Journal of Computing and Information Technology - CIT 17, 2009, 1, 37–50
doi:10.2498/cit.1001188

Stabilization of Nonholonomic Robot Formations: A First-state Contractive Model Predictive Control Approach

Feng Xie[1] and Rafael Fierro[2]

[1]School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, USA
[2]MARHES Lab, Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, USA

A model predictive control algorithm is developed for stabilizing a team of nonholonomic mobile robots navigating in formation within an obstacle-populated environment. In this scenario, the *leader* robot may need to execute abrupt maneuvers (i.e., sudden stops and backward motions) in order to avoid collisions and accomplish mission objectives. Moreover, *follower* robots should be capable of tracking their leaders and maintaining desired relative distance and orientation. To this end, nonholonomic trajectory tracking and point stabilization should be combined in a suitable way. Most proposed control algorithms for nonholonomic robots do not have simultaneous tracking and point stabilization capability; therefore, they may perform poorly when the leader robot executes aggressive maneuvers. In this paper, we address this problem by applying model predictive control (MPC). Motivated by the *contractive* MPC scheme developed in [14], the proposed algorithm guarantees its stability by adding a contractive constraint on the first state at the beginning of the prediction horizon. The resulting MPC scheme is denoted as *first-state contractive* MPC (FSC-MPC). Simulation results are provided to verify the effectiveness of the method.

Keywords: nonholonomic mobile robots, model predictive control, formation control, trajectory tracking, point stabilization

1. Introduction

Cooperative decision and control of mobile robot teams have been of great interest in civilian and military applications, as these teams are expected to be capable of performing many key roles in future civilian or battlefield missions. A lot of effort has been put into this area and the use of mobile robots is likely to increase rapidly in the near future. Applications for cooperative control of multi-robot systems include [31, 12]: Formation control, cooperative classification and surveillance, cooperative attack and rendezvous, environmental sampling, intelligent highways, and air traffic control.

Formation control was inspired by the emergent self-organization observed in nature, like flocking birds and schooling fish. Each animal in a herd benefits by maximizing the chance of finding food and minimizing its encounters with predators. Teams of UAVs moving in formations with precisely defined geometries lead to many advantages, such as energy saving when the vortex forces are taken into account. Several experimental studies have verified the energy saved when flying in close formations [37]. In addition, formation control allows for intelligent *leaders* and single agent planning while *followers* can focus on other tasks. The challenge here lies in designing a formation controller that is computationally simple, yet robust.

Leader-following is a common approach to build formations of multivehicle systems [13] and a nonholonomic model (*e.g.*, unicycle) is wildly adopted to describe vehicle's kinematics in this approach. However, how to stabilize the followers when the formation leader i) stops or ii) moves backward? This problem is often parried in the stabilization of nonholonomic robot formations with leader-following approach. The controller design is still a challenge. To handle this problem, we inevitably need to face the fundamental control problems of nonholonomic mobile robots, trajectory tracking and point stabilization.

During the past decades, trajectory tracking and point stabilization have received a lot of attention and numerous control algorithms can be found in the existing literature. For interested readers, reference [32] is a good starting point for understanding nonholonomic systems.

The trajectory tracking problem focuses on stabilizing robots to a time-varying trajectory. Nonlinear feedback controllers are mostly found in the literature. Early results on local controllers can be found in [22, 15] using Lyapunov direct method. The problem is globally solved in [36] by nonlinear feedback. Dynamic feedback linearization is applied in [33]. Other techniques include approximate linearization [38], sliding mode control [10] and backstepping [17]. However, a major restriction is that the tracked linear velocity and angular velocity must not converge to zero, which means the tracked trajectory must be continuously excited. This restriction makes it difficult for a single controller to handle the tracking problem and point stabilization problem simultaneously. Consequently, the range of applications of the above mentioned controllers is limited. In addition, the internal dynamics of the closed-loop system under output feedback linearization controllers exhibit unstable properties when robots track a trajectory moving backward. To overcome this issue, full-state tracking techniques are explored in [40]. Model predictive controllers are reported in [19] for trajectory tracking. The MPC approach is based on the quasi-infinite algorithm proposed by authors in [9].

The point stabilization problem, which considers stabilizing robots to a final goal posture, is more challenging. A smooth time-invariant feedback control law does not exist according to Brockett's theorem [7]. Most of the control algorithms for point stabilization can be classified into three categories: (i) Smooth time-varying algorithms, (ii) discontinuous or piecewise feedback laws, and (iii) hybrid strategies (discontinuous and time-varying). Smooth time-varying stabilization is pioneered by Samson [35] and explored by other authors in [34, 32]. For discontinuous feedback controllers, see [24, 2]. Hybrid strategies include [28]. In general, time-varying control laws are complex and only for a special class of nonholonomic systems a general design methodology is available. Moreover, time-varying control laws produce very slow convergence and are intrinsically oscillating. A comparative experimental study of time-varying and discontinuous controllers for nonholonomic mobile robots is reported in [25]. Other techniques, such as dynamic feedback linearization [33], predictive motion control [39], model predictive control [20], adaptive [11], neural networks [18], and backstepping [17] are also found in the literature. Interesting results are given in [1] and [21]. With a special choice of the system state variables in Polar-like coordinates, smooth feedback control laws can be globally stable. Furthermore, the problem of designing controllers with simultaneous tracking and stabilization capabilities can be found in [16, 30, 33, 19].

In this paper, we address the problem of stabilizing a team of robots in leader-following formation when the leader exhibits aggressive maneuvers such as abrupt stops and backward motions. These maneuvers may be required to avoid collisions when the team is navigating within a cluttered environment. We propose an MPC algorithm that guarantees its stability by adding a *contractive* constraint on the *first state* at the beginning of the prediction horizon. The key idea is to take advantage of the simultaneous tracking and stabilization capability of the proposed FSC-MPC.

The rest of the paper is organized as follows. Section 2 introduces the robot kinematic model and trajectory tracking and point stabilization problems of a nonholonomic mobile robot. A *first-state contractive* MPC algorithm is proposed in Section 3. Stability results of the proposed algorithm are found in Section 4. In Section 5, simulation results are provided to show the effectiveness of the method. An extension to multi-robot formations is described in Section 6. Finally, concluding remarks and future work are given in Section 7.

2. Preliminaries

This paper deals with the problem of designing control laws for the motion control of nonholonomic mobile robots. This section presents a brief introduction of the kinematic model used for the mobile robots and the two fundamental classes of problems considered herein, trajectory tracking and point stabilization.

2.1. Kinematic Model

Consider the planar motion of mobile robots under the nonholonomic constraint of pure rolling and non-slipping, the kinematic model is given as follows (see Figure 1)

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

where $(x, y) \in \mathbf{R}^2$ denotes the position of the robot in a Cartesian coordinate frame, $\theta \in (-\pi, \pi]$ represents the orientation of the robot with respect to the positive X axis, and $v \in \mathcal{V} \subseteq \mathbf{R}$ and $\omega \in \mathcal{W} \subseteq \mathbf{R}$ are the control inputs representing linear and angular velocities, respectively.

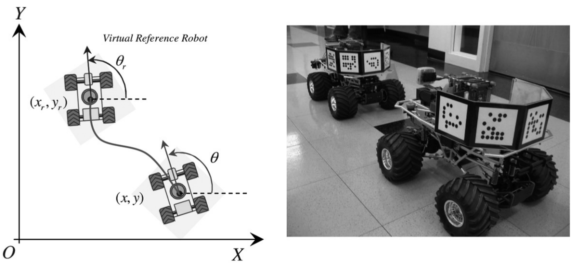

Figure 1. Nonholonomic mobile robots.

Although there is no consideration of motor dynamics and other mechanical effects, this simplified model is sufficient to describe the nonholonomic mobile robots' motion.

Since system (1) falls in the form of driftless systems

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{2}$$

and the accessibility rank condition is globally satisfied [5]

$$\begin{aligned} \operatorname{rank}\{g_1, g_2, [g_1, g_2]\} &= \operatorname{rank} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta & 0 & -\cos\theta \\ 0 & 1 & 0 \end{bmatrix} \\ &= 3, \end{aligned} \tag{3}$$

where $g_1 = [\cos\theta \;\; \sin\theta \;\; 0]^T$, $g_2 = [0 \;\; 0 \;\; 1]^T$ and $[g_1, g_2]$ is the Lie bracket of g_1 and g_2, system (1) is controllable. Note that, for nonlinear systems, the existence of continuous time-invariant state feedback control laws cannot be implied by controllability.

2.2. Trajectory Tracking

Let a triplet $\mathbf{z}_c = [x \;\; y \;\; \theta]^T$ describe the position and the orientation of a mobile robot. The reference trajectory can be described by a virtual reference robot with a state vector $\mathbf{z}_r = [x_r \;\; y_r \;\; \theta_r]^T$, an input vector $\mathbf{u}_r = [v_r \;\; \omega_r]^T$ and the kinematic model (see Figure 1)

$$\dot{\mathbf{z}}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_r. \tag{4}$$

Then the trajectory tracking problem can be stated [15].

Problem 1. *The trajectory tracking problem, under the assumption that the virtual reference robot is not at rest ($v_r = \omega_r = 0$) when $t \to \infty$, is to find a feedback control law $\mathbf{u} = [v \;\; \omega]^T$, such that*

$$\lim_{t\to\infty} (\mathbf{z}_r - \mathbf{z}_c) = 0,$$

with any initial robot posture $\mathbf{z}_c(0)$.

By transforming the reference state $\mathbf{z}_r$ in a local coordinate system attached to the tracking robot, an error state $\mathbf{z}_e$ can be defined as follows [22]

$$\mathbf{z}_e := \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{z}_r - \mathbf{z}_c). \tag{5}$$

Taking the derivative of (5) and rearranging with (1), (4), the error model becomes

$$\begin{aligned} \dot{x}_e &= \omega y_e - v + v_r \cos\theta_e, \\ \dot{y}_e &= -\omega x_e + v_r \sin\theta_e, \\ \dot{\theta}_e &= \omega_r - \omega. \end{aligned} \tag{6}$$

Let us define $\mathbf{u}_e$,

$$\mathbf{u}_e := \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -v + v_r \cos\theta_e \\ \omega_r - \omega \end{bmatrix}, \tag{7}$$

then the error model (6) becomes

$$\dot{\mathbf{z}}_e = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z}_e + \begin{bmatrix} 0 \\ v_r \sin\theta_e \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_e. \tag{8}$$

By linearizing system (8) about the the equilibrium point ($\mathbf{z}_e = 0, \mathbf{u}_e = 0$), we obtain

$$\dot{\mathbf{z}}_e = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z}_e + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_e. \qquad (9)$$

The controllability of system (9) can be easily checked. However, when the virtual reference robot stops ($v_r = \omega_r = 0$), the controllable property is lost.

2.3. Point Stabilization

For the point stabilization problem, one can use the following definition.

Problem 2. *Given an arbitrary constant reference position and orientation $\mathbf{z}_d = [x_d \;\; y_d \;\; \theta_d]^T$, the point stabilization problem is to find a feedback control law $\mathbf{u} = [v \;\; \omega]^T$, such that*

$$\lim_{t \to \infty} (\mathbf{z}_d - \mathbf{z}_c) = 0,$$

with any initial robot posture $\mathbf{z}_c(0)$.

Without loss of generality, we use $\mathbf{z}_d = [0 \;\; 0 \;\; 0]^T$. Then the problem becomes finding a feedback control law which drives system (1) back to the origin. It is a well-known fact that a smooth time-invariant feedback control law does not exist for the point stabilization problem [7]. However, with the analysis in Section 2.1, the system (1) is still controllable.

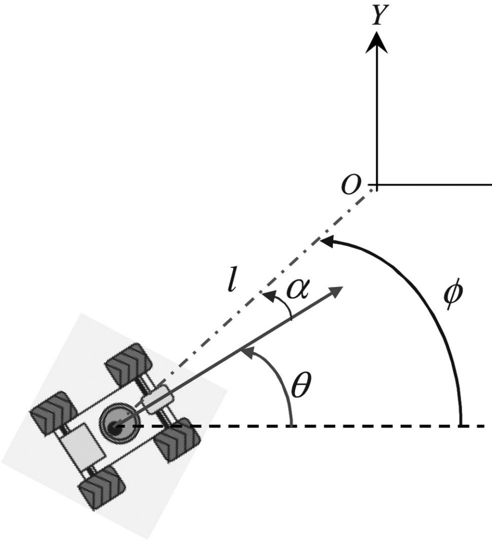

Figure 2. Coordinate transformation.

Consider a Cartesian to Polar coordinate transformation [1] (see Figure 2), a polar state $\mathbf{z}_q = [l \;\; \phi \;\; \alpha]^T$ can be defined by

$$\begin{aligned} l &:= \sqrt{x^2 + y^2} \\ \phi &:= \operatorname{arctan2}(y, x) \\ \alpha &:= \phi - \theta \end{aligned} \qquad (10)$$

Then the kinematic model (1) becomes

$$\begin{aligned} \dot{l} &= -v \cos \alpha \\ \dot{\phi} &= \frac{v \sin \alpha}{l} \\ \dot{\alpha} &= -\omega + \frac{v \sin \alpha}{l} \end{aligned} \qquad (11)$$

Note, when $l = 0$, which means that the robot reaches the origin, the new kinematic model is not defined.

3. First-state Contractive MPC

Without considering disturbances and model uncertainties, systems like (6) and (11) can be generally expressed by the following nonlinear set of differential equations

$$\dot{\mathbf{z}}(t) = \mathbf{h}(\mathbf{z}(t), \mathbf{u}(t)), \quad \mathbf{z}(0) = \mathbf{z}_0, \qquad (12)$$

with a state vector $\mathbf{z}(t) \in \mathbf{R}^m$ and an input vector $\mathbf{u}(t) \in \mathbf{R}^n$, $m, n \in \mathbf{N}$. Function $\mathbf{h} : \mathbf{R}^m \times \mathbf{R}^n \to \mathbf{R}^m$ is assumed to be continuous.

Since the control system is usually implemented on a computer in discrete time, (12) can be converted into

$$\mathbf{z}(k+1) = \mathbf{f}(\mathbf{z}(k), \mathbf{u}(k)), \quad \mathbf{z}(0) = \mathbf{z}_0, \qquad (13)$$

with a state vector $\mathbf{z}(k) \in \mathcal{Z}$ and an input vector $\mathbf{u}(k) \in \mathcal{U}$, $k \in \mathbf{Z}^*$. $\mathcal{Z} \subset \mathbf{R}^m$ is the state constraint which contains the origin in its interior. $\mathcal{U} \subset \mathbf{R}^n$ is the input constraint which is a compact subset of $\mathbf{R}^n$ containing the origin in its interior. Usually, we have $\mathcal{U} = \{\mathbf{u} \in \mathbf{R}^n : \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\}$. $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ are known constants in $\mathbf{R}^n$. Function $\mathbf{f} : \mathbf{R}^m \times \mathbf{R}^n \to \mathbf{R}^m$ is assumed to be continuous.

The control goal is to find a control sequence $\mathbf{u}(k)$ which drives the system (13) toward the equilibrium ($\mathbf{z}(k) = 0$ and $\mathbf{u}(k) = 0$). To obtain the current control $\mathbf{u}(k)$ at time t_k, where k is a non-negative integer ($k \in \mathbf{Z}^*$), a finite-horizon optimal control problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_H(\mathbf{z}, k, \mathbf{u}), \\ \text{subject to:} \quad & \mathbf{z}(k+1) = \mathbf{f}(\mathbf{z}(k), \mathbf{u}(k)), \\ & \mathbf{z}(k) \in \mathcal{Z}, \\ & \mathbf{u}(k) \in \mathcal{U}, \end{aligned} \tag{14}$$

must be solved online for an MPC algorithm. The performance index $J_H(\mathbf{z}, k, \mathbf{u})$ is defined as

$$J_H(\mathbf{z}, k, \mathbf{u}) := \sum_{i=1}^{H} L(\mathbf{z}(k+i;k), \mathbf{u}(k+i-1;k)), \tag{15}$$

where $H \in \mathbf{N}$ is the horizon length (for simplicity, in this paper the prediction horizon equals the control horizon). The incremental cost is defined by

$$L(\mathbf{z}, \mathbf{u}) := \|\mathbf{z}\|_Q^2 + \|\mathbf{u}\|_R^2, \tag{16}$$

where $\|\mathbf{z}\|_Q$ and $\|\mathbf{u}\|_R$ denote the weighted 2-norm, which are defined as $\|\mathbf{z}\|_Q^2 := \mathbf{z}^T Q \mathbf{z}$ and $\|\mathbf{u}\|_R^2 := \mathbf{u}^T R \mathbf{u}$. Q and R are positive definite symmetric matrices of appropriate dimensions.

Since a finite horizon is used, the controller found in (14) is not guaranteed to be stable. Many researchers have contributed to the stability of nonlinear MPC with some important methods. In [26], authors propose that by adding a *terminal state equality constraint* $\mathbf{z}(k+H) = 0$ to the optimal control problem, the stability can be guaranteed. However, strong assumptions on the optimal control problem are required and the computational load is huge in a nonlinear case. In order to overcome those difficulties, a *terminal state inequality constraint* $\mathbf{z}(k+H) \in \Omega$ can take the place of *terminal state equality constraint* [29]. Ω, a subset of $\mathbf{R}^m$ containing the origin in its interior is called *terminal region*. A *terminal controller* is switched in if the state vector is inside the *terminal region*. This method is denoted as *dual-mode* MPC. A similar approach, *quasi-infinite* MPC, is proposed in [9]. In addition to the *terminal inequality constraint*, a terminal penalty $g(\mathbf{z}(k+H))$ is added to the performance index to assure stability of the system. A *terminal controller* is still required, but it is never applied to the system in the *quasi-infinite* MPC scheme. Although the computational load can be reduced, the difficulty of *dual-mode* MPC and *quasi-infinite* MPC algorithms lies in calculating the *terminal region* (upper bounded by the *region of attraction* of the *terminal controller*). An MPC scheme without *terminal region* requirements is proposed by authors in [14]. A *terminal contractive constraint* $g(\mathbf{z}(k+H)) \leq \rho g(\mathbf{z}(k))$, where $\rho \in (0,\ 1)$ is called *contractive parameter*, takes the place of *terminal inequality constraint*. Global exponential stability of the closed-loop system can be guaranteed. The reader is referred to [4] for a formal definition and further details of contractive and invariant sets. Most recently a robust model predictive control algorithm blended with a reactive safety mode has been introduced in [8] for uncertain nonlinear systems. See [27] for a more detailed discussion on stability of MPC.

To achieve stability, the core idea behind the methods mentioned above is to add terminal state penalties in the performance index and impose constraints on the terminal state at the end of the prediction horizon. Therefore, those methods can be denoted as *terminal-state constrained* MPC (TSC-MPC). However, in the implementation of most MPC schemes, only the first control of the control sequence yield by optimization is applied to the plant at each sampling instant. All the other controls are discarded. Only the first state at the beginning of the prediction horizon is directly affected by this implementation.

Motivated by this observation and the *contractive* MPC scheme developed in [14], in this paper we obtain the current control $\mathbf{u}(k)$ at time t_k by solving the following finite-horizon optimal control problem online

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_H(\mathbf{z}, k, \mathbf{u}), \\ \text{subject to:} \quad & \mathbf{z}(k+1) = \mathbf{f}(\mathbf{z}(k), \mathbf{u}(k)), \\ & \mathbf{z}(k) \in \mathcal{Z}, \\ & \mathbf{u}(k) \in \mathcal{U}, \\ & \|\mathbf{z}(k+1)\|_{\hat{P}} \leq \rho \|\mathbf{z}(k)\|_{\hat{P}}, \end{aligned} \tag{17}$$

where $J_H(\mathbf{z}, k, \mathbf{u})$ is defined in (15) and (16). $\hat{P}$ is a positive definite symmetric matrix and $\rho \in (0,\ 1)$.

Note that the last inequality constraint in (17) can be called *first-state contractive* constraint.

This means that the first state at the beginning of the prediction horizon, $\mathbf{z}(k+1)$ is contracted in norm with respect to the current state, $\mathbf{z}(k)$. Therefore, the proposed MPC algorithm can be denoted as *first-state contractive* MPC (FSC-MPC). The user adjustable parameter ρ is called *contractive parameter*, which addresses the contraction rate.

The FSC-MPC controller can be implemented as follows.

FSC-MPC Algorithm

Data: prediction horizon $H \in \mathbf{N}$; sampling time $\delta_T \in \mathbf{R}^+$; weights Q, R, $\hat{P} > 0$; constraints $\mathbf{u}_{\min}, \mathbf{u}_{\max} \in \mathbf{R}^m$; contractive parameter $\rho \in (0,\ 1)$; $k \in \mathbf{Z}^*$.

Step 0: set $k = 0$; set initial control prediction $\hat{\mathbf{u}}(i;k) = 0, i \in [1, \ldots, H-1]$.

Step 1: measure the states $\mathbf{z}(k)$ at time t_k; with control prediction $\hat{\mathbf{u}}(i;k)$, solve the optimal control problem (17) and obtain a control sequence $\mathbf{u}^*(i;k)$.

Step 2: apply the first control $\mathbf{u}^*(1;k)$ in the control sequence $\mathbf{u}^*(i;k)$ to system (13) for the time interval $[t_k,\ t_{k+1}]$, where $t_{k+1} = t_k + \delta_T$.

Step 3: update the control prediction as follows

$$\hat{\mathbf{u}}(i;k) = \begin{cases} \mathbf{u}^*(i+1;k) & i \in [1, \ldots, H-2] \\ \mathbf{u}^*(i;k) & i = H-1 \end{cases};$$

set $k = k+1$; go back to **Step 1**.

Note, in **Step 1**, an assumption is that for all $k \in \mathbf{Z}^*$, a feasible solution of the optimal control problem (17), satisfying all the constraints, always exists. However, globally optimal solution is not strictly required here. Locally optimal or less optimal solutions are acceptable. This approach might compromise the performance, but the stability property of the algorithm will not be affected.

4. Stability Results

In this section, the stability of the FSC-MPC algorithm will be proven. Before we give the main result, let's make the following assumptions [14].

Assumption 3. *There exists a constant $\beta \in (0,\ \infty)$ such that for all $\mathbf{z}(k) \in B_\beta := \{\mathbf{z} \in \mathcal{Z} \,|\, \|\mathbf{z}\|_{\hat{P}} \le \beta\}$, a contractive parameter $\rho \in (0,\ 1)$ can be found so that at time t_k, a feasible solution of the optimal control problem (17), satisfying all the constraints, always exists for all $k \in \mathbf{Z}^*$.*

Note that Assumption 3 means that there is a nonempty convex and compact set of initial conditions such that the optimal control problem is feasible. It is restrictive but still acceptable. As pointed out by authors in [6], if $[(\partial\mathbf{h}/\partial\mathbf{z})(\partial\mathbf{h}/\partial\mathbf{u})] \in \Omega$, $\forall \mathbf{z}$, $\mathbf{u}$, where $\Omega \subseteq \mathbf{R}^{m\times(m+n)}$, the nonlinear system can be replaced by a time-varying linear system in *linear differential inclusions* form by using *global linearization* techniques. Then sufficient conditions for Assumption 3 can be found by using a single quadratic Lyapunov function (details can be found in [3], where sufficient conditions for exponential stability and an induced $L_2 - norm$ performance objective are given).

Assumption 4. *For all $t \in [t_k,\ t_{k+1}]$, $k \in \mathbf{Z}^*$, there exists a constant $\kappa \in (0,\ \infty)$, such that the transient state, $\mathbf{z}(t)$, satisfies $\|\mathbf{z}(t)\|_{\hat{P}} \le \kappa\|\mathbf{z}(k)\|_{\hat{P}}$.*

Note that Assumption 4 means that systems with finite escape time are not under consideration.

Theorem 5. *Suppose that the optimal control problem is feasible at time t_0 and Assumptions 3 and 4 are satisfied. The FSC-MPC algorithm described in Section 3 for system (12), (13) is exponentially stable in the sense that the state trajectory of the closed-loop system satisfies the following inequality*

$$\|\mathbf{z}(t)\|_{\hat{P}} \le \kappa\|\mathbf{z}(0)\|_{\hat{P}} e^{-\frac{(1-\rho)}{\delta_T}(t-t_0)}, \tag{18}$$

where δ_T is the sampling time.

Proof: Since the optimal control problem is feasible at time t_0, from Assumption 3, the optimal control problem is feasible at time t_k, $k \in \mathbf{Z}^*$. Therefore, we have

$$\|\mathbf{z}(k)\|_{\hat{P}} \le \rho\|\mathbf{z}(k-1)\|_{\hat{P}} \le \cdots \le \rho^k\|\mathbf{z}(0)\|_{\hat{P}}. \tag{19}$$

Now with Assumption 4 and (19), $\mathbf{z}(t)$ satisfies the following inequality

$$\|\mathbf{z}(t)\|_{\hat{P}} \leq \kappa \rho^{k} \|\mathbf{z}(0)\|_{\hat{P}}, \tag{20}$$

where $t \in [t_k,\ t_{k+1}]$, for all $k \in \mathbf{Z}^{*}$.

Since $\rho \in (0,\ 1)$, we have $e^{(\rho-1)} - \rho \geq 0$, which means $e^{(\rho-1)k} \geq \rho^{k} \geq 0$, for all $k \in \mathbf{Z}^{*}$. Inequality (20) can be rewritten as follows

$$\|\mathbf{z}(t)\|_{\hat{P}} \leq \kappa \|\mathbf{z}(0)\|_{\hat{P}} e^{-(1-\rho)k}. \tag{21}$$

Since $k = (t_k - t_0)/\delta_T$ and $(t - t_0)/\delta_T \leq (t_k - t_0)/\delta_T = k$, for all $t \in [t_0,\ t_k]$, we have

$$e^{-(1-\rho)k} \leq e^{-\frac{(1-\rho)}{\delta_T}(t-t_0)} \tag{22}$$

Therefore, from inequalities (21) and (22), we conclude

$$\|\mathbf{z}(t)\|_{\hat{P}} \leq \kappa \|\mathbf{z}(0)\|_{\hat{P}} e^{-\frac{(1-\rho)}{\delta_T}(t-t_0)}.$$

According to [23], the closed-loop system is exponentially stable, so is the FSC-MPC algorithm. ■

5. Simulation Results

The effectiveness of the FSC-MPC algorithm presented in Section 3 is investigated by numerical simulations. In the figures, each robot is depicted by an arrow within a circle (dotted circle for virtual reference robots). The orientation of the robot is shown by the orientation of the arrow.

The units used in the simulations are: for position, meter (m), for orientation, radian (rad), for linear velocity, meter/second (m/s), and for angular velocity, radian/second (rad/s).

5.1. Trajectory Tracking

In this section, the simulation results of our FSC-MPC controller, Kanayama's controller proposed in [22] and Samson's controller proposed in [15] are compared. Specifically, the controllers proposed in [22] and [15] are

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e + K_x x_e \\ \omega_r + v_r(K_y y_e + K_\theta \sin\theta_e) \end{bmatrix}, \tag{23}$$

and

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e + K_1 x_e \\ \omega_r + K_2 v_r \frac{\sin\theta_e}{\theta_e} + K_3 \theta_e \end{bmatrix}, \tag{24}$$

respectively.

The reference trajectory starts from posture $\mathbf{z}_r(0) = [0\ \ 0\ \ 0]^T$ with constant control inputs $[v_r\ \ \omega_r]^T = [1\ \ 0]^T$. In addition, we assume that two perturbations occur at times 10 s and 20 s which changes the orientation of the virtual reference robot from 0 rad to $\pi/2$ rad and from $\pi/2$ rad back to 0 rad, respectively. Total simulation time is 30 s.

The initial condition of tracking robot is $\mathbf{z}_c(0) = [0\ \ 3\ \ 0]^T$. Controller parameters are selected as follows. For Kanayama's controller, $K_x = 1$, $K_y = 4$ and $K_\theta = 4$. For Samson's controller, $K_1 = 1$, $K_2 = 4$ and $K_3 = 4$. Sampling time for these two controllers is $\delta_T = 0.1$ s. For FSC-MPC controller, $H = 6$, $\rho = 0.95$,

$$Q = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ R = I_{2\times 2},\ \hat{P} = I_{3\times 3}, \tag{25}$$

where $I_{r\times r}$ denotes the $r \times r$ identity matrix. Sampling time is $\delta_T = 0.5$ s. Control input constraints are

$$-4(\mathrm{m/s}) \leq v \leq 4(\mathrm{m/s}),$$
$$-0.8(\mathrm{rad/s}) \leq \omega \leq 0.8(\mathrm{rad/s}).$$

The system responses of the three controllers are shown in Figure 3. The FSC-MPC controller successfully stabilizes the tracking robot to the reference trajectory.

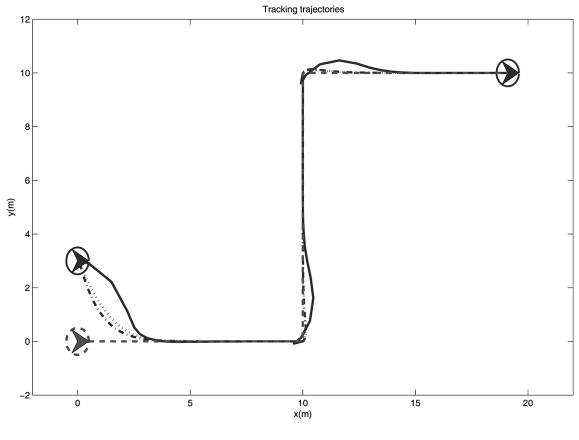

Figure 3. Tracking trajectories of three controllers. Dashed: Reference. Solid: FSC-MPC. Dotted: Samson. Dash-dot: Kanayama.

Initial Posture	Aicardi's Controller	FSC-MPC Controller
$[1 \;\; 0 \;\; \pi/2]^T$	1772.7122	10.4999
$[-0.5 \;\; 0.867 \;\; \pi/2]^T$	586.2641	6.1507
$[-0.5 \;\; -0.867 \;\; \pi/2]^T$	69.7075	3.7015

Table 1. The integral of norm squared actual control inputs for stabilization.

5.2. Point Stabilization

In this section, the simulation results of our FSC-MPC controller and Aicardi's controller proposed in [1] are compared. Specifically, the controller proposed in [1] is

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} K_1 e \cos\alpha \\ k_2\alpha + K_1 \frac{\cos\alpha\sin\alpha}{\alpha}(\alpha + K_3\phi) \end{bmatrix}. \tag{26}$$

Three initial robot postures are used in the simulation. They are

$$\mathbf{z}_c(0) = \left\{ \begin{bmatrix} 1 \\ 0 \\ \pi/2 \end{bmatrix}, \begin{bmatrix} -0.5 \\ 0.867 \\ \pi/2 \end{bmatrix}, \begin{bmatrix} -0.5 \\ -0.867 \\ \pi/2 \end{bmatrix} \right\}.$$

The final posture is $\mathbf{z}_d = [0 \quad 0 \quad 0]^T$. Controller parameters are selected as follows. For Aicardi's controller, K_1=3, K_2=6 and K_3=1. For FSC-MPC controller, controller parameters, sampling time and control input constraints are the same as those in Subsection 5.1.

The trajectories generated by Aicardi's controller and our FSC-MPC controller from different initial postures are shown in Figure 4. The FSC-MPC controller successfully stabilizes the robot at the desired final posture.

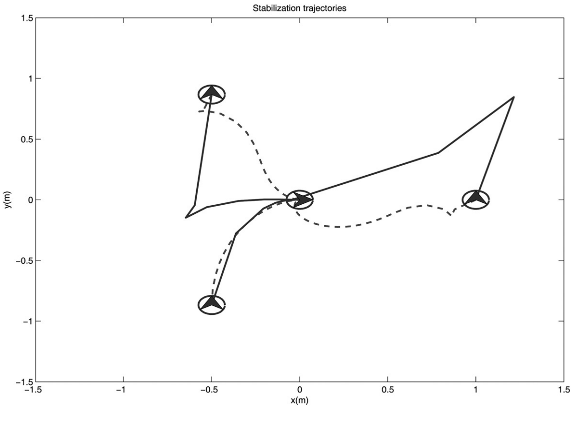

Figure 4. Stabilizing trajectories of two controllers. Solid: FSC-MPC. Dashed: Aicardi.

Specifically, we use the integral of norm squared actual control inputs ($\sum_1^k \|\mathbf{u}\|^2 \delta_T$) as a metric to evaluate the control energy. The control energy expended by each controller from different initial postures are shown in Table 1. The FSC-MPC requires much less control energy in comparison with the Aicardi's controller.

5.3. Simultaneous Tracking and Stabilization

In this section, numerical simulations illustrate that our FSC-MPC controller has the ability of simultaneous tracking and stabilization. Usually, simultaneous tracking and stabilization is not considered under a single controller approach. Most of the existing controllers for trajectory tracking of nonholonomic mobile robots will fail when the virtual reference robot stops or moves backward. However, in realistic scenarios, we do need to handle those situations.

In simulation, the virtual reference robot starts moving backward from posture $\mathbf{z}_r(0)$=[0 0 $\pi/2]^T$ with constant control inputs $[v_r \;\; \omega_r]^T = [-1 \;\; 0.1]^T$. Then, it stops at time $t = 5$ s. The initial condition of the tracking robot is $\mathbf{z}_c(0) = [10\;10\;\pi/2]^T$. We compare our FSC-MPC controller with Samson's controller (24). Controller parameters, sampling time, simulation time and control input constraints are the same as those in Subsection 5.1.

The result is shown in Figure 5. The FSC-MPC controller successfully stabilizes the tracking robot to the final posture where the reference robot stops. Meanwhile, Samson's controller experiences some extreme maneuvers and only stops the tracking robot to a neighboring position.

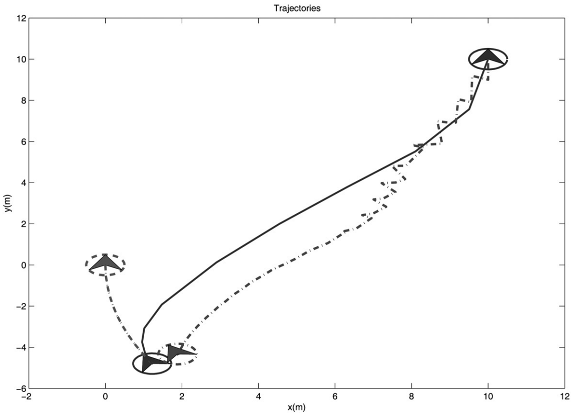

Figure 5. Trajectories of simultaneous tracking and stabilization. Dashed: Reference. Solid: FSC-MPC. Dash-Dot: Samson.

6. FSC-MPC Formation Control

In this section, we consider using the FSC-MPC algorithm proposed in Section 3 to solve the problem of controlling a team of mobile robots with nonholonomic constraints to leader-following formations. Since the FSC-MPC algorithm has simultaneous tracking and point stabilization capability, we expect that it can keep the formation in real-world scenarios. The effectiveness of the method is investigated by numerical simulations.

6.1. Formation Error

Let a triplet $p_i = [x_i \;\; y_i \;\; \theta_i]^T$ describe the position and the orientation of the i^{th} mobile robot. A leader-following formation configuration F_{ij}^d between robot i and j can be defined by the desired relative distances Δx_{ij}^d and Δy_{ij}^d in a local coordinate frame attached to the leader robot i, where $F_{ij}^d = \left[\Delta x_{ij}^d \quad \Delta y_{ij}^d\right]^T$.

However, to directly use the results in Section 3, we need to convert the formation control problem to a trajectory tracking problem. Let us define a reference robot, whose position and orientation are as follows

$$\begin{aligned} x_r &:= x_i + l\cos(\eta), \\ y_r &:= y_i + l\sin(\eta), \\ \theta_r &:= \theta_i, \end{aligned} \tag{27}$$

where

$$\begin{aligned} l &= \sqrt{\Delta x_{ij}^{d\,2} + \Delta y_{ij}^{d\,2}}, \\ \eta &= \operatorname{arctan2}(\Delta y_{ij}^d, \Delta x_{ij}^d) + \theta_i. \end{aligned} \tag{28}$$

Now we can define the formation error for the j^{th} robot, which is a tracking error defined by

$$x_j^e := \begin{bmatrix} x_r - x_j \\ y_r - y_j \\ \theta_r - \theta_j \end{bmatrix}. \tag{29}$$

Figure 6 shows this conversion.

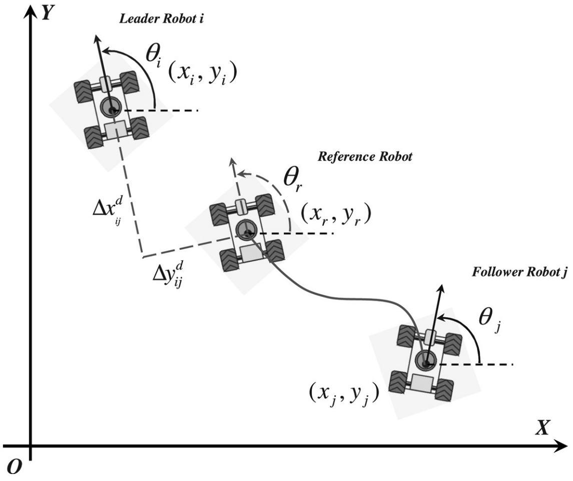

Figure 6. Convert formation control to trajectory tracking.

Let us define the optimal control problem for robot j, we have

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_j^H(\mathbf{x}_j^e, k, \mathbf{u}_j), \\ \text{subject to:} \quad & \mathbf{x}_j^e(k+1) = \mathbf{f}(\mathbf{x}_j^e(k), \mathbf{u}_j(k)), \\ & \mathbf{x}_j^e(k) \in \mathcal{X}, \\ & \mathbf{u}_j(k) \in \mathcal{U}, \\ & \|\mathbf{x}_j^e(k+1)\|_{\hat{P}} \le \rho \|\mathbf{x}_j^e(k)\|_{\hat{P}}, \end{aligned} \tag{30}$$

where $J_j^H(\mathbf{x}_j^e, k, \mathbf{u}_j)$ is defined as

$$J_j^H(\mathbf{x}_j^e, k, \mathbf{u}_j) := \sum_{i=1}^{H} L(\mathbf{x}_j^e(k+i;k), \mathbf{u}_j(k+i-1;k)), \tag{31}$$

where

$$L(\mathbf{x}_j^e, \mathbf{u}_j) := \|\mathbf{x}_j^e\|_Q^2 + \|\mathbf{u}_j\|_R^2. \tag{32}$$

The FSC-MPC formation controller can be implemented as follows.

Data: prediction horizon $H \in \mathbf{N}$; sampling time $\delta_T \in \mathbf{R}^+$; weights Q, R, $\hat{P} > 0$; constraints $\mathbf{u}_{\min}, \mathbf{u}_{\max} \in \mathbf{R}^m$; contractive parameter $\rho \in (0,\ 1)$; $k \in \mathbf{Z}^*$; desired formation F_{ij}^d.

Step 0: set $k = 0$; set initial control prediction $\hat{\mathbf{u}}_i(s;k) = \hat{\mathbf{u}}_j(s;k) = 0$, $s \in [1, \ldots, H-1]$.

Step 1: receive leader states $\mathbf{p}_i(k)$, control prediction $\hat{\mathbf{u}}_i(s;k)$ and measure the states $\mathbf{p}_j(k)$ at time t_k; with control prediction $\hat{\mathbf{u}}_i(s;k)$ and $\hat{\mathbf{u}}_j(s;k)$, desired formation F_{ij}^d and relations (27), (28) and (29), solve the optimal control problem (30) and obtain a control sequence $\mathbf{u}_j^*(s;k)$.

Step 2: apply the first control $\mathbf{u}_j^*(1;k)$ in the control sequence $\mathbf{u}_j^*(s;k)$ to robot j for the time interval $[t_k,\ t_{k+1}]$, where $t_{k+1} = t_k + \delta_T$.

Step 3: update the control prediction as follows

$$\hat{\mathbf{u}}_j(s;k) = \begin{cases} \mathbf{u}_j^*(s+1;k) & s \in [1, \ldots, H-2] \\ \mathbf{u}_j^*(s;k) & s = H-1 \end{cases};$$

set $k = k+1$; go back to **Step 1**.

6.2. Simulation Results

The effectiveness of the FSC-MPC formation control algorithm is investigated by numerical simulations.

6.2.1. Reconfiguration

In this simulation, we consider a formation reconfiguration scenario. We assume that the leader robot has the ability to sense the environment and communicate with each robot in the team. During the navigation, the leader detects a wall and sends commands to follower robots to change the formation to a line such that they can pass a small gate. After all the robots pass the gate, the leader sends commands to resume the previous formation.

The leader robot starts moving from posture $\mathbf{p}_1(0) = [0\ 0\ 0]^T$ with constant control inputs $[v_1\ \ \omega_1]^T = [1\ \ 0]^T$. Robot 2 and 3 start from $\mathbf{p}_2(0) = [-8\ \ -1\ \ \pi/4]^T$ and $\mathbf{p}_3(0) = [-15\ \ -3\ \ -\pi/4]^T$. The desired formation is $F_{12}^d = [-5\ \ 2]^T$ and $F_{23}^d = [-5\ \ -4]^T$. At time $t = 15$ s, the leader detects the wall and sends commands to team robots to change to a new formation $F_{12}^d = [-5\ \ 0]^T$ and $F_{23}^d = [-5\ \ 0]^T$, which is a line formations. At time $t = 30$ s, as all the robots have passed the gate, the leader sends commands again and changes the formation back to the original one $F_{12}^d = [-5\ \ 2]^T$ and $F_{23}^d = [-5\ \ -4]^T$.

The control parameters for the FSC-MPC formation controller are $H = 3$, $\rho = 0.999$. Q, R and $\hat{P}$ are the same as in (25). Control input constraints are

$$\begin{gathered} -4(\mathrm{m/s}) \le v \le 4(\mathrm{m/s}), \\ -0.5(\mathrm{rad/s}) \le \omega \le 0.5(\mathrm{rad/s}). \end{gathered}$$

Sampling time is $\delta_T = 0.5$ s and the total simulation time is 50 s.

Figure 7 depicts the trajectories of the robot team. We can see that the triangle formation is kept before and after the change to a line formation and the line formation is achieved when the team passes the gate.

Figures 8 and 9 show the control inputs and the relative position of Robot 2. In Figure 9, we can see that the actual relative position of Robot 2 converges to the desired formation $F_{12}^d = [-5\ \ 2]^T$ before $t = 15$ s and after $t = 30$ s and the desired formation $F_{12}^d = [-5\ \ 0]^T$ during time $[15\ \ 30]$. In Figure 8 we can see that, when the formation is achieved, the control inputs converge to $[1\ \ 0]^T$, which are the velocities of the leader robot.

Figures 10 and 11 show the control inputs and the relative position of Robot 3. In Figure 11, we can see that the actual relative position of Robot 3 converges to the desired formation $F_{23}^d = [-5\ \ -4]^T$ before $t = 15$ s and after $t = 30$ s and the desired formation $F_{23}^d = [-5\ \ 0]^T$ during time $[15\ \ 30]$. Figure 10 shows that, when the formation is achieved, the control inputs converge to $[1\ \ 0]^T$, which are the velocities of the leader robot. However, Robot 3 requires more control energy than Robot 2 does by comparing Figures 8 and 10.

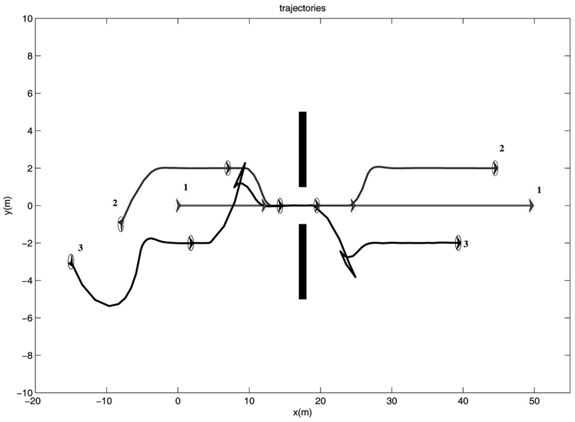

Figure 7. Reconfiguration during navigation.

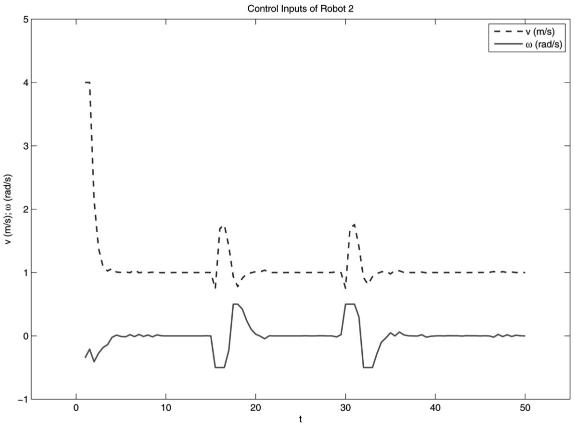

Figure 8. Control inputs of Robot 2.

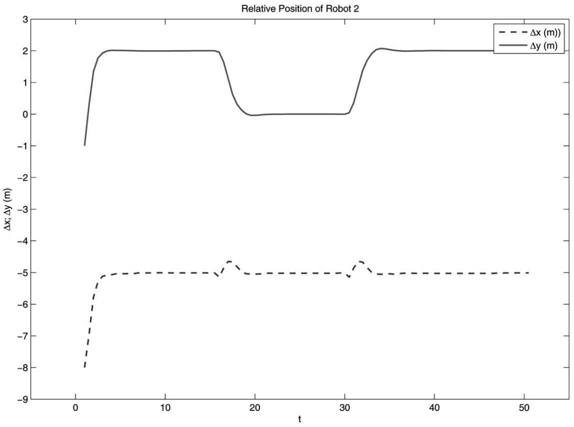

Figure 9. Relative position of Robot 2.

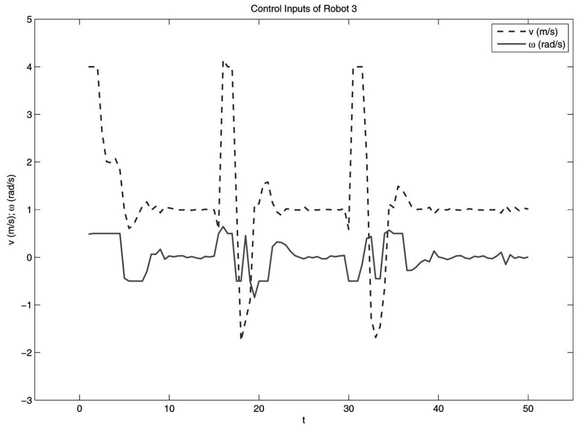

Figure 10. Control inputs of Robot 3.

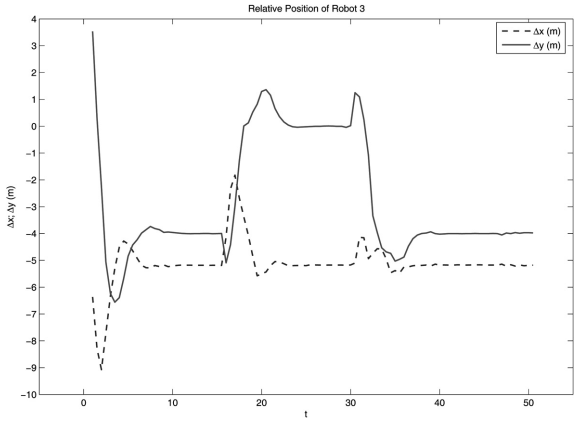

Figure 11. Relative position of Robot 3.

6.2.2. Obstacle Avoidance

In this simulation, we consider an obstacle avoidance scenario. We assume that the leader robot has the ability to sense the environment. During navigation, the leader robot detects an obstacle. It stops and then moves backward, stops and moves forward for several times to make sure the whole team will not collide with the obstacle. Finally, the team moves forward again in formation.

The leader robot starts moving from posture $\mathbf{p}_1(0) = [0 \;\; 0 \;\; 0]^T$. The leader robot's velocity inputs are

$$\begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} = \begin{cases} [1 \;\; 0]^T & t = [0 \;\; 15], \\ [0 \;\; 0]^T & t = [15.5 \;\; 20], \\ [-1 \;\; -0.15]^T & t = [20.5 \;\; 25], \\ [0 \;\; 0]^T & t = [25.5 \;\; 30], \\ [1 \;\; 0.15]^T & t = [30.5 \;\; 35], \\ [0 \;\; 0]^T & t = [35.5 \;\; 39.5], \\ [1 \;\; 0]^T & t = 40, \\ [-1 \;\; -0.15]^T & t = [40.5 \;\; 45], \\ [0 \;\; 0]^T & t = [45.5 \;\; 50], \\ [1 \;\; 0]^T & t = [50 \;\; 60], \end{cases}$$

Robots 2 and 3 start from $\mathbf{p}_2(0) = [-8 \;\; -1 \;\; \pi/4]^T$ and $\mathbf{p}_3(0) = [-15 \;\; -3 \;\; -\pi/4]^T$. The desired formation is $F_{12}^d = [-5 \;\; 2]^T$ and $F_{23}^d = [-5 \;\; -4]^T$.

Control parameters for the FSC-MPC formation controller are, $H = 4$, $\rho = 0.999$. Matrices Q, R, $\hat{P}$ are chosen as (25). Control input constraints are

$$-4(\mathrm{m/s}) \le v \le 4(\mathrm{m/s}),$$

$$-0.5(\mathrm{rad/s}) \le \omega \le 0.5(\mathrm{rad/s}).$$

Sampling time is $\delta_T = 0.5$ s and the total simulation time is 60 s.

Figure 12 shows the trajectories of the robot team. We can see that the triangle formation is kept. Figure 13 depicts the control inputs of Robot 2. In addition, Figures 14 and 15 present the control inputs and the relative position of Robot 3. In Figure 15, we can see that the actual relative position of Robot 3 converges to the desired formation $F_{23}^d = [-5 \;\; -4]^T$. However, by comparing Figures 13 and 14, we can see that Robot 3 requires more control authority than Robot 2.

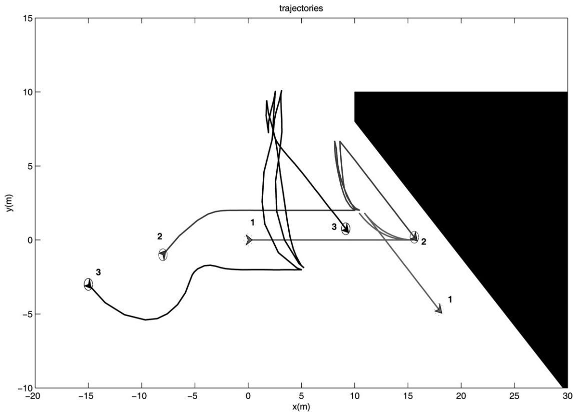

Figure 12. Obstacle avoidance during navigation - three robots.

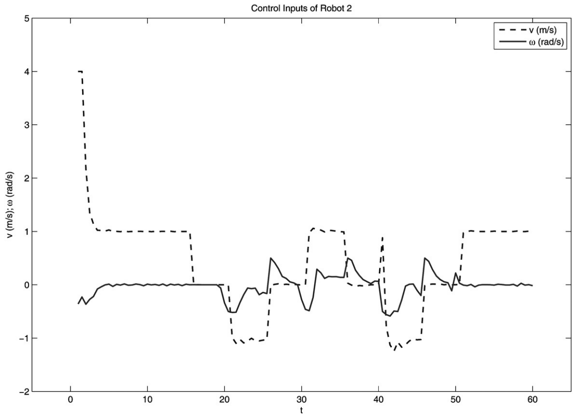

Figure 13. Control inputs of Robot 2.

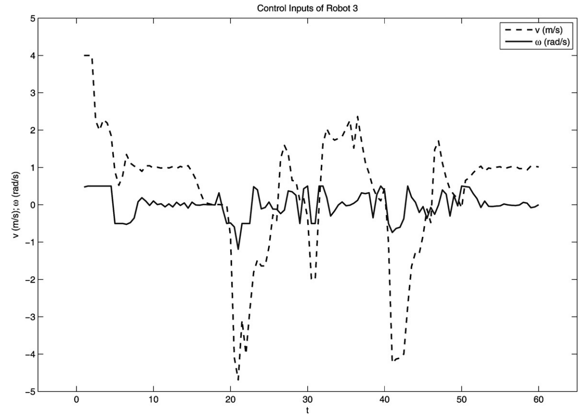

Figure 14. Control inputs of Robot 3.

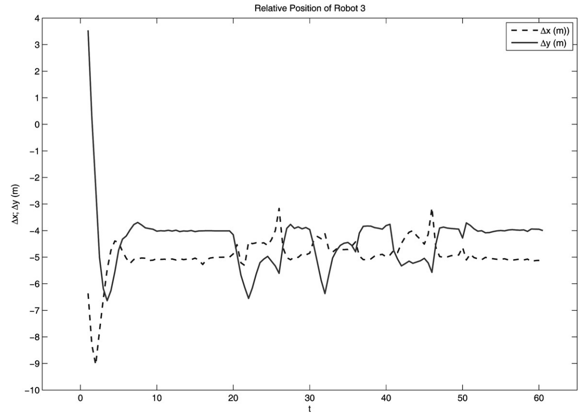

Figure 15. Relative position of Robot 3.

7. Conclusions

In this paper, a *first-state contractive* model predictive control (FSC-MPC) algorithm is developed for the trajectory tracking, point stabilization and formation problems of nonholonomic mobile robots. Simulation results show that the proposed FSC-MPC scheme can generate satisfactory system responses while it requires much less control energy in comparison with other well-known controllers. Moreover, we address the problem of tracking stationary and backward trajectories and show that our controller is capable of dealing with this problem.

For all simulations, an initial feasible solution is required for the proposed FSC-MPC controller. Like most of the MPC schemes, a trial-and-error approach is used. The choice of the contractive parameter is critical for the initial feasible solution. A value close to 1 is preferred. However, a small value will give faster convergence rate when the system approaches the equilibrium point. As part of our future work, we are investigating adaptive or time-varying schemes of the contractive parameter, and experimental verifications of the FSC-MPC on our experimental testbed [12].

Acknowledgment

This work is supported in part by NSF grants CAREER #0811347 and CNS #0709329, and the U.S. Army Research Office grant DAAD19-03-1-0142 (through the University of Oklahoma). We would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] M. AICARDI, G. CASALINO, A. BICCHI, AND A. BALESTRINO, Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robot. Automat. Mag.*, pages 27–35, March 1995.

[2] A. ASTOLFI. Exponential stabilization of a wheeeled mobile robot via discontinuous control. *J. Dynam. Syst. Meas.*, 121:121–126, 1999.

[3] G. BECKER, A. PACKARD, D. PHILBRICK, AND G. BALAS. Control of parametrically-dependent linear systems: A single quadratic lyapunov approach. In *Proceedings of the American Control Conference*, volume 3, pages 2795–2799, San Francisco, CA, June 1993.

[4] F. BLANCHINI. Set invariance in control. *Automatica*, 35:1747–1767, 1999. Survey paper.

[5] A. M. BLOCH. *Nonholonomic Mechanics and Control*. Springer, New York, NY, 2003.

[6] S. BOYD, L. E. GHAOUI, E. FERON, AND V. BALAKRISHNAN. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, 1994.

[7] R. W. BROCKETT. Asymptotic stability and feedback stabilization. In R. W. Brockett, R. S. Millman, and H. J. Sussmann, editors, *Diferential Geometric Control Theory*, pages 181–191, Birkhäuser, Boston, MA, 1983.

[8] J. CARSON, B. ACIKMESE, R. MURRAY, AND D. MACMYNOWSKI. A robust model predictive control algorithm with a reactive safety mode. In *International Federation of Automatic Control (IFAC) World Congress*, Seoul, Korea, July 6-11 2008. To appear.

[9] H. CHEN AND F. ALLGOWER. A quasi-infinite horizon nonlinear model predictive scheme with guaranteed stability. *Automatica*, 14(10):1205–1217, 1998.

[10] D. CHWA. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Trans. Contr. Syst. Technol.*, 12(4):637–644, 2004.

[11] R. COLBAUGH, E. BARANY, AND K. GLASS. Adaptive control of nonholonomic robotic systems. *J. Robotic Systems*, 15(7):365–393, 1998.

[12] D. CRUZ, J. MCCLINTOCK, B. PERTEET, O. ORQUEDA, Y. CAO, AND R. FIERRO. Decentralized cooperative control: A multivehicle platform for research in networked embedded systems. *IEEE Control Syst. Mag.*, 27(3):58–78, June 2007.

[13] A. K. DAS, R. FIERRO, V. KUMAR, J. P. OSTROWSKI, J. SPLETZER, AND C. J. TAYLOR. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825, October 2002.

[14] S. L. DE OLIVEIRA KOTHARE AND M. MORARI. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 45(6):1053–1071, 2000.

[15] C. CANUDAS DE WIT, H. KHENNOUF, C. SAMSON, AND O. J. SORDALEN. Nonlinear control design for mobile robots. In Y. F. Zheng, editor, *Recent Trends in Mobile Robot*, chapter 5, pages 121–156. World Scientific, 1993.

[16] W. E. DIXON, D. M. DAWSON, E. ZERGEROGLU, AND A. BEHAL. *Nonlinear control of wheeled mobile robots*. Springer, London, UK, 2001.

[17] R. FIERRO AND F. L. LEWIS. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3805–3810, New Orleans, LA, December 1995.

[18] R. FIERRO AND F. L. LEWIS. Robust practical point stabilization of a nonholonomic mobile robot using neural networks. *Journal of Intelligent and Robotic Systems*, (20):295–317, 1997.

[19] D. GU AND H. HU. A stabilizing receding horizon regulator for nonholonomic mobile robots. *IEEE Trans. Robot. Automat.*, 21(5):1022–1028, October 2005.

[20] D. GU AND H. HU. Receding horizon tracking control of wheeled mobile robots. *IEEE Trans. Contr. Syst. Technol.*, 14(4):743–749, July 2006.

[21] G. INDIVERI. Kinematic time-invariant control of a 2D nonholonomic vehicle. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2112–2117, Phoenix, AZ, December 1999.

[22] Y. KANAYAMA, Y. KIMURA, F. MIYAZAKI, AND T. NOGUCHI. A stable tracking control method for an autonomous mobile robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 384–389, 1990.

[23] H. K. KHALIL. *Nonlinear Systems*. Prentice Hall, third edition, 2002.

[24] H. KHENNOUF AND C. C. DE WIT. On the construction of stabilizing discontinuous controllers for nonholonomic systems. In *IFAC Nonlinear Control Syst. Design Symp.*, pages 747–752, Tahoe City, CA, 1995.

[25] B. KIM AND P. TSIOTRAS. Controllers for unicycle-type wheeled robots: Theoretical results and experimental validation. *IEEE Trans. Robot. Automat.*, 18(3):294–307, 2002.

[26] D. Q. MAYNE AND H. MICHALSKA. Receding horizon control of nonlinear systems. *IEEE Trans. Automat. Contr.*, 35(7):814–824, 1990.

[27] D. Q. MAYNE, J. B. RAWINGS, C. V. RAO, AND P. O. M. SCOKAERT. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[28] R. T. M'CLOSKEY AND R. M. MURRAY. Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Trans. Automat. Contr.*, 42(5):614–628, May 1997.

[29] H. MICHALSKA AND D. Q. MAYNE. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 38(11):1623–1633, November 1993.

[30] P. MORIN AND C. SAMSON. Practical stabilization of a class of nonlinear rsystems: Application to chain systems and mobile robots. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2144–2150, Sydney, Australia, 2000.

[31] R. M. MURRAY. Recent research in cooperative control of multivehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, September 2007.

[32] R. M. MURRAY, Z. LI, AND S. S. SASTRY. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1993.

[33] G. ORIOLO, A. DE LUCA, AND M. VENDITTELLI. WMR control via dynamic feedback linearization: Design, implementaion, and experimental validation. *IEEE Trans. Contr. Syst. Technol.*, 10(6):835–852, 2002.

[34] J. B. POMET. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems and Control Letters*, 18:147–158, 1992.

[35] C. SAMSON. Control of chained systems. Application to path following and time-varying point-stabilizaiton of mobile robots. *IEEE Trans. Automat. Contr.*, 40:64–77, Jan. 1995.

[36] C. SAMSON AND K. AIT-ABDERRAHIM. Feedback control of a nonholonomic wheeled cart in Cartesian space. In *IEEE Int. Conf. on Robotics and Automation*, pages 1136–1141, Sacramento, CA, 1991.

[37] M. J. VACHON, R. J. RAY, K. R. WALSH, AND K. ENNIX. F/a-18 performance benefits measured during the autonomous formation flight project. Technical Report NASA/TM-2003-210734, NASA Dryden Flight Research Center, Edwards, California., 2003.

[38] G. C. WALSH, D. TILBURY, S. SASTRY, R. MURRAY, AND J. P. LAUMOND. Stabilization of trajectory for systems with nonholonomic constraints. *IEEE Trans. Automat. Contr.*, 39:216–222, Jan. 1994.

[39] J. WAN, C. G. QUINTERO, N. LUO, AND J. VEHI. Predictive motion control of a MiRoSoT mobile robot. In *Proceedings of World Automation Congress*, volume 15, pages 325–330, 2004.

[40] D. WANG AND G. XU. Full-state tracking and internal dynamics of nonholonomic wheeled mobile robots. *IEEE/ASME Trans. Mechatronics*, 8(2):203–214, 2003.

Received: December, 2007
Revised: May, 2008
Accepted: June, 2008

Contact addresses:

Feng Xie
School of Electrical and
Computer Engineering
Oklahoma State University
Stillwater, USA
e-mail: fredfxie@yahoo.com

Rafael Fierro
The MARHES Lab, Department of Electrical
& Computer Engineering
MSC01 1100
1 University of New Mexico
Albuquerque, NM 87131-0001, USA
e-mail: rfierro@ece.unm.edu

FENG XIE received a B.Sc. degree in testing technology and instrumentation and a M.Sc. degree in control science and engineering from Zhejiang University, China in 1997 and 2000, respectively. He received a second M.S. degree in chemical engineering and a Ph.D. degree in electrical engineering from Oklahoma State University in 2004 and 2007, respectively. His research interests include cooperative control of mobile robots, model predictive control and nonlinear control theory.

RAFAEL FIERRO received a M.Sc. degree in control engineering from the University of Bradford, England and a Ph.D. degree in electrical engineering from the University of Texas at Arlington in 1990 and 1997, respectively. He held a postdoctoral appointment with the GRASP Lab, University of Pennsylvania, and a faculty position with the Department of Electrical and Computer Engineering, Oklahoma State University. Dr. Fierro is currently an associate professor in the Electrical and Computer Engineering Department, University of New Mexico. His research interests include hierarchical hybrid and embedded systems, optimization-based cooperative control, and robotics. He is the recipient of a Fulbright Scholarship and of the 2004 National Science Foundation CAREER Award.