High-speed Trajectory Tracking Based on Model Predictive Control for Omni-directional Mobile Robots

Zhiwen Zeng, Huimin Lu, Zhiqiang Zheng
College of Mechanics and Automation, National University of Defense Technology, Changsha, 410073, China
E-mail: z7z7w7@gmail.com

Abstract: In dynamic environments, mobile robots sometimes are required to perform accurate and stable tracking of high-speed trajectories, but rare researches are reported, which specifically deal with high-speed tracking problem. When tracking high-speed trajectories, mobile robots usually approach the top speed and acceleration limits, which indicates that the kinematic constraints can not be ignored. In order to cope with such situations, a novel trajectory tracking controller using model predictive control has been studied. Where it differs from other control strategies lies in that this control law is generated based on predicating the evolution state of mobile robots and it is able to handle hard constraints directly in the optimization process, so that mobile robots can track trajectories both quickly and safely. In order to cut down the computational cost for on-line applications, Laguerre polynomials are used in the design of model predictive control to reduce the number of parameters for optimization. The proposed algorithm is applied on a real soccer robot, and the experimental results show that the robot can track high-speed trajectories effectively with small tracking errors and tiny computation time.

Key Words: Trajectory Tracking, High-speed, Model Predictive Control, Laguerre Polynomials

1 INTRODUCTION

Trajectory tracking is concerned with the design of control laws that force a mobile robot to converge to a time parameterized reference[1]. Given the applications of mobile robots in industrial domains as well as in military situations, these controllers are required to perform accurate and stable tracking of complicated and high-speed trajectories in dynamic environments. For example, the automated guided vehicles (AGVs) in industrial load transportation use are supposed to promote handling capacity with shorter transport time. Inspite of their prosperous prospect, rare researches which specially deal with high-speed tracking problem can be found in literatures. Part of the reason is that classical controllers typically base on the closed-loop feedback structures, calculating the control signals based on the error between the current state and the reference one. In other words, the corrective measures are taken only after the error has occurred, which is unsuitable for tracking high-speed trajectories[2]. For instance, a popular technique that "chase" a target point using a PID controller drives mobile robots to follow a point ahead on the trajectory, but the tracking errors increase with higher speed and shaper curves[3]. In the meantime, the controller outputs will reach the actuators' saturation limits in practical application in cases of high-speed tracking. When the robots reach the top velocity and acceleration limits, these constraints firmly demand attention.

Considering the problems mentioned, this paper proposes model predictive control to solve the trajectory tracking problems. An important advantage of this type of control lies in its ability to achieve optimal control signals, which refer to the future information from the whole trajectory and from predicting the state of mobile. This improvement allows action to be taken before the error occurs, thus the model predictive controller wins out over others in tracking high-speed trajectory. Besides, it can directly handle the hard constraints on controls and states in the optimizing process, so that a superb tracking performance can be achieved under both the velocity and acceleration constraints. Many works have studied on using model predictive control for tracking problem, eg. the implementations based on linear model are presented in [5][6][7][8], while the nonlinear control scheme for the tracking problem are proposed in [2][9][10][11][12][13]. Because of the high computational cost, many researches present detailed analysis of predictive controller applied on mobile robots, but most of them only presents the simulation results[2][10][11][12][13]. In order to practice the application of model predicative control, cutting down the computation cost is of great importance, especially for those robots of limited resources. In this paper, a strategy using model predictive control is proposed to track trajectories with high speed. The first step is to obtain the linear error dynamics model around the reference, and the second step is to employ the Laguerre polynomials in model predictive control design. The unique advantage of using Laguerre polynomials here aims to reduce the parameter variables to be optimized. Generally, the computational load is strictly related to the control horizon N_c whereas stability and performance can be governed using the prediction horizon N_p[14]. By substituting a long horizon with a small number of parameters, the computation cost can be cut down for on-line use. On the other hand, the stability has been proved when terminal state constraints are used in this design[15].

The controller presented in this paper has been developed in the application in the soccer robots of the RoboCup middle size league(MSL), which specifically deals with highly dynamic and uncertain environment. In RoboCup scenarios, each robot is supposed to track high-speed trajectories with sudden changes in direction and orientation where the robots have to elbow its way.

The rest of the paper is organized as follows: in Section II, an omnidirectional mobile robot is modeled and its constraints are determined, then the trajectory tracking problem is formulated; model predictive control based on Laguerre polynomials is developed in Section III; Section IV shows the experimental results; Finally, the conclusions and future work are discussed in Section V.

2 SYSTEM KINEMATIC MODEL AND PROBLEM FORMULATION

Omnidirectional mobile robots can travel in all directions at any time regardless of the orientation, which helps to decrease the complexity of control system and enables faster and more accurate movement. In the middle size league of the annual RoboCup game, we developed our omnidirectional mobile robots(shown in Fig. 1).

2.1 Kinematic Model

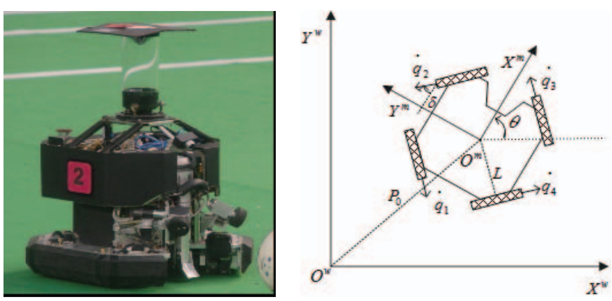

Figure 1: The model of omnidirectional mobile robot.

Omnidirectional mobile robot is formulated in the fixed coordinate system as shown in Fig. 1, both of the world coordinate system $\{X^wO^wY^w\}$ and body coordinate system $\{X^mO^mY^m\}$ are employed to describe the robot system. Angle θ between the axes X^w and X^m denotes the robot orientation; L represents the distance from the wheel to the mass point; δ refers to the wheel orientation in the body coordinate system and is equal to 45 degrees. The kinematic model of the robot can be described as follows:

$$\dot{X} = R(\theta)^T \begin{bmatrix} u \\ v \\ w \end{bmatrix}, R(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where $R(\theta)$ is the transformation matrix and $X = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ is the robot state vector, which is composed of the robot's position x, y and orientation θ under the world coordinate system. The inputs include the rotation velocity w and the translation velocities u, v with respect to the axes X^m and Y^m, respectively.

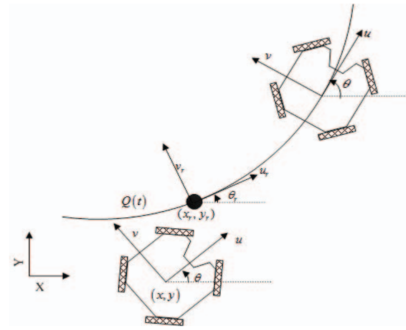

Figure 2: The formulation of trajectory tracking problem.

2.2 The Linear Error Dynamics Model

Trajectory tracking requires the robot to track the specified position and velocity defined at each given time. The reference trajectory which is parameterized in time t can be formulated as follows: $Q(t) = (X_r(t), Y_r(t))\,, t\epsilon\,[0, T]$. The description of trajectory tracking problem is shown in Fig. 2, where u_r refers to the tangential velocity and w_r denotes the angular velocity of the platform. Besides, v_r suggests the normal direction velocity and equals to zero.

According to the reference state vector $\begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T$, when mobile robot is driven on the a reference trajectory, the state-tracking error expressed in the frame of the body coordinate system can be formulated as:

$$X_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = R(\theta) \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{2}$$

Then taking the robot's kinematic model into account and deriving the relation (2), the following kinematic model can be obtained:

$$\begin{bmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{\theta_e} \end{bmatrix} = \begin{bmatrix} 0 & w & 0 \\ -w & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} 0 \\ u_r \sin\theta_e \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -u + u_r \cos\theta_e \\ -v \\ w_r - w \end{bmatrix} \tag{3}$$

Subsequently, by linearizing the error dynamics (3) around the reference trajectory $\begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T$, the following linear model results:

$$\begin{bmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{\theta_e} \end{bmatrix} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & u_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix},$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -u + u_r \cos\theta_e \\ -v \\ w_r - w \end{bmatrix} \tag{4}$$

The system is controllable as long as w_r and v_r are not equal to zero simultaneously.

Since constraints are imposed on almost each application, actuators are naturally limited on the force (or equivalent) they can offer. While tracking a trajectory, the robot's velocity commands should not exceed the max velocity constraints, so as the acceleration. In this study, the following constraints are considered:

$$\begin{bmatrix} -u_{max}+u_r(k)\cos\theta_e(k) \\ -v_{max} \\ w_r(k)-w_{max} \end{bmatrix} \leq \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix} \leq \begin{bmatrix} -u_{min}+u_r(k)\cos\theta_e(k) \\ -v_{min} \\ w_r(k)-w_{min} \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} (-a_{umax}\Delta t+u_r(k)\cos\theta_e(k)\cdots \\ \qquad -u_r(k-1)cos(\theta_e(k-1))) \\ -a_{vmax}\Delta t \\ w_r(k)-w_r(k-1)-a_{wmax}\Delta t \end{bmatrix} \leq \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_3(k) \end{bmatrix} \leq \begin{bmatrix} (-a_{umin}\Delta t+u_r(k)\cos\theta_e(k)\cdots \\ \qquad -u_r(k-1)cos(\theta_e(k-1))) \\ -a_{vmin}\Delta t \\ w_r(k)-w_r(k-1)-a_{wmin}\Delta t \end{bmatrix} \tag{6}$$

where Δt is the sampling period and $u_r(k)$ denotes the tangential velocity of the reference trajectory at time k; $w_r(k)$ refers to the reference rotation velocity; $\theta_e(k)$ is the angle error between the reference trajectory and the practical one; $u_r(k-1), w_r(k-1), \theta_e(k-1)$ represent the last step values.

2.3 CONTROLLER DESIGN

Model predictive control is suitable for tracking high-speed trajectories, because the controller enables quick response to the abrupt changes by optimizing future plant behaviour at each control interval and it ensures safety for robot driving at a high velocity under the kinematics constraints. However, the high computation cost of model predictive control makes it hard for on-line applications. In this paper, a set of Laguerre polynomials with small number of parameters are used to approximate the control trajectory, therefore reduce the computation time for real-time applications.

2.4 Introduction To Laguerre Polynomials

Laguerre polynomials offer many advantages such as good approximation capability for the variances of system time-delay, order and other structural parameters, low computational complexity, and the facility of online parameter identification[16].

The discrete-time Laguerre network can be shown as follows:

$$\Gamma_N(z) = \Gamma_{N-1}(z)\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}, \Gamma_1 = \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}} \tag{7}$$

where α called the scaling factor is the pole of the discrete-time Laguerre network, and $0 \leq \alpha < 1$ for stability of the network. N denotes the degree of the Laguerre network, $l_i(k)$ refers to the inverse z-transform of $\Gamma_i(z,\alpha)$. Laguerre polynomials can be expressed in a vector form as follows:

$$L(k) = \begin{bmatrix} l_1(k) & l_2(k) & \ldots & l_N(k) \end{bmatrix}^T \tag{8}$$

The set of discrete-time Laguerre polynomials have the following characteristics which are very useful in model predictive control design.

$$L(k+1) = AL(k) \tag{9}$$

$$\text{let} \quad \beta = 1-\alpha^2$$

$$A = \begin{bmatrix} \alpha & 0 & 0 & 0 & 0 \\ \beta & \alpha & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ (-1)^{N-1}\alpha^{N-3}\beta & \ldots & \beta & \alpha & 0 \\ (-1)^N\alpha^{N-2}\beta & (-1)^{N-1}\alpha^{N-3}\beta & \ldots & \beta & \alpha \end{bmatrix}$$

and the initial condition is given by

$$L(0) = \begin{bmatrix} 1 & -\alpha & \alpha^2 & -\alpha^3 & \ldots & (-1)^{N-1}\alpha^{N-1} \end{bmatrix}^T \tag{10}$$

Especially, when choosing the scale factor $\alpha = 0$, the Laguerre polynomials become a set of pulses

$$L(k) = \begin{bmatrix} \delta(k) & \delta(k-1) & \ldots & \delta(k-N+1) \end{bmatrix}^T \tag{11}$$

Another important characteristic is orthonormality, namely

$$\begin{cases} \sum_{k=0}^{\infty} l_i(k)l_j(k) = 0 \quad ,i \neq j \\ \sum_{k=0}^{\infty} l_i(k)l_j(k) = 1 \quad ,i = j \end{cases} \tag{12}$$

Suppose that the impulse response of a stable system in $H(k)$, then with a given number of N, $H(k)$ can be written as:

$$H(k) = c_1 l_1(k) + c_2 l_2(k) + \ldots + c_N l_N(k) \tag{13}$$

where $c_1, c_2, \ldots, c_N$ are the coefficients determined by the system.

2.5 Model Predictive Control Based on Laguerre Polynomials

The augmented model of the state-space model $\begin{pmatrix} A_m & B_m & C_m \end{pmatrix}$ will be used in the design of predictive control, which has the following form:

$$\begin{cases} x(k+1) = Ax(k) + B\Delta u(k) \\ y(k) = Cx(k) \end{cases} \tag{14}$$

$$A = \begin{bmatrix} A_m & O_m^T \\ C_m A_m & 1 \end{bmatrix}, B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}, C = \begin{bmatrix} O_m^T & 1 \end{bmatrix}$$

$$\Delta u(k) = u(k) - u(k-1) \tag{15}$$

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \tag{16}$$

$$x(k) = \left[\begin{array}{cc} \Delta x_m(k)^T & y(k) \end{array} \right]^T \quad (17)$$

at time k_i, the future control trajectory is

$$\Delta U = \left[\begin{array}{cccc} \Delta u(k_i) & \Delta u(k_i+1) & \ldots & \Delta u(k_i+N_c-1) \end{array} \right]^T \quad (18)$$

N_c denotes the control horizon. At time k_i, any elements within ΔU can be represented using the descrete δ-function in the conjuction with ΔU

$$\Delta u(k_i{+}i) = \left[\begin{array}{cccc} \delta(i) & \delta(i-1) & \ldots & \delta(i-N_c+1) \end{array} \right] \Delta U \quad (19)$$

When model predictive control is employed in real-time applications such as for implementing on the source-limited robots, the computation cost should not be ignored. The procedure of this algorithm is very time-consuming, so that most of the reported researches only present the simulation results. As we known, the control horizon N_c is a basic tuning parameter, and the computation time of model predictive control is strictly related with the length of the control horizon. To reduce the computation cost, a feasible way is to decrease the number of parameters being optimized. Aimed at this objective, Laguerre polynomials are employed to design the model predictive controller. By using our algorithm, a long control horizon can be realized with a small number of parameters thus the computation time is cut down.

Combining (13) and (19), $\Delta u(k_i+i)$ can be expressed as follows:

$$\Delta u(k_i+i) = \sum_{j=1}^{N} c_j(k_i) l_j(i) \quad (20)$$

namely

$$\Delta u(k_i+i) = L(i)^T \eta \quad (21)$$

$$\eta = \left[\begin{array}{cccc} c_1 & c_2 & \ldots & c_N \end{array} \right]$$

$L(i)^T$, η refer to the transposed Laguerre polynomials vector and the Laguerre coefficients vector. Within this design framework, the control horizon N_c is substituted by the number of terms N in conjunction with the parameter α. To achieve a long control horizon with a smaller number of parameters N, a larger α should be selected. It's worth noticing that, when $\alpha = 0, N = N_c$, this approach turns back to be the traditional model predictive control. By choosing a proper α and N, the dynamic performance of the controller can be guaranteed, and the number of decision variables being optimized can be reduced as much as possible. With a small number of parameters, the computation time can be cut down for on-line applications. From (14) and (21), by using the input signal with the initial state variable information $x(k_i)$, the prediction of the future state variable $x(k_i+m|k_i)$ at sampling instant m, becomes

$$x(k_i+m|k_i) = A^m x(k_i) + \sum_{i=0}^{m-1} A^{m-i-1} B L(i)^T \eta \quad (22)$$

$$y(k_i+m|k_i) = CA^m x(k_i) + \sum_{i=0}^{m-1} CA^{m-i-1} B L(i)^T \eta \quad (23)$$

In our case, the velocity and acceleration constraints are directly handled in the optimizing computation, and the terminal state constraint is employed to guarantee the global asymptotic stability. The following equation describes the cost function which is used to minimize the state error and control input.

$$J = \sum_{m=1}^{N_p} x(k_i{+}m|k_i)^T Q x(k_i{+}m|k_i) + \Delta U^T \overline{R} \Delta U \quad (24)$$

subject to:

$$x(k_i+N_p|k_i) = 0$$
$$\Delta U^{min} \le \Delta U \le \Delta U^{max}$$
$$U^{min} \le U \le U^{max}$$

where N_p denotes the prediction horizon; meanwhile, Q, R are weighting matrices, with $Q \ge 0$ and $R > 0$. We define:

$$\Delta u(k_i+m) = 0, N_c \le m \le N_p - 1 \quad (25)$$

Notice that:

$$\Delta U^T \overline{R} \Delta U = \sum_{m=0}^{N_p} \Delta u(k_i+m)^T r_w \Delta u(k_i+m) \quad (26)$$

where r_w is the diagonal element of R. According to the (12) and (21), with a sufficiently large of prediction horizon N_p, the cost function (24) can be reconstructed as follows:

$$J = \sum_{m=1}^{N_p} x(k_i+m|k_i)^T Q x(k_i+m|k_i) + \eta^T \overline{R} \eta \quad (27)$$

Accordingly, the constraints are reformed as

$$\begin{cases} M_1 \eta \le \Delta U^{max} \\ -M_1 \eta \le -\Delta U^{min} \end{cases} \quad (28)$$

$$\begin{cases} M_2 \eta \le U^{max} - \overline{u}(k_i-1) \\ -M_2 \eta \le U^{min} + \overline{u}(k_i-1) \end{cases} \quad (29)$$

where

$$M_1 = \left[\begin{array}{cccc} L_1(m)^T & o_2^T & \ldots & o_m^T \\ o_1^T & L_2(m)^T & \ldots & o_m^T \\ \ldots & \ldots & \ldots & \ldots \\ o_1^T & o_2^T & \ldots & L_m(m)^T \end{array} \right]$$

$$M_2 = \left[\begin{array}{cccc} \sum_{i=0}^{k-1} L_1(i)^T & o_2^T & \ldots & o_m^T \\ o_1^T & \sum_{i=0}^{k-1} L_2(i)^T & \ldots & o_m^T \\ \ldots & \ldots & \ldots & \ldots \\ o_1^T & o_2^T & \ldots & \sum_{i=0}^{k-1} L_m(i)^T \end{array} \right]$$

By solving the optimizing problem above, the controller can generate an optimal sequence, but only the first element of the sequence applies to the robot, and the controller then updates the state and re-solves the optimizing problem. For applying the proposed algorithm on real mobile robots, the Hildreth's Quadratic Programming is employed to solve the optimization procedure.

3 EXPERIMENTAL RESULTS

The proposed control algorithm is implemented on a real soccer robot on a standard RoboCup field with $18 \times 12m^2$. The omnidirectional soccer robot is shown in Fig. 1, which is equipped with an onboard PC with 1.66G HZ CPU and 1G RAM. The maximum transition velocity and rotation velocity of the robot are $3.5m/s$ and $13rad/s$, and the acceleration are $4.5m/s^2$ and $20rad/s^2$. The experimental trajectory is generated from mapping a parameterized geometric path $C(s)$ into time space, thus a time parameterized trajectory $C(s(t))$ is obtained. The procedure works as follows: at first, the Bezier Curve $(Bez(x(s), y(s)), s \in [0, 1])$ can be generated from the desired control points, and next a time parameterized trajectory can be obtained by the following equation:

$$s_{k+1} = s_k + T_s \frac{v_k}{|\frac{dBez}{ds}|} + \frac{T_s^2}{2}\left(\frac{a_k}{|\frac{dBez}{ds}|} - \frac{v_k^2(\frac{dBez^T}{ds}\frac{d^2Bez}{ds^2})}{|\frac{dBez}{ds}|^4}\right)$$

where T_s is the sampling period and $v_k = v(kT_s), a_k = a(kT_s)$ are the reference velocity and acceleration profile defined by the user. This method works very well in RoboCup competition and more detailed information can be found in [17]. In our experiment, the control points of Bezier Curve are chosen as:

$$p_0(-100, 540), p_1(-450, 150), p_2(0, -500), p_3(750, -200)$$

The acceleration profile is set as a constant value $a = 1m/s^2$ and when the velocity achieves $3.00m/s$, the acceleration becomes 0.

In the following trajectory tracking experiments, the parameters for model predictive control are assigned as:

$$Q = 3C_m^T C_m, R = 2.5I, T_s = 0.03s$$

$$\alpha = 0.5, N = 1, N_p = 35$$

in addition, the velocity and acceleration are bounded at $3.25m/s$ and $4.5m/s^2$.

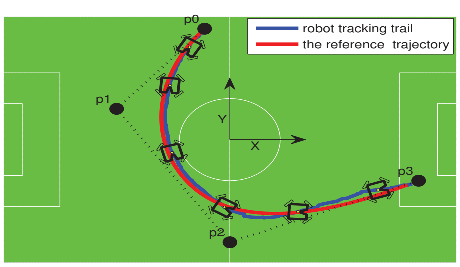

Figure 3: The controller based on Laguerre polynomials with $\alpha = 0.5, N = 1$ drives mobile robot to "reach" the reference with the algorithm's execution time 2.18ms.

Fig. 3 illustrates the results of tracking the given trajectory by using the proposed controller. The robot starts at control point p_0 and tracks the given trajectory with high

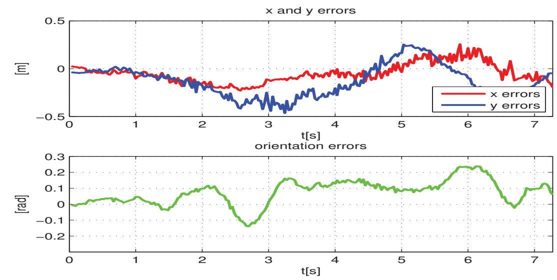

Figure 4: The tracking errors.

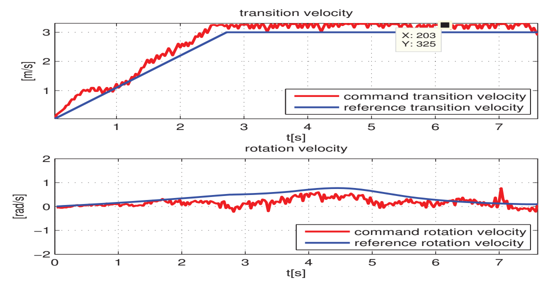

Figure 5: The tracking speed is bounded at $3.25m/s$ when applying the velocity constraints.

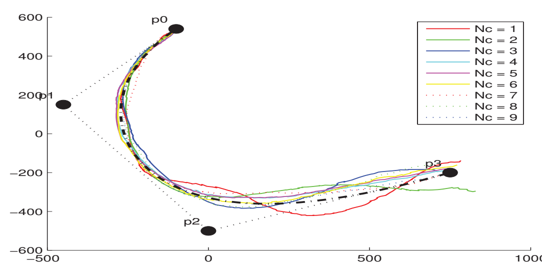

Figure 6: Tracking the given trajectory when using the traditional design of model predictive control with different N_c.

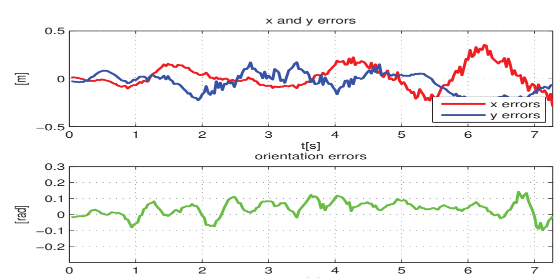

Figure 7: The tracking errors when using the traditional design of model predictive control with $N_c = 9$

Table 1: Control Horizon Influence on Computation Time

control horizon N_c	computation time(ms)
1	1.79
2	2.48
3	3.20
4	3.89
5	4.72
6	5.14
7	5.75
8	6.73
9	7.48

speed bounded at $3.25m/s$(shown in Fig. 5), and finally stops at p_3. With the parameters $\alpha = 0.5, N = 1$, the total tracking time is 7.2s and the execution time of our algorithm is only 2.18ms, which is sufficient small for real-time use even if the robot is source limited. Fig. 4 shows that, at most time steps, the tracking position errors range from -0.3m to 0.3m, and the orientation errors are from -0.2rad to 0.2rad, which indicates that good performance can be realized by using Laguerre-polynomials-based model predictive control on tracking high-speed trajectory.

Fig. 6 shows the results when traditional model predictive control works with different control horizons to solve the problem. The tracking performance can be improved as the control horizon grows larger, but the computation cost also increases as the control horizon grows up as shown in Table 1. Fig. 7 shows that when choosing $N_c = 9$ in the traditional model predictive control design, the performance is similar to that when using the proposed algorithm with $\alpha = 0.5$ and $N = 1$. In other words, a long control horizon can be realized by using $N = 1$, and the computation time in this way is 2.18ms rather than 7.48ms with $N_c = 9$. The computation time is cut down significantly, while maintaining a good performance, so the proposed algorithm is effective.

4 CONCLUSIONS

This paper proposes a model predictive control scheme for omnidirectional soccer robots in tracking high-speed trajectories. Laguerre polynomials are employed to design the model predictive controller, by substituting a long horizon with a small number of parameters, so that the computation cost can be cut down for practical applications. The proposed model predictive control has been tested on RoboCup soccer robots, and the experimental results show that, the model predictive control works well in tracking high-speed trajectory.

The proposed control scheme deals with the linearized model and only kinematics is taken into account. As we know, the basic limitation of the linearization is that the stability is only guaranteed in a neighborhood of the reference points. On the other hand, for high-speed tracking it is required to have high response rate of the actuators, the dynamics of the physical system should be learned. In future work, we are going to focus on implementing nonlinear model predictive control based on the dynamics model on soccer robots and improving the robustness of our algorithm.

REFERENCES

[1] Aguiar, A. P. and J. P. Hespanha (2007). Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles With Parametric Modeling Uncertainty. Automatic Control, IEEE Transactions on 52(8): 1362-1379.

[2] Ferreira, J. R. and A. P. Moreira. (2012). Non-linear model predictive controller for trajectory tracking of an omnidirectional robot using a simplified model.

[3] Hoffmann, G. M., C. J. Tomlin, et al. (2007). Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. American Control Conference, 2007. ACC '07.

[4] Mayne, D. Q., J. B. Rawlings, et al. (2000). Constrained model predictive control: Stability and optimality. Automatica 36(6): 789-814.

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

[6] F. Kuhne, W.F. Lages, J.M. Gomes da Silva Jr., Model predictive controlof a mobile robot using linearization, in: Proceedings of Mechatronics and Robotics 2004, Aachen, Germany, 2004.

[7] Maurovic, I., M. Baotic, et al. (2011). Explicit Model Predictive Control for trajectory tracking with mobile robots. Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on.

[8] Kanjanawanishkul, K. and A. Zell (2009). Path following for an omnidirectional mobile robot based on model predictive control. Robotics and Automation, 2009. ICRA '09. IEEE International Conference on.

[9] X. Li and A. Zell. Nonlinear Model Predictive Control of an Omnidirectional Mobile Robot In Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS 2008)

[10] Y. Zhu and Umit Ozguner. Constrained model predictive control for nonholonomic vehicle regulation problem. Proceedings of the 17th IFAC World Congress,pages 9552-9557, 2008.

[11] Gu, D. and H. Hu (2009). A model predictive controller for robots to follow a virtual leader. Robotica 27(06): 905-913.

[12] Yoon, Y., J. Shin, et al. (2009). Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. Control Engineering Practice 17(7): 741-750.

[13] Vougioukas, S. G. (2007). Reactive Trajectory Tracking for Mobile Robots based on Non Linear Model Predictive Control. Robotics and Automation, 2007 IEEE International Conference on.

[14] Kouvaritakis and Cannon (2001). Nonlinear predictive control, theory and practice. London: The IEE.

[15] Wang, L. (2004). Discrete model predictive controller design using Laguerre functions. Journal of Process Control 14(2): 131-142.

[16] T. Zhang, Z. H. Chen, et al. (2006). Adaptive predictive control algorithm based on Laguerre Functional Model. Int. J. Adaptive Control and Signal Processing, 20(2), 53-76.

[17] L. Biagiotti and C. Melchiorri. Trajectory Planning for Automatic Machines and Robots. Springer Berlin Heidelberg, 2008.