

A Model-Predictive Approach to Formation Control of Omnidirectional Mobile Robots

Kiattisin Kanjanawanishkul and Andreas Zell

Abstract—This paper presents a solution to the problem of steering a group of real omnidirectional mobile robots along a given path, while maintaining a desired formation pattern. This problem can be divided into a leader agent subproblem and a follower agent subproblem such that a leader agent follows a given path and each follower agent tracks a trajectory, estimated by using the leader’s information. In this paper, we exploit nonlinear model predictive control (NMPC) as a local control law for real-world experiments due to its advantages of taking the robot constraints and future information into account. To solve the path following problem for the leader agent, we propose to integrate the rate of progression of a virtual vehicle to be followed along that path into the local cost function of NMPC. After the open-loop optimization problem is solved, the optimal rate of progression at each time step in the future is obtained. This information and the leader’s current state are broadcasted to all follower agents. With respect to a desired formation configuration and a reference path, each follower agent can estimate its own reference trajectory by using the leader’s information and its time stamp. NMPC is also employed as a local control law to steer the follower agent to track that reference trajectory. Our approach was validated by experiments using three omnidirectional mobile robots.

I. INTRODUCTION

The fields of coordination and formation control for multiple mobile robots have been the object of recent research efforts in the last few years [1], [2]. The problem of formation control is defined as the coordination of a group of robots to follow a given path (in case of a given trajectory, to track a time parameterized reference trajectory) and to maintain a desired spatial formation, e.g., a column, a line, a triangle, etc. Solutions of this problem can be applied in a wide range of applications, such as search and rescue missions, remote terrain and space exploration, security patrols, and land mine removal.

This paper addresses the problem of path following and the problem of formation keeping. Various strategies and approaches have been investigated for solving the formation control problem. These approaches can be broadly classified into three groups, i.e., the leader-following approach [3], [4], the virtual structure approach [5], [6] and the behavior-based approach [7]. In our proposed method, we use the leader-following strategy. The main advantage of the leader-following approach is its simplicity in that formation maneuvers can be completely specified in terms of the leader’s path or trajectory, and the leader-following problem can be reduced to a tracking problem. Consequently, the

global problem of keeping the desired formation pattern can be decomposed into two subproblems, i.e., a leader agent subproblem and an individual follower agent subproblem.

To control motions of all robot agents, we employ NMPC as a local control law since it can explicitly account for robot constraints and easily handle nonlinear and time-varying systems. To solve the problem of path following for the leader agent, we propose to integrate explicitly the rate of progression \dot{s} of a virtual vehicle into the local cost function of NMPC. Thus, the optimal predicted trajectory can be produced after the open-loop optimization is solved. This information is sent out to all follower agents via broadcast communication. With respect to this optimal predicted trajectory, each follower agent can generate its own *estimated* reference trajectory. Then its local NMPC law uses this reference trajectory to find the optimal control input.

The rest of the paper is structured as follows: Section II describes how to combine the formation configuration with the path following problem. In Section III, the omnidirectional mobile robots, which have been used in our experiments, are modeled. Then NMPC is presented in Section IV. Section V shows the experimental results. Finally, our conclusions and future work are given in Section VI.

II. FORMATION CONFIGURATION

In this paper, we solve two problems, i.e., the path following problem and the formation keeping problem. In the path following problem, the path is normally parameterized by the path length s , instead of time, which is normally used in a trajectory tracking problem. To make a formation pattern, we employ an idea of formation configuration in a curvilinear coordinate system, proposed in [8]. When the formation is turning, the formation’s shape can be slightly modified. Follower agents on the outside speed up and follower agents on the inside slow down, which allows the formation to be shape compliant on route (see Fig. 1). In our method, only the path which the leader follows is generated, while each individual follower agent F_i in the group has a pre-specified offset $(p_i(s), q_i(s))$ in curvilinear coordinates relative to the reference point C , which the leader agent follows, as shown in Fig. 1.

In some situations, the collision-free path does not always guarantee the safety for the whole formation. For example, the width of the path could be too narrow to allow for more than one robot. Thus, the formation must be changed to a column (see Fig. 2(b)). However, as stated in [8], the width of the formation (q -offset) can only be changed if the second derivative d^2q/ds^2 exists, i.e., offset q_i must be adequately

K. Kanjanawanishkul and A. Zell are with Wilhelm-Schickard-Institute, Department of Computer Architecture, University of Tübingen, Sand 1, 72076 Tübingen, Germany {kiattisin.kanjanawanishkul, andreas.zell}@uni-tuebingen.de

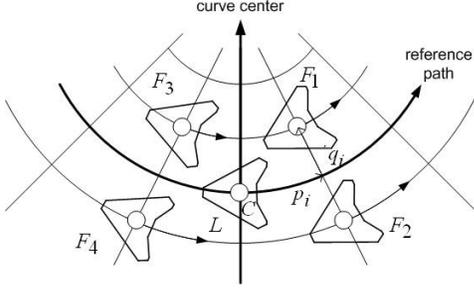


Fig. 1. Graphical depiction of a mobile robot path and accompanying offset quantities [8] when the formation is turning. L denotes a leader agent and $F_1 - F_4$ denote follower agents.

smooth with respect to the corresponding progenitor path during the transient from one configuration to another. To solve this problem, we propose to use a fifth-order (quintic) polynomial to join two path segments with different offset, so that the position, first and second derivatives at the starting and end points, where two path segments are joined, match. A quintic is the minimum order polynomial, which is able to give sufficient degrees of freedom and comply with the constraints on the slope. The general form of a quintic function is given by

$$q(s) = \sum_{j=0}^5 (a_j s_d^j), \quad (1)$$

subject to constraints on the conditions of the starting and end points, and its slope:

$$\begin{aligned} q(s_{start}) &= q_{start}, q(s_{end}) = q_{end}, \\ q'(s) &= \frac{dq(s)}{ds}, q'(s_{start}) = q'(s_{end}) = 0, \\ q''(s) &= \frac{d^2q(s)}{ds^2}, q''(s_{start}) = q''(s_{end}) = 0, \end{aligned} \quad (2)$$

where a_j are the coefficients of the function, (s, q) is the position on the offset curve at the path length s , $s_d = \frac{s - s_{start}}{s_{end} - s_{start}}$, (s_{start}, q_{start}) is the starting point of the quintic curve, and (s_{end}, q_{end}) is the end point of the quintic curve.

Applying (2) to (1) yields $a_0 = q_{start}$, $a_1 = a_2 = 0$, $a_3 = 10$, $a_4 = -15$, and $a_5 = 6$. The quintic function becomes:

$$q(s) = q_{start} + (q_{end} - q_{start})(6s_d^5 - 15s_d^4 + 10s_d^3). \quad (3)$$

Using the scenario of formation configuration in a curvilinear coordinate system allows us not only to have the offset-

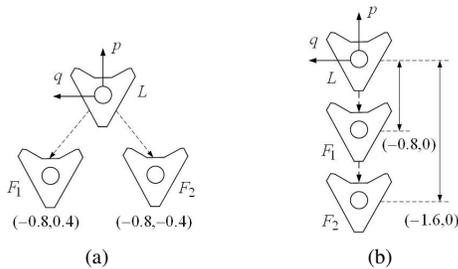


Fig. 2. Graphical description of formation configurations: (a) a triangle (b) a column. L , F_1 , and F_2 denote the leader agent, the follower agent 1 and the follower agent 2, respectively. Units are given in meter.

varying distance $q_i(s)$ but also to adjust the $p_i(s)$ coordinate. This is simply obtained by decreasing or increasing the velocity of a follower agent i in an appropriate manner. However, collision avoidance has to be taken into account. The timing of each follower agent has to be evaluated in order to ensure that no collision occurs during transition.

Let u_c be the translational velocity of point C , which the leader agent follows. In other words, u_c is the rate of progression of a virtual vehicle. Once the $(p_i(s), q_i(s))$ coordinates of a follower agent i have been determined, the path length of a follower agent s_i can be obtained by $s_i = s_c + p_i$, where s_c is the path length at point C . Then its velocity profile can be obtained by¹

$$u_i = H u_p, \quad (4)$$

$$\omega_i = k_i u_i, \quad (5)$$

where

$$\begin{aligned} k_i &= \text{sign}(b) \frac{\sqrt{a^2 + b^2}}{H^2}, \\ H &= \sqrt{(1 - k_p q)^2 + \left(\frac{dq}{ds}\right)^2}, \\ a &= -2k_p \frac{dq}{ds} - q \frac{dk_p}{ds} - (1 - k_p q) \frac{G}{H^2}, \\ b &= k_p - k_p^2 q + \frac{d^2q}{ds^2} - \frac{dq}{ds} \frac{G}{H^2}, \\ G &= (1 - k_p q) \left(-k_p \frac{dq}{ds} - q \frac{dk_p}{ds}\right) + \frac{dq}{ds} \frac{d^2q}{ds^2}, \end{aligned}$$

u_i , ω_i and k_i are the translational velocity, the rotational velocity and the curvature of the follower agent i , respectively, u_p is the translational velocity at s_i , which is usually equal to u_c . k_p is the curvature at s_i on the reference path, and $q_i(s)$ is the offset at s_i .

III. KINEMATIC MODEL OF OMNIDIRECTIONAL MOBILE ROBOTS

Omnidirectional mobile robots are becoming increasingly popular, since they have some distinguishing advantages over their nonholonomic counterparts. They have simultaneously and independently controlled rotational and translational motion capabilities. The annual RoboCup competition is an example of a highly dynamic environment where omnidirectional mobile robots have been exploited highly successfully (see the RoboCup Official Website: <http://www.robocup.org>).

There are two coordinate frames used in the modeling: the body frame (X_m, Y_m) and the world frame (X_w, Y_w) . The body frame is fixed at the moving robot with the origin at the center of the robot, whereas the world frame is fixed at the ground, as shown in Fig. 3. θ denotes the robot orientation, which is the angle of the axis X_m in the world coordinate system. φ denotes the robot moving direction in the world coordinate system. Each wheel has the same distance L_w to the robot's center of mass R . δ refers to the wheel orientation in the body frame. The kinematic model of an omnidirectional mobile robot is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}, \quad (6)$$

¹The derivations can be found at: <http://www.ra.cs.uni-tuebingen.de/mitarb/kanjana/IROS2008deriv.pdf>.

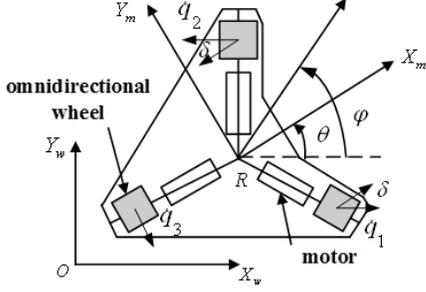


Fig. 3. Coordinate frames of an omnidirectional mobile robot.

where $[x, y, \theta]^T$ is the state vector in the world frame and $[u, v, \omega]^T$ is the vector of robot velocities observed in the body frame. u and v are the robot translational velocities and ω is the robot rotational velocity. When the wheel velocities are considered, the lower level kinematic model with respect to the robot coordinate can be described by

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \cos \delta & \sin \delta & L_w \\ -\cos \delta & \sin \delta & L_w \\ 0 & -1 & L_w \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}, \quad (7)$$

where $\dot{\mathbf{q}}(t) = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$ is the vector of wheel velocities, which is equal to the wheel's radius multiplied by the wheel's angular velocity.

Motion control of an omnidirectional robot has been well studied. Trajectory linearization control (TLC), which is based on linearization along the desired trajectory and inversion of the dynamics was implemented in [10]. Watanabe [9] introduced the PID control, self-tuning PID control, and fuzzy control of omni-directional mobile robots. In our proposed approach, NMPC combined with path following has been carried out for the leader agent, and is explained in details in the next section. NMPC is an attractive approach, because it can explicitly account for system constraints and easily handle nonlinear and time-varying systems.

IV. NONLINEAR MODEL PREDICTIVE CONTROL

Nonlinear model predictive control (NMPC) has become an increasingly popular control technique used in industry. The controller is based on a finite-horizon continuous time minimization of nonlinear predicted tracking errors with constraints on the control inputs and the state variables. It predicts system outputs based on current states and the system model, finds an open loop control profile by numerical optimization, and applies the first control signal in the optimized control profile to the system [11]. However, due to the use of a finite predictive control horizon, control stability becomes one of the main problems. To guarantee control stability, many approaches have been investigated, e.g., using so called terminal region constraints and/or a terminal penalty term. More explanations regarding NMPC can be found in [11], [12].

In the centralized system of formation control, the complete system is modeled and all the control inputs are computed in one optimization problem. Using this strategy, the

size of the state variables depends typically on the number of mobile robots. When the control horizon becomes larger, the number of variables, of which the agent has to find the value, increases rapidly. Also, the demands of computer power and memory are daunting for the real-time solution of systems with a large control horizon and a large number of mobile robots. Although short prediction horizons are desirable from a computational point of view, long prediction horizons are required for closed-loop stability and good performance. Thus, the research has led us to extending the single-agent MPC framework to the use of multiple agents by means of decomposing the centralized system into smaller subsystems that are independently controlled. This can be achieved by using distributed/decentralized control or hierarchical design (see [13], [14], [15], [16]). The main difference of these control approaches is the kind of interaction between two subsystems via state variables, constraints or objectives. In this paper, the task of formation keeping is divided into two subtasks: (i) the leader agent follows a given reference path, and (ii) each follower agent tracks its own *estimated* trajectory. Each agent computes a solution to its local problem. The leader agent communicates the most recent variables to its followers, each of which re-solves its optimization problem with the updated values.

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

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \text{subject to: } \mathbf{x}(t) &\in \mathcal{X}, \mathbf{u}(t) \in \mathcal{U}, \forall t \geq 0 \end{aligned} \quad (8)$$

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. In NMPC, the input applied to the system is usually given by the solution of the following finite horizon open-loop optimal control problem, which is solved at every sampling instant:

$$\min_{\mathbf{u}(\cdot)} \int_t^{t+T_p} F(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau + V(\mathbf{x}(t+T_p)), \quad (9)$$

$$\begin{aligned} \text{subject to: } \dot{\mathbf{x}}(\tau) &= \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) \\ \mathbf{u}(\tau) &\in \mathcal{U} \quad \forall \tau \in [t, t+T_c] \\ \mathbf{x}(\tau) &\in \mathcal{X} \quad \forall \tau \in [t, t+T_p] \\ \mathbf{x}(t+T_p) &\in \Omega \end{aligned} \quad (10)$$

where $F(\mathbf{x}, \mathbf{u}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}$, T_c and T_p are the control horizon and the prediction horizon with $T_c \leq T_p$. $V(\mathbf{x}(t+T_p))$ is the terminal penalty and Ω is the terminal region. The deviation from the desired values is weighted by the positive definite matrices Q and R . In our experiments, the control horizon T_c is equal to the prediction horizon T_p . A Lyapunov function for the terminal-state penalty can be defined as follows:

$$V(\mathbf{x}(t+T_p)) = \frac{1}{2} \mathbf{x}(t+T_p)^T \mathbf{x}(t+T_p) \quad (11)$$

under the terminal-state controller $\mathbf{u}^L(t)$ such that the following condition is satisfied:

$$\dot{V}(\mathbf{x}(t)) + F(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0, \quad (12)$$

for any state $\mathbf{x}(t)$ belonging to the terminal region Ω .

By our implementation, at initial time the internal clock of each agent has to be synchronized using clock synchronization. Furthermore, the sampling time is calculated based on the average past results [17]. Thus sampling time is varying. This leads us to work with asynchronous agents with different sampling time. This strategy allows agents to proceed at its own speed. In the next two subsections, we give more details about our controllers used in the leader agent and in the follower agent.

A. Leader Agent and Path Following Control

The main tasks for the leader agent are to steer itself to a given reference path, to produce an optimal predicted reference trajectory at each time instant, and to send out its information to all follower agents via broadcast communication. The first two tasks are simply handled by NMPC.

The path following problem is illustrated in Fig. 4. Since the translation and rotation of omnidirectional robots can be separately controlled [19], the intuitive explanation is that the controller finds (i) the distance from the robot to the path with the help of the rate of progression of a virtual vehicle, and (ii) the angle between the robot orientation θ and the desired orientation θ_b , and then reduces both to zero. The error kinematic model with respect to the path coordinate, adapted from [18] to omnidirectional mobile robots, can be given as

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & k(s)\dot{s} & 0 \\ -k(s)\dot{s} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} + \mathbf{u}_e, \quad (13)$$

$$\text{where } \mathbf{u}_e = \begin{bmatrix} -\dot{s} + u_o \cos \phi \\ u_o \sin \phi \\ \omega - \omega_b \end{bmatrix},$$

$\omega_b = \dot{\theta}_b$, \mathbf{x}_e is the vector of the pose error with respect to the path coordinate, $k(s)$ is the path curvature at s , and u_o is the desired translational velocity along the reference path. $\phi = \varphi - \theta_r$ and $\theta_e = \theta - \theta_b$, where φ is the angle of the moving direction of the robot with respect to the world frame and θ_r is the orientation angle of the tangent to the reference curve at s . In this paper, the desired orientation θ_b is simply given as θ_r .

To drive the error \mathbf{x}_e to zero, \mathbf{x} and \mathbf{u} in (9)-(12) are replaced by \mathbf{x}_e and \mathbf{u}_e , respectively and the terminal state feedback controller $\mathbf{u}^L = [u_1^L, u_2^L, u_3^L]^T$ is selected as follows:

$$\begin{aligned} u_1^L &= -\alpha x_{eT}, \\ u_2^L &= -\beta y_{eT}, \\ u_3^L &= -\gamma \theta_{eT}, \end{aligned} \quad (14)$$

where $\mathbf{x}_e(t + T_p) = [x_{eT}, y_{eT}, \theta_{eT}]^T$, $\alpha \geq 0$, $\beta \geq 0$, and $\gamma \geq 0$. All weight parameters have to be selected such that (12) is satisfied.

After the optimization problem at time t_k is solved, the current reference state $(s_{l,k})$, the optimal rate of progression at each time step in the future $(\dot{s}_{l,k|k}, \dot{s}_{l,k+1|k}, \dots, \dot{s}_{l,k+T_p-1|k})$, and the sampling time $\delta_{l,k}$ are transmitted to all follower agents. Each data packet is

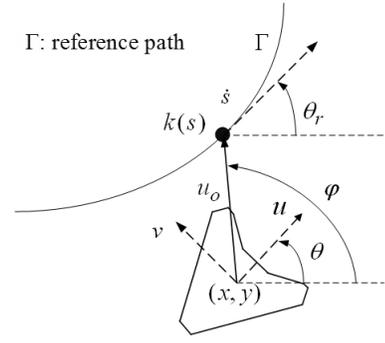


Fig. 4. Illustration of the path following problem.

time-stamped, so that the *age* of the information can be extracted at a follower controller.

B. Follower Agent and Trajectory Tracking Control

The task for each follower agent is to track its own *estimated* reference trajectory, based on the leader agent's information, the predefined formation configuration, and the given reference path. In practice, some problems may arise, e.g., the information time delay is not zero, the sampling time of the follower agent can be different (asynchronous timing conditions) from that of the leader agent or the data packet can be lost. To overcome these problems, first we calculate the *age* of the received information and then estimate the robot's own reference trajectory with the velocity profiles, computed by using (4) and (5). In case of packet loss, the missing information can be filled in by using the previous information received from the leader agent.

The error kinematic model of each follower agent i with respect to the robot coordinate, adapted from [20] to omnidirectional mobile robots, can be given as

$$\dot{\mathbf{x}}_{i,e} = \begin{bmatrix} \dot{x}_{i,e} \\ \dot{y}_{i,e} \\ \dot{\theta}_{i,e} \end{bmatrix} = \begin{bmatrix} 0 & \omega_i & 0 \\ -\omega_i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{i,e} \\ y_{i,e} \\ \theta_{i,e} \end{bmatrix} + \mathbf{u}_{i,e}, \quad (15)$$

$$\text{where } \mathbf{u}_{i,e} = \begin{bmatrix} u_{i,r} \cos \theta_{i,e} - u_i \\ u_{i,r} \sin \theta_{i,e} - v_i \\ \omega_{i,r} - \omega_i \end{bmatrix},$$

$\mathbf{x}_{i,e}$ is the vector of the pose error with respect to the robot frame, $[u_i, v_i, \omega_i]^T$ is the vector of robot velocities. $u_{i,r}$ and $\omega_{i,r}$ are the desired translational and rotational velocities, respectively. Since $\mathbf{x}_{i,e}$ is required to converge to zero, \mathbf{x} and \mathbf{u} in (9)-(12) are replaced by $\mathbf{x}_{i,e}$ and $\mathbf{u}_{i,e}$, respectively. We select the terminal penalty and the feedback controller in the same way as we do for the leader agent.

V. EXPERIMENTAL RESULTS

We implemented our algorithm on three omnidirectional mobile robots shown in Fig. 5. Each is equipped with a Pentium-M 2 GHz on-board PC with 1 GB RAM and its wheels are driven by three 60W Maxon DC motors. It has an omnidirectional camera as sole sensor, which is used for self localization. Thanks to [21], the self localization applied for the RoboCup field has been employed in our experiments.



Fig. 5. Omnidirectional mobile robots used in the formation control experiments.

This self-localization algorithm is based on probabilistic Monte-Carlo localization (MCL). In our experiments, one of the omnidirectional robots is defined as a leader and the others as followers. The reference path is given as

$$x_o(t) = \frac{2.3 \cos t}{1 + (\sin t)^2}, \quad y_o(t) = \frac{2.3 \sin t \cos t}{1 + (\sin t)^2}. \quad (16)$$

By our implementation, this given path was numerically parameterized by the path length s . All parameters used in our experiments are listed in Table I, ($\mathbf{I}_3 = \text{diag}(1, 1, 1)$). A sampling time of approximately $0.07s$ can be achieved. The free package DONLP2 [22] has been used to solve the online optimization problem and PID controllers have been implemented for motor velocity control.

Fig. 6 shows the superimposed snapshots of three mobile robots keeping and switching the formation, while the leader follows the reference path with the translational reference velocity u_o of 0.4 m/s and the rotational reference velocity ω_o of $k_o u_o$, where k_o is the curvature at the reference point. A formation was changed from a triangle (see Fig. 2(a)) to a column (see Fig. 2(b)) and then switched back to the triangle. The pose errors of the leader, of the follower 1, and of the follower 2 are shown in Fig. 7(a), Fig. 7(b), and Fig. 7(c), respectively. The velocities of the leader, of the follower 1, and of the follower 2, compared with their reference velocities, are shown in Fig. 8(a), Fig. 8(b), and Fig. 8(c), respectively. As can be seen, the leader robot can follow the reference path with the desired translational velocity and the follower robots can track the reference trajectory based on the leader robot's information. Also they can maintain the desired formation at any time.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present formation control of real omnidirectional mobile robots using a local NMPC law. NMPC is an attractive control method as not only it can handle the state and input constraints but also utilize future information to generate a trajectory of optimal control input at each time

TABLE I
PARAMETERS USED IN OUR EXPERIMENTS

Parameters	Leader	Follower1	Follower2
Q	$\mathbf{1I}_3$	$\mathbf{1I}_3$	$0.5\mathbf{I}_3$
R	$0.1\mathbf{I}_3$	$0.2\mathbf{I}_3$	$0.1\mathbf{I}_3$
T_p	3	3	3
α, β, γ	2	2	2

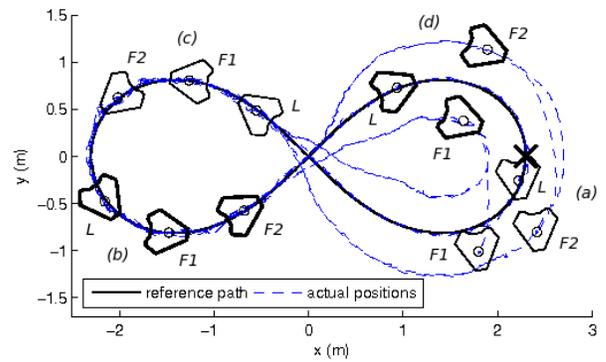


Fig. 6. The snapshots are taken at the following time: (a) original configuration (thin line) at $t = 0s$, (b) column configuration (thick line) at $t = 15.4s$, (c) column formation (thin line) obtained at $t = 23.1s$, and (d) triangle formation (thick line) obtained at $t = 39.5s$. L denotes leader, $F1$ denotes the follower 1 and $F2$ denotes the follower 2. \times denotes the starting position.

step. However, the well-known disadvantages of NMPC are a control stability problem and high computational time. By our implementation, we employ a Lyapunov function for the terminal-state penalty and terminal constraints to guarantee the stability of a local NMPC law and we have shown that NMPC is a promising control approach, applied for real-time applications of mobile robots as seen in our experimental results.

The two key points, which are employed to solve the path following problem and formation keeping problem in this paper, are that (i) the rate of progression of a virtual vehicle is integrated into the local cost function of the leader agent, and (ii) each follower agent computes its own reference trajectory with velocity profiles, estimated by using the leader's information, a given path, and a desired formation pattern. Velocity profiles for each follower agent can be computed by using (4) and (5) when $(p_i(s), q_i(s))$ coordinates have been determined.

Since each follower agent has to estimate its own reference trajectory by using the leader's information, the mismatch between what the leader agent is actually doing and what the follower agents believe that the leader agent is doing plays an important role in the stability problem. Reducing the uncertainty and estimation mismatch made about the leader's information is currently under our investigation. Furthermore, we will integrate obstacle avoidance as coupling constraints and analyze a condition for the proposed approach to ensure the feasibility of the optimization problem.

REFERENCES

- [1] Y. Q. Chen and Z. M. Wang, "Formation Control: a Review and a New Consideration", in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug. 2005, pp. 3181-3186.
- [2] R. M. Murray, Recent Research in Cooperative-control of Multivehicle Systems, *Journal of Dynamics, Systems, Measurement and Control*, vol. 129, no. 5, 2007, pp. 571-583.
- [3] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, A Vision-based Formation Control Framework, *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, Oct. 2002, pp. 813-825.

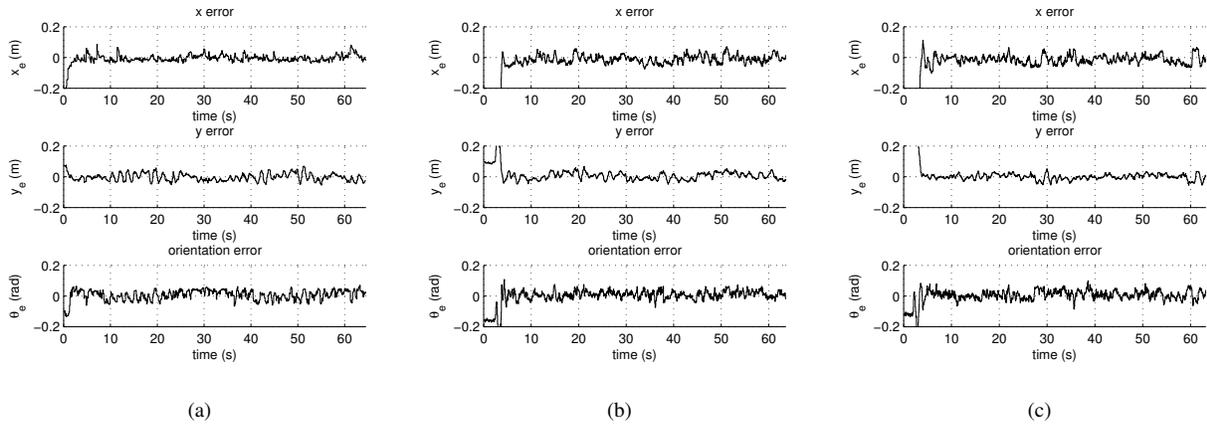


Fig. 7. Pose errors of (a) the leader, (b) the follower 1, and (c) the follower 2.

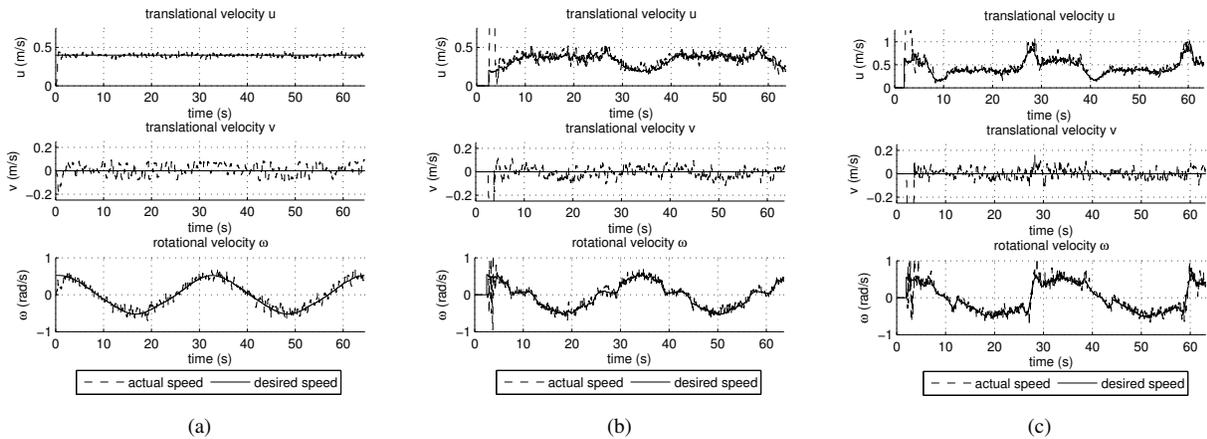


Fig. 8. Velocities of (a) the leader, (b) the follower 1, and (c) the follower 2.

- [4] M. Egerstedt and X. Hu, Formation Constrained Multi-agent Control, *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, Dec. 2001, pp. 947-951.
- [5] M. A. Lewis and K. -H. Tan, High Precision Formation Control of Mobile Robots using Virtual Structures, *Autonomous Robots*, vol. 4, no. 4, 1997, pp. 387-403.
- [6] 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, no. 1, Jan. 2004, pp. 73-82.
- [7] T. Balch and R. C. Arkin, Behavior-based Formation Control for Multirobot Teams, *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, 1998, pp. 1-15.
- [8] T. D. Barfoot and C. M. Clark, Motion Planning for Formations of Mobile Robots, *Robotics and Autonomous Systems*, vol. 46, no. 2, 2004, pp. 65-78.
- [9] K. Watanabe, "Control of Omnidirectional Mobile Robot", in *2nd Int. Conf. on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, Apr. 1998, pp. 51-60.
- [10] Y. Liu, X. Wu, J. Jim Zhu, and J. Lew, "Omni-directional Mobile Robot Controller Design by Trajectory Linearization", in *Proc. of American Control Conference*, Denver, Colorado, Jun. 2003, pp. 3423-3428.
- [11] F. Allgöwer, R. Findeisen, and Z. K. Nagy, Nonlinear Model Predictive Control: from Theory to Application, *Journal of Chinese Inst. of Chem. Engrs.*, vol. 35, no. 3, 2004, pp. 299-315.
- [12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, Constrained Model Predictive Control: Stability and Optimality, *Automatica*, vol. 36, no. 6, 2000, pp. 789-814.
- [13] W. B. Dunbar and R. M. Murray, Distributed Receding Horizon Control for Multi-vehicle Formation Stabilization, *Automatica*, vol. 42, no. 4, 2006, pp. 549-558.
- [14] A. G. Richards and J. P. How, "A Decentralized Algorithm for Robust Constrained Model Predictive Control", in *Proc. of American Control Conference*, Boston, Massachusetts, June 30 - July 2, 2004, pp. 4261-4266.
- [15] T. Keviczky, F. Borrelli, and G. J. Balas, "A Study on Decentralized Receding Horizon Control for Decoupled Systems", in *Proc. of American Control Conference*, Boston, Massachusetts, June 30 - July 2, 2004, pp. 4921-4926.
- [16] M. G. Earl and R. D'Andrea, A Decomposition Approach to Multi-vehicle Cooperative Control, *Robotics and Autonomous Systems*, vol. 55, no. 4, 2007, pp. 276-291.
- [17] R. Franz, M. Milam, and J. Hauser, "Applied Receding Horizon Control of the Caltech Ducted Fan", in *Proc. of American Control Conference*, Anchorage AK, May 2002, pp. 3735-3740.
- [18] D. Soeanto, L. Lapiere, and A. Pascoal, "Adaptive Non-singular Path-following, Control of Dynamic Wheeled Robots", in *Proc. of International Conference on Advanced Robotics*, Coimbra, Portugal, June 30 - July 3, 2003, pp. 1387-1392.
- [19] G. Campion, G. Bastin, and B. D'Andréa-Novel, Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots, *IEEE Trans. on Robotics and Automation*, vol. 12, no. 1, Feb. 1996, pp. 47-62.
- [20] D. Gu and H. Hu, Receding Horizon Tracking Control of Wheeled Mobile Robots, *IEEE Trans. on Control Systems Technology*, vol. 14, no. 4, 2006, pp. 743-749.
- [21] P. Heinemann, J. Haase, and A. Zell, "A Combined Monte-Carlo Localization and Tracking Algorithm for RoboCup", in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 1535-1540.
- [22] P. Spellucci, An SQP Method for General Nonlinear Programs using only Equality Constrained Subproblems, *Mathematical Programming*, vol. 82, no. 3, 1998, pp. 413-448.