Nonlinear Model Predictive Control of an Omnidirectional Mobile Robot

Xiang LI [a,1], Kiattisin KANJANAWANISHKUL [a] and Andreas ZELL [a]
[a] *Wilhelm-Schickard-Institute, Department of Computer Architecture, University of Tübingen, Sand 1, 72076 Tübingen, Germany*

Abstract. This paper focuses on motion control problems of an omnidirectional robot based on the Nonlinear Model Predictive Control (NMPC) method. Although NMPC has been studied in many mobile robots applications due to the advantages of taking the robot constraints into account and increasing the robot performance with future information, the high computational requirement makes NMPC difficult to be utilized in the real systems with fast sampling time. In order to reduce the computational effort, many works either eliminate the computations which are necessary to keep the control stability, or linearize the nonlinear models only with local stability. Lots of research only presents the simulation results with NMPC. The main contributions of this paper are not only to analyze and design NMPC controllers with guaranteed stability to nonlinear kinematic models, but also to show the feasibility of NMPC with a real fast moving omnidirectional robot.

Keywords. Nonlinear Model Predictive Control, Omnidirectional Robot, Path Following, Trajectory Tracking

Introduction

Recently, omnidirectional wheeled robots have received more attention in mobile robots applications, because they have full mobility in the plane, which means that they can move at each instant in any direction without any reorientation [5]. Unlike nonholonomic robots, such as car-like robots, having to rotate before implementing any desired translation velocity, omnidirectional robots have higher maneuverability and are widely used in dynamic environments, for example, in the middle-size league of the annual RoboCup competition.

Two fundamental motion control problems of mobile robots, trajectory tracking and path following, have been well studied. The aim of the trajectory tracking problem is to control robots to track a given trajectory parameterized in time t. In the path following problem, robots are required to converge to a reference path described by a parameter s. Normally, solving these problems is first to formulate error kinematic models, which present the errors between the real robot states and their desired states according to a reference path or trajectory. Then feedback controllers are designed to drive the errors to zero. With respect to the nonlinear characteristics of error kinematic models, many

[1]Corresponding Author: Xiang Li, Wilhelm-Schickard-Institute, Department of Computer Architecture, University of Tübingen, Sand 1, 72076 Tübingen, Germany; E-mail:xiang.li@uni-tuebingen.de.

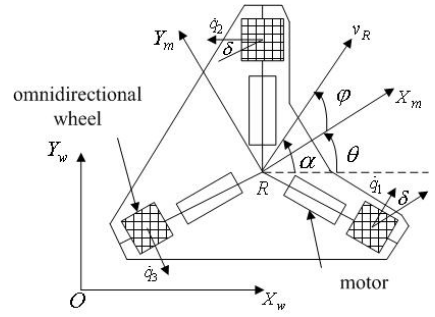

Figure 1. Kinematics diagram of the base of an omnidirectional robot

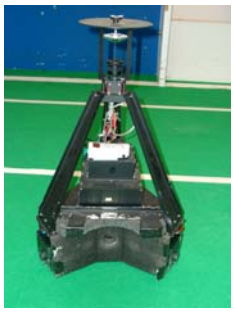

Figure 2. The real omnidirectional robot

nonlinear controllers have been presented ([1,2,6,7,9,17,18]). However, many methods rarely take the robot constraints into account, which are the crucial factors capable of degrading the robot performance, even destroying the control stability ([11,19]). Moreover, only the errors between the current robot states and the desired states are considered in most control laws, which ignore the potential opportunity of improving the control performance by considering more information of the given path or trajectory.

Motivated by the above considerations, we apply Nonlinear Model Predictive Control (NMPC) to solve the path following and trajectory tracking problem of an omnidirectional robot. As NMPC can easily take robot constraints into account, and utilize the future information to get current control inputs, it has been used in many robotics applications. Because the high computational requirement of NMPC, some works eliminate the computations which are necessary to keep the control stability ([4,13,14]). Some methods linearize the error kinematics to reduce the computational effort in NMPC, but they can only guarantee local stability ([3,20]). Many researches present detailed analysis of NMPC with mobile robots, but the applications only in simulation ([10,15,16]). The main contributions of this paper are to analyze and design NMPC controllers with respect to nonlinear kinematic models with guaranteed stability, and to show the feasibility of applying NMPC on a real omnidirectional robot.

1. Robot Kinematic Model

Figure 1 shows the base of our omnidirectional robot. Besides the fixed world coordinate system $\{W\}$ composed of axes X_w and Y_w, a moving robot coordinate system $\{M\}$ consisting of axes X_m and Y_m is defined. Angle θ between the axis X_m and X_w denotes the robot orientation. Angles α and φ denote the robot moving direction in the world and robot coordinate systems, respectively. Each wheel has the same distance L_w to the robot's center of mass R. δ refers to the wheel orientation in the robot coordinate system and is equal to 30 degrees. The kinematic model of the robot is as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}, \tag{1}$$

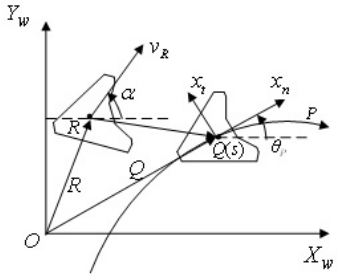

Figure 3. Path following problem with the desired position determined by parameter s.

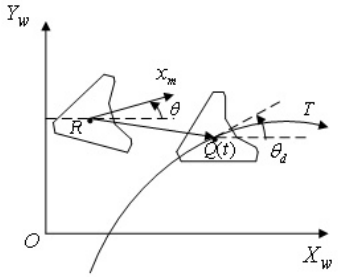

Figure 4. Trajectory tracking problem with the desired position determined by time t.

where $\mathbf{x}$ is the robot state vector with respect to the world coordinate system, which is composed of the robot position x, y and the robot orientation θ; The inputs include the robot rotation velocity ω and the robot translation velocities u and v with respect to the axes X_m and Y_m of the robot coordinate system, respectively.

When we consider the wheel velocities, the lower level kinematic model of the robot can be deduced as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos(\theta+\delta) & \sin(\theta+\delta) & L_w \\ -\cos(\theta-\delta) & -\sin(\theta-\delta) & L_w \\ \sin\theta & -\cos\theta & L_w \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}, \tag{2}$$

where $\dot{\mathbf{q}}$ is the vector of wheel velocities $[\dot{q}_1\ \dot{q}_2\ \dot{q}_3]^T$, and $\dot{q}_i (i = 1, 2, 3)$ denotes the i-th wheel velocity, which is equal to the wheel's radius multiplied by the wheel's angular velocity. As the motor's voltage and current are magnitude limited, the maximum wheel velocity is limited by $\dot{q}_m$, namely $|\dot{q}_i| \leq \dot{q}_m$.

It is important to notice that the transformation matrices in the above kinematic models are all full rank, which denotes that the translation and rotation of the omnidirectional robot are decoupled, and guarantees the separate control of these two movements.

2. Problem Formulation

The path following and trajectory tracking problems are illustrated in Figures 3 and 4, respectively. P and T denote the given path and trajectory, respectively. Point Q is the desired point of the robot. v_R refers to the robot translation velocity. The main difference between these two problems is that the desired position of the robot are determined by the parameter s and time t, respectively.

2.1. Path Following Problem

In the path following problem, a path coordinate system $\{P\}$ is introduced, which moves along the path P. The coordinate axes x_t and x_n direct the tangent and normal directions at point Q, respectively. θ_P denotes the path tangent direction at point Q. If we use vectors $\mathbf{R}$ and $\mathbf{Q}$ to describe the positions of R and Q in the world coordinate system, ${}^wR_m$ and ${}^wR_p$ to present the transformation matrices from $\{M\}$ to $\{W\}$ and from $\{P\}$ to $\{W\}$, respectively, we can get the following equation:

$$\mathbf{R} = \mathbf{Q} +^{w} R_p(\begin{matrix} x_e \\ y_e \end{matrix}), \tag{3}$$

where x_e and y_e denote the robot positions with respect to the path coordinate system $\{P\}$. With derivatives and some simple calculations, the error kinematic model can be deduced as,

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\alpha}_e \end{bmatrix} = \begin{bmatrix} (y_e c(s) - 1)\dot{s} + v_R \cos\alpha_e \\ -x_e c(s)\dot{s} + v_R \sin\alpha_e \\ \dot{\alpha} - c(s)\dot{s} \end{bmatrix}, \tag{4}$$

where the error vector $\mathbf{x}_e$ is with respect to the path coordinate system $\{P\}$, α_e presents the angular error between the robot moving direction α and the path tangent direction θ_p, the $c(s)$ denotes the path curvature at point Q, $\dot{\alpha}$ is the corresponding angular velocity of α.

It is noticed from Eq. (4) that the controlling of v_R is decoupled from controlling of $\dot{s}$ and $\dot{\alpha}$, because the errors can stay on the equilibrium ($\mathbf{x}_e = 0$) when $\dot{s}$ approaches v_R. Therefore, the path following problem is to find suitable values of $\dot{s}$ and $\dot{\alpha}$ driving errors x_e, y_e and α_e to zero with v_R to be steered to track any desired velocity.

2.2. Trajectory Tracking Problem

Unlike the path following problem, the desired translation velocity of the robot can not be chosen freely in the trajectory tracking problem, but is determined by the reference trajectory parameterized in time t. Therefore, the trajectory tracking problem requires the robot to track the specified position and velocity defined by each given time. Similar to the robot model (1), a reference trajectory T for an omnidirectional robot can be described as follows,

$$\dot{\mathbf{x}}_d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos\theta_d & -\sin\theta_d & 0 \\ \sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_d \\ v_d \\ \omega_d \end{bmatrix}, \tag{5}$$

where the reference state $\mathbf{x}_d = [x_d, y_d, \theta_d]^T$ and corresponding reference input velocities u_d, v_d and ω_d are determined by time t.

If we control the robot orientation to track the tangent direction of the reference trajectory, namely $v_d = 0$, and the robot coordinate system as the reference frame, the following error kinematic model of the trajectory tracking problem can be deduced by combining models (1) and (5):

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - u + u_d \cos\theta_e \\ -\omega x_e - v + u_d \sin\theta_e \\ \omega_d - \omega \end{bmatrix}, \tag{6}$$

where the error vector $\mathbf{x}_e$ is with respect to the robot coordinate system $\{M\}$, x_e and y_e denote the distance error between the robot and the trajectory T, θ_e presents the orientation error between the robot orientation θ and the trajectory tangent direction θ_d.

With respect to the error model (6), the objective of trajectory tracking is to choose suitable robot inputs u, v and ω, such as the tracking error $\mathbf{x}_e$ converges to zero.

3. Nonlinear Model Predictive Control

As an attractive optimal control method, Nonlinear Model Predictive Control has been used in our control problems, because it can easily handle the system constraints and take future information into the controller design ([8]).

With respect to a normal nonlinear system described by the following differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \;\; \text{subject to} \;\; \mathbf{u}(t) \in U, \;\; \mathbf{x}(t) \in X, \;\; \forall t \geq 0, \tag{7}$$

where $\mathbf{x}(t) \subseteq R^n$ and $\mathbf{u}(t) \subseteq R^m$ are the n dimensional state vector and m dimensional input vector, respectively, the sets X and U include the feasible states and inputs, especially the system equilibrium ($\mathbf{x}(t) = 0$ and $\mathbf{u}(t) = 0$), the basic idea of NMPC is to execute the following steps :

1. predict the system's future behavior over a prediction horizon T_p at each time step t;
2. find optimal inputs $\bar{u}(\cdot)\colon \; [t, t+T_p] \rightarrow U$ to minimize the value of the following objective function,

$$\mathbf{J}(t, \mathbf{x}(t), \bar{\mathbf{u}}(\cdot)) = \int_t^{t+T_p} \mathbf{F}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau, \tag{8}$$

   subject to:

   - $\dot{\bar{\mathbf{x}}}(\tau) = \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)), \; \bar{\mathbf{x}}(0) = \mathbf{x}(0),$
   - $\bar{\mathbf{u}}(\tau) = \bar{\mathbf{u}}(\tau + T_c), \; \forall \tau \in [t + T_c, t + T_p]\,,$
   - $\bar{\mathbf{u}}(\tau) \in U, \;\; \bar{\mathbf{x}}(\tau) \in X, \;\; \forall t \in [t, t + T_p]\,,$

   where T_c is the control horizon with $T_c \leq T_p$, $\mathbf{F}$ is the cost function specifying the desired control performance, the bar denotes the predicted values in the future, which are not same as the real values;
3. take the first optimal input value $\bar{\mathbf{u}}(t)$ as the current input.

It is well known that the above finite horizon strategy can not guarantee closed-loop stability. In many proposed methods, adding a terminal penalty $E(\,\bar{\mathbf{x}}(t + T_p)$ to the objective function and constraints $\bar{\mathbf{x}}(t + T_p) \in \mathbf{\Omega}$ to the terminal state is a computationally feasible method to achieve closed-loop stability ([8,10]), where the objective function becomes

$$\mathbf{J}(t, \mathbf{x}(t), \bar{\mathbf{u}}(.)) = \int_t^{t+T_p} \mathbf{F}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau + E(\bar{\mathbf{x}}(t + T_p)). \tag{9}$$

The following stability theorem presented in [10] provides a way to find the suitable terminal penalty and constraints,

Theorem Suppose

- the cost function is continuous with $\mathbf{F}(\mathbf{0}, \mathbf{0}) = 0$ and $\mathbf{F}(\mathbf{x}, \mathbf{u}) > 0$ for every state $\mathbf{x}$ and $\mathbf{u}$,

- the open-loop optimization problem has a solution at time $t = 0$,
- for a terminal penalty $E(\mathbf{x})$ with $E(\mathbf{0}) = 0$, and a closed region $\Omega \subseteq X$ including the origin, if there is a control law $\mathbf{k}(\mathbf{x}) \in U$ with $\mathbf{k}(\mathbf{0}) = 0$ such that

$$\dot{E}(\mathbf{x}) + \mathbf{F}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \leq 0, \forall \mathbf{x} \in \Omega, \tag{10}$$

the former described NMPC method guarantees asymptotical stability of the closed-loop system.

Although the high computational demands of solving the nonlinear finite optimization problem make NMPC hard to be implemented in applications with fast sampling time and limited computational resources ([8]), many research results show the possibility of applying nonlinear predictive controllers in some real-time processes ([13,12]). As one often used method, sequential quadratic programming (SQP) is utilized in our project to solve the online open-loop nonlinear optimization problem. We use the software *donlp2-intv-dyn* written by P. Spellucci ([21]), which is a general purpose nonlinear optimizer and can be found at *http://plato.la.asu.edu/donlp2.html*.

3.1. Path Following Control

As the translation and rotation of an omnidirectional robot are completely decoupled, the angular error α_e in Eq. (4) can be directly controlled. Therefore, the path following problem of an omnidirectional robot can be solved by finding suitable $\dot{s}$ and α_e. With the other freedom of controlling the robot orientation, we can drive the robot orientation θ to any desired orientation θ_d at the same time. Defining the orientation error θ_e, and a new vector $\mathbf{u}_e = [u_1, u_2, u_3]^T$, we get the following error kinematic model having the equilibrium at $\mathbf{x}_e = 0$ and $\mathbf{u}_e = 0$,

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & c(s)\dot{s} & 0 \\ -c(s)\dot{s} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -\dot{s} + v_R \cos\alpha_e \\ v_R \sin\alpha_e \\ \dot{\omega}_d - \omega \end{bmatrix}, \tag{11}$$

where $\theta_e = \theta_d - \theta$, ω_d denotes the desired rotation velocity.

As the errors are required to converge to zero, we select the following cost function:

$$F(\mathbf{x}, \mathbf{u}) = \mathbf{x}_e^T \mathbf{Q} \mathbf{x}_e + \mathbf{u}_e^T \mathbf{R} \mathbf{u}_e + g(\mathbf{x}_e(t + T_p)), \tag{12}$$

with positive weight matrices

$$\mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_{11} & 0 & 0 \\ 0 & r_{22} & 0 \\ 0 & 0 & r_{33} \end{bmatrix}, \tag{13}$$

where $q_{11} > 0, q_{22} > 0, q_{33} > 0$ and $r_{11} > 0, r_{22} > 0, r_{33} > 0$.

To guarantee the control stability, the following Lyapunov function can be selected as the terminal penalty:

$$g(\mathbf{x}_e(t+T_p)) = \frac{1}{2}\mathbf{x}_e(t+T_p)^T\mathbf{x}_e(t+T_p), \tag{14}$$

where $\mathbf{x}_e(t+T_P) = [x_{eT_P}, y_{eT_P}, \theta_{eT_P}]^T$ denotes the terminal state.

When we choose the terminal feedback controller $\mathbf{u}^L = [u_1^L, u_2^L, u_3^L]^T$ as:

$$\begin{array}{c} u_1^L = -\alpha x_{eT_P}, \\ u_2^L = -\beta y_{eT_P}, \\ u_3^L = -\gamma\theta_{eT_P} \end{array} \tag{15}$$

with parameters $\alpha \geq 0$, $\beta \geq 0$, and $\gamma \geq 0$, the stability condition (10) becomes

$$\begin{aligned} \dot{g}(x_e(t+T)) + F(t+T) =& x_{e_T}^2(-\alpha + q_{11} + \alpha^2 r_{11}) \\ &+ y_{e_T}^2(-\beta + q_{22} + \beta^2 r_{22}) \\ &+ \theta_{e_T}^2(-\gamma + q_{33} + \gamma^2 r_{33}). \end{aligned} \tag{16}$$

To satisfy the stability condition (10), the following requirements are necessary:

$$\begin{array}{c} \alpha - q_{11} - \alpha^2 r_{11} \geq 0, \\ \beta - q_{22} - \beta^2 r_{22} \geq 0, \\ \gamma - q_{33} - \gamma^2 r_{33} \geq 0. \end{array} \tag{17}$$

Furthermore, the outputs of feedback controllers (15) have to satisfy the system constraints, which are the bounded wheel velocities in our case. With the definitions of the input vector $\mathbf{u}_e$, terminal feedback controller $\mathbf{u}^L$, and the robot kinematic model (1), we get the second part of terminal constraints as

$$-\begin{bmatrix} \dot{q}_m \\ \dot{q}_m \\ \dot{q}_m \end{bmatrix} \leq \begin{bmatrix} \cos(\theta+\delta) & \sin(\theta+\delta) & L_w \\ -\cos(\theta-\delta) & -\sin(\theta-\delta) & L_w \\ \sin\theta & -\cos\theta & L_w \end{bmatrix} \begin{bmatrix} -\alpha x_{eT} + \dot{s} \\ -\beta y_{eT} \\ -\gamma\theta_{eT} + \omega_d \end{bmatrix} \leq \begin{bmatrix} \dot{q}_m \\ \dot{q}_m \\ \dot{q}_m \end{bmatrix}, \tag{18}$$

3.2. Trajectory Tracking Control

With the same idea of solving the path following problem, we transfer the error kinematic model of the trajectory tracking problem (6) by introducing the following control variables u_1, u_2 and u_3:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -u + u_R\cos\theta_e \\ -v + u_R\sin\theta_e \\ \dot{\omega}_d - \omega \end{bmatrix}. \tag{19}$$

When we select the cost function, the terminal penalty and the feedback controller with the same forms as (12), (14) and (15), the corresponding terminal constraints can be deduced following the above process.

4. Experimental Results

The real-world experiments were made in our robot laboratory having a carpet covered field with a size of $5.1 \times 4.2\ m^2$. Figure 2 shows our middle-sized RoboCup robot, which is equipped with a Pentium-M 2GHz on-board PC with 1GB RAM. The wheels are driven by three $60W$ Maxon DC motors and have the maximum wheel velocity $\dot{q}_m = 1.9m/s$.

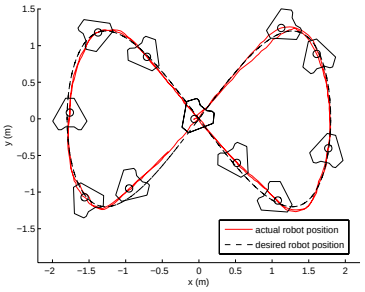

Figure 5. Reference trajectory and robot trajectory.

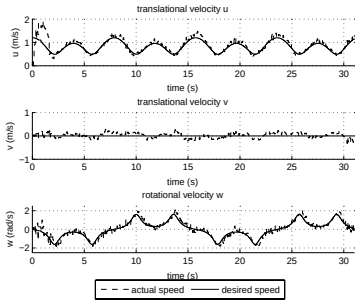

Figure 6. Velocities with respect to the robot frame.

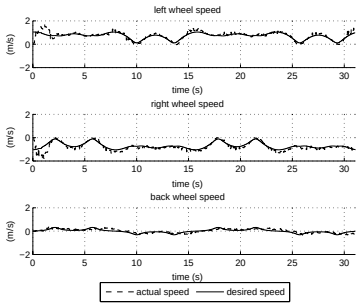

Figure 7. Wheel Velocities.

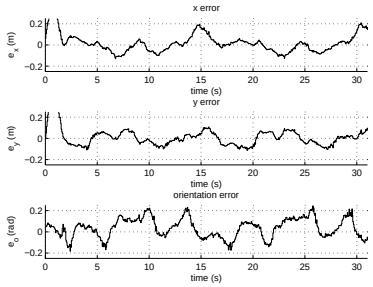

Figure 8. Tracking errors.

Two kinds of experiments were made to test the above NMPC method: one is the path following control with a constant desired velocity of $v_R = 1.0m/s$ and the desired robot orientation $\theta_d = \theta_P$, the other is the trajectory tracking control, where the desired velocity of v_R is changing with time t. An eight-shaped curve is adopted as the reference path and trajectory, because its geometrical symmetry and sharp changes in curvature make the test challenging. With respect to the world coordinate system, the coordinates of the reference path (x_P, y_P) and trajectory (x_T, y_T) are given by

$$\begin{cases} x_P(s) = 1.8\sin(s) \\ y_P(s) = 1.2\sin(2s) \end{cases} \quad \begin{cases} x_T(t) = 1.8\sin(0.4t) \\ y_T(t) = 1.2\sin(0.8t). \end{cases} \tag{20}$$

In the path following and trajectory experiments, we chose the same parameters as:

$$\mathbf{Q} = 0.5\mathbf{I}_3,\ \mathbf{R} = 0.1\mathbf{I}_3,\ \alpha = \beta = \gamma = 2,\ \tau = 0.07s,\ T_P = T_c = 3, v_R = 1.0m/s.$$

Figures 5 and 9 illustrate the results of trajectory tracking and path following experiments. Figures 6 and 10 show that the desired translation velocity is changing according

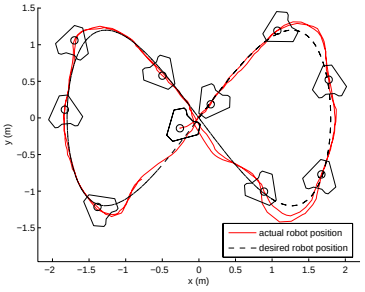

Figure 9. Reference path and robot path.

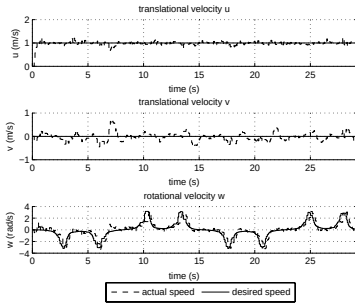

Figure 10. Velocities with respect to the robot frame.

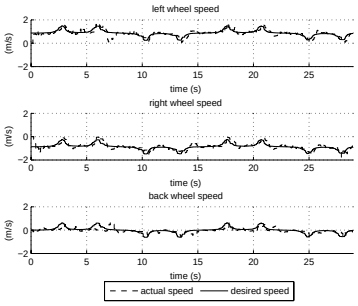

Figure 11. Wheel Velocities.

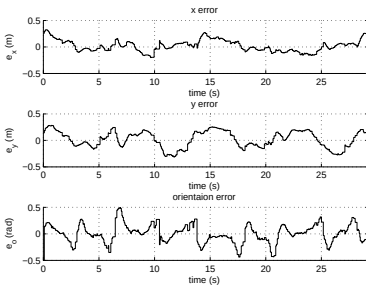

Figure 12. Following errors.

to the trajectory's curvature, but keeps constant in the path following problem, which increases the difficulty in following the sharp turning part of the given path. Figures 7 and 11 show that the controller guarantees the required wheel velocities under the boundaries. Figures 8, 12 show that the NMPC solved these real-time motion control problems with good performance, because at most time steps, the distance errors are less than 0.15m and 0.3m, the orientation errors are less than 0.2rad and 0.5rad in the trajectory tracking and path following tasks, respectively.

5. Conclusions

This paper focuses on motion control problems of an omnidirectional robot. Considering the constraints of robot systems and utilizing more known information to increase the robot performance, we use Nonlinear Model Predictive Control in this work. According to analyzing the path following and trajectory tracking problems, we formulate NMPC with respect to the corresponding error kinematic models, especially present details of choosing suitable terminal penalties and terminal constraints to guarantee the control stability. Experimental results with a real omnidirectional robot show that the Nonlinear Model Predictive Control not only can solve the path following and trajectory tracking problems with good performance, but also is feasible to be applied on a fast moving robot platform.

References

[1] A. P. Aguiar and J. ao Pedro Hespanha. Logic-based switching control for trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. In *Proceedings of the 2004 American Control Conference ACC*, Boston Massachusetts, USA, 2004.

[2] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotovic. Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239, 2005.

[3] M. Bak, N. K. Poulsen, and O. Ravn. Path following mobile robot in the presence of velocity constraints. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2001.

[4] A. Bauchspiess. A predictive algorithm with fine interpolation for vision-guided robots. In *Proceedings of the 5th IFAC Workshop*, pages 97–102, 1998.

[5] G. Campion, G. Bastin, and B. D'Andréa-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12, 1996.

[6] F. D. del Río, G. Jiménez, J. L. Sevillano, S. Vicente, and A. Civit-Balcells. A generialization of path following for mobile robots. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, volume 1, pages 7–12, 1999.

[7] M. Egerstedt, X. Hu, and A. Stotsky. Control of mobile platforms using a virtual vehicle approach. *IEEE Transactions on Automatic Control*, 46(11):1777–1782, 2001.

[8] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, Veldhoven, Netherlands, 2002.

[9] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Coordinated path following control of multiple wheeled robots with directed communication links. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

[10] D. Gu and H. Hu. Receding horizon tracking control of wheel mobile robots. *IEEE Transaction on Control Systems Technology*, 14(4), july 2006.

[11] G. Indiveri, J. Paulus, and P. G. Plöger. Motion control of swedish wheeled mobile robots in the presence of actuator saturation. In *Proceedings of the 10th annual RoboCup International Symposium*, 2006.

[12] T. Keviczky and G. J. Balas. Software-enabled receding horizon control for autonomous uav guidance. In *Proceeding of the AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, California USA, 2005.

[13] T. Keviczky, P. Falcone, R. Borrelli, J. Asgari, and D.Hrovat. Predictive control approach to autonomous vehicle steering. In *Proceedings of the 2006 American Control Conference ACC*, 2006.

[14] B. Kim, D. Necsulescu, and J. Sasiadek. Model predictive control of an autonomous vehicle. In *Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, volume 2, pages 1279–1284, 2001.

[15] F. Kühne, J. J.M.G. da Silva, and W. Lages. Mobile robot trajectory tracking using model predictive control. In *Proceedings of 2005 Latin-American Robotics Symposium*, 2005.

[16] F. Kühne, W. Lages, and J. J.M.G. da Silva. Point stabilization of mobile robots with nonlinear model predictive control. In *Proceedings of 2005 IEEE International Conference on Mechatronics and Automation*, volume 3, pages 1163–1168, 2005.

[17] X. Li and A. Zell. Motion control of an omnidirectional mobile robot. In *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics*, Angers, France, 2007.

[18] K. Maček, I. Petrović, and R. Siegwart. A control method for stable and smooth path following of mobile robots. In *Proceedings of the European Conference on Mobile Robots 2005*, 2005.

[19] A. Scolari Conceição, A. Moreira, and P. j. Costa. Trajectory tracking for omni-directional mobile robots based on restrictions of the motor's velocities. In *Proceedings of the 8th International IFAC Symposium on Robot Control*, 2006.

[20] M. Seyr and S. Jakubek. Mobile robot predictive trajectory tracking. In *Proceedings of 2005 ICINCO*, pages 112–119, 2005.

[21] P. Spellucci. An "sqp" method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82:413–448, 1998.