WALAILAK JOURNAL
http://wjst.wu.ac.th Engineering and Physical Sciences

Path Following with a Time-Convergence Penalty Term for a Mobile Robot

Kiattisin KANJANAWANISHKUL

Mechatronics Research Laboratory, Faculty of Engineering, Mahasarakham University, Kamriang, Kantharawichai, Mahasarakham 44150, Thailand

(Corresponding author's e-mail: kiattisin.k@msu.ac.th)

Received: 31 July 2013, Revised: 28 October 2013, Accepted: 29 November 2013

Abstract

In this paper, we focus on a combination of path following and trajectory tracking for a mobile robot. Both are basic motion control schemes for a robot. The advantages of path following over trajectory tracking are that path following can avoid the use of large control signals for large path errors; therefore, it can eliminate aggressiveness by converging to the path smoothly, and control inputs are less likely to be forced to saturation. However, there is no temporal specification for path following. Therefore, in this work, we propose to add a time-convergence penalty term into the optimization problem of model predictive control (MPC) that we use to control robot motion. MPC can handle the saturation of control signals explicitly. As a result, the robot can move safely. However, the major concern in the use of MPC is whether such an open-loop control scheme can guarantee system stability. To solve this problem, we apply the idea of a contractive constraint to guarantee the stability of our MPC framework. To illustrate its effectiveness, numerous simulation scenarios have been conducted. Furthermore, we depict remarkable advantages of path following over trajectory tracking.

Keywords: Path following, contractive model predictive control, mobile robots, trajectory tracking, time-convergence

Introduction

Trajectory tracking is defined as a mobile robot tracking a time-parameterized reference [1]. The major disadvantage of this basic motion is that large control signals are required for large position error, resulting in aggressive movement and saturation in control signals. To avoid such problems, path following is employed instead. Like trajectory tracking, path following is one of the fundamental motion control schemes of a robot. In general, path following is defined as a mobile robot required to converge to and to follow a path-parameterized reference, without any temporal specifications. Typically, to achieve path following control, the robot's forward velocity tracks a desired velocity profile, while the controller determines the robot's heading direction to drive it to the desired path. However, this standard path following control cannot meet the requirements of trajectory tracking, because there is no consideration of temporal specifications. Therefore, the objective of this work is to achieve both path following and trajectory tracking because we would like to gain the advantages of path following and to achieve the requirements of trajectory tracking.

For path following, a virtual vehicle concept is employed in this work. It is defined as a reference point moving along the given path; the real robot tries to catch up with this virtual vehicle. To gain the advantage of path following, the desired behavior should be as follows: when the path error (the difference between the position of the real robot and the position of the virtual vehicle) is large, the virtual vehicle will wait for the real one; when the path error is small, the virtual vehicle will move along the path at a speed close to the desired speed assignment. This behavior is suitable in practice, because it avoids the use of large control signals for large path errors. It can eliminate the aggressiveness of the trajectory tracking counterpart by converging to the path smoothly, and control inputs are less likely forced to saturation [2]. To obtain such a behavior mentioned above, a model predictive control (MPC) scheme is implemented. Furthermore, to meet the requirement of trajectory tracking, we add a time-convergence penalty term into the objective function of MPC. As a result, the reference point of path following and the reference point of trajectory tracking is compromised via the optimization problem of MPC.

The path following problem

Typically, control laws of path following [3] are determined to steer a robot in order to reach and to follow a reference path, i.e., a manifold parameterized by a continuous scalar s, while the secondary goal is to command the robot to move along the path to satisfy some additional dynamic specifications, e.g., time, speed, or acceleration assignments [4]. This setting is more flexible than the standard trajectory tracking problem, since the path variable s can be used as an extra degree of freedom for the secondary goal.

Diaz del Rio *et al*. [5] proposed a method called error adaptive tracking, in which the tracking adapts to the errors. They defined the function of $\dot{s}$ as $\dot{s} = g(e)$, where e is the distance error. They also proposed $\dot{s} = g(t,e)$ in order to preserve time determinism of trajectory tracking. Soeanto *et al*. [6] controlled $\dot{s}$ by modeling the kinematic equations of motion with respect to the Frenet frame. A virtual vehicle concept was also employed by Egerstedt *et al*. [7], whose control law ensures global stability by determining the motion of the virtual vehicle on the desired path via a differential equation containing error feedback.

To achieve our goal, mentioned previously, we adapt the idea of [8] to obtain optimal motion of the virtual vehicle by using MPC. Since an MPC algorithm employs an explicit model of the plant, which is used to predict the future output behavior, the kinematic model of a differential-drive robot is given as follows (see **Figure 1**);

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

where $[x_m \;\; y_m \;\; \theta_m]^T$ is the state vector in the world frame. v_m and ω_m stand for the linear and angular velocities, respectively.

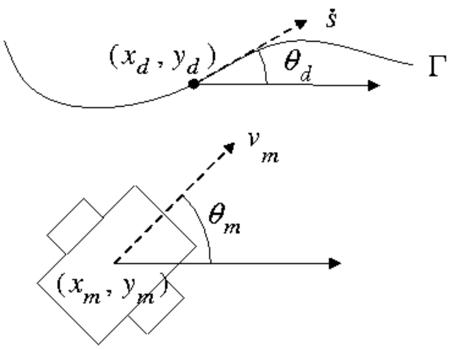


Figure 1 A graphical representation of a differential-drive mobile robot and a reference path.

In general, we wish to find the control laws of $\dot{s}$ and ω_m such that the robot follows a virtual vehicle with position $[x_d \;\; y_d \;\; \theta_d]^T$. The error state vector between the robot state vector $[x_m \;\; y_m \;\; \theta_m]^T$ and a virtual vehicle's state vector $[x_d \;\; y_d \;\; \theta_d]^T$ 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;

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

where κ is the path curvature.

Due to the requirement of a time convergence for trajectory tracking, we introduce an acceleration control input a_m, where $a_m = \dot{v}_m$ and, with $\eta_e = v_m - v_d$, we then obtain $\dot{\eta}_e = a_m - \dot{v}_d$. Thus, we can redefine the control signals as follows;

$$\mathrm{u}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -\dot{s} + v_m \cos\theta_e \\ \omega_m - \dot{s}\kappa \\ a_m - \dot{v}_d \end{bmatrix} \tag{4}$$

and the error state dynamic model then becomes;

$$\dot{\mathrm{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \\ \dot{\eta}_e \end{bmatrix} = \begin{bmatrix} 0 & \dot{s}\kappa & 0 & 0 \\ -\dot{s}\kappa & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \\ \eta_e \end{bmatrix} + \begin{bmatrix} u_1 \\ v_m \sin\theta_e \\ u_2 \\ u_3 \end{bmatrix} \tag{5}$$

Besides steering the robot to the desired path, the forward velocity v_m can be used as an extra degree of freedom. In this paper, it conforms to the desired velocity v_d at specific time of trajectory tracking.

Contractive model predictive control

As the name suggests, an MPC algorithm employs an explicit model of the plant which is used to predict the future output behavior. This prediction capability allows computing of a sequence of manipulated variable adjustments in order to solve optimal control problems online, where the future behavior of a plant is optimized over a future horizon, possibly subject to constraints on the manipulated inputs and outputs [9,10].

Important issues of linear MPC theory are by now well developed. However, as many systems are nonlinear, nonlinear MPC must be used [9]. The major concern in the use of nonlinear MPC is whether such an open-loop control can guarantee system stability. Mayne *et al.* [10] presented essential principles for the stability of MPC in constrained dynamical systems. We intentionally do not collect all published contributions because of the large number of publications. We refer the reader to [9,10].

Although MPC is apparently not a new control method, studies dealing with MPC of path following problems are rare. Recently, Faulwasser *et al*. [11] proposed a nonlinear MPC approach that is equivalent to setpoint stabilization in different coordinates, and they used the path as the terminal region. Yu *et al*. [12] presented a nonlinear MPC, where a polytopic linear differential inclusion (PLDI) based algorithm is used to choose the suitable terminal penalty and terminal constraint. A comprehensive survey paper related to MPC for a mobile robot can be found in [13]. However, this paper differs from other MPC schemes, because the objective of this paper is to combine path following and trajectory tracking.

Problem formulation

A nonlinear system is, in general, described by the following nonlinear differential equation;

$$\dot{\mathrm{x}}(t) = \mathrm{f}(\mathrm{x}(t), \mathrm{u}(t)) \tag{6}$$

subject to;

$$\mathrm{x}(t) \in X, \quad \mathrm{u}(t) \in U, \ \forall t \geq 0$$

where $\mathrm{x}(t) \in R^n$, $\mathrm{u}(t) \in R^m$ are the n dimensional state vector and the m dimensional input vector of the system, respectively. $X \subseteq R^n$ and $U \subseteq R^m$ denote the set of feasible states and inputs of the system, respectively. The input applied to the system is given by the solution of the following finite horizon open-loop optimal control problem, which is solved at each time instant;

$$\min_{\bar{\mathrm{u}}(\bullet)} \left\{ \int_t^{t+T_p} F\big(\bar{\mathrm{x}}(\tau), \bar{\mathrm{u}}(\tau)\big) d\tau + V_t \right\} \tag{7}$$

subject to;

$$\dot{\bar{\mathrm{x}}}(\tau) = \mathrm{f}\big(\bar{\mathrm{x}}(\tau), \bar{\mathrm{u}}(\tau)\big) \tag{8a}$$

$$\bar{\mathrm{u}}(\tau) \in U, \ \forall \tau \in \left[t, t+T_c\right] \tag{8b}$$

$$\bar{\mathrm{x}}(\tau) \in X, \ \forall \tau \in \left[t, t+T_P\right] \tag{8c}$$

$$\left\| \bar{\mathrm{x}}(t+T_p) \right\|_P \leq \alpha \left\| \bar{\mathrm{x}}(t) \right\|_P \quad \alpha \in [0,1) \tag{8d}$$

where $F(\bar{\mathrm{x}}, \bar{\mathrm{u}}) = \bar{\mathrm{x}}^T Q \bar{\mathrm{x}} + \bar{\mathrm{u}}^T R \bar{\mathrm{u}}$ and $\left\|\bar{\mathrm{x}}\right\|_p = \sqrt{\bar{\mathrm{x}}^T P \bar{\mathrm{x}}}$ with positive definite matrix P. 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 ($T_c \leq T_p$). The deviations from the desired values are weighed by the positive definite matrices Q, and R.

In (7), V_t is a time-convergence penalty term, integrated into the objective function of our MPC framework in order to fulfill the requirement of trajectory tracking. It is defined as follows;

$$V_t = K_t \big(s_t - \bar{s}(t+T_p)\big)^2 \tag{9}$$

where K_t is a positive constant. This constant weighs the relative importance of convergence in time over spatial convergence to the path. s_t is the path length at the time-parameterized reference plus the length of the predictive horizon, while $\bar{s}(t+T_p)$ is the internal path length at $t+T_p$. Note that this penalty term is not applied at each time instant along the future horizon, since it leads to aggressive motions.

The constraints in (8b) denote bounded control inputs. From (4), we have the following system control inputs $\dot{s}$, ω_m and a_m;

$$\begin{bmatrix} \dot{s} \\ \omega_m \\ a_m \end{bmatrix} = \begin{bmatrix} -u_1 + v_m \cos\theta_e \\ u_2 + \dot{s}\kappa \\ u_3 + \dot{v}_d \end{bmatrix} \tag{10}$$

and their boundaries;

$$\begin{bmatrix} 0 \\ \omega_{m,\min} \\ \Delta\omega_{m,\min} \\ v_{m.\min} \\ a_{m,\min} \end{bmatrix} \le \begin{bmatrix} \dot{s} \\ \omega_m \\ \Delta\omega_m \\ v_m \\ a_m \end{bmatrix} \le \begin{bmatrix} \dot{s}_{\max} \\ \omega_{m,\max} \\ \Delta\omega_{m,\max} \\ v_{m,\max} \\ a_{m,\max} \end{bmatrix} \tag{11}$$

The last inequality end constraint in (8d) is a so-called contractive constraint [14]. It requires that, at time t, the system states at the end of the predictive horizon, $\bar{\mathrm{x}}(t+T_p)$ are contracted in norm with respect to the states at the beginning of the prediction, $\bar{\mathrm{x}}(t)$. The 2 additional parameters which determine how much contraction is required are the contractive parameter, $\alpha \in [0,1)$ and the positive definite matrix P.

Stability analysis

The following assumptions based on [14] are needed to ensure stability:

Assumption 1 There exists a constant $\rho \in (0, \infty)$ such that for all $\mathrm{x}(t) \in B_\rho := \{\mathrm{x} \in R^n \mid \|\mathrm{x}\|_P \le \rho\}$, we can find a contractive parameter $\alpha \in [0,1)$ so that with the chosen finite horizon T_P all the constraints on the inputs and states can be satisfied, and the objective function is finite.

Note that, if $x(t_0) \in B_\rho$, then $x(t_k) \in B_{\alpha^k\rho} \subset B_\rho$ where $t_k = t_0 + kT_p$. Thus, using Assumption 1, if the optimal control problem is feasible at time t_0, then the sequence of control problems at $t > t_0$ is feasible as well.

Assumption 2 There exists a constant $\beta \in (0, \infty)$ such that the transient state, $\mathrm{x}(\tau)$, of the model satisfies $\|\mathrm{x}(\tau)\|_P \le \beta\|\mathrm{x}(t)\|_P$, for all $\tau \in [t, t+T_p]$.

Note that, since u(t) is constrained, Assumption 2 is always satisfied, except for systems with finite escape time. Then, the theorem based on [14] can now be given.

Theorem 1 Suppose Assumptions 1 and 2 hold; the MPC algorithm described in Subsection Problem Formulation is exponentially stable, in such a way that the resulting trajectory of the closed-loop system satisfies the following inequality;

$$\| \mathrm{x}(t) \|_P \leq \beta \| \mathrm{x}(t_0) \|_P e^{-(1-\alpha)(t-t_0)/T_P} \tag{12}$$

Proof. From Assumption 1, if the optimal control problem is feasible at time t_0, the optimal control problem is feasible at time $t > t_0$. Thus, we have;

$$\|\mathrm{x}(t_k)\|_P \leq \alpha \|\mathrm{x}(t_{k-1})\|_P \leq \cdots \leq \alpha^k \|\mathrm{x}(t_0)\|_P \tag{13}$$

where $t_k = t_0 + kT_p$ and k belongs to the set of nonnegative integers. From Assumption 2, x(t) satisfies the following inequality;

$$\|\mathrm{x}(t)\|_P \leq \beta\alpha^k \|\mathrm{x}(t_0)\|_P \quad \forall t \in [t_k, t_{k+1}]. \tag{14}$$

Since $e^{-(1-\alpha)} - \alpha \geq 0$, which means $0 \leq \alpha^k \leq e^{-(1-\alpha)k}$ inequality (14) results in;

$$\|\mathrm{x}(t)\|_P \leq \beta \|\mathrm{x}(t_0)\|_P e^{-(1-\alpha)k}. \tag{15}$$

Since $k = (t_k - t_0)/T_P$ and $(t - t_0)/T_P < (t_k - t_0)/T_P$ $\forall t \in [t_0, t_k)$, we have;

$$\beta e^{-(1-\alpha)k} \leq \beta e^{-(1-\alpha)(t-t_0)/T_P}. \tag{16}$$

Therefore, using (15) and (16), we finally have;

$$\|\mathrm{x}(t)\|_P \leq \beta \|\mathrm{x}(t_0)\|_P e^{-(1-\alpha)(t-t_0)/T_P}. \tag{17}$$

This concludes the proof. ■

Simulation results and discussion

To assess our proposed MPC framework, the following 8-shaped curve is chosen as a reference path, since its geometrical symmetry and sharp changes in curvature make the test challenging;

$$\begin{aligned} x_d(t) &= 1.8\sin(0.1t) \\ y_d(t) &= 1.2\sin(0.2t) \end{aligned} \tag{18}$$

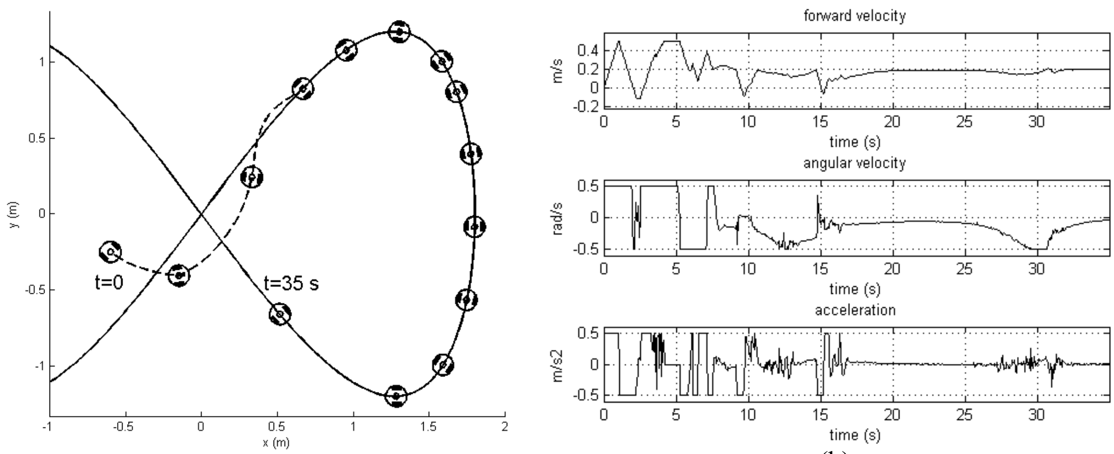
where t is time in the case of trajectory tracking, while this reference is numerically parameterized by the path variable s in case of the path following problem. All the parameters of our framework are set as follows;

Q = diag (200, 800, 0.5, 0.5), R = diag (0.01, 0.01, 0.01), P = diag (1, 1, 0.01, 0.01),
$K_t = 1$, $N = 10$, $T_c = T_p = 0.5$ s, δ (sampling time) = 0.05 s, $v_d = 0.2$ m/s, $s(0) = 0$ m,
$\dot{s}_{\max} = 0.5$ m/s, $\alpha = 0.999$, $v_{\mathrm{m,max}} = 0.5$ m/s, $v_{\mathrm{m,min}} = -0.5$ m/s, $\omega_{\mathrm{m,max}} = 0.5$ rad/s, $\omega_{\mathrm{m,min}} = -0.5$ rad/s,
$\Delta\omega_{\mathrm{m,max}} = 0.5$ rad/s, $\Delta\omega_{\mathrm{m,min}} = -0.5$ rad/s, $a_{\mathrm{m,max}} = 0.5$ m/s^2 and $a_{\mathrm{m,min}} = -0.5$ m/s^2.

The following simulation scenarios have been conducted to show the effectiveness of our proposed control scheme. Note that the circles in all figures below are snapshots of robot location at every 2.5 s and the robot trajectories are shown as dashed lines.

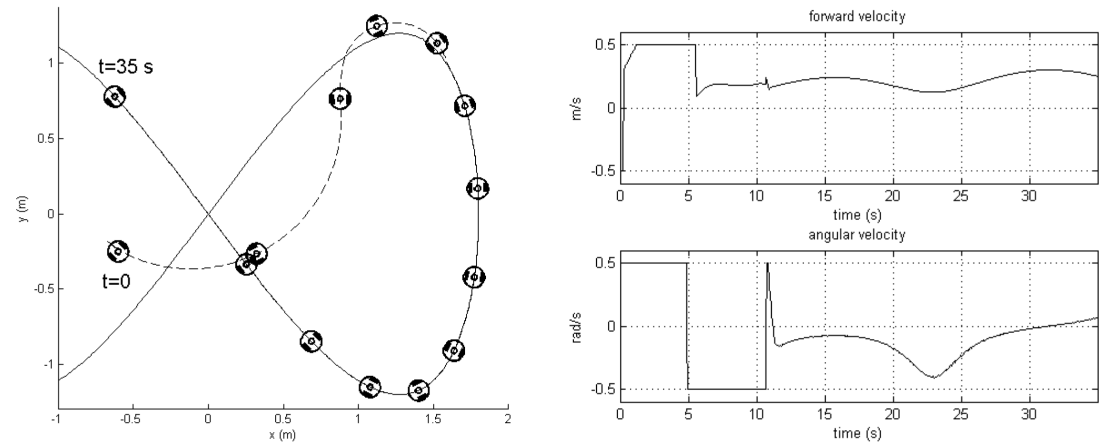
Time-convergence penalty

An additional penalty term V_t in (7) is integrated into the objective function in order to fulfill the requirement of trajectory tracking. The simulation results of path following without a time-convergence penalty term, of trajectory tracking (see [15] for details), and of path following with a time-convergence penalty term, are illustrated in **Figure 2a**, **Figure 3a**, and **Figure 4a**, respectively. As seen in **Figures 3a** and **3b**, the robot moves aggressively, due to the large distance error and because the control signals are forced to saturation at the beginning, while in **Figures 2a** and **2b**, the robot converges to the path smoothly, and control signals are less likely pushed to saturation. Although robot motion in the case of path following is less aggressive, the time constraints are not achieved. With V_t in (7), the robot converges smoothly to the desired path, similar to the results in **Figure 2a**, and then reacts to achieve zero trajectory tracking error, i.e., the robot reaches the same position and the same velocity as the results of trajectory tracking, shown in **Figure 4a**, at the same time. Therefore, in this work, we can achieve both smooth spatial convergence and time convergence by penalizing the objective function with V_t.



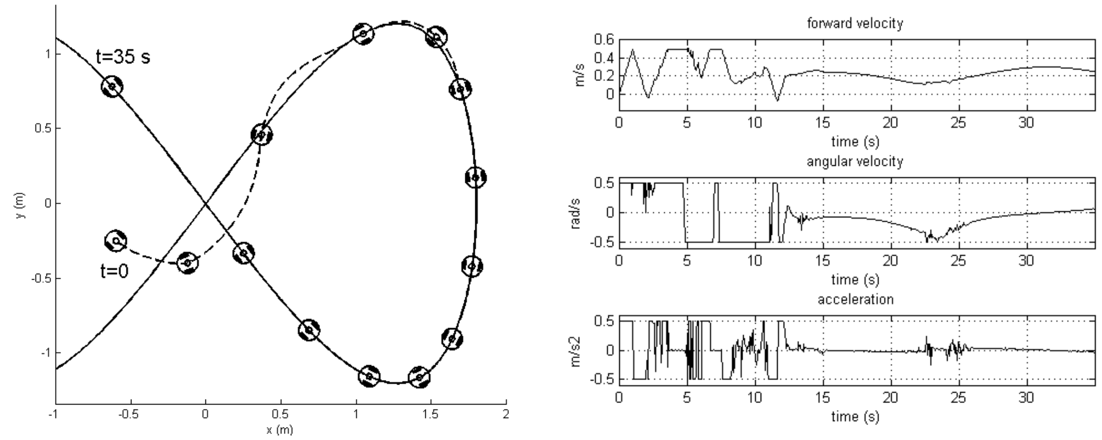
(a) (b)

Figure 2 The simulation results with the initial posture set to $[-0.6, -0.25, -\pi/4]^T$: (a) superimposed snapshots of path following without a time-convergence penalty term, and (b) velocity and acceleration corresponding to (a).



(a) (b)

Figure 3 The simulation results with the initial posture set to $[-0.6,-0.25,-\pi/4]^T$: (a) superimposed snapshots of trajectory tracking, and (b) the velocity profiles corresponding to (a).

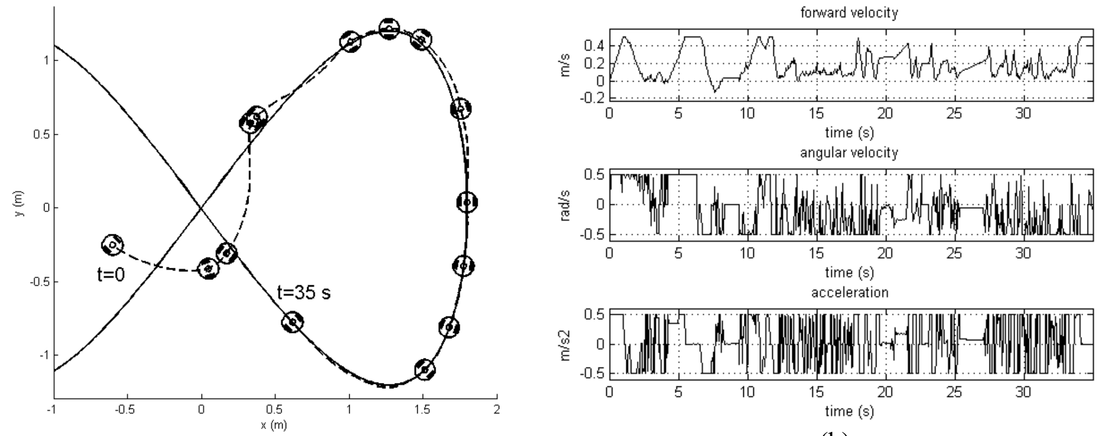


(a) (b)

Figure 4 The simulation results with the initial posture set to $[-0.6,-0.25,-\pi/4]^T$: (a) superimposed snapshots of path following with a time-convergence penalty term, and (b) velocity and acceleration corresponding to (a).

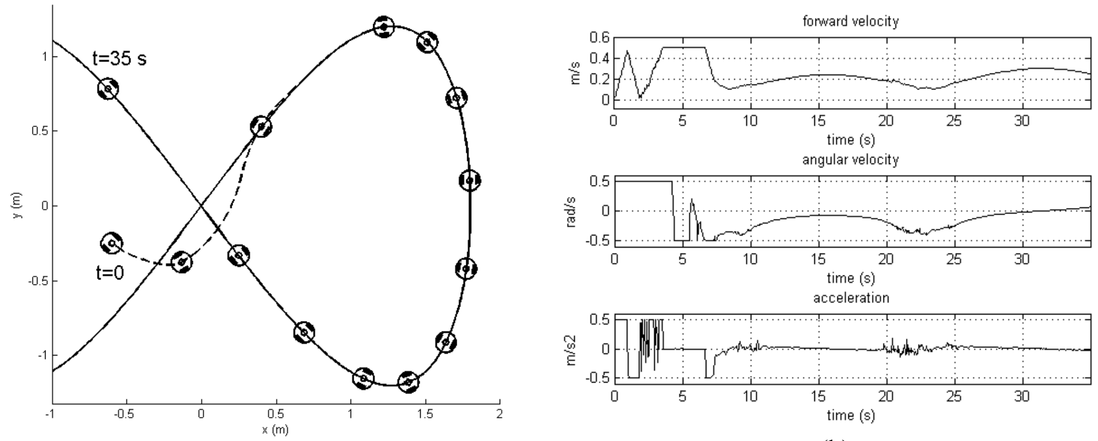
Parameter turning

We only test our MPC algorithm with 3 different values of the control horizon T_c (note that, in this work, we choose $T_c = T_p$), while the other tuning parameters, i.e., the sampling period δ and the penalty weight matrices Q, R, P and K_t, are fixed. The results of $T_c = 0.25$ s (i.e., $N = 5$ steps), $T_c = 0.5$ s (i.e., $N = 10$ steps), and $T_c = 1.0$ s (i.e., $N = 20$ steps) are shown in **Figure 5a**, **Figure 4a** and **Figure 6a**, respectively. As obviously seen in the results, the shorter control horizon causes worse performance, while the longer control horizon improves performance but leads to an increase of online computation. Thus, the control horizon must be chosen to compromise between performance and computation.



(a) (b)

Figure 5 The simulation results of path following with a time-convergence penalty term using T_c = 0.25 s (a) superimposed snapshots, and (b) velocity and acceleration corresponding to (a).



(a) (b)

Figure 6 The simulation results of path following with a time-convergence penalty term using T_c = 1.0 s (a) superimposed snapshots, and (b) velocity and acceleration corresponding to (a).

Conclusions and future work

In this paper, we presented a new approach to achieve both path following and trajectory tracking. The key idea is to add a time-convergence penalty term into the object function of the MPC scheme used to control robot motions. Furthermore, we also satisfied the following objectives using our proposed MPC framework: (i) path following control with stability guarantee, (ii) optimal forward velocity for a virtual vehicle, and (iii) bounded control signals, i.e., the MPC scheme is used to produce a sequence of control inputs by taking into account input boundaries, a contractive constraint, and time-convergence requirements.

Currently, a real mobile robot which can be used to validate our control law in real-world environments is being developed. We also would like to extend our controller to accomplish the path following task in dynamic environments with static and moving obstacles.

References

[1] P Morin and C Samson. *Motion Control of Wheeled Mobile Robot*. *In*: B Siciliano and O Khatib (ed.). Springer Handbook of Robotics. Springer, Berlin, 2008, p. 799-826.

[2] SA Al-Hiddabi and NH McClamroch. Tracking and maneuver regulation control for nonlinear non-minimum phase systems: application to flight control. *IEEE Trans. Contr. Syst. Tech.* 2002; **10**, 780-92.

[3] A Aguiar, D Dacic, J Hespanha and P Kokotovic. Path-following or reference-tracking? An answer relaxing the limits to performance. *In*: Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 2004.

[4] R Skjetne, T Fossen and P Kokotovic. Robust output maneuvering for a class of nonlinear systems. *Automatica* 2004; **40**, 373-83.

[5] FD del Rio, G Moreno, J Ramos, C Rodriguez and A Balcells. A new method for tracking memorized paths: Application to unicycle robots. *In*: Proceedings of the 10th IEEE Mediterranean Conference on Control and Automation, Lisbon, Portugal, 2002.

[6] D Soeanto, L Lapierre and A Pascoal. Adaptive non-singular path-following, control of dynamic wheeled robots. *In*: Proceedings of the International Conference on Advanced Robotics, Coimbra, Portugal, 2003, p. 1387-92.

[7] M Egerstedt, X Hu and A Stotsky. Control of mobile platforms using a virtual vehicle approach. *IEEE Trans. Automat. Contr*. 2001; **46**, 1777-82.

[8] K Kanjanawanishkul and A Zell. Distribured model predictive control for coordinated path following control of omnidirectional mobile robots. *In*: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Singapore, 2008, p. 3120-5.

[9] F Allgower, R Findeisen and Z Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Eng*. 2004; **35**, 299-315.

[10] DQ Mayne, JB Rawlings, CV Rao and POM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica* 2000; **36**, 789-814.

[11] T Faulwasser, B Kern and R Findeisen. Model predictive path-following for constrained nonlinear systems. *In*: Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, China, 2009, p. 8642-7.

[12] SY Yu, X Li, H Chen and F Allgower. Nonlinear model predictive control for path following problems. *In*: Proceeding of the 4th IFAC Nonlinear Model Predictive Control Conference, Noordwikkerhout, the Netherlands, 2012, p. 145-50.

[13] K Kanjanawanishkul. Motion control of a wheeled mobile robot using model predictive control: A survey. *KKU Res. J*. 2012; **17**, 811-37.

[14] S Kothare and M Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Automat. Contr*. 2000; **45**, 1053-71.

[15] F Xie and R Fierro. First-state contractive model predictive control of nonholonomic mobile robots. *In*: Proceedings of the American Control Conference, Seattle, WA, 2008, p. 3494-9.