Smooth Reference Tracking of a Mobile Robot using Nonlinear Model Predictive Control

Kiattisin Kanjanawanishkul Marius Hofmeister Andreas Zell
Computer Science Department, University of Tübingen, Tübingen, Germany

***Abstract*— In this paper, path following control and trajectory tracking control of a mobile robot have been studied. Reference convergence in a path following problem and time convergence in a trajectory tracking problem are considered in the cost function of the nonlinear model predictive control framework. The benefit of path following control 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 incorporate this benefit to the trajectory tracking problem to achieve smooth convergence to the reference and to achieve time convergence of trajectory tracking. Furthermore, by using nonlinear model predictive control, input constraints can be handled straightforwardly in the optimization problem so that the robot can travel safely. Our controller was validated by simulation and real-world experiments with a unicycle-type mobile robot were also conducted.**

***Index Terms*— Path following, trajectory tracking, mobile robot, nonlinear model predictive control**

I. Introduction

Three generic problems of motion control of a vehicle addressed in the literature can be described below [1]:

- point stabilization, where the objective is to stabilize a vehicle at a desired robot posture,
- trajectory tracking, where the vehicle is required to track a time-parameterized reference, and
- path following, where the vehicle is required to converge to and follow a desired path-parameterized reference, without any temporal specifications.

Point stabilization is very different from the problems of path following and trajectory tracking. A central aspect of the problem, which triggered much of the subsequent research on the control of nonholonomic systems, is that asymptotic stabilization of fixed points cannot be achieved by using continuous feedbacks which depend on the state only. This is a consequence of an important result due to Brockett in 1983 [2].

The trajectory tracking problem for fully actuated systems is now well studied. However, when it comes to underactuated vehicles, i.e., when the vehicle has less actuators than state variables to be tracked, the problem is still a very active topic of research.

In path following control, a path following controller should look at (i) the distance from the vehicle to the reference path and (ii) the angle between the vehicle's velocity vector and the tangent to the path, and then reduce both to zero, without any consideration in temporal specifications. Pioneering work in this area can be found in [3] as well as [4]. Typically, smoother convergence to a path is achieved compared to trajectory tracking controllers, and the control signals are less likely pushed to saturation. The solutions of this path following problem have been applied in a wide range of applications. For example, Samson [5] described a path following problem for a car pulling several trailers. In [6], Altafini addressed a path following controller for an n trailer vehicle. Furthermore, path following controllers for aircraft and marine vehicles have been reported in [7] and [8], respectively.

Let $\Gamma(s) \in \mathbb{R}^2$ be a desired geometric path parameterized by the curvilinear abscissa $s(t) \in \mathbb{R}$. We then have the freedom to select a temporal specification for $s(t)$. In particular, the rate of progression ($\dot{s}$) of a virtual vehicle, considered as an addition control input, has been controlled explicitly (e.g. [9, 10, 11, 12]). Stringent initial condition constraints that are present in a number of path following control strategies have been overcome, as stated in [9].

In this paper, we wish to achieve smooth spatial convergence to the trajectory as well as time convergence using the advantage of path following control. This is accomplished by modifying the cost function of the model predictive control framework through the addition of a time dependent penalty term. Based on this concept, our controller is able to optimize the reference point between the virtual vehicle (path-parameterized) and the trajectory point (time-parameterized) and also takes into account input constraints. Furthermore, in the presence of obstacles, the controller deviates from the reference by incorporating obstacle information from range sensors into the optimization, while respecting motion constraints.

This paper is organized as follows: Section II describes the mathematical model of a mobile robot and explains the basic principle in path following and trajectory tracking. The control law based on nonlinear model predictive control (NMPC) is developed in Section III. In Section IV, simulation results are shown, and then real-world experiments with a unicycle-type mobile robot are given in Section V. Finally, our conclusions and future work are drawn in Section VI.

II. Problem Formulation

A simple kinematic model of a unicycle-type mobile robot is the following:

$$\begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} v_m \cos\theta_m \\ v_m \sin\theta_m \\ \omega_m \end{bmatrix} , \tag{1}$$

where $\mathbf{x}_m(t) = [x_m, y_m, \theta_m]^T$ is the state vector in the world frame. v_m and ω_m stand of the linear and angular velocities, respectively.

We first consider the path following control problem of a mobile robot. We wish to find control law $\dot{s}$ and ω_m such that the robot follows a virtual vehicle with position $\mathbf{x}_d = [x_d, y_d, \theta_d]^T$. The kinematic model of a mobile robot can be formulated with respect to a Serret-Frenet frame moving along the reference path. This frame plays the role of the body frame of a virtual vehicle that must be followed by the real robot as depicted in Figure 1(b), together with a spatial path Γ. In the path following problem, we normally let the forward velocity v_m track a desired velocity profile v_d, while the rate of progression of a virtual vehicle $\dot{s}$ converges to v_m. The error state vector $\mathbf{x}_e$ between the robot state vector $\mathbf{x}_m$ and a virtual vehicle's state vector $\mathbf{x}_d$ can be expressed in the frame of the path coordinate as follows

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_m - x_d \\ y_m - y_d \\ \theta_m - \theta_d \end{bmatrix} . \tag{2}$$

Using (1) and (2), the error state dynamic model chosen in a rotated coordinate frame becomes

$$\begin{aligned} \dot{x}_e &= y_e \dot{s}\kappa - \dot{s} + v_m \cos\theta_e \\ \dot{y}_e &= -x_e \dot{s}\kappa + v_m \sin\theta_e \quad , \\ \dot{\theta}_e &= \omega_m - \dot{s}\kappa \end{aligned} \tag{3}$$

where κ is the path curvature and $\dot{s}$ is the velocity of a virtual vehicle, bounded by $0 \leq \dot{s} \leq \dot{s}_{\max}$.

However, the robot's translation velocity v_m has to be controlled in order to achieve trajectory tracking. Thus, we introduce the acceleration control input a_m, where $a_m = \dot{v}_m$ and we then obtain

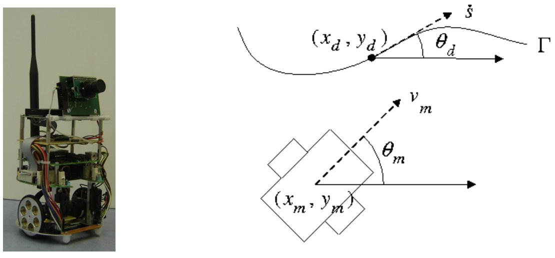
$$\dot{\eta}_e = a_m - \dot{v}_d, \tag{4}$$

where $\eta_e = v_m - v_d$.

Similar to [20], we redefine the control signals

$$\mathbf{u}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -\dot{s} + v_m \cos\theta_e \\ \omega_m - \dot{s}\kappa \\ a_m - \dot{v}_d \end{bmatrix} . \tag{5}$$

The control input vector $\mathbf{u}_e$ is used as the control input in our NMPC framework. When the open-loop optimal control problem is solved, the system control input signals $\dot{s}$, a_m, ω_m can be obtained by (5).



(a) (b)

Fig. 1. (a) A unicycle-type mobile robot (12 cm diameter) used in experiments and (b) a graphical representation of a unicycle mobile robot and a reference path.

Subsequently, the error state dynamic model becomes

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \\ \dot{\eta}_e \end{bmatrix} = \begin{bmatrix} 0 & \dot{s}\kappa & 0 & 0 \\ -\dot{s}\kappa & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \\ \eta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ v_m \sin\theta_e \\ u_2 \\ u_3 \end{bmatrix} . \tag{6}$$

III. Nonlinear Model Predictive Control Design

Nonlinear model predictive control (NMPC) is based on a finite-horizon continuous time minimization of nonlinear predicted tracking errors with constraints on the control inputs and the state variables. It predicts system outputs based on current states and the system model, finds an open-loop control profile by numerical optimization, and applies the first control signal in the optimized control profile to the system. However, due to the use of a finite horizon, control stability becomes one of the main problems. In general, the terminal region constraint and/or the terminal penalty in the cost function are employed to enforce stability. Basically, the terminal penalty is assumed to be a control Lyapunov function for the system in the terminal region, enforcing a decrease in the value function. 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 [13, 14, 15, 16] for more details).

Most model predictive controllers use a linear model of mobile robot kinematics to predict future system outputs. In [17, 18], a model-predictive control based on a linear, time-varying description of the system was used for trajectory tracking control. Generalized predictive control (GPC) was used to solve the path following problem in [19]. The nonlinear predictive controller scheme for a trajectory tracking problem was proposed in [20, 21]. Recently, Falcone et al. [22] implemented an MPC-based approach for active steering control design. They presented two approaches, i.e., MPC using a nonlinear vehicle model and MPC based on successive online linearization of the vehicle model. The differences of this paper from other work are that this paper (i) deals with path following control, which can provide the optimal velocity of a virtual vehicle to be followed along the path, (ii) achieves smooth convergence to the reference with time constraints, and (iii) takes into account obstacle avoidance.

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

$$\begin{aligned} &\dot{\mathbf{x}}_e(t) = f(\mathbf{x}_e(t), \mathbf{u}_e(t)), \\ &\text{subject to:} \quad \mathbf{x}_e(t) \in \mathcal{X}, \; \mathbf{u}_e(t) \in \mathcal{U}, \; \forall t \geq 0 \end{aligned} \tag{7}$$

where $\mathbf{x}_e(t) \in \mathbb{R}^n$, $\mathbf{u}_e(t) \in \mathbb{R}^m$ are the n dimensional state vector and the m dimensional input vector of the system, respectively. $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ denote the set of feasible states and inputs of the system, respectively. In 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:

$$\min_{\mathbf{u}_e(\cdot)} \int_t^{t+T_p} F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau))\, d\tau + V(\mathbf{x}_e(t+T_p)), \tag{8}$$

$$\text{subject to:} \quad \begin{aligned} &\dot{\mathbf{x}}_e(\tau) = f(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) \\ &\mathbf{u}_e(\tau) \in \mathcal{U} \quad \forall \tau \in [t, t+T_c] \\ &\mathbf{x}_e(\tau) \in \mathcal{X} \quad \forall \tau \in [t, t+T_p] \\ &\mathbf{x}_e(t+T_p) \in \Omega \end{aligned} \tag{9}$$

where $F(\mathbf{x}_e, \mathbf{u}_e) = \mathbf{x}_e^T Q \mathbf{x}_e + \mathbf{u}_e^T R \mathbf{u}_e$. T_c and T_p are the control horizon and the prediction horizon, respectively, with $T_c \leq T_p$. $V(\mathbf{x}_e(t+T_p))$ 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. Similar to [20], a Lyapunov function for the terminal-state penalty can be selected as follows:

$$V(\mathbf{x}_e(t+T_p)) = \tfrac{1}{2}\mathbf{x}_e(t+T_p)^T P \mathbf{x}_e(t+T_p), \tag{10}$$

where P is a positive definite matrix, under the terminal-state controller $\mathbf{u}_e^L(t)$ such that the following condition is satisfied:

$$\dot{V}(\mathbf{x}_e(t)) + F(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \leq 0, \quad \forall \mathbf{x}_e(t) \in \Omega. \tag{11}$$

The terminal state feedback controller $\mathbf{u}_e^L = [u_1^L, u_2^L, u_3^L]^T$ is defined as follows:

$$\begin{aligned} u_1^L &= -\alpha x_{eT} \\ u_2^L &= -\beta \theta_{eT} \\ u_3^L &= -\gamma \eta_{eT} \end{aligned} \ , \tag{12}$$

where $\alpha,\ \beta,\ \gamma > 0$, and $\mathbf{x}_e(t+T_p) = [x_{eT}, y_{eT}, \theta_{eT}, \eta_{eT}]^T$. The subscript T denotes the terminal state. All weight parameters have to be selected such that (11) is satisfied.

The positive definite weight matrices of the F function in (8) are selected as follows:

$$Q = \text{diag}(q_{11}, q_{22}, q_{33}, q_{44}), \ R = \text{diag}(r_{11}, r_{22}, r_{33}) \ , \tag{13}$$

and the positive definite weight matrix of the V function in (10) is given by

$$P = \text{diag}(p_{11}, p_{22}, p_{33}, p_{44}) \ . \tag{14}$$

Then, the stability condition becomes

$$\begin{aligned} &\dot{V}(x_e(t+T_p)) + F(t+T_p) \\ &\quad = p_{11} x_{eT} \dot{x}_{eT} + p_{22} y_{eT} \dot{y}_{eT} + p_{33} \theta_{eT} \dot{\theta}_{eT} + p_{44} \eta_{eT} \dot{\eta}_{eT} \\ &\qquad + F(t+T_p) \\ &\quad = p_{11} x_{eT} u_1^L + p_{22} y_{eT} v_m \sin\theta_{eT} + p_{33} \theta_{eT} u_2^L + p_{44} \eta_{eT} u_3^L \\ &\qquad + F(t+T_p) \\ &\quad = p_{11} x_{eT} u_1^L + p_{22} y_{eT} v_m \sin\theta_{eT} + p_{33} \theta_{eT} u_2^L + p_{44} \eta_{eT} u_3^L \\ &\qquad + q_{11} x_{eT}^2 + q_{22} y_{eT}^2 + q_{33} \theta_{eT}^2 + q_{44} \eta_{eT}^2 \\ &\qquad + r_{11} u_1^{L^2} + r_{22} u_2^{L^2} + r_{33} u_3^{L^2} . \end{aligned} \tag{15}$$

Substituting the terminal state feedback controller (12) into (15), we get

$$\begin{aligned} \dot{V}(x_e(t+T_p)) + F(t+T_p) = &x_{eT}^2(-p_{11}\alpha + q_{11} + \alpha^2 r_{11}) \\ &+ \theta_{eT}^2(-p_{33}\beta + q_{33} + \beta^2 r_{22}) \\ &+ \eta_{eT}^2(-p_{44}\gamma + q_{44} + \gamma^2 r_{33}) \\ &+ p_{22} y_{eT} v_m \sin\theta_{eT} + q_{22} y_{eT}^2 . \end{aligned} \tag{16}$$

Similar to [20], to have a negative derivative of the value function, the following conditions for the weight parameters are required:

$$\begin{aligned} p_{11}\alpha - q_{11} - \alpha^2 r_{11} &\geq q_{22} \\ p_{33}\beta - q_{33} - \beta^2 r_{22} &\geq 0 \\ p_{44}\gamma - q_{44} - \gamma^2 r_{33} &\geq 0 \end{aligned} \tag{17}$$

and the terminal-state region is defined as follows:

$$\begin{aligned} |x_{eT}| &\geq |y_{eT}| \\ p_{22} y_{eT} v_m \theta_{eT} &< 0 \end{aligned} \ . \tag{18}$$

The terminal-state region is further bounded by the control signal region. From (5) and (12), we have the following results at the terminal state:

$$\begin{bmatrix} u_1^L \\ u_2^L \\ u_3^L \end{bmatrix} = \begin{bmatrix} -\alpha x_{eT} \\ -\beta \theta_{eT} \\ -\gamma \eta_{eT} \end{bmatrix} = \begin{bmatrix} -\dot{s}^L + v_m^L \cos\theta_{eT} \\ \omega_m^L - \dot{s}^L \kappa \\ a_m^L - \dot{v}_d \end{bmatrix} . \tag{19}$$

Then, the system control inputs $\dot{s}^L$, ω_m^L, a_m^L at the terminal state can be obtained by

$$\begin{bmatrix} \dot{s}^L \\ a_m^L \\ \omega_m^L \end{bmatrix} = \begin{bmatrix} \alpha x_{eT} + v_m \cos\theta_{eT} \\ \beta \theta_{eT} + \dot{s}\kappa \\ -\gamma \eta_{eT} + \dot{v}_d \end{bmatrix} , \tag{20}$$

with the control input constraints

$$\begin{bmatrix} \omega_{\min} \\ a_{\min} \end{bmatrix} \leq \begin{bmatrix} \omega_m^L \\ a_m^L \end{bmatrix} \leq \begin{bmatrix} \omega_{\max} \\ a_{\max} \end{bmatrix} . \tag{21}$$

A. Time Parameterized Reference

In the trajectory tracking problem, the vehicle is required to track a time parameterized reference. We normally feed a desired posture $\mathbf{x}_{d,t}$ to a tracking controller. In this work, we wish to combine trajectory tracking behaviors in a path following control law, thus achieving smooth spatial convergence to the trajectory as well as time convergence. We penalize the cost function with

$$F_t = (\mathbf{x}_m(T_p) - \mathbf{x}_{d,t}(T_p))^T K_t (\mathbf{x}_m(T_p) - \mathbf{x}_{d,t}(T_p)), \tag{22}$$

where K_t is a positive definite matrix. This matrix weighs the relative importance of convergence in time over spatial convergence to the path. If $K_t = 0$ is chosen, pure path following is achieved.

B. Obstacle Avoidance

Typically, the desired reference is generated by a planning algorithm based on a map of the environment and this reference is assumed to be collision-free. During the actual motions it is possible that obstacles appear in the vehicle's path, which had not been present in the planning phase. This may also happen because of imprecision in the map, or vehicle localization errors. In this work, we assume that the simulated sensors mimic infra-red sensors placed in a ring around the robot, spaced by 30° and they have a distance range of 50 cm. The obstacle information is then incorporated into the cost function, so that the computed control follows the desired reference, while staying away from the obstacles. In case of moving obstacles, the information such as their position and velocity can be used to predict the information over the next T_p horizon and then the cost function can be computed. It has to be noted that we consider only convex polygonal obstacles.

The obstacle points detected by sensors contribute to the cost function with a term which penalizes states as follows

$$F_{obs} = \sum_{i=1}^{N_p} \sum_{j=1}^{N_s} K_{obs} \frac{e^{-c_1 |\theta_{obs,ij}|}}{e^{c_2 d_{obs,ij}}}, \tag{23}$$

where K_{obs}, c_1, and c_2 are positive constants. N_s is the number of range sensors. N_p is the number of predictive steps, given by $N_p = T_p/\delta$, where δ is the sampling time. θ_{obs} is the angle of the obstacle with respect to the robot frame and d_{obs} is the distance between the robot and the obstacle.

IV. Simulation Results

Our NMPC controller was first implemented in Matlab and numerous simulations were performed. All the elements of our NMPC framework were set as follows:

$Q = \text{diag}(0.2, 2, 0.01, 0.01)$,
$R = \text{diag}(0.0001, 0.0001, 0.0001)$,
$P = \text{diag}(1, 1, 0.015, 0.015)$,
$K_t = \text{diag}(1, 2, 0.01, 0.01)$,
$K_{obs} = 1.5, \quad c_1 = 0.01, \quad c_2 = 10$,
$N_p = 3, \quad T_c = T_p = 0.15$ s, $\quad \delta = 0.05$ s, $\quad s(0) = 0$ m,
$\alpha = 2.5, \quad \beta = 1, \quad \gamma = 1$.

The circle reference is

$$x_d(t) = R\cos\frac{s(t)}{R}, \; y_d(t) = R\sin\frac{s(t)}{R},$$

where $R = 1$ m and the desired translation and rotation were $v_d = 0.5$ m/s and $\omega_d = 0.5$ rad/s, respectively. The maximum and minimum control inputs were set to $v_{\text{max}} = 2$ m/s, $v_{\text{min}} = -2$ m/s, $\omega_{\text{max}} = 2$ rad/s, $\omega_{\text{min}} = -2$ rad/s, $a_{\text{max}} = 2$ m/s^2 and $a_{\text{min}} = -2$ m/s^2.

The performance achieved with pure path following, pure trajectory tracking (see [20] for details), and for combined trajectory tracking and path following was assessed. Figure 2(a) and Figure 2(b) show the simulation results of pure path following control and pure trajectory tracking control, respectively, with four different initial postures. The velocities and the posture errors of pure path following are depicted in Figure 3(a) and Figure 4(a), respectively, while those of pure trajectory tracking are plotted in Figure 3(b) and Figure 4(b), respectively when the initial posture of both cases was set to $(1.5, -0.5, \pi)$. Obviously, the control signals of path following control were less likely pushed to saturation and motions were less aggressive. However, time constraints were not achieved in the path following control. Figure 2(c) shows the simulation results of the combination of path following control and trajectory tracking control, and the velocities and the posture errors are shown in Figure 3(c) and Figure 4(c), respectively when the initial posture was set to $(1.5, -0.5, \pi)$. This controller can achieve both reference convergence and time convergence with smooth motions. As seen in the results, the robot converged smoothly to the desired path and then it reacted to achieve zero trajectory tracking error. Interesting enough, in case of path following control, the predicted system states can reach the terminal region in finite time with less effort than in case of trajectory tracking control because the trajectory is the time-parameterized reference and the conditions in (18), depending on time, need to be satisfied.

Next, a convex polygonal obstacle was introduced in a position which prohibited path following. As it is shown in Figure 5, the controller deviated from the desired reference in order to safely avoid the obstacle and time convergence could still be achieved. In Figure 6, 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 could find a way to stay away from the obstacles and to follow the reference.

V. Real-World Experiments

A unicycle-type mobile robot, shown in Figure 1(a) was used in real-world experiments in this paper. The robot controller is an ATMEGA644 microprocessor with 64 KB flash program memory, 16MHz clock frequency and 4 KB SRAM. The robot orientation was measured by a Devantech CMPS03 compass. 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, but with $v_d = 0.2$ m/s, $\omega_d = 0.2$ rad/s and $\delta = 0.1$ s. The free package DONLP2 [23] was used to solve the online optimization problem. However, incorporating the terminal penalty and the terminal-state constraints degrades performance because of high computational time. Therefore we did not include them in the cost function in the real-world experiments. However, we still penalized the cost function with the time dependent penalty

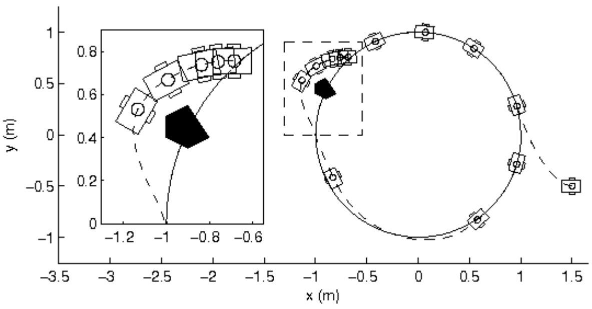


Fig. 5. The simulation results when a static polygonal obstacle was present.

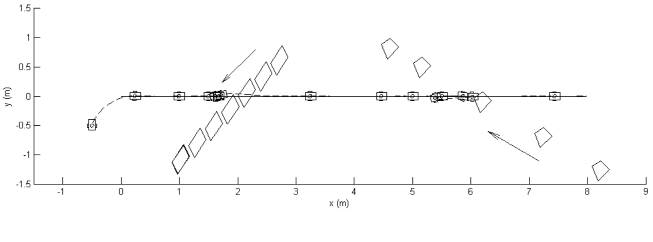
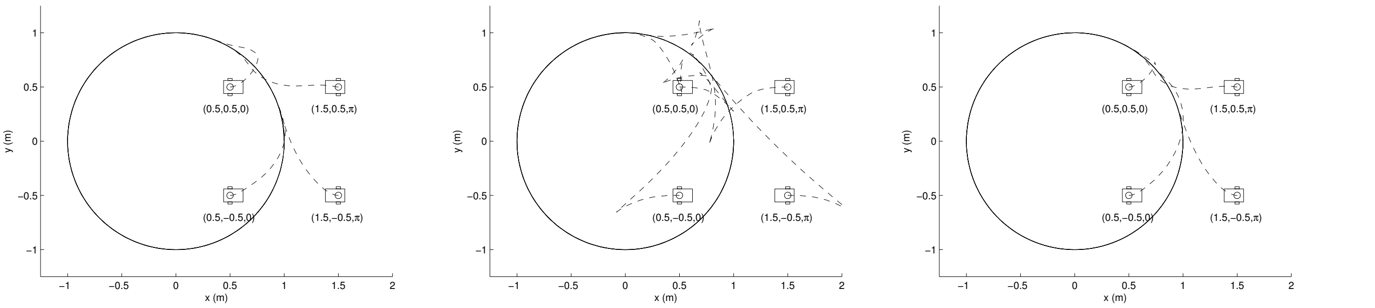
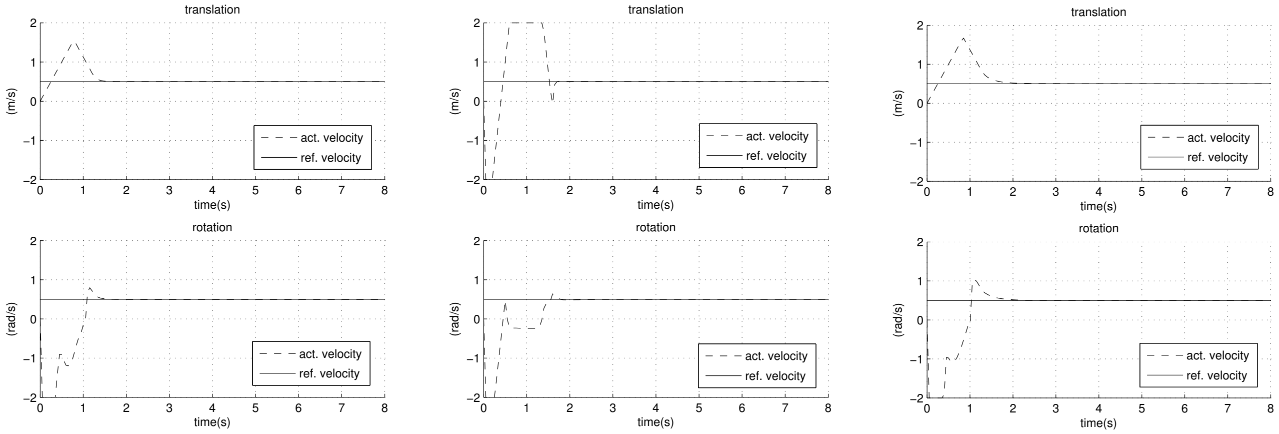


Fig. 6. The simulation results when two moving polygonal obstacles were present.



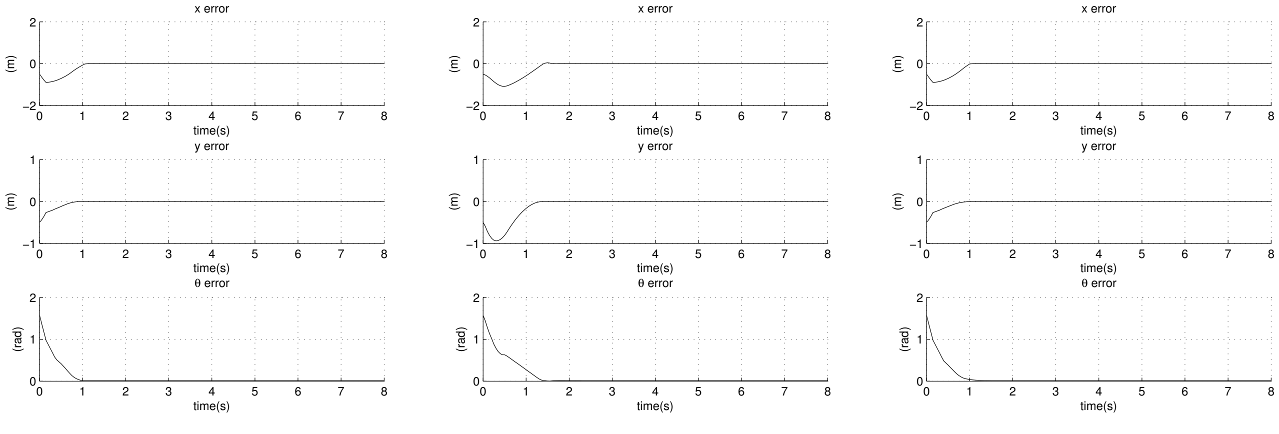
(a) (b) (c)

Fig. 2. The simulation results with four different initial postures: (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking.



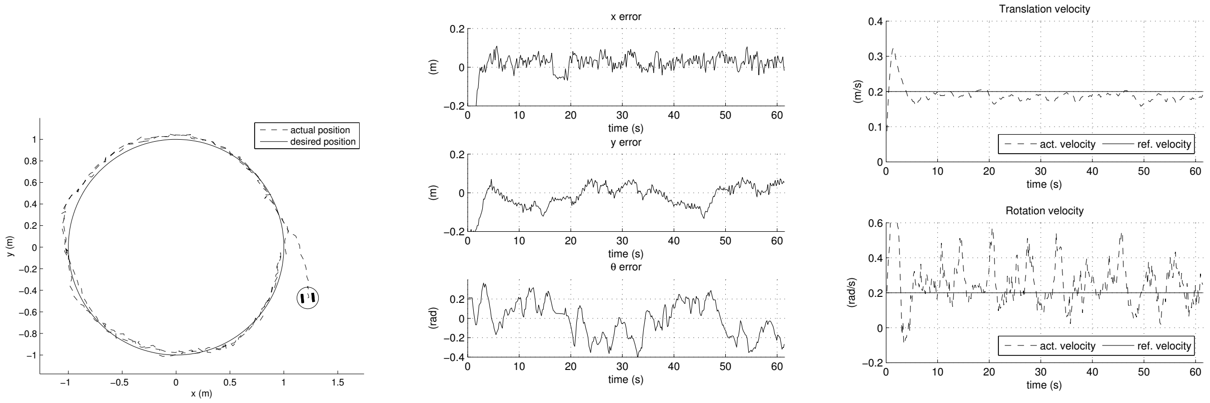
(a) (b) (c)

Fig. 3. The robot velocities when the initial posture was set to $(1.5, -0.5, \pi)$: (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking.



(a) (b) (c)

Fig. 4. The posture errors when the initial posture was set to $(1.5, -0.5, \pi)$: (a) pure path following, (b) pure trajectory tracking, and (c) the combination of path following and trajectory tracking.



(a)

(b)

(c)

Fig. 7. The experimental results by using our NMPC law: (a) the robot positions and its reference, (b) the posture errors, and (c) the robot's velocities.

term in order to satisfy time constraints. Furthermore, obstacle avoidance will be considered in our future work because of high computational time demand under real-time constraints. The results are shown in Figure 7.

VI. Conclusions and Future Work

In this paper, we presented a solution to the problem of combined trajectory tracking and path following for a mobile robot. Our approach based on the NMPC framework can control a mobile robot to smoothly converge to a reference with time and control input constraints.

However, the computation is one of the problems to use NMPC in real-time systems. Improving the computation efficiency is still under our investigation. Since the initial feasibility to the optimization has been assumed in order that subsequent feasibility can be implied, feasibility analysis is one of our further research. In addition, we will extend our controller to accomplish coordination tasks with a larger number of mobile robots in a complex environment.

References

[1] P. Morin and C. Samson, "Motion control of wheeled mobile robot," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 799-826, Springer Berlin Heidelberg, 2008.

[2] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory* (R. W. Brockett, R. S. Millman, and H. J. Sussmann, eds.), pp. 181-191, Birkhauser, Boston, 1983.

[3] A. Micaelli and C. Samson, "Trajectory-tracking for unicycle-type and two-steering-wheels mobile robots," *Technical Report No. 2097*, INRIA, Sophia-Antipolis, Nov. 1993.

[4] C. Canudas de Wit, C. Samson, H. Khennouf, and O. J. Sørdalen, "Nonlinear control design for mobile robots," in *Recent trends in mobile robots* (Y. F. Zheng, ed.), vol. 11, pp. 121-156, World Scientific Publishing, 1993.

[5] C. Samson, "Control of chained systems: Application to path-following and time-varying point stabilization of mobile robots," *IEEE Trans. on Automatic Control*, vol. 40, no. 1, pp. 64-77, Jan. 1995.

[6] C. Altafini, "Following a path of varying curvature as an output regulation problem," *IEEE Trans. on Automatic Control*, vol. 47, no. 9, pp. 1551-1556, Sep. 2002.

[7] S. A. Al-Hiddabi and N. H. McClamroch, "Tracking and maneuver regulation control for nonlinear non-minimum phase systems: application to flight control," *IEEE Trans. on Control Systems Technology*, vol. 10, no. 6, pp. 780-792, 2002.

[8] P. Encarnação and A. Pascoal, "3D path following for autonomous underwater vehicle," in *Proc. of 39th IEEE Conference on Decision and Control CDC2000*, Sydney, Australia, Dec. 2000, pp. 2977-2982.

[9] D. Soeanto, L. Lapierre, and A. Pascoal, "Adaptive non-singular path-following, control of dynamic wheeled robots," in *Proc. of International Conference on Advanced Robotics*, Coimbra, Portugal, June 30 - July 3, 2003, pp. 1387-1392.

[10] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Trans. on Automatic Control*, vol. 46, no. 11, pp. 1777-1782, Nov. 2001.

[11] L. Lapierre, R. Zapata, and P. Lepinay, "Combined path-following and obstacle avoidance control of a wheeled robot," *International Journal of Robotics Research*, vol. 26, no. 4, pp. 361-376, 2007.

[12] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, pp. 27-35, March 1995.

[13] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive. control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205-1218, 1998.

[14] H. Michalska and D.Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 38, no. 11, pp. 1623-1633, Nov. 1993.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, pp. 789-814, 2000.

[16] A. Jadbabaie, J. Yu, and J. Hauser, "Stabilization receding horizon control of nonlinear systems: A control Lyapunov function approach," in *Proc. of American Control Conference*, San Diego, CA, 1999, pp. 1535-1539.

[17] W. F. Lages and J. A. V. Alves, "Real-time control of a mobile robot using linearized model predictive control," in *Proc. of 4th IFAC Symposium on Mechatronic Systems*, Heidelberg, Germany, Sep. 2006, pp. 968-973.

[18] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460-469, 2007.

[19] A. Ollero and O. Amidi, "Predictive path tracking of mobile robots. Application to the CMU Navlab," in *Proc. of International Conference on Advanced Robotics*, Pisa, Italy, Jun. 1991, pp. 1081-1086.

[20] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 743-749, Jul. 2006.

[21] R. Hedjar, R. Toumi, P. Boucher, and D. Dumur, "Finite horizon nonlinear predictive control by the Taylor approximation: application to robot tracking trajectory," *Int. Journal of Applied Mathematics and Computer Science*, vol. 15, no. 4, pp. 527-540, 2005.

[22] P Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. on Control Systems Technology*, vol. 15, no. 3, pp. 566-580, May 2007.

[23] P. Spellucci, "An SQP method for general nonlinear programs using only equality constrained subproblems," *Mathematical Programming*, vol. 82, no. 3, pp. 413-448, 1998.