KKU Res. J. 2012; 17(5):811-837
http : //resjournal.kku.ac.th

Motion Control of a Wheeled Mobile Robot Using Model Predictive Control: A Survey

Kiattisin Kanjanawaniskul *

Department of Mechatronics Engineering, Faculty of Engineering, Mahasarakham University
**Correspondent author: kiattisin_k@hotmail.com*

Abstract

Model predictive control (MPC) has been one of the most promising control strategies in industrial processes for decades. Due to its remarkable advantages, it has been extended to many areas of robotic research, especially motion control. Therefore, the goal of this paper is to review motion control of wheeled mobile robots (WMRs) using MPC. Principles as well as key issues in real-time implementations are first addressed. We then update the current literature of MPC for motion control. We also classify publications by using three criteria, i.e., MPC models, robot kinematic models, and basic motion tasks. MPC models categorized here include nonlinear MPC, linear MPC, neural network MPC, and generalized predictive control (GPC), while robot kinematic models we focus on consist of unicycle-type vehicles, car-like vehicles, and omnidirectional vehicles. Basic motion tasks, in general, are classified into three groups, i.e., trajectory tracking, path following, and point stabilization. To show that MPC strategies are capable of real-time implementations, some experimental scenarios from our previous work are given. We also conclude by identifying some future research directions.

Keywords: wheeled mobile robots, model predictive control, motion control, trajectory tracking, path following

1. Introduction

Robots have become increasingly more important in human daily lives in the last decade and apparently the number of robots will increase and get more involved in the human society in the near future (1). Real-world applications employing robots have already shown the effectiveness and usefulness of robots, especially in industry. However, many unsolved problems still exist in many robotic research areas.

In general, basic tasks in robotic research are mapping, controlling, planning and localizing (2). Usually, a robot creates a map of the environment. Using this map, it can localize itself. Then it plans the reference if it wants to travel. The controller is designed to move it to the target. However, accomplishing those missions is not an easy task. In this paper, we address only the problem of motion control of wheeled mobile robots (WMRs). Motion control of WMRs has been, and still is, the subject of numerous research studies. Many nonlinear

techniques have been proposed in the literature, e.g., dynamic feedback linearization (3), sliding mode control (4), backstepping techniques (5), etc., to name some. For this paper, the goal is to present a survey of model predictive control (MPC) applied to WMRs. Although MPC is not a new control method, works dealing with MPC of WMRs are sparse.

Model predictive control (MPC), also referred to as receding horizon control (RHC) and moving horizon optimal control, has been widely adopted in process control industry for decades because control objectives and operating constraints can be integrated explicitly in the optimization problem that is solved at each instant. Many successful MPC applications have been reported in the last three decades (6, 7). Although it is traditionally applied to plants with dynamics slow enough to permit computations between samples, recently, due to the combination of advanced research results and the advent of faster computers, it has become possible to extend the implementation of MPC design to systems governed by faster dynamics, including WMRs.

The rest of the paper is structured as follows: in Section 2, principles and relevant literature of MPC are introduced. Major practical issues of MPC are discussed in Section 3. The current literature of MPC for motion control is updated and classified into three basic motion tasks, i.e., trajectory tracking, path following, and point stabilization, in Section 4. Section 5 illustrates experimental scenarios from our previous work, where MPC techniques were implemented in real-time applications. We also suggest some future research directions in Section 6 and finally, we close our review with some conclusions in Section 7.

2. Principles and Formulation

The conceptual structure of MPC is illustrated in Figure 1. As its name suggests, an MPC algorithm employs an explicit model of the plant to be controlled which is used to predict the future output behavior. This prediction capability allows computing a sequence of manipulated variable adjustments in order to solve optimal control problems in real time, where the future behavior of a plant is optimized over a future horizon, possibly subject to constraints on the manipulated inputs and outputs (8-12). The result of the optimization is applied according to a receding horizon philosophy: At time t only the first input of the optimal command sequence is actually applied to the plant. The remaining optimal inputs are discarded, and a new optimal control problem is solved at time $t+\delta$, where d is the sampling period. As new measurements are collected from the plant at each time t, the receding horizon mechanism provides the controller with the desired feedback characteristics.

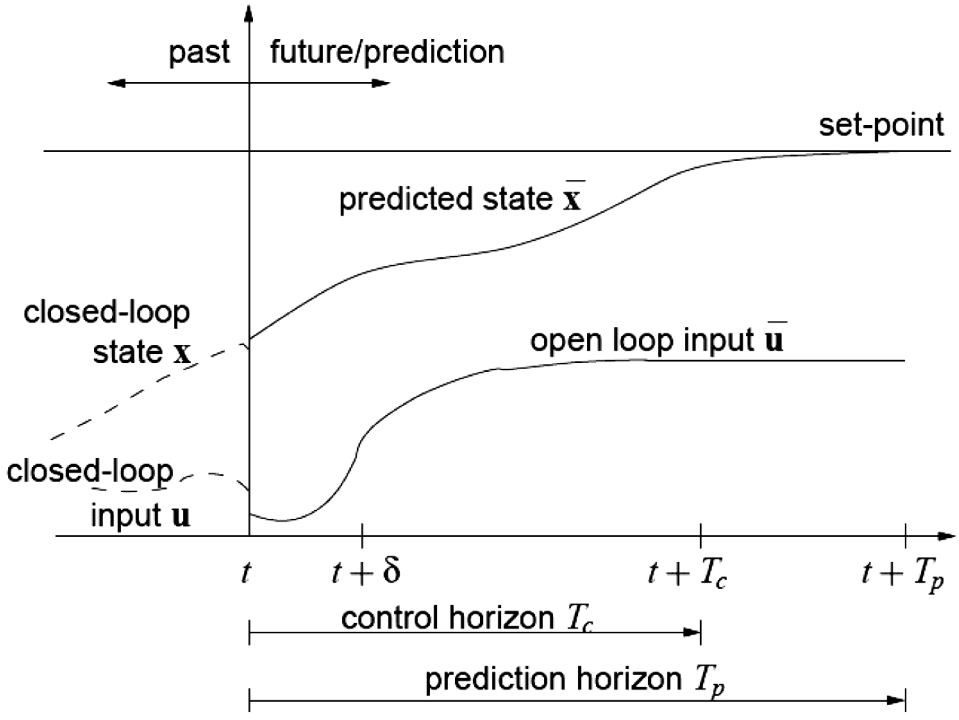


Figure 1. Principle of model predictive control (8).

When the model is linear, the optimization problem is a convex quadratic programming (QP) problem if the performance index is expressed through the l_2-norm, or a linear programming problem if expressed through the l_1/l_∞-norm. It has a unique, global minimum which can be quickly and reliably computed numerically in a constrained case. In an unconstrained case the solution can be computed analytically as a linear feedback control law. If a process model is in the form of a discrete transfer function or equivalently a difference equation (e.g., an ARX-type model), generalized predictive control (GPC) (13, 14) can be derived. By now, important issues of linear MPC theory are well addressed (6, 15). However, many systems are inherently nonlinear and linear MPC is inadequate for highly nonlinear systems. Therefore, nonlinear models must be used (8). However, the optimization problem is certainly not linear or quadratic, it is generally a nonconvex when the model is nonlinear. For such problems, there are no sufficiently fast and reliable numerical optimization procedures. Therefore, many attempts have been made to construct simplified (and generally suboptimal) nonlinear MPC algorithms avoiding full online nonlinear optimization. One possibility is to use model linearization or multiple linear models, in which only a QP problem is solved online (6). There are also many designs of predictive algorithms based on nonlinear optimization and also using neural network techniques.

A nonlinear system is normally described by the following nonlinear differential equation:

$$\dot{\underline{x}}(t) = f\left(\underline{x}(t), \underline{u}(t)\right) \tag{1}$$

subject to: $\underline{x}(t) \in X, \quad \underline{u}(t) \in U, \quad \forall t \geq 0$

where $\underline{x}(t) \in \Re^n$, $\underline{u}(t) \in \Re^m$ are the n dimensional state vector and the m dimensional input vector of the system, respectively. $X \subseteq \Re^n$ and $U \subseteq \Re^m$

denote the set of feasible states and inputs of the system, respectively. In nonlinear MPC (NMPC), the input applied to the system is usually given by the solution of the following finite horizon open-loop optimal control problem, which is solved at every sampling instant:

$$\min_{\bar{u}(\cdot)} \int_t^{t+T_p} F(\bar{x}(\tau),\bar{u}(\tau))d\tau + V(\bar{x}(t+T_p)) \quad (2)$$

$$\text{subject to:} \quad \begin{aligned} &\bar{x}(t) = f(\bar{x}(\tau),\bar{u}(\tau)) \\ &\bar{u}(\tau) \in U \quad \forall \tau \in [t, t+T_c] \\ &\bar{x}(\tau) \in X \quad \forall \tau \in [t, t+T_p] \\ &\bar{x}(t+T_p) \in \Omega \end{aligned} \quad (3)$$

where $F(\bar{x},\bar{u}) = \bar{x}^T Q \bar{x} + \bar{u}^T R \bar{u}$. The bar denotes an internal controller variable. T_p represents the length of the prediction horizon or output horizon, and T_c denotes the length of the control horizon or input horizon ($T_c \le T_p$). When $T_p = \infty$, we refer to this as the infinite horizon problem, and similarly, when T_p is finite, as a finite horizon problem. $V(\bar{x}(t+T_p))$ is the terminal penalty and Ω is the terminal region. The deviation from the desired values is weighted by the positive definite matrices Q and R.

A standard MPC scheme works as follows (8):

1) Obtain measurements/estimates of the states of the system at time instant *t*.

2) Calculate an optimal input series $\bar{u}(t, t+T_c)$ by minimizing the desired cost function over the predictive horizon in the future using the system model, the generated predictive state sequence $\bar{x}(t, t+T_p)$ from $\bar{u}(t, t+T_c)$ should contain the terminal state $\bar{x}(t+T_p)$ that falls in the required terminal state region.

3) Implement the first part of the optimal input series $u = \bar{u}$ until the new measurement/estimates of the states are available.

4) Continue with 1) at the next time instant $t = t + \delta$.

As mentioned before that MPC has been one of the most promising control strategies. The reason is due to the following remarkable advantages of MPC over conventional control schemes:

- its ability to incorporate generic models, linear and nonlinear, and constraints in the optimal control problem;
- its formulation that can be extended to handle multiple-variable, nonlinear, time-varying plants in a single control formulation;
- its ability to redefine cost functions and constraints as needed to reflect changes in the system and/or the environment;
- its ability to use future values of references when they are available, allowing MPC to improve performance in navigation;
- its ability to tune parameters that are directly related to a cost function.

3. Practical Issues of Nonlinear MPC

In this section, we review some practical issues, i.e., feasibility, stability and real-time optimization. They are some of the most important aspects in NMPC implementations.

3.1 Feasibility

Typically one assumes feasibility at time t = 0 and chooses the cost function and the stability constraints such that feasibility is preserved at the following time steps. This can be done, for example, by ensuring that the shifted optimal sequence $\{\bar{u}(t+\delta),...,\bar{u}(t+T_p),0\}$ is feasible at time t+δ. Furthermore, typically the constraints in [3] which involve state components are treated as soft constraints, for instance by adding the slack variable ε, while input constraints in [3] are maintained as hard because they come from actuator saturation and/or physical, safety or economical requirements. Relaxing the state constraints removes the feasibility problem. Keeping them tight does not make sense from a practical point of view because of the presence of noise, disturbances, and numerical errors.

3.2 Stability

The next major concern in the use of MPC is that whether such an open-loop control can guarantee system stability. It is shown that an infinite predictive control horizon can guarantee stability of a system, but the infinite predictive horizon may not be feasible for a nonlinear system in practice (8). Mayne et al. (10) have presented the essential principles for the stability of MPC of constrained dynamical systems. Different approaches to attain closed-loop stability using finite horizon lengths exist. We review some of the popular techniques proposed in the literature to enforce stability. For reasons of a simple presentation, no detailed proofs are given.

Most recent MPC controllers use a terminal cost and enforce a terminal constraint set. The following stability theorem given in (8) provides a way to find the suitable terminal penalty and constraints.

Theorem 1: Suppose

1) $U \subseteq \Re^m$ is compact, $X \subseteq \Re^n$ is connected and the origin is contained in the interior of $U \times X$.

2) The vector field $f : \Re^n \times \Re^m \to \Re^n$ is continuous in $\underline{u}$ and locally Lipschitz in $\underline{x}$ and satisfies $f(0,0) = 0$.

3) $F : \Re^n \times U \to \Re$ is continuous in all arguments with $F(0,0) = 0$ and $F(x,u) > 0$, $\forall (x,u) \in \Re^n \times U \setminus \{0,0\}$.

4) The terminal penalty $V : \Omega \to \Re$ is continuous with $V(0) = 0$ and that the terminal region Ω is given by $\Omega := \{x \in X \mid V(x) \le e_1\}$ for some $e_1 > 0$ such that $\Omega \subset X$.

5) There exists a continuous local control law $u = k(x)$ such that $k(x) \in U$ for all $x \in \Omega$ and

$$\frac{\partial V}{\partial x} f\big(x, k(x)\big) + F\big(x, k(x)\big) \le 0, \quad \forall x \in \Omega. \quad (4)$$

6) The NMPC open-loop optimal control problem [2] has a feasible solution for t = 0.

Then for any sampling time $0 < \delta \le T_p$, the nominal closed-loop system is asymptotically stable and the region of attraction is given by the set of states for which the open-loop optimal control problem has a feasible solution.

Many NMPC schemes follow this theorem to guarantee stability. Generally, they differ in how the terminal region and terminal penalty terms are obtained. Basically, the terminal penalty and the terminal region are determined off-line such that the cost function gives an upper bound on the infinite horizon cost and guarantees a decrease in the value function as the horizon recedes in time. Various ways to determine a suitable terminal penalty term and terminal region exist. Examples are the use of a control Lyapunov function as a terminal penalty (16, 17) for the system in the terminal region, enforcing a decrease in the value function, or the use of a local nonlinear or linear control law to determine a suitable terminal penalty and a terminal region (18-20). The terminal region constraint is added to enforce that if the open-loop optimal control problem is feasible once, that it will remain feasible, and to allow establishing the decrease using the terminal penalty (see (10, 18, 19, 21) for more details). In general, it is not necessary to find always an optimal solution in order to guarantee stability (16, 19, 22). Only a feasible solution resulting in a decrease in the value function is necessary. This can be utilized to decrease the necessary online solution time (8).

3.3 Optimization Solvers

Although stability results for NMPC have been well established, it is not applicable in practical implementation. Since a constrained nonlinear optimization problem has to be solved online, the heavy online computational burden causes two important issues in implementation of NMPC (8). One is the computational delay. The other is the global optimization solution which cannot be guaranteed in each optimization procedure

since it is, in general, a nonconvex, constrained nonlinear optimization problem.

In practice, linear models are most often used and the resulting optimizations are linear or quadratic programs. In the nonlinear constrained optimization, the objective criterion is optimized directly by discretizing the original problem to finite dimensional approximation (8). The discretized version of the optimal control problem (OCP) can be solved with well known nonlinear programming (NLP) algorithms, such as sequence quadratic programming (SQP) (23, 24) or interior-point (IP) methods (25). For more details on optimization solvers, the reader is referred to (8, 23, 26).

4. Motion Control Using MPC

In Section 2 and 3, an overview of the theoretical and practical aspects of MPC has shown some of the challenging issues. Although MPC is suitable for low-process systems, such as chemical factories, with new optimization solvers, more powerful computers and more advanced MPC frameworks, MPC can be implemented in real-time applications, as seen in this section.

Recently, MPC strategies for path planning and local navigation have become increasingly popular in robotic research, but they are beyond the scope of this paper. In this paper, we focus on three criteria used to classify publications of motion control, i.e., MPC models, robot kinematic models, and basic motion tasks, as summarized in Table 1. Since an MPC algorithm employs an explicit model of the plant (the plant here is a mobile robot) to be controlled which is used to predict the future output behavior, we review the three most popular kinematic models in the literature, i.e., unicycle-type vehicles, car-like vehicles, and omnidirectional vehicles (27).

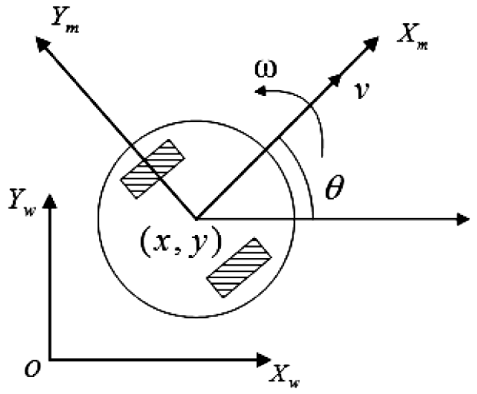
A unicycle-type vehicle, shown in Figure 2(a), has two identical parallel rear wheels, which are controlled by two independent motors on the same axle and one caster wheel. It is assumed that the center of mass of the mobile robot is located in the middle of the axis connecting the rear wheels. Based on this wheel configuration, the following kinematic model of a unicycle-type mobile robot can be obtained:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \quad (5)$$

where (x, y) indicates the position of the robot center in the world frame (X_w, Y_w) and θ is the heading angle of the robot. v and w stand for the linear and angular velocities, respectively. In (29), it is shown that the nonlinear, nonholonomic system [5] is fully controllable, i.e., it can be steered from any initial state to any final state in finite time by using finite inputs. Nonholonomic constraints mean the perfect rolling constraints without longitudinal or lateral slipping of the wheels. In the case of a trajectory-tracking controller, a linear time-varying system is obtained by approximate linearization around the trajectory. The linearization obtained is shown to be controllable as long as the trajectory does not come to stop, which implies that the system can be asymptotically stabilized by smooth linear or nonlinear feedback. Furthermore, due to Brockett's theorem (29), the asymptotic stabilization of a fixed point, where a position must be reached with a given orientation, is mainly achieved via discontinuous feedback and/or continuous time-varying feedback. An extensive review of nonholonomic control problems can be found in (30).



(a)



(b)

Figure 2. (a) A unicycle mobile robot (12 cm diameter), and (b) coordinate frames of a unicycle mobile robot (28).

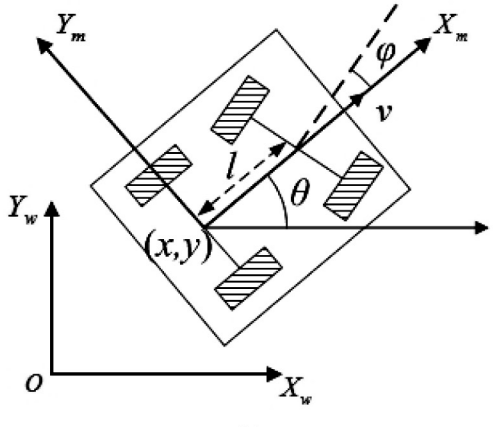
A car-like vehicle is shown in Figure 3(a). The robot is a rear-wheel-drive vehicle and its front wheels are used for steering. Based on Figure 3(b), the kinematic model is hence described as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{v}{l}\tan\varphi \end{bmatrix} \quad (6)$$

where (x, y) are the world reference frame coordinates, v is the forward velocity at the middle of the front axis, θ is yaw angle, φ is the front-wheel steering angle, and l is the distance between the wheels. The car-like robot is also a nonholonomic vehicle and it has a mechanical constraint, which imposes a maximum curvature (or minimum turning radius) of the path executed by the robot. Furthermore, for such a vehicle to move sideways requires a parking maneuver consisting of repeated changes in direction forward and backward. However, the limited maneuverability of car-like steering has an important advantage, i.e., its directionality and steering geometry provide it with very good lateral stability in high-speed turns (27).



(a)



(b)

Figure 3. (a) A car-like mobile robot, and (b) coordinate frames of a car-like mobile robot.

An omnidirectional vehicle, shown in Figure 4(a), becomes increasingly popular in mobile robot applications, since they have some distinct advantages over nonholonomic mobile robots. They have a full omnidirectionality with simultaneously and independently controlled rotational and translational motion capabilities, i.e., they can move at each instant in any direction without reorientation (31). The omnidirectional motion is enabled via special wheels used in mobile robot design. One of the most popular arrangements utilizes so-called Swedish wheels mounted on the periphery of the chassis, thus allowing freedom of motion. Based on the basic architecture of the wheeled platform illustrated in Figure 4(b), the velocity component with respect to the world frame is obtained by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} \quad (7)$$

where the point (x, y) is the position of the center of the robot on the axes (X_w, Y_w) and θ is the angular position with respect to the axis X_w. The input signals are given by u, v, ω with u, v being two orthogonal velocity vectors, where u is aligned with the reference axis of the robot. ω corresponds to the rotational velocity of the robot.

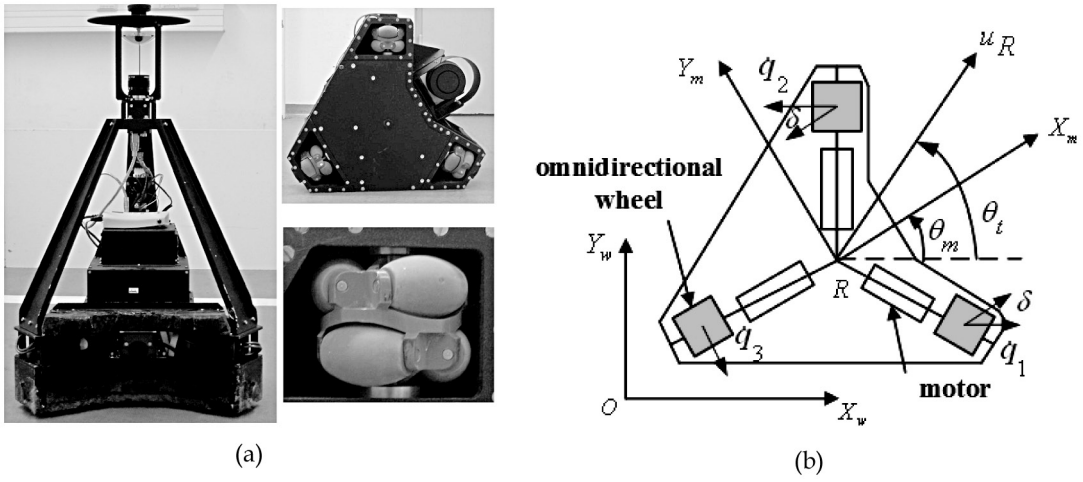


Figure 4. (a) The structure of an omnidirectional mobile robot with Swedish wheels that contain a series of rollers attached to its circumference, and (b) coordinate frames of an omnidirectional mobile robot (32).

These robot kinematic models can be used directly with MPC, called nonlinear MPC (NMPC). However, we can reduce the complexity of the nonlinear model by the following three possibilities:

- Linearization methods to linearize the nonlinear model into the linearized time-varying model;
- System identification techniques for linear GPC to approximate the nonlinear system;
- Neural network approaches to form neural network models.

The following three subsections are classified according to motion control tasks treated in the literature (2). Table 1 provides a summary of publications of motion control according to our three classification criteria.

Table 1. Classification of MPC for robot motion control tasks

	Kinematic Models			Motion Tasks			MPC Models			
	Unicy-cle	Car-like	Omni-direc-tional	Trajectory Tracking	Path Fol-lowing	Point Stabili-zation	Nonlinear	Linearized	GPC	Neural Network
Lages and Alves (33)	✓			✓				✓		
Klancar and Skrjanc (34)	✓			✓				✓		
Jiang et al. (35)	✓			✓				✓		
Seyr and Jakubek (36)	✓			✓			✓			
Hedjar et al. (37)	✓			✓			✓			
Gu and Hu (38)	✓			✓		✓	✓			
Essen and Nijmeijer (39)	✓			✓		✓	✓			
Xie and Fierro (40)	✓			✓		✓	✓			
Araujo et al. (41)			✓	✓			✓			
Chen and Li (42)	✓			✓						✓
Pan and Wang (43)	✓			✓						✓
Maurovic et al. (44)	✓			✓				✓		
Ollero and Amidi (45)		✓			✓				✓	
Normey-Rico et al. (46)	✓				✓				✓	
Vougioukas (47)		✓			✓		✓			
Conceicao et al. (48)			✓		✓		✓			
Falcone et al. (49)		✓			✓		✓	✓		

Table 1. Classification of MPC for robot motion control tasks (cont.)

	Kinematic Models			Motion Tasks			MPC Models			
	Unicycle	Car-like	Omni-directional	Trajectory Tracking	Path Following	Point Stabilization	Nonlinear	Linearized	GPC	Neural Network
Bak et al. (50)	✓				✓			✓		
Raffo et al. (51)		✓			✓			✓	✓	
Yang et al. (52)		✓			✓					✓
Gomez-Ortega and Camacho (53)	✓				✓					✓
Gu and Hu (54)		✓			✓					✓
Alves and Lages (55)	✓					✓	✓			
Kuhne et al. (56)	✓					✓	✓			
Wei et al. (57)	✓					✓	✓			
Li et al. (32)			✓	✓	✓		✓			
Kanjanawanishkul and Zell (58)			✓		✓			✓		
Kanjanawanishkul et al. (28)	✓			✓	✓		✓			

4.1 Trajectory Tracking

Typically, trajectory tracking problems for mobile robots are solved by designing control laws that make the robots track given time varying trajectories, i.e., trajectories that specify the time evolution of the position, orientation, as well as the linear and angular velocities (59). However, this approach suffers from the drawback that usually the robots' dynamics exhibit complex nonlinear terms and significant uncertainties, which make the task of computing a feasible trajectory difficult. Also, in the presence of tracking errors, the controller attempts to make the outputs catch up with the time-parameterized desired outputs. This may lead to too large control signals. One approach to eliminate such problems is to use a path following controller instead of a tracking controller, as explained in the next subsection.

In the field of mobile robotics, MPC approaches to trajectory tracking seem to be very promising because the reference trajectory is known beforehand. In the literature, most MPC controllers use a linear model of mobile robot kinematics to predict future system outputs. Lages and Alves (33) used a successive linearization approach, yielding a linear, time-varying description of the system that can be controlled through linear MPC. Then, the optimization problem can be transformed into a QP problem and easily solved by numerically robust solvers, leading to global optimal solutions at each sampling time. An MPC trajectory tracking algorithm with a robot model that is linearized around the reference trajectory was also proposed by Klancar and Skrjanc (34). Their analytic control law is explicitly obtained without using any optimization solver, while the bounded velocity and acceleration constraints are considered in low-level control. Jiang et al. (35) presented a tracking method, where the predictive control is used to predict the position and the orientation of the robot and the fuzzy control is used to deal with the nonlinear characteristics of the system.

Seyr and Jakubek (36) solved a nonholonomic control problem consisting of NMPC in conjunction with kinematics of a unicycle-type mobile robot under consideration of side slip and tangential wheel slip. Based on a Gauss-Newton algorithm, predicted future position errors are minimized by numerical computation of an optimal sequence of control inputs using pre-specified shape functions. Hedjar et al. (37) presented a finite-horizon nonlinear predictive controller using the Taylor approximation. One of the main advantages of their control schemes is that they do not require on-line optimization and asymptotic tracking of the smooth reference signal is guaranteed. Gu and Hu (38) presented a stabilizing model predictive controller for tracking control of a nonholonomic mobile robot. A terminal-state region and its corresponding local controller are developed to guarantee the stability of controlled systems. The proposed model predictive controller can be used for simultaneous tracking control and point stabilization problems. Essen and Nijmeijer (39) developed an NMPC algorithm, which is applied to both problems of point stabilization and trajectory tracking. An application of their NMPC to the stabilization of a kinematic model of a unicycle-type mobile robot with input and state constraints was studied. Xie and Fierro (40) proposed a first-state contractive model predictive control (FSC-MPC) algorithm for the trajectory tracking and point stabilization problems of nonholonomic mobile robots. Stability of the proposed MPC scheme is guaranteed by adding a first-state contractive constraint.

Araujo et al. (41) presented a methodology for state feedback MPC synthesis applied to the trajectory tracking control problem of a three-wheeled omnidirectional mobile robot. Closed loop system stability is guaranteed by deriving LMI constraints for the monotonicity of the upper bound of the cost function. Chen and Li (42) enhanced computational efficiency

of Neural Network Predictive Control (NNPC) using Particle Swarm Optimization with Controllable Random Exploration Velocity (PSO-CREV) for searching optimal solutions so that NNPC can be used in the systems with rapid dynamics. Pan and Wang (43) proposed a recurrent neural network (RNN) approach to NMPC. By using decomposition, the original optimization associated with NMPC is reformulated as a QP problem with unknown parameters. To solve the QP problem, an improved dual neural network with less complexity is applied. Recently, Maurovic et al. (44) developed an explicit MPC scheme, where the solution to the MPC minimization problem can be calculated off-line and expressed as a piecewise affine function of the current state of a mobile robot, thus avoiding the need for online minimization. By obtaining such optimal controller, which has a form of a look-up table, there is no need for expensive and large computational infrastructure.

4.2 Path Following

Path following has recently been formulated to replace the standard trajectory tracking as it is more suitable for certain applications (59). As illustrated in (60), with path following, the time dependence of the problem is removed, smoother convergence to the path is achieved, and the control signals are less likely pushed into saturation when compared to trajectory tracking.

Path following problems (59) are primarily concerned with design of control laws that steer an object (robot arm, mobile robot, ship, aircraft, etc.) to reach and to follow a geometric path, i.e., a manifold parameterized by a continuous scalar *s* (called a geometric task), while a secondary goal is to force the object moving along the path to satisfy some additional dynamic specifications (called a dynamic assignment task). This dynamic behavior is further specified via time, speed, or acceleration assignments (60).

Ollero and Amidi (45) used GPC to solve the path following problem to obtain an appropriate steering angle taking into account the vehicle velocity. A GPC approach using a Smith predictor to cope with an estimated system time delay was presented by Normey-Rico et al. (46). In (45, 46), it is assumed that the control acts only in the angular velocity, while the linear velocity is constant. Vougioukas (47) presented a reactive path tracking controller based on NMPC, along with an iterative gradient descent algorithm for its real-time implementation. In the presence of obstacles, the controller deviates from the reference trajectory by incorporating into the optimization obstacle-distance information from range sensors. Conceicao et al. (48) proposed a nonlinear model based predictive controller for an omnidirectional mobile robot. The optimization algorithms, mainly the methods based on conjugate gradients, present good times of minimization of the cost function, allowing its use in the predictive controller.

Falcone et al. (49) presented two approaches with different computational complexities for controlling an active front steering system in an autonomous vehicle. In the first approach, the MPC problem is formulated by using a nonlinear vehicle model. The second approach is based on successive online linearization of the vehicle model, resulting in a linear time-varying (LTV) system. Bak et al. (50) proposed a fast real-time receding horizon controller with velocity constraints to avoid excessive overshooting and to have time to decelerate when turning. The presented controller is based on a strategy that forecasts the turning using a receding horizon approach where the controller predicts the posture of the robot and together with knowledge of an upcoming inter section compensates the control signals. Raffo et al. (51) proposed a controller architecture considering both kinematic and dynamic control in a cascade structure. Two different MPCs are compared: 1) a state space

formulation based on the linearized kinematic model of the error between the real vehicle and a reference vehicle and 2) a GPC scheme based on a local linear model and approximation paths. They found that the GPC strategy presents better compromise between performance and computational complexity.

A neural network also helps to solve the optimization problem. Yang et al. (52) solved a path following problem by using a neural network model of a car-like robot to predict the future vehicle posture according to the current posture and control variables. The modeling errors are corrected by an on-line learning algorithm. Gomez-Ortega and Camacho (53) presented a neural network approach for mobile robot dynamics, where a neutral network multilayer perceptron is trained to reproduce NMPC behaviors in a supervised manner. Unexpected static obstacles present in the robot environment are also considered in their implementation. Gu and Hu (54) presented a path tracking scheme for a car-like mobile robot based on neural predictive control, where a multi-layer back-propagation neural network is employed to model nonlinear kinematics of a mobile robot.

4.3 Point Stabilization (Parking, Regulation)

In point stabilization, a mobile robot should be moved from an arbitrary starting pose (i.e., position and orientation) and stabilized to a desired goal pose. The point stabilization is a hard task due to the existence of a nonholonomic constraint. Due to Brockett's conditions (29), a continuously differentiable, smooth feedback control law cannot be used to stabilize a nonholonomic system at a given configuration. To overcome these limitations discontinuous (non-smooth) and time-varying control laws have been proposed.

Gu and Hu (38) developed a stabilizing receding horizon controller with simultaneous tracking and regulating capability. The switching between tracking control and regulation is not necessary. Alves and Lages (55) presented an MPC technique using polar coordinates to the problem of point stabilization of a nonholonomic mobile robot. Unlike the Cartesian coordinate counterpart, the problem described in polar coordinates generates a feedback system with no steady state error. Kuhne et al. (56) also formulated a cost function of MPC in polar coordinates to solve a point stabilization problem for a nonholonomic wheeled mobile robot. Wei et al. (57) studied the problem of stabilizing WMRs subject to wheel slippage from an initial state to a final state. When slippage of the wheels occurs, WMRs can be modeled as hybrid systems. Thus the hybrid optimal control can be formulated as a smooth MPC problem and thus effectively solved using numerical methods.

5. Experimental Scenarios

In this section, we present three experimental scenarios from our previous work to show that MPC can be applied to real-time applications. The first experiment shows a comparison between trajectory tracking and path following of an omnidirectional mobile robot (32). The linearized model of an omnidirectional mobile robot is used in the second experiment (58). In this case, a time varying convex quadratic optimization problem is formulated and solved at each time step, leading to the reduction of the computational burden. The last experiment compares trajectory tracking and path following of a unicycle-type mobile robot, including obstacle avoidance and a time-parameterized penalty (28).

5.1 Experiment 1: NMPC of an Omnidirectional Mobile Robot

Two kinds of experiments were performed to test our proposed NMPC method for an omnidirectional mobile robot (32): One was path following control with a constant desired velocity of 1.0 m/s. Here, the desired

robot orientation was the path tangent direction. The other was trajectory tracking control, where the desired robot orientation was changing with respect to time. The following eight-shaped curve was selected as a reference because its geometrical symmetry and sharp changes in curvature make the test challenging:

$$x_d(t) = 1.8\sin(0.1t), \quad y_d(t) = 1.2\sin(0.2t) \qquad (8)$$

where t is time in case of trajectory tracking, while this reference is numerically parameterized by the path variable s in case of the path following problem. For the path following problem, the error state vector $\mathbf{x}_e$ can be defined as follows:

$$\mathrm{x}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 & 0 \\ -\sin\theta_d & \cos\theta_d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \\ \alpha - \alpha_d \end{bmatrix} \qquad (9)$$

where $[x, y, \theta, \alpha]^T$ is the desired state vector and $[x_d, y_d, \theta_d, \alpha_d]^T$ is the robot state vector. α and α_d represent the moving direction of the robot and of the virtual vehicle, respectively. By using the error state and the kinematic model [7], the error state dynamic model with respect to the rotated coordinate frame becomes (32):

$$\begin{aligned} \dot{x}_e &= y_e \kappa \dot{s} - \dot{s} + u_r \cos\alpha_e \\ \dot{y}_e &= -x_e \kappa \dot{s} + u_r \sin\alpha_e \\ \dot{\theta}_e &= \dot{\theta} - \dot{\theta}_d \\ \dot{\alpha}_e &= \dot{\alpha} - \kappa \dot{s} \end{aligned} \qquad (10)$$

where K denotes the path curvature and u_r refers to the forward velocity. The resulting model is used to predict the future output behavior of our MPC algorithm (32). Some vital parameters used in our experiments were as follows: $Q = \mathrm{diag}(0.5,0.5,0.5)$, $R = \mathrm{diag}(0.1,0.1,0.1)$, $\delta = 0.07$s, and prediction step = 3.

Figure 5 and Figure 6 illustrate results of trajectory tracking and path following experiments, respectively. However, the desired translation velocity remains constant in the path following problem, which increases difficulties in following the sharp turning part of the given path.

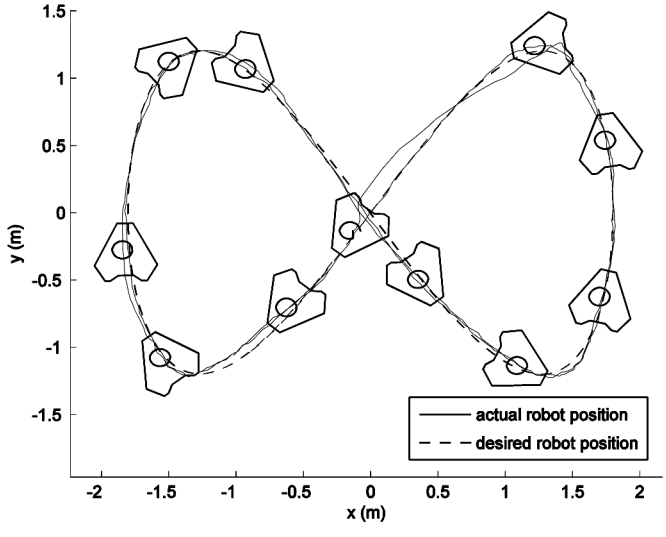


Figure 5. Experimental results of trajectory tracking control using NMPC.

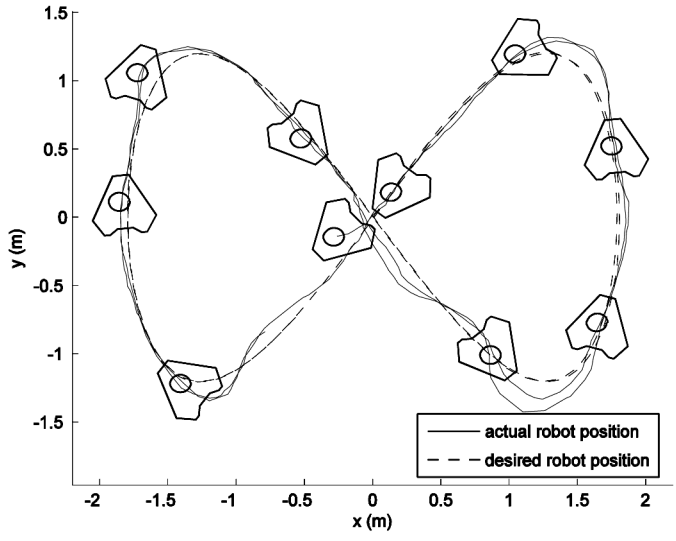


Figure 6. Experimental results of path following control using NMPC.

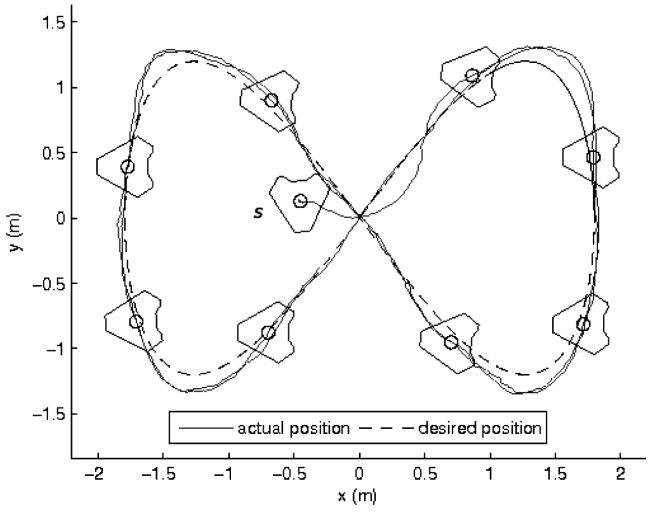
5.2 Experiment 2: A Linearized Version of MPC for an Omnidirectional Mobile Robot

One possibility to reduce computational time of solving nonlinear optimization problems is to use linearization techniques. With the linearized time-varying system, the optimization problem can be transformed into a QP problem. Since it turns into a convex problem, solving the QP problem results in global optimal solutions. This linear MPC controller is computationally effective and can be easily used in fast real-time implementations. From Subsection 5.1, we linearized the error state dynamic model [10] around the reference path. We then obtain the following linear model:

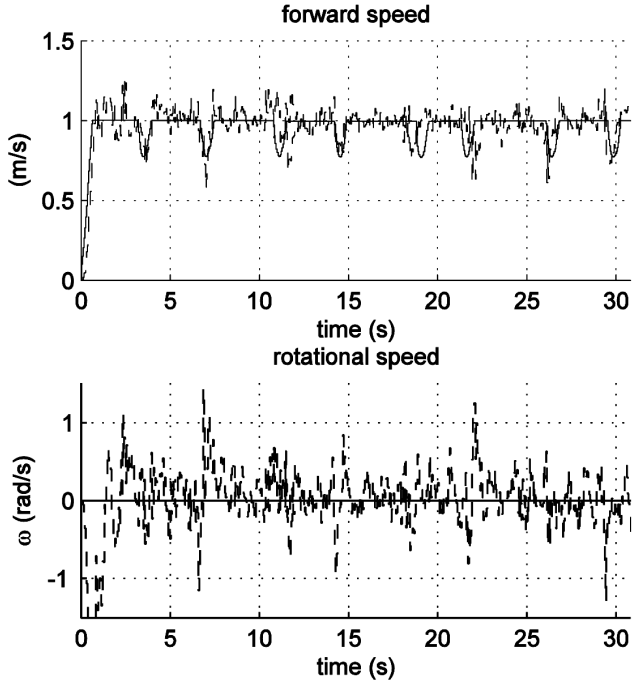
$$\begin{aligned} \dot{x}_e &= y_e \kappa_d u_r + u_1 \\ \dot{y}_e &= -x_e \kappa_d u_r + u_r \alpha_e \\ \dot{\theta}_e &= u_2 \\ \dot{\alpha}_e &= u_3 \end{aligned} \tag{11}$$

where $\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -\dot{s} + u_r \cos\alpha_e \\ \dot{\theta} - \dot{\theta}_d \\ \dot{\alpha} - \kappa \dot{s} \end{bmatrix}$, κ_d is the reference curvature. The resulting model was implemented into the QP problem with input constraints (58). Some important parameters used in our experiments were given as follows: Q = diag(300,300,7,70), R = diag(1,0.001,3), δ = 0.05 s, and prediction step = 3. These parameters were different from those in Subsection 5.1 because the different problem formulation and the different solvers were employed in our implementation.

As seen in Figure 7, using the path [8] as a reference, we achieve a real-time implementation of our control law. The forward velocity decreased in order to preserve the curvature radius when the robot made sharp turns, while the velocity commands did not exceed the velocity constraints, as expected.



(a)



(b)

Figure 7. Experimental results using the linear MPC law: (a) the superimposed snapshots, and (b) the forward velocity and the rotational velocity.

5.3 Experiment 3: NMPC of a Unicycle-type Robot

In this subsection, path following control and trajectory tracking control of a unicycle-type robot are compared. The advantage of the path following controller is that the path following controller eliminates aggressiveness of the tracking controller by forcing convergence to the desired path in a smooth way. Thus,

we incorporated this benefit to the trajectory tracking problem to achieve smooth convergence to the reference and to achieve time convergence of trajectory tracking. This was accomplished by modifying the cost function of the MPC framework through an addition of a time dependent penalty term. Based on this concept, our controller was able to optimize the reference point between the virtual vehicle (path-parameterized) and the trajectory point (time-parameterized). Furthermore, in the presence of obstacles, the controller deviated from the reference by incorporating obstacle information from range sensors into the optimization, while respecting motion constraints. Numerous simulations were performed to evaluate performance of our system. The following circle was used as a reference path:

$$x_d(t) = R\cos\left(\frac{s}{R}\right) \qquad y_d(t) = R\sin\left(\frac{s}{R}\right) \qquad (12)$$

where R is the radius of the circle. Using [5] and the following error state vector $\mathbf{x}_e$:

$$\mathrm{x}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix} \qquad (13)$$

where $[x, y, \theta]^T$ is the robot state vector and $[x_d, y_d, \theta_d]^T$ is the desired state vector, the error state dynamic model for a unicycle-type robot with respect to the rotated coordinate frame becomes

$$\begin{aligned} \dot{x}_e &= y_e\kappa\dot{s} - \dot{s} + v\cos\theta_e \\ \dot{y}_e &= -x_e\kappa\dot{s} + v\sin\theta_e \\ \dot{\theta}_e &= \omega - \kappa\dot{s} \end{aligned} \qquad (14)$$

where κ is the path curvature. However, the robot's translation velocity v has to be varied in order to achieve trajectory tracking. Thus, we introduced an acceleration control input a, where $a = \dot{v}$. Then, we obtained $\dot{\eta}_e = v - \dot{v}_d$, where $\eta_e = v - v_d$ and v_d is the desired translation velocity. In our implementation, some significant parameters were set as follows: Q = diag(0.2,2,0.01,0.01), R = diag(0.0001,0.0001,0.0001), prediction step = 3 and δ = 0.1 s. More details of our problem formulation and controller design can be found in (28).

The performance achieved with pure path following, pure trajectory tracking (see (38) for details), and for combined trajectory tracking and path following is assessed. Figure 8(a) and Figure 8(b) show simulation results of pure path following control and pure trajectory tracking control, respectively, with four different initial poses. The velocities of pure path following are depicted in Figure 9(a), while those of pure trajectory tracking are plotted in Figure 9(b) when the initial pose of both cases was set to (1.5,-0.5,π). As seen from the results, in case of path following control, the robot motions are less aggressive while the robot is approaching the reference path (see Figure 8(a)) and the control signals are less likely saturated. Figure 8(c) shows the simulation results of the combination of path following control and trajectory tracking control. The velocities are shown in Figure 9(c) when the initial pose was set to (1.5,-0.5,π). This controller is able to achieve both reference convergence and time convergence with smooth motions. As seen in the results, the robot converges smoothly to the desired path and then it reacts to achieve zero tracking error.

In Figure 10, two moving obstacles were present. The velocity of the first obstacle was 0.2 m/s at -135°, while the velocity of the second obstacle was 0.6 m/s at 150°. In the simulation results, the robot moved backward to avoid the collision and waited until it was able to find a way to stay away from the obstacles and to follow the reference.

Next, a unicycle-type mobile robot, shown in Figure 2(a) was used in real-world experiments. The robot controller is an ATMEGA644 microprocessor with 64 KB flash program memory, 16MHz clock frequency

and 4 KB SRAM. The localization was given by a camera looking down upon the robot's workplace and a PC was used to compute the control inputs and then sent these inputs to the robot via WLAN. The same reference used in simulation was employed in this experiment. From experimental results shown in Figure 11, system performance degradation compared to the simulation results was mainly caused by time delay originated from computation time of the control algorithm, the vision-based tracking system, and the wireless connection.

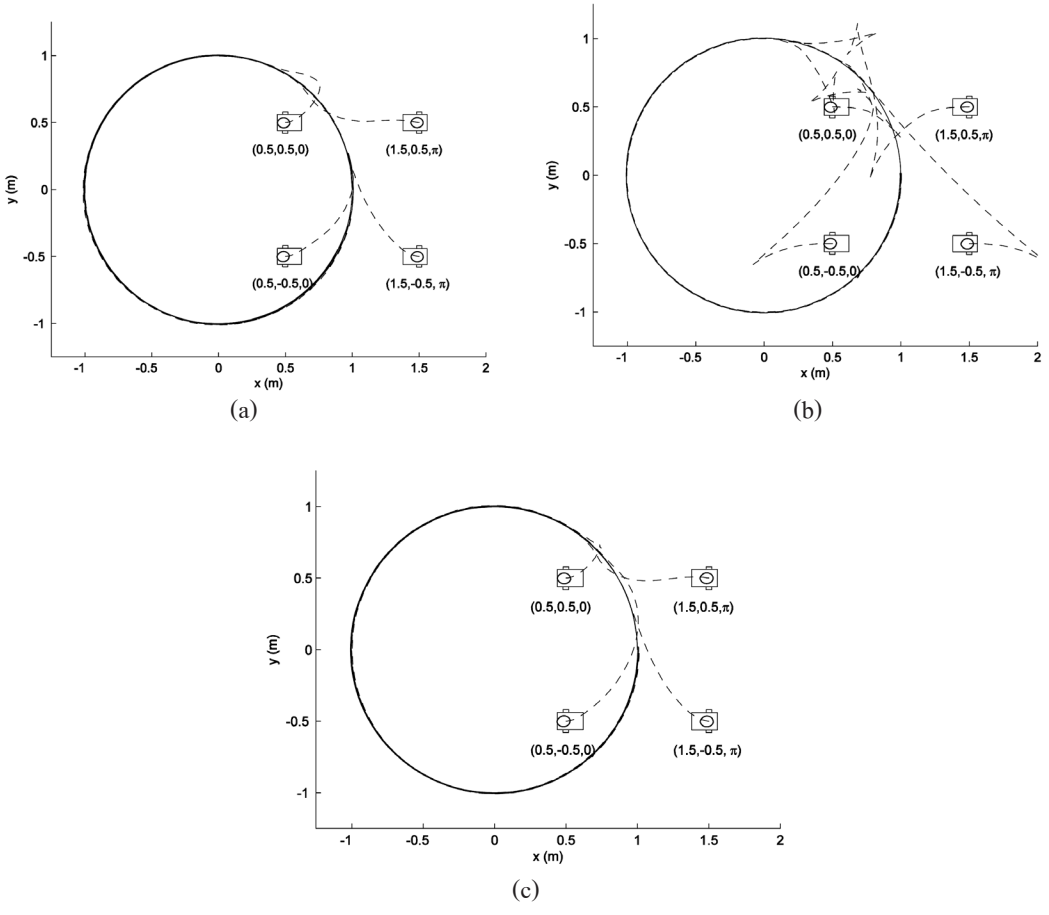
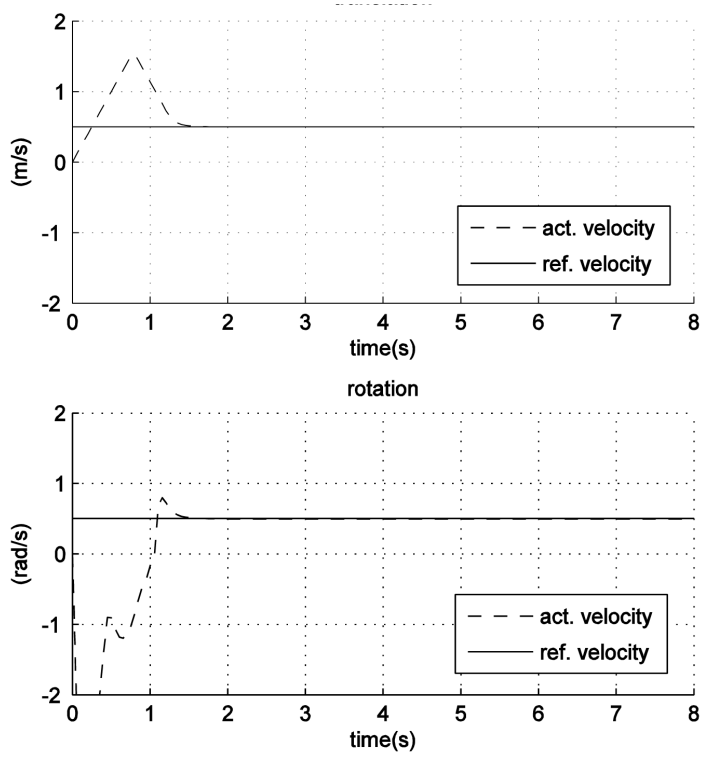
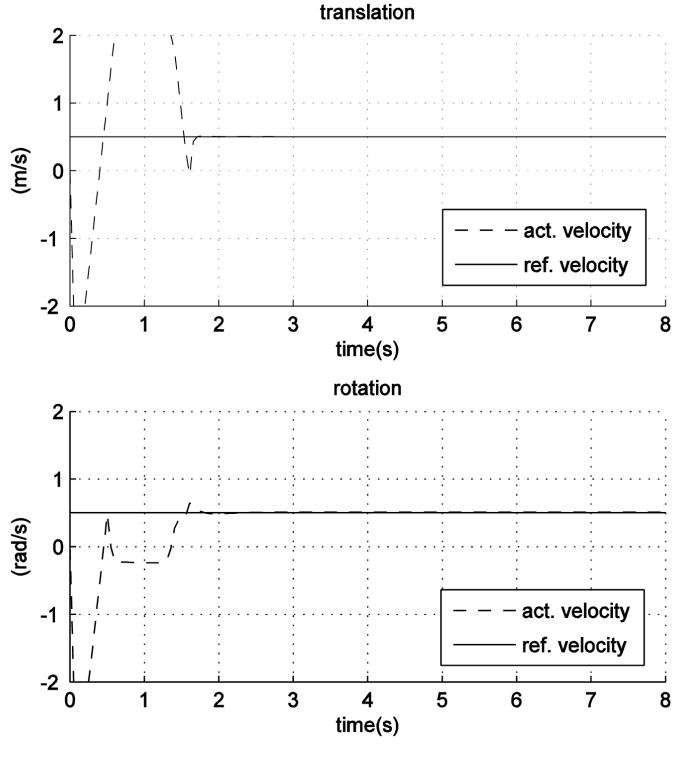


Figure 8. Simulation results with four different initial poses: (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking.

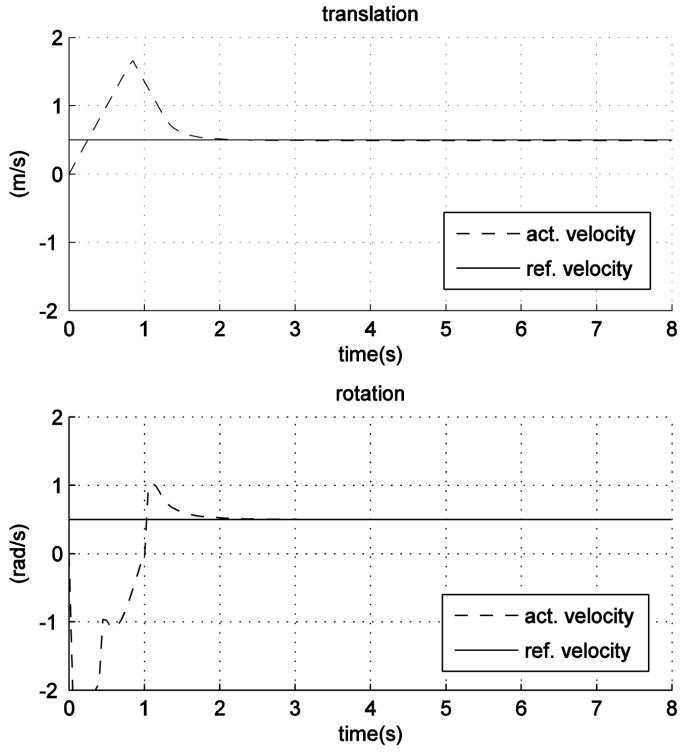


(a)



(b)

Figure 9. The robot velocities when the initial pose was set to (1.5, -0.5, π): (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking.



(c)

Figure 9. The robot velocities when the initial pose was set to (1.5, -0.5, π): (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking. (cont.)

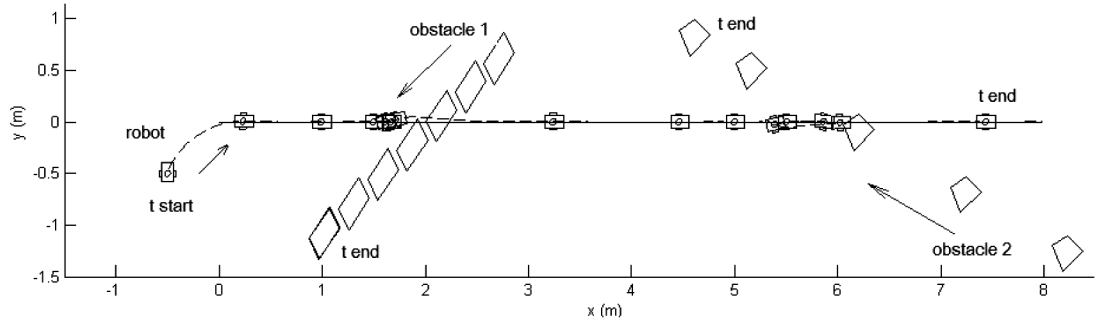
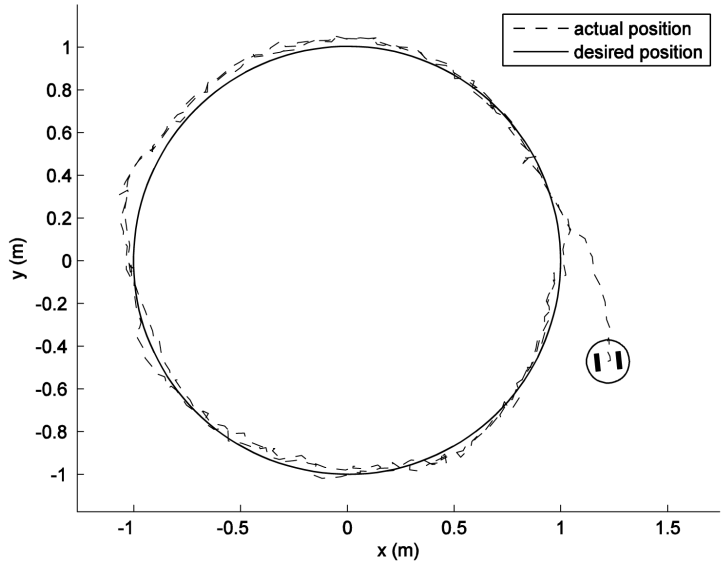
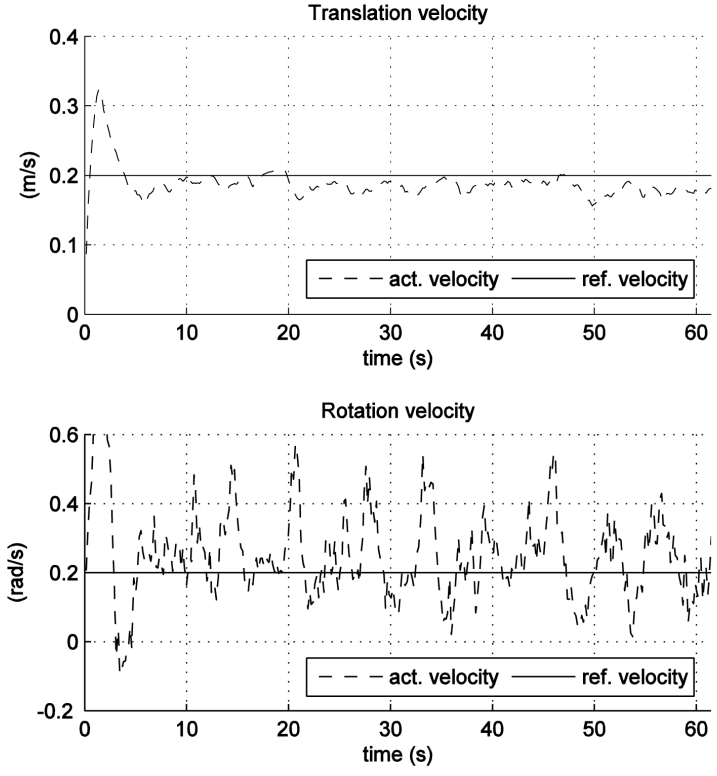


Figure 10. The simulation results when two moving polygonal obstacles were present.



(a)



(b)

Figure 11. Experimental results using our NMPC law: (a) the robot positions and its reference, and (b) the robot's velocities.

6. Future Research Perspectives

The topics listed here are not intended to be exhaustive, but rather to be indicative of the classes of problems which we are interested in.

6.1 Decentralized MPC

Formation control is one of the most active research topics in multi-robot systems (61). The goal of formation control is that a group of robots has to maintain a desired formation shape, while tracking or following a reference. One way to solve this problem is to formulate it as a centralized MPC scheme, i.e., one MPC controller has the full knowledge about the entire robot system and computes all the control inputs for the entire robot system. However, in general, the centralized implementation is not practical since the size of the state variables depends typically on the number of mobile robots. When the control horizon becomes larger, the number of variables, of which the robot has to find the value, increases rapidly. Also, the demands of computational power and memory are daunting for the real-time solution of systems with a large control horizon and a large number of mobile robots. Thus, the research has led to decomposing the centralized system into smaller subsystems, which are independently controlled in the MPC framework.

The main challenge is that stability and feasibility of decentralized schemes are very difficult to prove and/or too conservative (62). Even if we assume T_p to be infinite, the decentralized MPC approach does not guarantee that solutions computed locally are globally feasible and stable. The decentralization of the control is further complicated when disturbances act on the subsystems making the prediction of future behavior uncertain. The key point to guarantee feasibility and stability is that when decisions are made in a decentralized fashion, the actions of each subsystem must be consistent with those of the other subsystems (63). Thus, decisions taken independently do not lead to a violation of the coupling constraints. Some approaches based on this strategy were proposed by Dunbar and Murray (64), and Kanjanawanishkul and Zell (65), and references therein.

6.2 An Improved Real-time MPC Framework

As shown in our experimental scenarios, the MPC framework can be used beyond process control. However, the main obstacle in applying the MPC technology to real-time applications, e.g., WMRs is that the optimization problem is computationally quite demanding, especially for nonlinear systems. In order to reduce the online computational requirements, there are a number of directions in which future research on this problem can proceed. The first direction is to apply function approximations, e.g., artificial neural networks, which can be trained off-line to represent the optimal control law. Second, explicit MPC techniques, such as multi-parametric quadratic programming (mp-QP) approaches, may be employed since they can handle constrained MIMO linear models as well as constrained MIMO piecewise linear models (66), where part of the computations are performed off-line. The third direction to reduce the online computational requirements relates to open-loop optimization solvers. We should improve and test several nonlinear optimization algorithms which can enlarge the range of conditions for which a nonlinear MPC controller becomes real-time implementable.

7. Conclusions

The main objective of this paper is to review MPC schemes that are applied to motion control tasks of WMRs. We classify publications based on three criteria, i.e., MPC models, robot kinematic models, and motion tasks, as seen in Table 1. With the development

of increasingly faster processors and efficient numerical algorithms, the use of an MPC controller in faster applications (e.g., WMRs) becomes possible. Successful real-time motion tasks of WMRs have already been shown in Section 5. Furthermore, the comparison between path following and trajectory tracking for an omnidirectional mobile robot and a unicycle-type mobile robot has also been shown and discussed.

Although MPC approaches of WMRs have been particularly well studied, as seen in our survey, much work remains to be done to develop strategies capable of yielding better performance in case of mobile robots with parameter uncertainties, partially-known or unknown environments, and multi-robot cooperation.

8. References

(1) Broxvall M, Seo BS, Kwon WY. The PEIS kernel: A middleware for ubiquitous robotics. Proceedings of the IROS-07 Workshop on Ubiquitous Robotics Space Design and Applications; 2007 Oct 29-Nov 2; San Diego, CA; 2007.

(2) Morin P, Samson C. Motion control of wheeled mobile robot. In: Siciliano B, Khatib O, editors. Springer Handbook of Robotics. Berlin Heidelberg: Springer; 2008. p. 799-826.

(3) Oriolo G, Luca AD, Vendittelli M. WMR control via dynamic feedback linearization: design, implementation and experimental validation. IEEE Transactions on Control Systems Technology. 2002;10(6): 835-852.

(4) Solea R, Nunes U. Trajectory planning with velocity planner for fully-automated passenger vehicles. Proceedings of the IEEE Intelligent Transportation Systems Conference; 2006 Sep 17-20; Toronto, Canada; 2006. P. 474-480.

(5) Soeanto D, Lapierre L, Pascoal A. Adaptive non-singular path-following, control of dynamic wheeled robots. Proceedings of the International Conference on Advanced Robotics; 2003 Jun 30-Jul 3; Coimbra, Portugal, 2003. P. 1387–1392.

(6) Morari M, Lee J. Model predictive control: Past, present, and future. Computers and Chemical Engineering. 1999;23(4/5): 667–682.

(7) Qin SJ, Badgwell T. A survey of industrial model predictive control technology. Control Engineering Practice. 2003;11(7): 733–764.

(8) Allgower F, Findeisen R, Nagy ZK. Nonlinear model predictive control: from theory to application. Journal of Chinese Institute of Chemical Engineers. 2004;35(3): 299–315.

(9) Bemporad A, Morari M. Control of systems integrating logic, dynamics, and constraints. Automatica. 1999;35(3): 407–427.

(10) Mayne DQ, Rawlings JB, Rao CV, Scokaert POM. Constrained model predictive control: stability and optimality. Automatica. 2000;36: 789–814.

(11) Fontes F. A general framework to design stabilizing nonlinear model predictive controllers. System and Control Letters. 2001;42(2): 85–93.

(12) Magni L, Raimondo DM, Allgower F, editors. Nonlinear model predictive control – towards new challenging applications. Berlin: Springer-Verlag, 2009.

(13) Clarke DW, Mohtadi C, Tuffs PS. Generalized predictive control, part I: The basic algorithm. Automatica. 1987;23(2): 137–148.

(14) Clarke DW, Mohtadi C, Tuffs PS. Generalized predictive control, part II: Extension and interpretations. Automatica. 1987;23(2): 149–160.

(15) Kwon WH, Han S. Receding horizon control: Model predictive control for state models. London: Springer-Verlag; 2005.

(16) Jadbabaie A, Yu J, Hauser J. Unconstrained receding horizon control of nonlinear systems. IEEE Transactions on Automatic Control. 2001;46(5): 776–783.

(17) Primbs JA, Nevistic V, Doyle JC. A receding horizon generalization of pointwise min-norm controllers. IEEE Transactions on Automatic Control. 2000;45(5): 898–909.

(18) Michalska H, Mayne DQ. Robust receding horizon control of constrained nonlinear systems. IEEE Transactions on Automatic Control. 1993;38(11): 1623–1633.

(19) Chen H, Allgower F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica. 1998;34(10): 1205–1218.

(20) Magni L, Scattolini R. State-feedback MPC with piecewise constant control for continuous-time systems. Proceedings of the IEEE Conference on Decision and Control; 2002 Dec 10-13; Las Vegas, USA; 2002. P. 4625–4630.

(21) Jadbabaie A, Yu J, Hauser J. Stabilizing receding horizon control of nonlinear systems: A control Lyapunov function approach. Proceedings of the American Control Conference; 1999 Jun 2-4; San Diego, CA; 1999. P. 1535–1539.

(22) Scokaert POM, Mayne DQ, Rawlings JB. Suboptimal model predictive control (feasibility implies stability). IEEE Transactions on Automatic Control. 1999;44(3): 648–654.

(23) Bartlett RA, Wachter A, Biegler LT. Active set vs. interior point strategies for model predictive control. Proceedings of the American Control Conference; 2000 Jun 28-30; Chicago, IL; 2000. P. 4229–4233.

(24) Tenny M, Wright S, Rawlings J. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. Computational Optimization and Applications. 2004;28(1): 87–121.

(25) Cervantes A, Wachter A, Tutuncu R, Biegler L. A reduced space interior point strategy for optimization of differential algebraic systems. Computers & Chemical Engineering. 2000;24(1): 39–51.

(26) Biegler LT, Grossmann IE. Retrospective on optimization. Computers & Chemical Engineering. 2004;28(8): 1169–1192.

(27) Siegwart R, Nourbakhsh I. Introduction to autonomous mobile robots. The MIT Press; 2004.

(28) Kanjanawanishkul K, Hofmeister M, Zell A. Smooth reference tracking of a mobile robot using nonlinear model predictive control. Proceedings of the 4th European Conference on Mobile Robots (ECMR); 2009 Sept 23-25; Mlini/ Dubrovnik, Croatia; 2009. P. 161–166.

(29) Brockett RW. Asymptotic stability and feedback stabilization. Differential Geometric Control Theory. Boston: Birkhauser; 1983, P. 181–191.

(30) Kolmanovsky I, McClamroch NH. Developments in nonholonomic control problems. IEEE Control Systems Magazine. 1995;15(6): 20–36.

(31) Campion G, Bastin G, D´Andrea-Novel B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. IEEE Transactions on Robotics and Automation. 1996;12(1): 47–62.

(32) Li X, Kanjanawanishkul K, Zell A. Nonlinear model predictive control of an omnidirectional mobile robot. Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS2008); 2008 Jul 23-25; Baden-Baden, Germany; 2008. P. 92–99.

(33) Lages WF, Alves JAV. Real-time control of a mobile robot using linearized model predictive control. Proceedings of the 4th IFAC Symposium on Mechatronic Systems; 2006 Sep 12-14; Heidelberg, Germany; 2006. P. 968–973.

(34) Klancar G, Skrjanc I. Tracking-error model-based predictive control for mobile robots in real time. Robotics and Autonomous Systems. 2007;55(6): 460–469.

(35) Jiang X, Motai Y, Zhu X. Predictive fuzzy control for a mobile robot with nonholonomic constraints. Proceedings of the 12th International Conference on Advanced Robotics; 2005 Jul 18-20; Seattle, WA; 2005. P. 58–63.

(36) Seyr M, Jakubek S. Mobile robot predictive trajectory tracking. Proceedings of the International Conference on Informatics in Control; 2005 Sep 13-14; Barcelona, Spain; 2005. P. 112–119.

(37) Hedjar R, Toumi R, Boucher P, Dumur D. Finite horizon nonlinear predictive control by the Taylor approximation: application to robot tracking trajectory. International Journal of Applied Mathematics and Computer Science. 2005;15(4): 527–540.

(38) Gu D, Hu H. Receding horizon tracking control of wheeled mobile robots. IEEE Transactions on Control Systems Technology. 2006;14(4): 743–749.

(39) Essen HV, Nijmeijer H. Non-linear model predictive control of constrained mobile robots. Proceedings of the European Control Conference; 2001 Sep 4-7; Porto, Portugal; 2001. P. 1157–1162.

(40) Xie F, Fierro R. First-state contractive model predictive control of nonholonomic mobile robots. Proceedings of the American Control Conference; 2008 Jun 11-13; Seattle, WA; 2008. P. 3494–3499.

(41) Araujo H, Conceicao A, Oliveira G, Pitanga J. Model predictive control based on LMIs applied to an omni-directional mobile robot. Proceedings of the 18th IFAC World Congress; 2011 Aug 28-Sep 2; Milano, Italy; 2011. P. 8171–8176.

(42) Chen X, Li Y. Neural network predictive control for mobile robot using PSO with controllable random exploration velocity. International Journal of Intelligent Control and Systems. 2007;12(3): 217–229.

(43) Pan Y, Wang J. A neurodynamic optimization approach to nonlinear model predictive control. Proceedings of the IEEE International Conference on Systems Man and Cybernetics (SMC); 2010 Oct 10-13; Istanbul, Turkey; 2010. P. 1597–1602.

(44) Maurovic I, Baotic M, Petrovic I. Explicit model predictive control for trajectory tracking with mobile robots. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011); 2011 Jul 3-4; Budapest, Hungary; 2011. P. 712–717.

(45) Ollero A, Amidi O. Predictive path tracking of mobile robots: Application to the CMU Navlab. Proceedings of the International Conference on Advanced Robotics; 1991 Jun 19-22; Pisa, Italy; 1991. P. 1081–1086.

(46) Normey-Rico JE, Gomez-Ortega J, Camacho EF. A Smith-predictor based generalized predictive controller for mobile robot path tracking. Control Engineering Practice. 1999;7(6): 729–740.

(47) Vougioukas SG. Reactive trajectory tracking for mobile robots based on non linear model predictive control. Proceedings of the IEEE International Conference on Robotics and Automation; 2007 Apr 10-14; Roma, Italy; 2007. P. 3074–3079.

(48) Conceicao A, Oliveira H, Silva AS, Oliveira D, Moreira A. A nonlinear model predictive control of an omni-directional mobile robot. Proceedings of the IEEE International Symposium on Industrial Electronics; 2007 Jun 4-7; Vigo, Spain; 2007. P. 2161 – 2166.

(49) Falcone P, Borrelli F, Asgari J, Tseng HE, Hrovat D. Predictive active steering control for autonomous vehicle systems. IEEE Transactions on Control Systems Technology. 2007;15(3): 566–580.

(50) Bak M, Poulsen NK, Ravn O. Path following mobile robot in the presence of velocity constraints (Technical Report): Informatics and Mathematical Modeling, Technical University of Denmark; 2001.

(51) Raffo G, Gomes G, Normey-Rico J, Kelber C, Becker L. A predictive controller for autonomous vehicle path tracking. IEEE Transactions on Intelligent Transportation Systems. 2009;10(1): 92–102.

(52) Yang X, He K, Guo M, Zhang B. An intelligent predictive control approach to path tracking problem of autonomous mobile robot. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics; 1998 Oct 11-14; San Diego, CA; 1998. P. 3301–3306.

(53) Gomez-Ortega J, Camacho E. Mobile robot navigation in a partially structured static environment using neural predictive control. Control Engineering Practices. 1996;4(12): 1669-1679.

(54) Gu D, Hu H. Neural predictive control for a car-like mobile robot. Robotics and Autonomous Systems. 2002;39(2): 73–86.

(55) Alves J, Lages W. Real-time point stabilization of a mobile robot using model predictive control. Proceedings of the 13th IASTED International Conference Robotics and Application; 2007 Aug 29-31; Wuerzburg, Germany; 2007. P. 58–63.

(56) Kuhne F, Lages W, da Silva Jr J. Point stabilization of mobile robots with nonlinear model predictive control. Proceedings of the IEEE International Conference on Mechatronics and Automation; 2005 Jul 29-Aug 1; Niagara Falls, Canada; 2005. P. 1163–1168.

(57) Wei S, Zefran M, Uthaichana K, DeCarlo R. Hybrid model predictive control for stabilization of wheeled mobile robots subject to wheel slippage. Proceedings of the IEEE International Conference on Robotics and Automation; 2007 Apr 10-14; Roma, Italy; 2007. P. 2373 – 2378.

(58) Kanjanawanishkul K, Zell A. Path following for an omnidirectional mobile robot based on model predictive control. Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009); 2009 May 12-17; Kobe, Japan; 2009. P. 3341 – 3346.

(59) Aguiar AP, Dacic DB, Hespanha JP, Kokotovic P. Path-following or reference-tracking? An answer relaxing the limits to performance. Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles; 2004 Jul 5-7; Lisbon, Portugal; 2004.

(60) Skjetne R, Fossen T, Kokotovic P. Robust output maneuvering for a class of nonlinear systems. Automatica. 2004;40(3): 373–383.

(61) Murray RM. Recent research in cooperative-control of multivehicle systems. Journal of Dynamics, Systems, Measurement and Control. 2007;129(5): 571–583.

(62) Keviczky T, Borrelli F, Balas GJ. A study on decentralized receding horizon control for decoupled systems. Proceedings of the American

Control Conference; 2004 Jun 30-Jul 2; Boston, Massachusetts; 2004. P. 4921–4926.

(63) Richards AG, How JP. A decentralized algorithm for robust constrained model predictive control. Proceedings of the American Control Conference; 2004 Jun 30-Jul 2; Boston, Massachusetts; 2004. P. 4261–4266.

(64) Dunbar WB, Murray RM. Distributed receding horizon control for multi-vehicle formation stabilization. Automatica. 2006;42(4): 549–558.

(65) Kanjanawanishkul K, Zell A. Distributed model predictive control for coordinated path following control of omnidirectional mobile robots. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008); 2008 Oct 12-15; Singapore; 2008. P. 3120–3125.

(66) Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. Automatica. 2002;38(1): 3–20.