Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 55(3), 287–298(2014)

Kiattisin Kanjanawanishkul

Coordinated Path Following for Mobile Robots Using a Virtual Structure Strategy with Model Predictive Control

DOI 10.7305/automatika.2014.12.460
UDK 004.896:004.93
IFAC 4.6.2; 2.8.3

Original scientific paper

In this paper, we propose a novel coordinated path following controller based on model predictive control (MPC) for mobile robots. The strategy is based on a virtual structure approach where the entire formation is considered as a rigid body and the control laws for a virtual leader vehicle and for actual follower robots are optimized by considering the dynamics of the virtual structure and the desired motion of each vehicle. Besides, we also fulfill time convergence for trajectory tracking by integrating an additional penalty term into our model predictive control scheme. However, the major concern in the use of model predictive control is whether such an open-loop control scheme can guarantee system stability. In this case, we apply the idea of a contractive constraint to guarantee the stability of our MPC framework. Although our approach is centralized, numerous simulation scenarios have been conducted to illustrate its effectiveness and its superior performance for a small group of mobile robots. Furthermore, we show that path following control can offer a number of advantages over its trajectory tracking counterpart.

Key words: Coordinated Path Following Control, Model Predictive Control, Mobile Robots, Virtual Structure, Motion Control

Koordinirano slijeđenje putanje za mobilne robote zasnovano na strategiji virtualne strukture i modelskom prediktivnom upravljanju. U ovome članku predlaže se novi algoritam upravljanja koordiniranim slijeđenjem putanje za mobilne robote zasnovan na modelskom prediktivnom upravljanju. Strategija se zasniva na pristupu s virtualnom strukturom gdje se cijela formacija robota smatra krutim tijelom, dok se za virtualno vodeće vozilo i za slijedeće robote optimiziraju zakoni upravljanja vodeći računa o dinamici virtualne strukture i željenom gibanju svakog od vozila. Također, algoritam ispunjava uvjet vremena konvergencije radi praćenja trajektorija na način da integrira dodatani član unutar algoritma modelskog prediktivnog upravljanja. Međutim, koristeći modelsko prediktivno upravljanje postavlja se pitanje može li ovakav pristup u otvorenom upravljačkom krugu jamčiti stabilnost. Radi toga, primijenjuje se ideja sužavajućeg ograničenja radi jamčenja stabilnosti predloženog riješenja. Iako je predloženi pristup centraliziran, provedeni su brojni simulacijski eksperimenti kako bi se ilustrirala učinkovitost i superiorno vladanje na primjeru male grupe mobilnih robota. Nadalje, pokazuje se da upravljanje slijeđenjem putanje pruža brojne prednosti u usporedbi s praćenjem trajektorije.

Ključne riječi: upravljanje koordiniranim slijeđenjem putanje, modelsko prediktivno upravljanje, mobilni roboti, virtualna struktura, upravljanje gibanjem robota

1 INTRODUCTION

The main characteristics of three basic motion tasks of an autonomous vehicle [1] are as follows: (i) point stabilization, where the objective is to stabilize a vehicle at a desired robot posture, (ii) trajectory tracking, where the vehicle is required to track a time-parameterized reference, and (iii) path following, where the vehicle is required to converge to and follow a desired path-parameterized reference, without any temporal specifications. Typically the path following controller eliminates the aggressiveness of the trajectory tracking controller by converging to the path smoothly and control inputs are less likely forced to saturation [2].

The path following problem becomes more difficult if a group of mobile robots are required to follow a path and to maintain a desired formation shape at the same time. This is a so-called coordinated path following problem. Current application areas include search and rescue operations, security patrols, landmine removal, remote terrain and space exploration, control of arrays of satellites and UAVs, area coverage and reconnaissance in military missions [3].

Although the coordinated path following problem has been well studied, it is still the subject of numerous research studies. In this work, we wish to achieve the following four objectives through our model predictive control (MPC) framework: (i) coordinated path following control with stability guarantee and with formation feedback, (ii) bounded control signals, (iii) time-constraint requirements, and (iv) optimal forward velocity of a virtual leader vehicle. The first objective can be simply fulfilled via a contractive constraint that can be used to guarantee the stability of MPC, whereas the other objectives can be accomplished by treating them as either constraints or penalty terms of the MPC scheme. For the third objective, trajectory tracking, where the time evolution of the position, orientation, as well as the linear and angular velocities is specified, can be achieved by integrating an additional penalty term, thus fulfilling both path following and trajectory tracking. Furthermore, for the path following problem, assigning a velocity profile to the virtual vehicle [4] is a crucial task. In this paper, the velocity is optimized according to robots' pose errors and desired forward velocity of formation along some lookahead distance corresponding to the prediction horizon of the MPC scheme.

This paper is organized as follows: Section 2 describes the mathematical model of a mobile robot and explains the basic principle of coordinated path following. The control law based on contractive MPC and the virtual structure strategy is developed in Section 3. In Section 4, our controller is validated by extensive simulation scenarios. Likewise, a comparison between trajectory tracking and path following is performed to show the effectiveness of our proposed control scheme. Finally, our conclusions and future work are drawn in Section 5.

2 THE COORDINATED PATH FOLLOWING PROBLEM

Typically, control laws used for solving trajectory tracking problems make robots track predetermined feasible trajectories, i.e., trajectories that assign the time evolution of the position, orientation (spatial dimension), as well as the linear and angular velocities (temporal dimension) [5]. However, the drawback of this approach is that finding a feasible trajectory is a difficult task because of robots' complex nonlinear dynamics and significant uncertainties. In addition, in the presence of large tracking errors, the controller tries to produce large control signals to catch up with time-parameterized reference points, thus resulting in closed-loop performance difficulties and too large control signals. One approach to eliminate such problems is to use a path following controller instead, as seen in our numerical simulations.

Recently, trajectory tracking control has been replaced by path following control as it is more suitable for certain applications where the vehicle is not required to be on a given point of the trajectory at a given time instant. In particular, Aguiar et al. [5] highlighted a fundamental difference between path following and trajectory tracking by demonstrating that performance limitations due to unstable zero-dynamics can be removed in the path following problem. Besides, with path following, smoother convergence to the path is attained, the time-dependence of the problem is obviated, and the control signals are less likely forced into saturation when compared to trajectory tracking.

In general, control laws for path following problems are designed to steer an object (robot arm, mobile robot, ship, aircraft, etc.) to reach and to follow a geometric path, i.e., a manifold parameterized by a continuous path variable s (called a geometric task), while a secondary goal is to force the object moving along the path to satisfy some additional dynamic specifications (called a dynamic assignment task) [5]. This dynamic behavior can be determined by time, speed, or acceleration assignments [6]. This setting is apparently more general than the trajectory tracking problem in such a way that the path variable s is left as an extra degree of freedom for the secondary goal.

To determine the path variable s, a numerical projection from the current state onto the path was used by Micaelli and Samson [7]. This point on the path plays the role of a virtual vehicle that will be tracked by the actual vehicle. The main problem of this idea is that singularities occur when the distance to the path is not well-defined. This problem can be avoided by explicitly controlling the timing law for s along the path.

In general, the choice of the timing law for s has the following desired behavior: When path following errors are large, the virtual vehicle will slow down or wait for the actual vehicle, when path errors are small, the virtual vehicle will move at the speed close to the desired speed assignment. Since it avoids the use of large control signals for large path errors, it is suitable in practice. Diaz del Rio et al. [8] 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. [4] controlled $\dot{s}$ explicitly by modeling the kinematic equations of motion with respect to the Frenet frame. A virtual vehicle concept was also employed by Egerstedt et al. [9], 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.

In this work, we adapt the idea of [10] to obtain optimal motion of the virtual vehicle by using MPC. One of its well-known advantages is that input constraints are handled straightforwardly in the optimization problem so that the robot can travel safely. In contrast, other conventional control techniques are usually designed under the assumption that there are no limitations on inputs or states. Likewise, future information can be employed in order to improve system performance because the reference path is known beforehand.

A so-called coordinated path following problem makes the path following problem more difficult because a group of mobile robots are required to follow a path and to maintain a desired formation shape at the same time. This problem can be seen as a subset of formation control in the research area of multi-robot systems [3]. In the literature, the approaches to solve this problem are roughly categorized into three strategies: leader-following, behavior-based, and virtual-structure. Each approach has its own advantage and disadvantage. In this work, we choose a virtual structure approach where it considers the entire formation as a rigid body. The control law for each vehicle of the virtual structure is determined by considering the dynamics of the virtual structure and the desired motion of each vehicle. The main advantages of this approach are that the virtual structure can move as a whole in a given direction with some given orientation and the coordinated behavior for the whole group can be easily described and effectively maintained during maneuvers. However, this virtual-structure strategy may not be suitable for applications that the formation shape needs to be frequently reconfigured.

The virtual structure approach has been used for formation control of mobile robots [11–14], spacecraft [15] and marine vehicles [16]. Lewis and Tan [11] proposed an algorithm that iteratively fits the virtual structure to the robots' positions, displaces the virtual structure in some desired direction and updates the robots' positions. Their method included formation feedback, but they cannot guarantee that a formation converges to a desired configuration. Egerstedt and Hu [12] defined the formation through mathematical constraints that model the formation shape. The path for a virtual leader including formation feedback is computed as a reference point for the robots to follow. Similarly, Young et al. [13] included a specific term of formation feedback into the coordination variable evolution. The virtual structure slows down and stops as the robots get out of formation and it moves towards its final goal if the robots are maintaining formation. Recently, Sadowska et al. [14] proposed a controller that contains so-called mutual coupling terms between the robots to ensure robustness of the formation with respect to perturbations.

Besides the virtual structure approach for solving the coordinated path following problem of mobile robots, Ghabcheloo et al. [17] presented a solution to the problem of steering a group of wheeled mobile robots along given spatial paths, while holding a desired inter-vehicle formation pattern with bidirectional communication constraints. Do [18] designed a cooperative controller to force a group of mobile robots with limited sensing ranges to perform desired formation tracking, and to guarantee no collisions between robots using potential functions. Kanjanawanishkul et al. [19] developed a coordinated path following controller based on a Lyapunov function and a second-order consensus protocol with a reference velocity. Xiang et al. [20] addressed the problem of simultaneous path following control, obstacle avoidance and collision free for coordinated vehicles with speed adaptation. Recently, Ghomman et al. [21] developed a decentralized feedback control law that drives each robot to its desired path while adjusting its speed to the nominal velocity profile based on the exchange of information with its neightbors.

In this paper, we introduce a solution for the coordinated path following problem based on a centralized nonlinear MPC approach and a virtual structure strategy. Unlike most MPC controllers which have been employed to solve a trajectory tracking problem, our MPC controller offers two key advantages: (i) integrating the velocity of a virtual leader, $\dot{s}$, into the cost function explicitly to solve the path following problem and (ii) achieving time-convergence requirements.

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 unicycle mobile robot we use in our numerical simulations is as follows:

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

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 denote the linear and angular velocities, respectively.

To make a formation pattern where N unicycle mobile robots, namely $R_i, i = 1, ..., N$, to follow a reference path and to maintain a desired formation shape, we employ an idea of formation configurations in a curvilinear coordinate system [22] where the formation shape can be slightly modified when the formation is turning in such a way that robots on the outside speed up and robots on the inside slow down. This allows the formation to be shape compliant on route as seen in Fig. 1. In our method, the reference path which the virtual leader vehicle follows is first prespecified and parameterized by the path variable s_c. Then each actual follower robot i in the group has a predetermined coordinate $(p_i(s), q_i(s))$ where q_i is the offset distance perpendicular to the reference path at s_i (see Fig. 1) relative to the reference point C (the location of the virtual leader robot) such that $s_i = s_c + p_i(s)$. Thus, we can

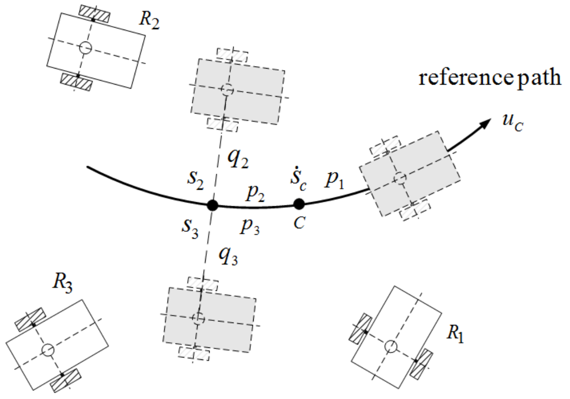


Fig. 1. Graphical depiction of mobile robots making a formation and following a reference path with offset quantities. Point C denotes the location of a virtual leader robot and $R_1 - R_3$ denote actual follower robots.

compute the desired pose of robot i by using the following equations:

$$\mathbf{x}_{i,r} = \begin{bmatrix} x_{i,r} \\ y_{i,r} \\ \theta_{i,r} \end{bmatrix} = \begin{bmatrix} x_{i,p} - q_i \sin\theta_{i,p} \\ y_{i,p} + q_i \cos\theta_{i,p} \\ \theta_{i,p} \end{bmatrix} \tag{2}$$

where $[x_{i,p},\ y_{i,p},\ \theta_{i,p}]^T$ is the desired state vector of robot i at s_i and $[x_{i,r},\ y_{i,r},\ \theta_{i,r}]^T$ is the vector of the desired reference pose at s_i with offset q_i (see Fig. 2). Then the desired velocity of follower robot i can be obtained by

$$\dot{l}_i = H_i \dot{s}_i \tag{3}$$

$$\dot{x}_{i,r} = \dot{l}_i \cos\theta_{i,r} \tag{4}$$

$$\dot{y}_{i,r} = \dot{l}_i \sin\theta_{i,r} \tag{5}$$

$$\omega_i = \kappa_i \dot{l}_i \tag{6}$$

where

$$\begin{aligned} &\kappa_i = \text{sign}(b)\tfrac{\sqrt{a^2+b^2}}{H^2} \\ &H = \sqrt{(1-\kappa_p q)^2 + (\tfrac{dq}{ds})^2} \\ &a = -2\kappa_p \tfrac{dq}{ds} - q\tfrac{d\kappa_p}{ds} - (1-\kappa_p q)\tfrac{G}{H^2} \\ &b = \kappa_p - \kappa_p^2 q + \tfrac{d^2q}{ds^2} - \tfrac{dq}{ds}\tfrac{G}{H^2} \\ &G = (1-\kappa_p q)(-\kappa_p \tfrac{dq}{ds} - q\tfrac{d\kappa_p}{ds}) + \tfrac{dq}{ds}\tfrac{d^2q}{ds^2} \end{aligned}$$

$\dot{l}_i$, ω_i and κ_i are the desired translational velocity, the desired rotational velocity and the desired curvature of follower robot *i*, respectively. κ_p is the curvature at s_i on the reference path. The derivations can be found in [23]. It has to be noted that (3) and (6) are not valid if cusp or singularity at the robot's path is present.

Furthermore, to deal with path following problems, we introduce an additional variable state η_e where $\eta_e = \dot{s}_c - u_c$ and u_c is the desired forward velocity of the whole formation moving along the reference path. We can then obtain the following equation:

$$\dot{\eta}_e = \ddot{s}_c - \dot{u}_c \tag{7}$$

where $\ddot{s}_c$ is an acceleration control input of the virtual leader vehicle.

Therefore, we generally wish to find the control laws of $\ddot{s}_c$, $v_{i,m}$ and $\omega_{i,m}$ such that the robot follows the desired reference pose with $\mathbf{x}_{i,r} = [x_{i,r}, y_{i,r}, \theta_{i,r}]^T$. The kinematic model of a mobile robot can be formulated with respect to the Frenet frame moving along the reference path. This frame plays the role of the body frame of the virtual leader vehicle that must be followed by the entire formation of actual robots as depicted in Fig. 1. The error state vector between $\mathbf{x}_{i,m}$ and $\mathbf{x}_{i,r}$ can be expressed in the frame of the path coordinate as follows:

$$\begin{bmatrix} x_{i,e} \\ y_{i,e} \\ \theta_{i,e} \end{bmatrix} = \begin{bmatrix} \cos\theta_{i,r} & \sin\theta_{i,r} & 0 \\ -\sin\theta_{i,r} & \cos\theta_{i,r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i,m} - x_{i,r} \\ y_{i,m} - y_{i,r} \\ \theta_{i,m} - \theta_{i,r} \end{bmatrix}. \tag{8}$$

Using (1) - (8), the error state dynamic model chosen in a rotated coordinate frame becomes

$$\begin{aligned} \dot{x}_{i,e} &= y_{i,e} H_i \dot{s}_i \kappa_i - H_i \dot{s}_i + v_{i,m}\cos\theta_{i,e} \\ \dot{y}_{i,e} &= -x_{i,e} H_i \dot{s}_i \kappa_i + v_{i,m}\sin\theta_{i,e} \\ \dot{\theta}_{i,e} &= \omega_{i,m} - H_i \dot{s}_i \kappa_i \\ \ddot{s}_i &= \ddot{s}_c + \ddot{p}_i \\ \dot{\eta}_e &= \ddot{s}_c - \dot{u}_c \end{aligned} . \tag{9}$$

We redefine the control signals as follows:

$$\mathbf{u}_{i,e} = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ u_{i,3} \end{bmatrix} = \begin{bmatrix} -H_i \dot{s}_i + v_{i,m}\cos\theta_{i,e} \\ \omega_{i,m} - H_i \dot{s}_i \kappa_i \\ \ddot{s}_c - \dot{u}_c \end{bmatrix}, \tag{10}$$

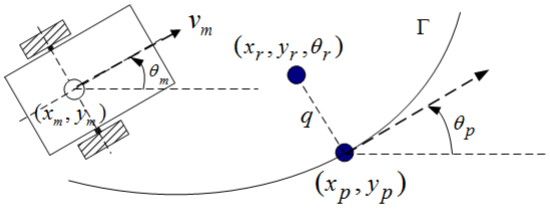


Fig. 2. A graphical representation of a unicycle mobile robot and a reference pose.

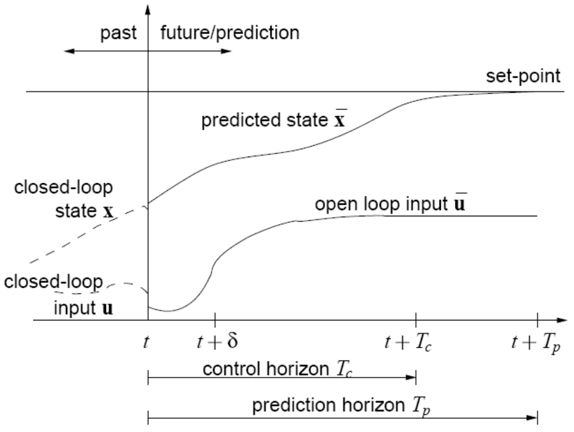

past
future/prediction
set-point
predicted state $\bar{\mathbf{x}}$
closed-loop state $\mathbf{x}$
open loop input $\bar{\mathbf{u}}$
closed-loop input $\mathbf{u}$
t
$t+\delta$
$t+T_c$
$t+T_p$
control horizon T_c
prediction horizon T_p

Fig. 3. Principle of model predictive control [24].

and the error state dynamic model then becomes

$$\dot{\mathbf{x}}_{i,e} = \begin{bmatrix} \dot{x}_{i,e} \\ \dot{y}_{i,e} \\ \dot{\theta}_{i,e} \\ \dot{\eta}_e \end{bmatrix} = \begin{bmatrix} 0 & H_i \dot{s}_i \kappa_i & 0 & 0 \\ -H_i \dot{s}_i \kappa_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{i,e} \\ y_{i,e} \\ \theta_{i,e} \\ \eta_e \end{bmatrix} + \begin{bmatrix} u_{i,1} \\ v_{i,m} \sin\theta_{i,e} \\ u_{i,2} \\ u_{i,3} \end{bmatrix}. \tag{11}$$

3 THE VIRTUAL STRUCTURE STRATEGY WITH MPC

The conceptual structure of MPC is depicted in Fig. 3. As its name suggests, an MPC algorithm uses an explicit model of the plant for predicting the future output behavior. This prediction capability allows computing a sequence of manipulated variables 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 [24,25]. The result of the optimization is used according to a receding horizon philosophy as follows: At time t only the first input of the optimal command sequence is applied to the plant. The remaining optimal inputs are discarded and then a new optimal control problem is solved at time $t + \delta$, where δ is the sampling period. As new measurements are collected from the plant at each update time, the receding horizon mechanism provides the controller with the feedback characteristics.

Although important issues of linear MPC theory are well addressed by now, linear MPC is not suitable for highly nonlinear systems. Therefore nonlinear models must be used [24]. Besides computation burdens of nonlinear optimization problems, another major concern in the use of MPC is whether such an open-loop controller can guarantee system stability. It is shown that an infinite predictive control horizon can guarantee stability of a system, but the infinite predictive horizon may not be feasible for a nonlinear system in practice [24]. Mayne et al. [25] have presented essential principles for the stability of MPC of constrained dynamical systems. Different approaches to attain closed-loop stability using finite horizon lengths exist. We intentionally do not collect all published contributions because of a large number of publications, we refer the reader to [24, 25].

Although MPC is apparently not a new control method, research studies dealing with MPC of path following problems are rare. Some of them are as follows: Vougioukas [26] presented a reactive path tracking controller based on nonlinear MPC, along with an iterative gradient descent algorithm for its real-time implementation. In the presence of obstacles, the controller deviates from the reference trajectory by incorporating obstacle-distance information from range sensors into the optimization problem. Falcone et al. [27] presented two approaches with different computational complexities for controlling an active front steering system in an autonomous vehicle. In the first approach, the MPC problem is formulated by using a nonlinear vehicle model. The second approach is based on successive online linearization of the vehicle model, resulting in a linear time-varying (LTV) system. Bak et al. [28] proposed a model predictive controller with velocity constraints to avoid excessive overshooting and to have time to decelerate when turning. Recently, Raffo et al. [29] proposed two MPC schemes considering both kinematic and dynamic control in a cascade structure: (1) a state space formulation based on the linearized kinematic model of the error between the real vehicle and a reference vehicle and (2) a generalized predictive control (GPC) scheme based on a local linear model and approximation paths.

This paper differs from other MPC schemes as follows: (i) our MPC scheme can produce an optimal velocity of a virtual leader vehicle to solve path following problems, (ii) we achieve time-convergence requirements for trajectory tracking, and (iii) stability can be guaranteed via a contractive constraint.

3.1 Contractive MPC

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

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \text{subject to:} \quad & \mathbf{x}(t) \in \mathcal{X},\ \mathbf{u}(t) \in \mathcal{U},\ \forall t \geq 0 \end{aligned} \tag{12}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ 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. To achieve our goals, both pose errors of all robot members and path following errors of the formation are included into an objective function, which is minimized at each update time. The input applied to the system is then given by the solution of the following finite horizon open-loop optimal control problem:

$$\min_{\bar{\mathbf{u}}(\cdot)} \int_{t}^{t+T_p} F(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau + V_t \tag{13}$$

$$\begin{aligned} \text{subject to:}\quad & \dot{\bar{\mathbf{x}}}(\tau) = \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)) && \text{(14a)}\\ & \bar{\mathbf{u}}(\tau) \in \mathcal{U} \ \ \forall \tau \in [t, t+T_c] && \text{(14b)}\\ & \bar{\mathbf{x}}(\tau) \in \mathcal{X} \ \ \forall \tau \in [t, t+T_p] && \text{(14c)}\\ & \|\bar{\mathbf{x}}(t+T_p)\|_P \le \alpha \|\bar{\mathbf{x}}(t)\|_P \ \ \alpha \in [0,1) && \text{(14d)} \end{aligned}$$

where

$$F(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \sum_{i=1}^{N} (\bar{\mathbf{x}}_{i,e}^T Q_i \bar{\mathbf{x}}_{i,e} + \bar{\mathbf{u}}_{i,e}^T R_i \bar{\mathbf{u}}_{i,e}). \tag{15}$$

The bar denotes an internal controller variable. T_p represents the length of the prediction horizon or output horizon, and T_c denotes the length of the control horizon or input horizon ($T_c \le T_p$). When $T_p = \infty$, we refer to this as the infinite horizon problem, and similarly, when T_p is finite, as a finite horizon problem. The deviation from the desired values is weighted by the positive definite matrices Q and R.

To achieve both smooth spatial convergence and time convergence, we penalize the objective function (13) with the following time-convergence penalty term, V_t, so that trajectory tracking requirements are fulfilled:

$$V_t = K_t (s_t - \bar{s}(t+T_p))^2 \tag{16}$$

where a positive constant, K_t, weighs the relative importance of convergence in time over spatial convergence to the path. If K_t is set to zero, pure path following is obtained. 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 only applied at end of the predictive horizon in order to avoid aggressive motion.

The constraints in (14a) and (14b) denote the following bounded states and control inputs:

$$\begin{bmatrix} 0 \\ v_{m,\min} \\ \Delta v_{m,\min} \\ \omega_{m,\min} \\ \Delta\omega_{m,\min} \end{bmatrix} \le \begin{bmatrix} \dot{s}_c \\ v_{i,m} \\ \Delta v_{i,m} \\ \omega_{i,m} \\ \Delta\omega_{i,m} \end{bmatrix} \le \begin{bmatrix} \dot{s}_{\max} \\ v_{m,\max} \\ \Delta v_{m,\max} \\ \omega_{m,\max} \\ \Delta\omega_{m,\max} \end{bmatrix}, \tag{17}$$

where $\Delta v_{i,m} = v_{i,m}(t_n) - v_{i,m}(t_{n-1})$ and $\Delta\omega_{i,m} = \omega_{i,m}(t_n) - \omega_{i,m}(t_{n-1})$.

The last inequality end constraint in (14d) is a so-called contractive constraint [30]. This constraint imposed in the optimization at time t requires that the system states at the end of the predictive horizon, $\bar{\mathbf{x}}(t+T_p)$ are contracted in norm with respect to the states at the beginning of the prediction, $\bar{\mathbf{x}}(t)$. The two additional controller parameters which determine how much contraction is required are the so-called contractive parameter, $\alpha \in [0,1)$ and the positive definite matrix P.

3.2 Stability Analysis

Before the stability proof of our MPC framework will be given, the following assumptions based on [30] are needed to ensure stability:

Assumption 1 *There exists a constant $\rho \in (0,\infty)$ such that for all $\boldsymbol{x}(t) \in B_\rho := \{\boldsymbol{x} \in \mathbb{R}^n | \ \|\boldsymbol{x}\|_P \le \rho\}$, 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 Assumption 1 holds and 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, $\boldsymbol{x}(\tau)$, of the model satisfies $\|\boldsymbol{x}(\tau)\|_P \le \beta\|\boldsymbol{x}(t)\|_P$, for all $\tau \in [t, t+T_p]$.*

Note that, if Assumption 2 holds, systems with finite escape time are ruled out. Then, the theorem based on [30] can now be given.

Theorem 1 *Suppose Assumptions 1 and 2 hold, the MPC algorithm described in Subsection 3.1 is exponentially stable in such a way that the resulting trajectory of the closed-loop system satisfies the following inequality*

$$\|\boldsymbol{x}(t)\|_P \le \beta\|\boldsymbol{x}(t_0)\|_P e^{-(1-\alpha)(t-t_0)/T_p}. \tag{18}$$

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

$$\|\mathbf{x}(t_k)\|_P \le \alpha\|\mathbf{x}(t_{k-1})\|_P \le \cdots \le \alpha^k \|\mathbf{x}(t_0)\|_P, \tag{19}$$

where $t_k = t_0 + kT_p$ and k belongs to the set of non-negative integers. From Assumption 2, $\mathbf{x}(t)$ satisfies the following inequality:

$$\|\mathbf{x}(t)\|_P \le \beta\alpha^k \|\mathbf{x}(t_0)\|_P, \ \ \forall t \in [t_k, t_{k+1}]. \tag{20}$$

Since $e^{(\alpha-1)} - \alpha \geq 0$, which means $0 \leq \alpha^k \leq e^{-(1-\alpha)k}$, inequality (20) results that

$$\|\mathbf{x}(t)\|_P \leq \beta \|\mathbf{x}(t_0)\|_P e^{-(1-\alpha)k}. \tag{21}$$

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

$$\beta e^{-(1-\alpha)k} \leq \beta e^{-(1-\alpha)(t-t_0)/T_P}. \tag{22}$$

Therefore, using (21) and (22), we finally have

$$\|\mathbf{x}(t)\|_P \leq \beta \|\mathbf{x}(t_0)\|_P e^{-(1-\alpha)(t-t_0)/T_P}. \tag{23}$$

This concludes the proof. ■

4 SIMULATION RESULTS

Our coordinated path following control framework with a contractive constraint and formation feedback has been evaluated through extensive computer simulations. The following eight-shaped curve is selected as a reference path because its geometrical symmetry and sharp changes in curvature make the test challenging:

$$x_p(t) = \frac{2.3\cos(0.15t)}{1+(\sin(0.15t))^2} \quad y_p(t) = \frac{2.3\sin(0.15t)\cos(0.15t)}{1+(\sin(0.15t))^2}$$

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

$Q_i = \text{diag}(0.1, 10, 0.1, 0.1)$,
$R_i = \text{diag}(0.01, 0.01, 0.01)$,
$P_i = \text{diag}(1, 1, 0.01)$,
$K_t = 1$, $T_c = T_p = 0.15$ s, $\delta = 0.05$ s,
$u_c = 0.2$ m/s, $s(0) = 0$ m, $\dot{s}_{\max} = 1.0$ m/s, $\alpha = 0.99$,
$v_{\text{m,min}} = -1.0$ m/s, $v_{\text{m,max}} = 1.0$ m/s,
$\omega_{\text{m,min}} = -1.0$ rad/s, $\omega_{\text{m,max}} = 1.0$ rad/s,
$\Delta v_{m,\text{min}} = -0.2$ m/s, $\Delta v_{m,\text{max}} = 0.2$ m/s,
$\Delta \omega_{m,\text{min}} = -0.2$ rad/s and $\Delta \omega_{m,\text{max}} = 0.2$ rad/s.

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

4.1 Optimal Forward Velocity of the Virtual Leader Vehicle

Since the forward velocity of the virtual leader vehicle can be used as an extra degree of freedom in this work, we show that it can be optimized in the sense that it can decrease or increase in order to minimize the objective function including robots' pose errors, the deviation from the desired control inputs and the deviation from the desired forward velocity (u_c) of the formation. Figure 4 shows simulation results that three mobile robots from initial pose: $R_1(2.0, -0.2, \pi/2)^T$, $R_2(2.5, -0.3, 2\pi/3)^T$, and $R_3(2.5, 0.3, 0)^T$ are required to follow the given reference path and to form a triangular shape, where $(p_1, q_1) = (-0.2, 0.2)$, $(p_2, q_2) = (-0.2, -0.2)$ and $(p_3, q_3) = (0.2, 0)$ for robot R_1, R_2, and R_3, respectively. The forward velocity and the acceleration control input of the virtual leader vehicle are shown in Fig. 4(d) and Fig. 4(e), respectively. As expected, the forward velocity of the virtual vehicle can be adjusted to meet our requirements. Furthermore, the control input boundaries are not violated as seen in Fig. 4(b) (the results of robot R_1 and R_3 are omitted because of limited space), while the robots' pose errors become close to zero as depicted in Fig. 4(c).

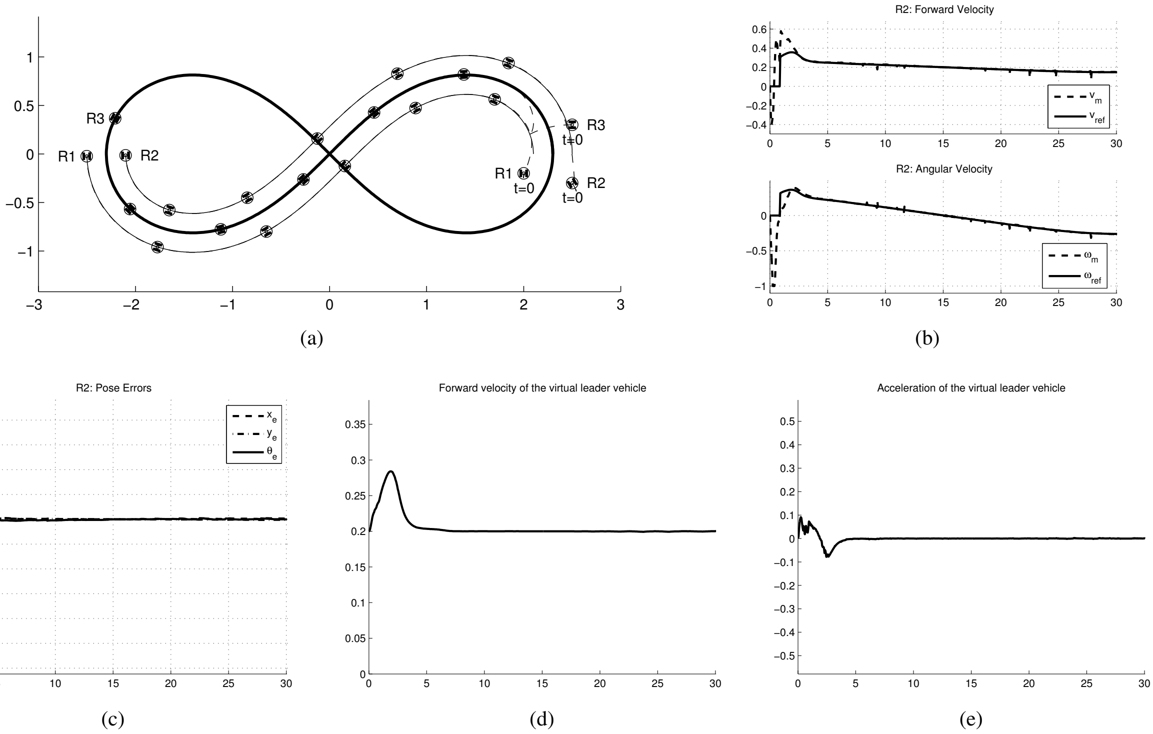
4.2 Parameter Turning

The important tuning parameters of the MPC algorithm include the sampling period δ, the control horizon T_c, the prediction horizon T_p, and the penalty weight matrices Q, R, P, and K_t. To obtain good closed-loop performance, the sampling period should be small enough to capture the process dynamics, while the control horizon must be chosen to provide a balance between performance and computation. Also, stability is strongly affected by the prediction horizon length. The weighting matrices Q, R, P, and K_t can be the most difficult tuning parameters because their values depend both on the scaling of the problem and the relative importance of the variables.

In this subsection, we test our MPC algorithm with three different values of the control horizon (Note that, in this work, we choose $T_c = T_p$). The results of $T_c = 0.15$ s (i.e., 3 steps), $T_c = 0.25$ s (i.e., 5 steps), and $T_c = 0.5$ s (i.e., 10 steps) are shown in Fig. 4(a), Fig. 5(a), and Fig. 5(d), respectively. As seen in Fig. 4(b), Fig. 5(b) and Fig. 5(e), the longer control horizon results in better performance in velocity tracking and better transient response, but leads to an increase of online computation. Thus, the control horizon must be carefully chosen to compromise between performance and computation.

4.3 The Number of Mobile Robots

Since our centralized controller processes all the information needed to achieve the desired control objectives, apparently it will not be suitable for a large number of robots because of high computation time of optimization problems and the nature of inter-robot communication networks. Although it is widely claimed that decentralized systems under local sensing, control, and interactions among robots and environments offer several advantages, including robustness/fault tolerance against single robot failures, natural exploitation of parallelism, and scalability [31], in formation control problems, centralized



(a) (b)

(c) (d) (e)

Fig. 4. The coordinated path following results with $u_c = 0.2$ m/s and 3 steps of predictive horizon (i.e., T_p is 0.15 s): (a) the superimposed snapshots of the formation, (b) the velocity profiles of robot R_2, (c) the pose errors of robot R_2, (d) the forward velocity of the virtual leader vehicle, and (e) the acceleration control input of the virtual leader vehicle.

control laws can ideally yield superior performance and optimal decisions for both the individual members and the formation as a whole. Therefore, in this work, we address the coordinated path following problem of a small number of mobile robots. Figure 6(a) and Figure 6(c) show simulation results of controlling six and twelve mobile robots, respectively, to maintain a desired formation shape. Note that, in the presence of perturbation or under transient conditions, the forward velocity of the virtual leader vehicle is more affected by the number of robots that participate in performing coordinated path following tasks as shown Fig. 6(b) and Fig. 6(d).

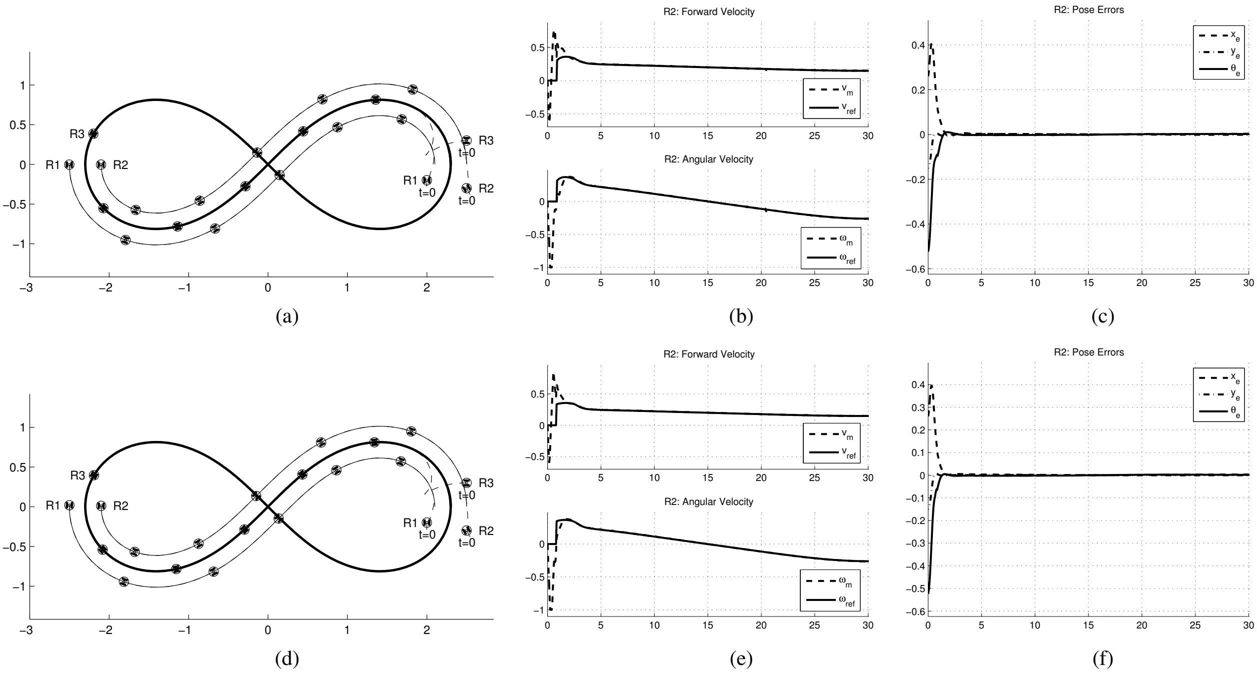
4.4 Time-convergence Penalty

This scenario shows that our controller can achieve both reference convergence in the path following problem and time convergence in the trajectory tracking problem. As mentioned previously, a group of mobile robots is required to track a time-parameterized reference for trajectory tracking. This can be achieved by integrating an additional penalty term V_t in (13) into the objective function, thus fulfilling both path following and trajectory tracking.

Figure 4(a) and Figure 7(a) show the simulation results of pure path following control and pure trajectory tracking control, respectively. Notice that, during transients of path following control, less control signals are pushed to input boundaries ($v_{\text{m,min}} = -1.0$ m/s, $v_{\text{m,max}} = 1.0$ m/s, $\omega_{\text{m,min}} = -1.0$ rad/s, $\omega_{\text{m,max}} = 1.0$ rad/s) and motions are less aggressive as shown in Fig. 4(b), while opposite behaviors of control signals occur in trajectory tracking control as illustrated in Fig. 7(b). However, time constraints are not achieved in the path following control. On the other hand, Figure 7(c) shows the simulation results of our proposed method that can achieve both reference convergence of path following and time convergence of trajectory tracking. As seen in the results, the robots converge smoothly to the desired path, similar to the results in Fig. 4(a) and then they react to achieve zero trajectory tracking error, i.e., the robots reach the same positions and the same velocities as the results of trajectory tracking, shown in Fig. 7(b), at the same time. Furthermore, less control signals are pushed to input boundaries as shown in Fig. 7(d).

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a new approach to satisfy the following four objectives: (i) to solve coordinated path following control with stability guarantee and with formation feedback, (ii) to produce optimal forward velocity of



(a) (b) (c)

(d) (e) (f)

Fig. 5. The path following results by using two different values of the control horizon: (a) $T_c = 0.25$ s (i.e., 5 steps of the control horizon), (b) the velocity profiles of robot R_2 corresponding to (a), (c) the pose errors of robot R_2 corresponding to (a), (d) $T_c = 0.5$ s (i.e., 10 steps of the control horizon), (e) the velocity profiles of robot R_2 corresponding to (d), and (f) the pose errors of robot R_2 corresponding to (d).

a virtual leader vehicle, (iii) to bound control signals, and (iv) to fulfill time convergence requirements. All these objectives are achieved through our MPC framework where the MPC law is used to produce a sequence of control inputs by considering input boundaries, formation feedback, a contractive constraint, and time-convergence requirements. Furthermore, a comparison between path following and trajectory tracking has also been shown and discussed.

Currently, we are developing real mobile robots which can be used to validate our control law in real-world environments. In addition, we will extend our controller to accomplish the coordinated path following task in dynamic environments with static and moving obstacles.

REFERENCES

[1] P. Morin and C. Samson, *Springer Handbook of Robotics*, ch. 34. Motion control of wheeled mobile robot, pp. 799–826. Springer Berlin Heidelberg, 2008.

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

[3] K. Kanjanawanishkul, "Formation control of mobile robots : Survey," *UBU Engineering Journal*, vol. 4, pp. 50–64, Jan.-Jun. 2011.

[4] 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), pp. 1387–1392, June 2003.

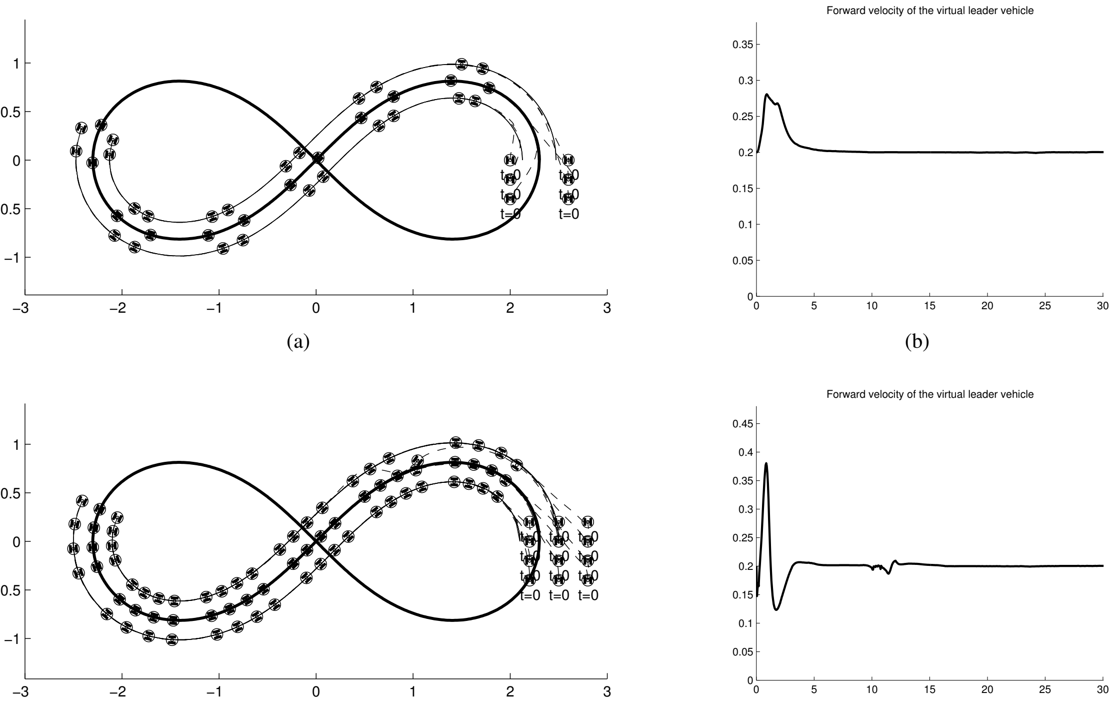
[5] A. P. Aguiar, D. B. Dačić, J. P. Hespanha, and P. Kokotović, "Path-following or reference-tracking? An answer relaxing the limits to performance," in *Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, (Lisbon, Portugal), July 2004.

[6] R. Skjetne, T. Fossen, and P. V. Kokotović, "Robust output maneuvering for a class of nonlinear systems," *Automatica*, vol. 40, no. 3, pp. 373–383, 2004.

[7] A. Micaelli and C. Samson, "Trajectory-tracking for unicycle-type and two-steering-wheels mobile robots," Tech. Rep. 2097, INRIA Sophia-Antipolis, November 1993.

[8] F. Diaz del Rio, G. J. Moreno, J. L. S. Ramos, C. A. A. Rodriguez, and A. A. C. 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), July 2002.

[9] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control*, vol. 46, pp. 1777–1782, November 2001.



(a) (b)

(c) (d)

Fig. 6. (a) The simulation retults with six mobile robots, (b) the forward velocity of the virtual leader vehicle corresponding to (a), (c) the simulation results with twelve mobile robots, and (d) the forward velocity of the virtual leader vehicle corresponding to (c).

[10] K. Kanjanawanishkul, "Formation control of omnidirectional mobile robot using distributed model predictive control," in *Proceedings of the 2nd International Conference on Robot Communication and Coordination (ROBOCOMM)*, (Odense, Denmark), pp. 1–7, March 2009.

[11] M. A. Lewis and K. H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[12] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 947–951, December 2001.

[13] B. Young, R. W. Beard, and J. Kelsey, "A control scheme for improving multivehicle formation maneuvers," in *Proceedings of the American Control Conference*, (Arlington, VA), pp. 704–709, June 2001.

[14] A. Sadowska, T. van den Broek, H. Huijberts, N. van de Wouw, D. Kostic, and H. Nijmeijer, "A virtual structuce approach to formation control of unicycle mobile robots using mutual couplings," *International Journal of Control*, vol. 84, pp. 1886–1902, Nov. 2011.

[15] W. Ren and R. W. Beard, "A decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, pp. 73–82, January 2004.

[16] I.-A. F. Ihle, J. Jouffroy, and T. I. Fossen, "Formation control of marine craft using constraint functions," in *Proceedings of the MTS/IEEE OCEANS*, (Washington D.C.), pp. 1023–1028, September 2005.

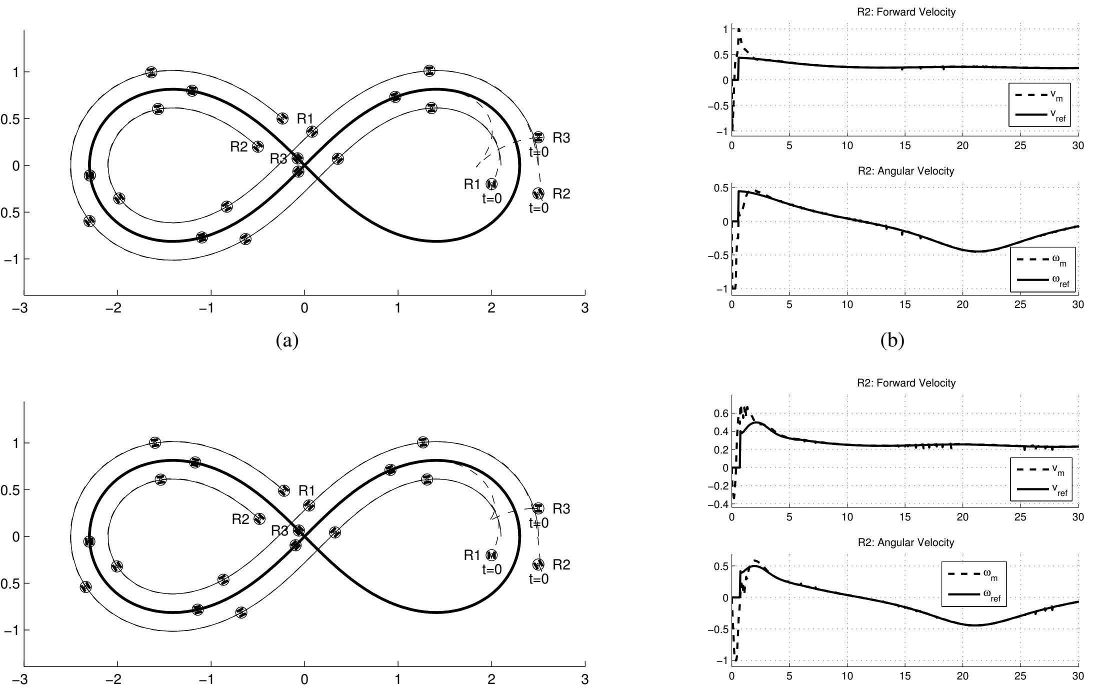
[17] R. Ghabcheloo, A. Pascoal, C. Silvestre, and D. Carvalho, "Coordinated motion control of multiple autonomous underwater vehicles," in *Proceedings of the International Workshop on Underwater Robotics*, (Genoa, Italy), pp. 41–50, November 2005.

[18] K. D. Do, "Formation tracking control of unicycle-type mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Roma, Italy), pp. 2391–2396, April 2007.

[19] K. Kanjanawanishkul, M. Hofmeister, and A. Zell, "Coordinated path following for mobile robots," in *Proceedings of Fachgespraech Autonome Mobile Systeme (AMS2009)*, (Karlsruhe, Germany), pp. 185–192, December 2009.

[20] X. Xiang, L. Lapierre, and B. Jouvencel, "Guidance based collision avoidance of coordinated nonholonomic autonouमous vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Taipei, Taiwan), pp. 6064–6069, October 2010.

[21] J. Ghomman, H. Mehrjerdi, M. Saad, and F. Mnif, "Adaptive coordinated path following control of non-holonomic mobile robots with quantised communication," *IET Control*



(a) (b)

(c) (d)

Fig. 7. The simulation results: (a) pure trajectory tracking, (b) the velocity profiles of robot R_2 corresponding to (a), (c) combined path following and trajectory tracking, and (d) the velocity profiles of robot R_2 corresponding to (c). Note that the robots can reach the same position at the same time in both cases.

Theory and Applications, vol. 5, no. 17, pp. 1990–2004, 2011.

[22] T. D. Barfoot and C. M. Clark, “Motion planning for formations of mobile robots,” *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, 2004.

[23] K. Kanjanawanishkul, X. Li, and A. Zell, “Nonlinear model predictive control of omnidirectional mobile robot formations,” in *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS 2008)*, (Baden-Baden, Germany), pp. 41–48, July 2008.

[24] F. Allgöwer, R. Findeisen, and Z. K. Nagy, “Nonlinear model predictive control: from theory to application,” *Journal of Chinese Institute of Chemical Engineers*, vol. 35, no. 3, pp. 299–315, 2004.

[25] 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.

[26] S. G. Vougioukas, “Reactive trajectory tracking for mobile robots based on non linear model predictive control,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Roma, Italy), pp. 3074–3079, April 2007.

[27] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.

[28] M. Bak, N. K. Poulsen, and O. Ravn, “Path following mobile robot in the presence of velocity constraints,” tech. rep., Informatics and Mathematical Modeling, Technical University of Denmark, 2001.

[29] G. Raffo, G. Gomes, J. Normey-Rico, C. Kelber, and L. Becker, “A predictive controller for autonomous vehicle path tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 92–102, March 2009.

[30] S. L. D. Kothare and M. Morari, “Contractive model predictive control for constrained nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 45, pp. 1053–1071, June 2000.

[31] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, “Cooperative mobile robotics: antecedents and directions,” *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.

Kiattisin Kanjanawanishkul was born in Trang, Thailand, in 1977. He received the B.Eng. in Electrical Engineering from Prince of Songkla University, Thailand in 2000. He received the M.Sc. in Mechatronics from University of Siegen, Germany in 2006. He received his Ph.D. in Computer Science from University of Tuebingen, Germany in 2010. Since 2010, he has been employed at the Faculty of Engineering, University of Mahasarakham, Thailand. His current research interests include cooperative and distributed control, model predictive control, intelligent control, multi-robot systems, and robotic motion control.

AUTHORS' ADDRESSES

Kiattisin Kanjanawanishkul, Ph.D.
Mechatronics Research Unit,
Faculty of Engineering,
Mahasarakham University,
Kamriang, Kantharawichai, Mahasarakham
44150, Thailand
email: kiattisin.k@msu.ac.th

Received: 2013-01-13
Accepted: 2014-07-25