, and angular speeds related with the steering angle φ or the differences of left and right wheel velocities, see equations (5) and (8). Due to the similarities, differential driven WMR can be used to analyze the car-like a robot case. In our work a differential driven robot that uses incremental encoders in order to obtain position and orientation coordinates is used. Figure 5-d shows two consecutive configurations that can be analyzed using discrete time representation for *k* and *k+1* time instants. The robot positioning can be described as a function of the radius of the left and right wheels (R_L, R_R), and the angular incremental position (θ_L, θ_R), with *L'* being the distance between both wheels and *ΔS* the incremental displacement of the WMR. The position and angular incremental displacements can be expressed:

$$\Delta S = \frac{R_R \Delta\theta_R + R_L \Delta\theta_L}{2} \quad \Delta\theta = \frac{R_R \Delta\theta_R - R_L \Delta\theta_L}{L'} \qquad (9)$$

Using incremental discrete time representation, equations (6), (7), and (8) can be expressed as follows:

$$\begin{aligned} x(k+1) &= x(k) + \Delta S\cos(\theta(k) + \Delta\theta) \\ y(k+1) &= y(k) + \Delta S\sin(\theta(k) + \Delta\theta) \\ \theta(k+1) &= \theta(k) + \Delta\theta \end{aligned} \qquad (10)$$

In this work, we compute the incremental position of the WMR using the odometer system through the available encoder information from (9) and (10).

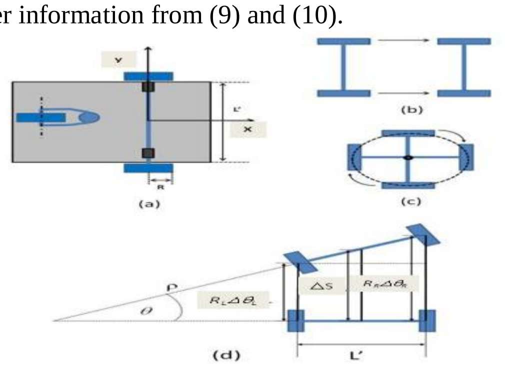


Fig 5. (a) Differential driven vehicle. (b) Pure translation case. (c) Pure rotation case. (d) Robot configuration as function of the rotation of each wheel. Usually, R_R and R_L have the same value denoted by *R*.

III. LOCAL PATH-PLANNING AND MOTION CONTROL

Local path-planning and motion control are introduced by using kinematic models that represent nonholonomic vehicles. With the aim of representing the forward movement of the car-like a robot case, Class 2 kinematic models are constrained to have only advancing velocities at both wheels. The kinematic model is an important constraint in order to perform local path-planning. The desired path to be followed can be composed of a sequence of straight lines and circle segments [12]. Thus, instead of regulating the robot about one point, the sequence of points to track is solved moving the WMR from point to point in the state space by using feedback combined with some path planning [13]. The importance of the results arises in the fact that linear control is suitable by using motion discontinuities as a path-tracking function. Within this scope, PID solutions are widely adopted as low level DC (direct current) motor speed controllers [6]. Path-following achievement is accomplished by developing high level strategies for motion planning [14]-[15]. In this work, it is proposed the use of experimental dynamic models as a suitable control solution that uses transfer functions relating the controlled outputs with the control inputs. The knowledge of a set of dynamic models can be used either for designing classical low level PID speed controllers or new methodologies as MPC (model predictive control). In this section experimental modeling is introduced by using the available differential driven WMR [16].

C. Path-Planning and Motion Control Using Kinematics Models

Dubins car case is proposed as a way of constraining the set of possible speeds [17]:

$$0 \le v_v \le k; \varphi_{\min} \le \varphi \le \varphi_{\max} \qquad (11)$$

The translational speed is constrained to a maximal positive value so just forward movements are allowed. Steering angle, *φ*, is constrained by maximum and minimum turning angles, $-\pi/2 < \varphi_{min}$ and $\varphi_{max} < \pi/2$. Using the maximum steering angle, the minimum turning radius, $\rho_{min} = L/\tan\varphi_{max}$, is obtained. Motion control using kinematic models has also to include velocity and acceleration constraints. In this way, the path-planning problem consists of finding safe goal approaching paths that use the free of collision bi-dimensional space. Denoting by *U* the set of possible inputs that can produce allowable trajectories within and interval of time (0, +∞), then the reachable set consists of (x, y) coordinates that belongs to any feasible trajectory obtained by integrating the set of possible *U* inputs [3]. Normally the time is limited; Figure 6-(a) shows an example

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

of time-limited reachable set. Motion control is done by searching for goal approaching and obstacle avoidance trajectories within the reachable space. Figure 6-(b) depicts, using a black dashed line, a right turning trajectory that can be done by overlapping reachable time-limited sets. Thus, path-planning can be achieved by selecting straight lines and curves belonging to consecutive reachable time limited sets.

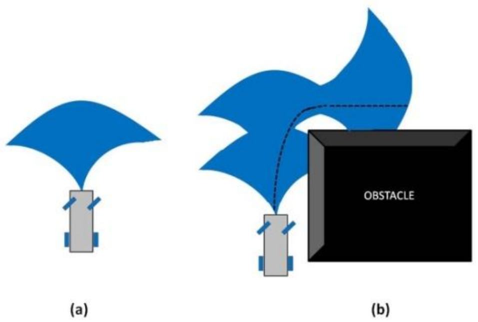


Fig. 6. (a) Time-limited reachable set for a car-like robot vehicle. (b) Partial reachable tree for right turning case when a right obstacle is considered.

D. System Identification and Dynamics Models

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 [18] and [19]. The parameter estimation is done by using a PRBS (Pseudo Random Binary Signal) 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 speed commands that correspond to low, medium, and high speeds. 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 linear regression, the estimated output can be written as:

$$\hat{y} = \theta\varphi \qquad (12)$$

with $\hat{y}$ being the estimated output vector, θ the vector of estimated parameters and φ the vector of measured input and output variables. The nonholonomic system dealt with in this work is considered initially to be a MIMO (multiple input multiple output) system, as shown in Figure 7, 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. By using the coupled system structure, the transfer function of the robot can be expressed as follows:

$$\begin{pmatrix} V_R \\ V_L \end{pmatrix} = \begin{pmatrix} G_{RR} & G_{LR} \\ G_{RL} & G_{LL} \end{pmatrix} \begin{pmatrix} U_R \\ U_L \end{pmatrix} \qquad (13)$$

Where V_R and V_L represent the velocities of right and left wheels, and U_R and U_L the corresponding speed commands, respectively. G_{LL} is the transfer function of the left wheel speed related to the left wheel speed commands, G_{RR} is the transfer function of the right wheel speed related to the right wheel speed commands, G_{RL} is the transfer function of left wheel speed related to right wheel speed commands, and G_{LR} is the transfer function of right wheel speed related to left wheel speed commands. G_{LR} and G_{RL} are coupled terms that denote the influence of left wheel speed commands on right wheel speeds and the influence of right wheel speed commands on left wheel speeds. Output speeds are computed by using the available on-robot encoders placed on each wheel. In order to know the dynamics of system, the matrix of transfer function should be identified. In this way, left and right wheel speed responses to PRBS input signals are analyzed. The system is identified by using the identification toolbox “ident” of MATLAB for the second order models which represent the dynamics of a DC motor controlled robot. The coupling effects should be studied as a way of obtaining a reduced-order dynamic model. When the transfer functions are analyzed, we observe that the dynamics of two DC motors are different and the steady gains of coupling terms are relatively small less than 20%. Thus, it is reasonable to neglect the coupling dynamics so as to obtain a simplified model.

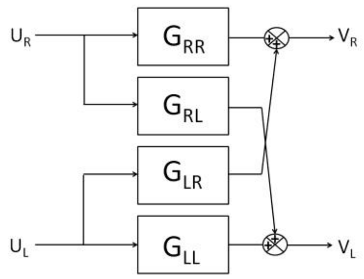


Fig. 7. The MIMO system structure

Experimental results confirm 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 1 shows the first order transfer functions obtained.

Table 1. The reduced WMR models

Linear Transfer Function	High velocities (0.5m/s)	Medium velocities (0.37m/s)	Low velocities (0.25m/s)
G_{DD}	$\frac{0.95}{0.42s+1}$	$\frac{0.92}{0.41s+1}$	$\frac{0.82}{0.46s+1}$
G_{EE}	$\frac{0.91}{0.24s+1}$	$\frac{0.92}{0.27s+1}$	$\frac{0.96}{0.33s+1}$

IV. MODEL PREDICTIVE CONTROL FOR LOCAL PATH-PLANNING AND MOTION CONTROL

Model predictive control, MPC, has many interesting aspects for its application to mobile robot control. It is one of

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

the most effective advanced control techniques, as compared to the standard PID control, that has made a significant impact to the industrial process control [8]. 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.
- Receding horizon idea is used in such a way that 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 use of mobile robot kinematics to predict future system outputs were proposed in most of the different research developed [20]-[21]. The use of kinematics should include velocity and acceleration constraints to prevent WMR of unfeasible path-tracking objectives. MPC applicability for vehicle guidance has been mainly addressed for path-tracking using different on-field fixed trajectories based on kinematics models. However, when dynamic environments or obstacle avoidance policies are considered, the navigation path planning should be constrained to the robot neighborhood where reactive behaviors are expected [14] and [22]. Due to the unknown environment uncertainties, short prediction horizons are proposed [9]. 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. 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. LMPC and contractive constraint formulations as well as the algorithms and simulations implemented are introduced in the next subsections.

A. 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, this work uses on-robot an encoder based odometer system for computing the WMR location. LMPC is proposed in order to use the available local 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:

$$J(n,m)=\min_{\left\{U(k+i|k)\right\}_{i=0}^{i=m-1}}\left\{\begin{array}{l}\left[X(k+n|k)-X_{ld}\right]^T P\left[X(k+n|k)-X_{ld}\right]\\+\sum_{i=1}^{n-1}\left[X(k+i|k)-\overline{X_{ld}X_{l0}}\right]^T Q\left[X(k+i|k)-\overline{X_{ld}X_{l0}}\right]\\+\sum_{i=1}^{n-1}\left[\theta(k+i|k)-\theta_{ld}\right]^T R\left[\theta(k+i|k)-\theta_{ld}\right]\\+\sum_{i=0}^{m-1}U^T(k+i|k)SU(k+i|k)\end{array}\right\} \quad (14)$$

The first term of (14) 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 term can be considered as an orientation term and is related to the distance between the predicted robot positions and the path segment given by a straight line between the initial path point coordinates $X_{l0}=(x_{l0}, y_{l0})$ and the desired local position, X_{ld}. This line orientation is denoted by θ_{ld} and denotes a desired orientation towards the local objective. $X(k+i/k)$ and $\theta(k+i/k)$ $(i=1,\ldots n-1)$ represent 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. System constraints, such as the convergence towards the desired configuration and the fact that the signal increment is kept fixed within the control horizon, are considered:

$$\left\{\begin{array}{ll}G_0<|U(k)|\le G_1 & \alpha\in(0,1]\\ |X(K+n/k)-X_{ld}|\le\alpha|X(k)-X_{ld}| & \\ \text{or } |\theta(k+n/k)-\theta_{ld}|\le\alpha|\theta(k)-\theta_{ld}| & \end{array}\right\} \quad (15)$$

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 [23], which result in the convergence of coordinates or orientation to the objective, and should be accomplished at each control step.

B. Algorithms and Simulated Results

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

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

The use of interior point methods can solve the above problem [24]-[25]. Gradient descent method and complete

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

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 [8]. Thus, the set of available input is reduced to one value, as it is shown in Figure 7.

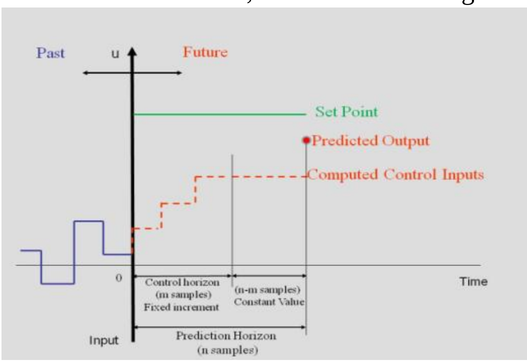
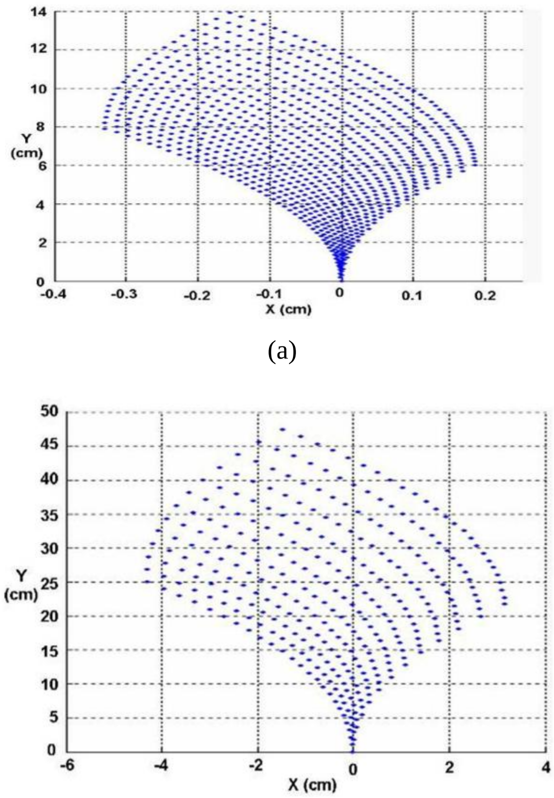


Fig. 7. LMPC strategy with fixed increment of the input during the control horizon and constant value for the remaining time

Once LMPC strategy is stated, coordinate prediction can be done by selecting control and prediction horizons. Figure 8 depicts the possible coordinates available for prediction when the horizons are set to (*n=5, m=3*) and (*n=10, m=5*). The sampling period of time is set to 0.1s. The obtained results show the predicted coordinates after 0.5s or 1s. In fact, it is obtained two different time-limited reachable set of coordinates that can be used to perform local path planning by computing the LMPC law cost given by (14). Gradient descent method and complete input search methods perform accurate path-tracking. Simulated results show the discrepancy between the two methods in which the subinterval gradient descent method usually does not give the optimal solution [9]. In this way, local minimum can be obtained for gradient descent methods. Due to the results obtained from simulations, complete input search is selected for the on-robot experiences developed in this work. The evaluation of the LMPC performance is made by using different parametric values in the proposed cost function (14). In this way, when only the desired coordinates are considered, (*P=1, Q=0, R=0, S=0*), the path-tracking is done with the inputs that can minimize the cost function by shifting the robot position to the left, see Figure 8. 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 path to be tracked 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. Obtained results show the need of *R* parameter when meaningful orientation errors are produced.



(b)

Fig. 8. (a) The horizon of available coordinates (m=3, n=5). (b) Predicted coordinates from speed zero, n=10, m=5.

Larger horizons of prediction 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, see Figure 8. For these reasons, a short horizon strategy (*n=5, m=3*) is proposed for implementing experimental results. 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 implemented using C language running in a Intel® core 2.7 GHz PC. Real on-robot algorithm time performance is also compared for different prediction horizons by using the embedded computer based on a VIA C3 EGBA 733/800 MHz CPU running under LINUX Debian OS and additional hardware system. Table 2 shows the LMPC processing time for different horizons of prediction when complete optimal values search or the gradient descent method are used.

Table 2. LMPC processing times

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

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

V. PATH-TRACKING EXPERIMENTAL RESULTS

In this section, path-tracking problem and the cost function parameter weights are analyzed. 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. Analysis of robustness was carried out experimentally by considering reduced model parameters. Parametric uncertainty studies revealed a robust behavior for model parameter variations of 20% of time-constant, and 11% of gain.

A. Path-Tracking Approach Using LMPC Methods

Path-tracking performance is improved by the adequate choice of a cost function that is derived from (14) 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 (14).
- 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 (14).
- 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 (14).
- The last parameter refers to changes allowed to the input signal. It corresponds with the parameter "*S*" of the LMPC cost function given by (14).

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. In this study the cost function parameters are P=1, Q=1, R=1, S=0. Control law tuning is carried out as a function of the kind of path being tracked (straight, wide left turn, slight left turn, wide right turn, and slight right turn). A trajectory is considered straight when the orientation error is less than 5º. A slight turn should be commanded when the left or right heading error is equal to or larger than 5º and less than 25º. Finally a wide turn should be commanded when the orientation error is larger than or equal to 25º. Factor tuning of the APD (approaching-point distance), OD (orientation deviation) and TDD (trajectory-deviation distance) parameters is done by considering experimental data [26]. High and low values are set by slightly modifying the values corresponding to sensitive parameters and verifying their validity for a set of trajectories. Heuristic experiments for selecting high and low values of APD, OD, and TDD have depicted a suitable interval up to 50%. 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 trajectories are tested. From each trajectory parameters such as time (T), RMS trajectory error (TE), travelled distance (TD) and averaged speed (AS) are measured. Path-tracking performance is analyzed using the means of the different factor weights. Due to the fact that experiments reveal similar results at each trial, for each combination of factors, different runs were tried. The averaged value of the three runs makes a statistical analysis of each factor combination possible. From these standard deviations, the importance of the factor effects can be determined using a rough rule that considers the effects when the value differences are close to or greater than two or three times their standard deviations. A detailed description is shown in [26]. Using previously obtained statistical data, an analysis of the main and lateral effects of APD (P), OD (Q) and TDD (R) is obtained.

B. Analysis of the Experimental Results

Once factorial analysis is carried out, this subsection compares path-tracking performance by using different control strategies. Previous results are used to perform function factor adjustments with the aim of securing better control laws. Quantitative results are computed, the analysis is performed by taking into consideration the different trajectories. Considering the information obtained, it is proposed a control law that is a function of the path to be followed because this flexibility improves the results. Table 3 shows the different factors selected as a function of the trajectory being tracked; L depicts a selection of a low value and H a selection of a high value.

	Trajectories				
factors	straight line	wide left turn	wide right turn	slight left turn	slight right turn
APD	H	H	H	H	L
OD	L	H	H	H	H
TDD	L	L	L	L	L

Table 3. Flexible factor values as trajectory function

The control results obtained previously are tested by commanding two different trajectories. These have to include the five different basic ones studied previously. In order to achieve this, a set of twelve points with a shape similar to a dodecagon is proposed. The path-tracking is tested in clockwise and anticlockwise directions in order to test both right and left turns. The experiments are carried out for fixed

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

and flexible LMPC controllers, with all the parameters of fixed control laws having high values. In order to analyze trajectory accuracy the sequence of points is commanded each time the previous point is reached. It should be noted that while this produces a reduction in speed each time the WMR is close to the final commanded point; better path-tracking accuracy is achieved. The measured parameters are time (T), RMS trajectory error (TE), travelled distance (TD) and averaged speed (AS). Analysis of LMPC for flexible and fixed control laws reveals significant effects in time reduction when the travelled distance and averaged speed are increased. Flexible control laws improve the performance of fixed control laws. Figure 9 (a) shows a sample of the results obtained for fixed and flexible clockwise dodecagon analysis. An anticlockwise dodecagon is proposed for analyzing left turn trajectories. Figure 9 (b) shows a sample of the path-tracking results obtained for fixed and flexible anticlockwise dodecagon analysis. Results show that flexible factor strategy improves a 12% and a 22% the total time, T, of the fixed factor strategy for clockwise and anticlockwise trajectories. Analysis of RMS TE parameter depicts that both flexible and fixed control laws perform accurate path following without remarkable differences. Clockwise analysis for TD and AS parameters reveal that flexible factor strategy increases a 1% the traveled distance while the averaged speed is increased a 15%. Anticlockwise analysis for TD and AS parameters show that flexible factor strategy decreases a 1.7% the travelled distance while the averaged speed is increased a 25%.

VI. CONCLUSION

The methodology used for performing nonholonomic motion control is shown. 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. Control system performance can be improved by considering different factor weights as a function of path to be followed. From experimental results, it is obtained that total time is reduced and averaged speed is increased for flexible cost functions. In this work, experiences are conducted by considering trajectories that include basic motions consisting of go straight, less turning, and wide turning. Furthermore, clockwise and anticlockwise directions are experienced in order to test both right and left turns.

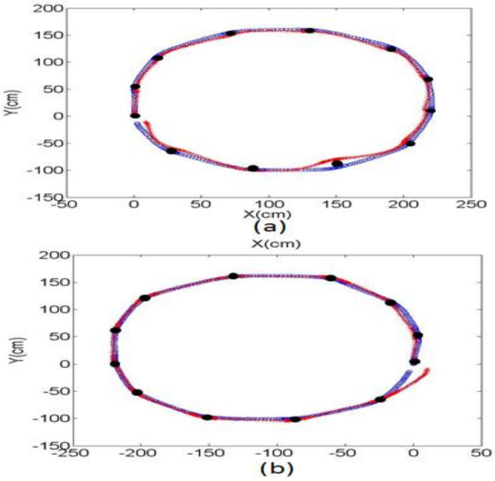


Fig. 9. Clockwise and anticlockwise experimental samples, the trajectory points are represented by twelve black dots. (a) Clockwise dodecagon results for flexible LMPC (empty circles) and fixed LMPC (crosses). (b) Anticlockwise dodecagon results for flexible (empty circles) and fixed LMPC (crosses).

REFERENCES

[1] R. D. Schraft, G. Schmierer, “Serviceroboter – Produkte Szenarien, Visionen” Springer-Verlag, Berlin, 1998.

[2] J. Dixon, O. Henlich, “Mobile Robots Navigation”, Imperial College, London, Available: http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/. Accessed: 2014 September 11.

[3] M. Lavalle, “Planning Algorithms,” Cambridge University Press, Available: http://planning.cs.uiuc.edu. Accessed 2012 May 8.

[4] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun, “Principles of Robot Motion: Theory, Algorithms, and Implementation,” MIT Press, April 2005.

[5] K. K. Tan, T. H. Lee and S. Huang, “Precision motion control: Design and implementation”, 2nd ed., London, Springer, 2008.

[6] K. Ogata, “Modern Control Engineering”, fifth edition, New Jersey, USA, Prentice Hall, 2009.

[7] A. M. Bloch, “Nonholonomic Mechanics and Control”, vol. 24, Interdisciplinary Applied Mathematics, New York, USA, Publisher Springer, 2003.

[8] Maciejowski, J.M. (2002). Predictive Control with Constraints, Ed. Prentice Hall, Essex (England)

[9] L. Pacheco, N. Luo, “Mobile robot local trajectory tracking with dynamic model predictive control techniques”, International Journal of Innovative Computing, Information and Control, Vol.7, No.6, pp. 3457-3483, June 2011.

[10] G. Campion, G., Bastin, B. D’Andréa-Novel, “Structural Properties and Classification of Kinematic and Dynamic models of Wheeled Mobile Robots,” IEEE Transaction on Robotics and Automation, vol.12, no.1, pp. 47-62, 1996.

[11] B. Siciliano, O. Khatib, “Springer Handbook of robotics, chapter 17 wheeled robots”, Siciliano Khatib editors, Springer-Verlag, Heidelberg, Germany, 2008.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 6, December 2014

[12] J. A. Reeds, L. A., Shepp, "Optimal paths for a car that goes forwards and backward" Pacific Journal of Mathematics, vol. 145, 1990.

[13] Astolfi, A. (1996) Discontinuous Control of Nonholonomic Systems, Systems and Control Letters, No. 27, pp. 37-45.

[14] P. Ögren, N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," IEEE Transaction on Robotics, Vol. 21, No. 2, pp. 188-195, April 2005.

[15] G. Capi, S. Kaneko, "Evolution of Neural Controllers in Real Mobile Robots for Task Switching Behaviors," International Journal of Innovative Computing, Information and Control, vol.5, no.11 (A), pp.4017-4024, 2009.

[16] L. Pacheco, N. Luo, I. Ferrer, X. Cufí, "Interdisciplinary Knowledge Integration through an Applied Mobile Robotics Course", Int. Journal of Engineering Education, Vol. 25, No. 4, pp. 830-840, 2009.

[17] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". American Journal of Mathematics, 79:497–516, 1957.

[18] J. Norton, "An Introduction to Identification," Academic Press, London and New York, 1986.

[19] L. Ljung, "Issues in System Identification," IEEE Control Systems Magazine, Vol. 11, No.1, and pp: 25-29, January 1991.

[20] 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, pp. 200-214, 2005.

[21] F. Kühne, W. F. Lages, J. M. Gomes da Silva Jr., "Model predictive control of a mobile robot using input-output linearization," Proceedings of Mechatronics and Robotics, July 2005

[22] 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.

[23] J. Wan, J., "Computational reliable approaches of contractive MPC for discrete-time systems," PhD Thesis, University of Girona, 2007.

[24] I. E. Nesterov, A. Nemirovskii, Y. Nesterov, Y., "Interior Point Polynomial Methods in Convex Programming," Siam Studies in Applied Mathematics, Vol 13, 1994.

[25] S. Boyd, L. Vandenberghe, L., "Convex Optimization," Cambridge University Press, New York (USA), 2004.

[26] G. E. P Box, J. S. Hunter, W. G. Hunter, "Statistics for Experimenters," Ed. John Wiley & Sons, New Jersey (USA), 2005.

AUTHOR'S PROFILE

Lluís Pacheco. Bachelor in Electronic Engineering from "Universitat Politecnica de Catalunya" of Girona in 1990, and graduate in Physics Sciences from "Universidad Nacional de Educación a Distancia" of Madrid in 1999, Doctor in Communication Sciences from University of Girona in 2009. He has been a faculty member and a Professor assistant in Computer Sciences in the University of Girona since 1992. His main interests are in the field of electronic engineering and include: hardware dessign, sensor systems, robotics, control systems and computer vision.

Ningsu Luo. Bachelor in Electronic Engineering and Master in Control Engineering from University of Science and Technology of China in 1982 and 1985, Doctor in Control Engineering from Southeast University in 1990 and Doctor in Physics Sciences from University of the Basque Country in 1994, respectively. He was a Visiting Professor in Technical University of Catalonia from 1994 to 1996 and in University of the Basque Country from 1996 to 1998. He has been a faculty member and a Professor in Systems Engineering and Control in the University of Girona since 1998. His main interests are in the field of analysis and design of controllers for systems with complex dynamics such as parametric uncertainties, unknown disturbances, time-delays, nonlinearities, dynamic couplings, actuator dynamics, etc, with application to networked control systems, mechatronic systems and mobile robotics, and vibration control of civil engineering structures and automotive and aeronautic systems and off-shore wind farms.