2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Model Predictive Control of Multi-Robot Formation Based on the Simplified Dual Neural Network

Xinzhe Wang, Zheng Yan, and Jun Wang

***Abstract*—This paper is concerned with formation control problems of multi-robot systems in framework of model predictive control. The formation control of robots herein is based on the leader-follower scheme. The followers are controlled by torques to track the desired trajectories to form and keep a formation. A model predictive control approach is proposed for solving the formation control problem, where the control problem is formulated as a dynamic quadratic optimization problem. A one-layer recurrent neural network called the simplified dual network is applied for computing the optimal control input in real time. Simulation results substantiate that the formation of robots can be well controlled by the proposed approach.**

I. Introduction

Formation control is an important issue for a group of robots. In many complicated tasks such as searching and rescuing operations, moving in formation offers many advantages such as reduced cost, increased flexibility, and improved robustness. In recent years, formation control has become a research focus. There are several methods for formation control such as leader-follower strategy [1-2], behavior based strategy [3-4], virtual structure strategy [5-6], graph theory based strategy [7-10], synchronization control strategy [11] and so on. The most popular scheme is the leader-follower strategy due to its implementability, scalability and reliability. In this work, the leader-follower strategy is considered for the formation control of a multi-agents system.

Much of the early formation control research based on the leader-follower strategy only considered the kinematic model where the formation is controlled by linear and angle velocities [12-16]. Formation control based on kinematic models needs strong assumptions on perfect velocity tracking to achieve the desired formation with convergent errors. In practice, the dynamics should be considered to guarantee the robots can move with the desired velocities. Some research studied formation control based on dynamic models. For example, in [17], the dynamics of followers are modeled by a neural network. In [18], a decentralized approach is proposed based on virtual points, potential functions and the individual robots abilities. In [19], a centralized scheme is proposed where a PD controller is developed . In [20], a dynamic formation control approach is presented using on the leader-follower strategy. It is shown that the dynamics of leaders are important for the followers [21].

In recent research, neural network control and adaptive control are widely studied for the formation control of multi-robot systems. In [22], a neural network output feedback control scheme is developed where neural networks are applied for learning the dynamics of followers and the formation. In addition, a neural network observer is introduced to estimate the linear and angular velocities of the follower and its leader. In [23], a leader-follower based adaptive formation control method is proposed with limited information. An adaptive observer is developed to estimate the velocity information and uncertainties. Then an adaptive control law is derived using dynamic surface control design procedures.

Despite of the recent progress, some issues remain unaddressed. First, many algorithms rely on the linear velocities and angular velocities of the leader [21], [24]. However, the velocity information is not easily obtained. Formation control based on position information would be more preferred in practice [25]. Secondly, some formation control algorithms are too complex to be applied for real-world applications [22]. A simple and valid control algorithm is always desirable.

In this paper, formation control of multi-robot systems is studied based on the leader-follower scheme. The leader position is the only information utilized for formation control. Meanwhile, the kinematics and dynamics are both considered. The control inputs are the torques of followers. To move in and keep the desired formation, a model predictive control approach is proposed where the formation control problem is formulated as a sequential quadratic programming. A recurrent neural network called he simplified dual network is applied for solving the quadratic programming in real time.

The rest of this paper is organized as follows. In Section II, preliminaries on the robot dynamics and formation control are described. In Section III, the neural network based model predictive control scheme is proposed. In section IV, the simulation results are presented and discussed. Finally, Section V concludes this paper.

II. Preliminaries

A. Formation Control

Cooperation is an essential character of a multi-agent system. A certain formation should be maintained when a multi-agents system implements a cooperative work. In this

X. Z. Wang and J. Wang are with Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Liaoning, China (email: wxzagm@dlut.edu.cn).

Z. Yan and J. Wang are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (email: {zyan, jwang}@mae.cuhk.edu.hk).

This work was supported the Fundamental Research Funds for the Central Universities (DUT12RC(3)97), the Research Grants Council of the Hong Kong Special Administrative Region under Grants CUHK416812E, and the National Natural Science Foundation of China under Grant 61273307.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.

paper, the formation control is considered based on the leader-follower approach where one agent is selected as the leader and others are the followers. To maintain the formation, each follower needs to maintain the relative position with respect to the leader. There are two types of control schemes to maintain the relative position. The first one is based on the desired distance and desired angle between the leader and the follower. The second one is based on desired distance between each agent pairs. We choose the first control scheme herein, as shown in Fig. 1.

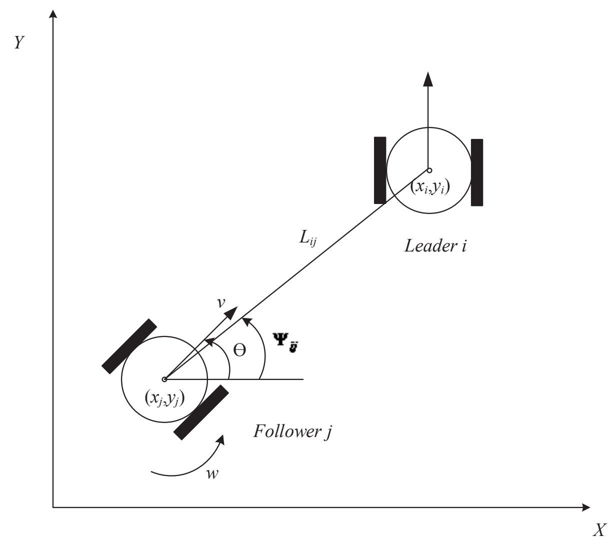

Fig. 1. Leader-follower formation

A follower is denoted with a subscript j, whereas the leader is denoted with the subscript i. In Fig. 1, L_{ij} and Ψ_{ij} are respectively the distance and angle between leader i and follower j. Given the desired distance L_{dij} and angle Ψ_{dij}, proper velocities can be found to keep the formation of a multi-agents system. The objective of the formation control can be described as follows:

$$\begin{aligned} &\lim_{t\to\infty}(L_{dij}-L_{ij})=0,\\ &\lim_{t\to\infty}(\Psi_{dij}-\Psi_{ij})=0. \end{aligned} \tag{1}$$

In other words, the distance and angle between follower and leader are convergent to the desired values.

It is assumed that the real-time coordinates of the leader can be obtained by each follower. Each follower tracks a trajectory based on the desired distance and angle.

B. Robot Dynamics

A robot can be modeled by its features of kinematics and kinetics. The kinematics describe the relation between its position and velocity. The kinetics describe the relationships between velocity and torque.

The kinematics equation for the jth robot can be written as:

$$\dot{q}_j = J_j(q_j)v_j \tag{2}$$

where $q_j = [x_j, y_j, \psi_j]^T$ is the position vector of the robot, x_j and y_j are the position coordinates, ψ_j is the angle of the robot; $v_j = [u_j, \theta_j]^T$ is the velocity vector, u_j and θ_j are the linear velocity and angle velocity respectively, and

$$J_j(q_j) = \begin{bmatrix} \cos\psi_j & 0 \\ \sin\psi_j & 0 \\ 0 & 1 \end{bmatrix}. \tag{3}$$

Then, the kinematics feature can be described as:

$$\begin{aligned} \dot{x}_j &= u_j\cos\psi_j,\\ \dot{y}_j &= u_j\sin\psi_j,\\ \dot{\psi}_j &= \theta_j. \end{aligned} \tag{4}$$

The kinetics equation for the jth robot can be written as [26]:

$$\dot{v}_j = f_j\tau_j \tag{5}$$

where

$$\tau_j = \begin{bmatrix} \tau_{jr} \\ \tau_{jl} \end{bmatrix}, \tag{6}$$

τ_{jr} and τ_{jl} are the torque of the right and left wheel respectively, and $f = (S^TMS)^{-1}S^TE$ where

$$S = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_Z \end{bmatrix}$$

$$E = \begin{bmatrix} \cos\psi_j/r & \cos\psi/r \\ \sin\psi/r & \sin\psi/r \\ b/r & -b/r \end{bmatrix}$$

M is the mass of the robot, r is the radius of the robot's wheels, b is half the distance between two wheels, and I_z is the moment of inertia of the robot about its center of mass.

Finally, the kinetics feature can be described as:

$$\begin{aligned} \dot{u}_i &= \frac{\tau_r}{mr} + \frac{\tau_l}{mr},\\ \dot{r}_i &= \frac{b\tau_r}{rI_Z} - \frac{b\tau_l}{rI_Z}. \end{aligned} \tag{7}$$

III. Control Strategies

A. Model prediction control

Consider a discrete-time nonlinear system as follows:

$$\begin{aligned} &x(k+1) = f(x(k)) + g(x(k))u(k)\\ &y(k) = Cx(k). \end{aligned} \tag{8}$$

The nonlinear system (8) is subject to the following constraints:

$$\begin{aligned} &u_{\min} \le u(k) \le u_{\max},\\ &\Delta u_{\min} \le \Delta u(k) \le \Delta u_{\max},\\ &y_{\min} \le y(k) \le y_{\max}, \end{aligned}$$

where $x(k) \in \Re^n$ is the state vector, $u(k) \in \Re^m$ is the input vector, $y(k) \in \Re^p$ is the output vector, $f(\cdot)$, $g(\cdot)$ are nonlinear functions, and $u_{\min} \le u_{\max}$, $\Delta u_{\min} \le \Delta u_{\max}$, $y_{\min} \le y_{\max}$ are vectors of lower and upper bounds.

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.

MPC is an iterative optimization technique: at each sampling time k, measure or estimate the current state, and then obtain the optimal input vector by solving an optimization problem. At each interactive control step, the following cost function can be used:

$$\begin{aligned} J(k) = &\sum_{j=1}^{N} \|r(k+j) - y(k+j\,|k)\|_Q^2 \\ &+ \sum_{j=1}^{N_u-1} \|\Delta u(k+j\,|k)\|_R^2, \end{aligned} \tag{9}$$

where $r(k+j)$ denotes the reference trajectory of output signal, $y(k+j\,|k)$ denotes the predicted output, and $\Delta u(k+j\,|k)$ denotes the input increment, and $\Delta u(k+j\,|k) \quad = u(k+j|k) - u(k+j-1|k)$. N and N_u are prediction horizon $(1 \le N)$ and control horizon $(0 < N_u < N)$, respectively. Q and R are appropriate weighting matrices. $\|\cdot\|$denotes the Euclidean norm of the corresponding vector. It can be obtained according to the model (8)

$$\begin{aligned} x(k+1|k) =& f(x(k|k-1)) + g(x(k|k-1))(u(k-1) \\ &+ \Delta u(k|k)), \\ x(k+2|k) =& f(x(k+1|k-1)) + g(x(k+1|k-1)) \\ &(u(k-1) + \Delta u(k|k) + \Delta u(k+1|k)), \\ &\vdots \\ x(k+N|k) =& f(x(k+N-1|k-1)) + g(x(k+N-1| \\ &k-1))(u(k-1) + \Delta u(k|k) + \ldots + \\ &\Delta u(k+N_u-1|k)). \end{aligned}$$

Define the following vectors:

$$\bar{r}(k) = [r(k+1) \cdots r(k+N)]^T,$$

$$\bar{y}(k) = [y(k+1\,|k) \cdots y(k+N\,|k)]^T,$$

$$\bar{u}(k) = [u(k\,|k) \cdots u(k+N_u-1\,|k)]^T,$$

$$\bar{x}(k) = [x(k+1\,|k) \cdots x(k+N\,|k)]^T,$$

$$\Delta\bar{u}(k) = [\Delta u(k\,|k) \cdots \Delta u(k+N_u-1\,|k)]^T, \tag{10}$$

where $\bar{r}(k)$ is known in advance. The predicted output $\bar{y}(k)$ is then expressed in the following form:

$$\bar{y}(k) = \tilde{C}\bar{x}(k) = \tilde{C}(G\Delta\bar{u}(k) + \tilde{f} + \tilde{g}) \tag{11}$$

where

$$\tilde{C} = \begin{bmatrix} C & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & C \end{bmatrix} \in \Re^{Np\times Nn},$$

$$G = \begin{bmatrix} g(x(k|k-1)) & \ldots & 0 \\ g(x(k+1|k-1)) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ g(x(k+N-1|k-1)) & \ldots & g(x(k+N-1|k-1)) \end{bmatrix} \in \Re^{Nn\times N_u m},$$

$$\tilde{f} = \begin{bmatrix} f(x(k|k-1)) \\ f(x(k+1|k-1)) \\ \vdots \\ f(x(k+N-1|k-1)) \end{bmatrix} \in \Re^{Nn},$$

$$\tilde{g} = \begin{bmatrix} g(x(k|k-1))u(k-1) \\ g(x(k+1|k-1))u(k-1) \\ \vdots \\ g(x(k+N-1|k-1))u(k-1) \end{bmatrix} \in \Re^{Nn}.$$

Hence, the original optimization problem (9) becomes:

$$\min \left\|\bar{r}(k) - \tilde{C}\tilde{f} - \tilde{C}\tilde{g} - \tilde{C}G\Delta\bar{u}(k)\right\|_Q^2 + \|\Delta\bar{u}(k)\|_R^2 \tag{12}$$

$$\begin{aligned} \text{s.t.} \qquad & \bar{u}_{\min} \le \bar{u}(k-1) + \tilde{I}\Delta u(k) \le \bar{u}_{\max} \\ & \Delta\bar{u}_{\min} \le \Delta\bar{u}(k) \le \Delta\bar{u}_{\max} \\ & \bar{y}_{\min} \le \tilde{C}\tilde{f} + \tilde{C}\tilde{g} + \tilde{C}G\Delta\bar{u}(k) \le \bar{y}_{\max} \end{aligned}$$

where

$$\tilde{I} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \in R^{Num\times Num}.$$

By defining $v = \Delta\bar{u}(k)$, problem (12) can be rewritten as a quadratic programming (QP) problem:

$$\begin{aligned} &\min \ \tfrac{1}{2}v^T W u + c^T v \\ &\text{s.t. } l \le Eu \le h \end{aligned} \tag{13}$$

where the coefficients are:

$$W = 2(G^T\tilde{C}^T Q\tilde{C}G + R) \in \Re^{N_u m\times N_u m},$$

$$c = -2G^T\tilde{C}^T Q(\bar{r}(k) - \tilde{C}\tilde{g} - \tilde{C}\tilde{f}) \in \Re^{N_u m},$$

$$E = \begin{bmatrix} -\tilde{I} & \tilde{I} & -\tilde{C}G & \tilde{C}G & I \end{bmatrix}^T \in \Re^{(3N_u m+2Np)\times N_u m},$$

$$b = \begin{bmatrix} -\bar{u}_{\min} + \bar{u}(k-1) \\ \bar{u}_{\max} - \bar{u}(k-1) \\ -\bar{y}_{\min} + \tilde{C}\tilde{f} + \tilde{C}\tilde{g} \\ \bar{y}_{\max} - \tilde{C}\tilde{f} - \tilde{C}\tilde{g} \end{bmatrix} \in \Re^{2N_u m+2Np},$$

$$l = \begin{bmatrix} -\infty \\ \Delta\bar{u}_{\min} \end{bmatrix} \in \Re^{3N_u m+2Np},$$

$$h = \begin{bmatrix} b \\ \Delta\bar{u}_{\max} \end{bmatrix} \in \Re^{3N_u m+2Np}.$$

The solution to the QP problem (13) gives the vector of control action $\Delta\bar{u}(k)$ whose first element can be used to calculate the optimal control input.

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.

B. Simplified dual neural network

In recent years, various neural network models have been developed as goal-seeking solvers for QP problems, e.g. [27-31]. The essence of neural computation lies in its parallel and distributed information processing. In particular, the simplified dual network developed in [27] showed superior performances in MPC applications [33]. This neural network model is applied for solving (13), whose dynamic equations can be described as:
State equation

$$\varepsilon\frac{dz}{dt} = -Ev + h(Ev - z).$$

Output equation

$$v = W^{-1}(E^T z - c). \quad (14)$$

where z is the state vector, v is the output vector and h is an activation function defined as

$$h(x_i) = \begin{cases} l_i, \ x_i < l_i; \\ x_i, \ l_i \le x_i \le h_i; \\ h_i, \ x_i > h_i. \end{cases} \quad (15)$$

The simplified dual network has a single-layer structure with totally $3N_u m + 2Np$ neurons. According to the convergence analysis in [27], it is Lyapunov stable and globally convergent to the optimal solution of any strictly convex QP problem.

C. Overall MPC scheme

The MPC scheme for formation control of multi-robot systems based on the simplified dual network is summarized as follows:

1) Let k=1. Set control time terminal T, prediction horizon N, Control horizon N_u, sample period t, weight matrices Q and R.
2) Calculate process model matrices G, $\tilde{f}$,$\tilde{g}$, $\tilde{C}$ and neural network matrices W, c, E.
3) Solve the convex quadratic minimization problem (13) by using the simplified dual neural network to obtain the optimal control action Δu_k.
4) Calculate the optimal input vector $\bar{u}(k)$ and implement the first element $u(k\,|k)$.
5) If $k < T$, set $k = k$+1, go to step 2;otherwise end.

IV. Simulation results

In this section, simulation results on an example are presented to validate the method. The discrete-time state space equation of the controlled system is

$$\begin{bmatrix} x_j(k+1) \\ y_j(k+1) \\ \psi_j(k+1) \\ u_j(k+1) \\ \theta_j(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cos\psi_j(k) & 0 \\ 0 & 0 & 0 & \sin\psi_j(k) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_j(k) \\ y_j(k) \\ \psi_j(k) \\ u_j(k) \\ \theta_j(k) \end{bmatrix} + h \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{mr} & \frac{1}{mr} \\ \frac{b}{rI_Z} & -\frac{b}{rI_Z} \end{bmatrix} \begin{bmatrix} \tau_{jr}(k) \\ \tau_{jl}(k) \end{bmatrix}.$$

In the simulation, we assume that the formation consists of one leader and two followers. The leader and followers have the same mass and size. The mass is $m = 2kg$, the radius of the robot's wheels $r = 0.05m$, half of the distance between two wheels $b = 0.05m$, the sampling time is $h = 0.05s$ and mass moment of inertia I_Z=1.

The desired trajectory of the leader is composed of a straight section and a section of quadratic curve. We also assume the leader velocity along the x-axis is 1 m/s. So in the second section, the linear velocity is accelerated.

The desired formation information and MPC parameters are shown in TABLE I. Simulation results are shown in Figs. 2-6. Fig. 2 demonstrates the formation trajectories. Fig. 3 and Fig. 4 show the variation of the torques of two wheels. The torque values lie in a small range. For the right wheel torque, τ_{rl} is in the range of [-0.1214, 0.1457], τ_{r1} is in the range of $[-0.1288, 0.2282]$, and τ_{r2} is in the range of $[-0.0891, 0.1780]$. For the left wheel torque, τ_{rl} is in the range of $[-0.1210, 0.1460]$, τ_{r1} is in the range of $[-0.0616,\ 0.1361]$, and τ_{r2} is in the range of $[-0.1266,\ 0.1710]$. Fig. 5 and Fig. 6 show the angular velocities and the linear velocities of the robots. Because of the accelerated moving in the tangential direction, the linear velocities are gradually increased. The angular velocities tend to a stable range. For the angular velocities, w_1 is in the range of [0.018, 0.0125], w_1 is in the range of $[0.0019, 0.0297]$ and w_2 is in the range of $[0.0019, 0293]$.

TABLE I
RELEVANT PARAMETERS

Parameter	VALUE	Representation
L_{d1}	2.1m	Distance between leader and follower 1
Ψ_{d1}	16.7^o	Angle between leader and follower 1
L_{d2}	2.1m	Distance between leader and follower 2
Ψ_{d2}	163.3^o	Angle between leader and follower 2
N	5	Prediction horizon
N_u	5	Control horizon
Q	Diag(100,500,0.1)	Weighting matrix
R	0.1I	Weighting matrix

V. Conclusion

This paper presents a neural network based model predictive control approach to formation control of multi-robot systems. Position of the leader is assumed to be available for each follower. The formation control problem is formulated as a sequential quadratic programming in MPC framework. The simplified dual network is applied for solving the quadratic programming in real-time. Simulation results show that the robots formation can be well controlled by using the proposed control method. Further works aim at solving formation control problems using local information only.

References

[1] Y. Q. Chen and Z. Wang, "Formation control: A review and a new consideration," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3181–3186.

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.

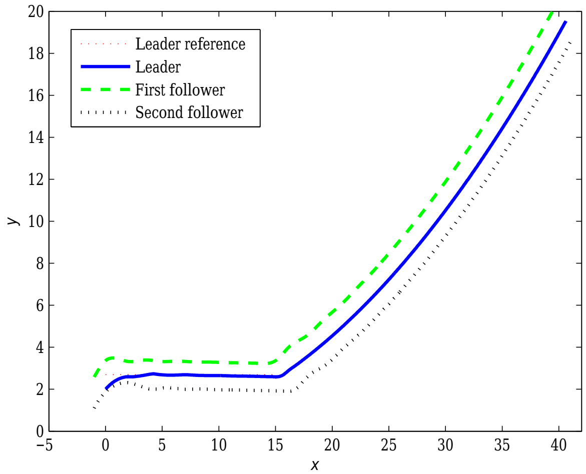

Fig. 2. Formation trajectories

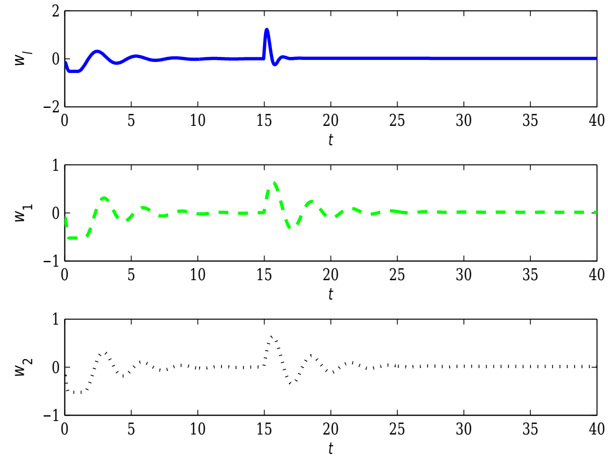

Fig. 5. Angular velocities of the agents

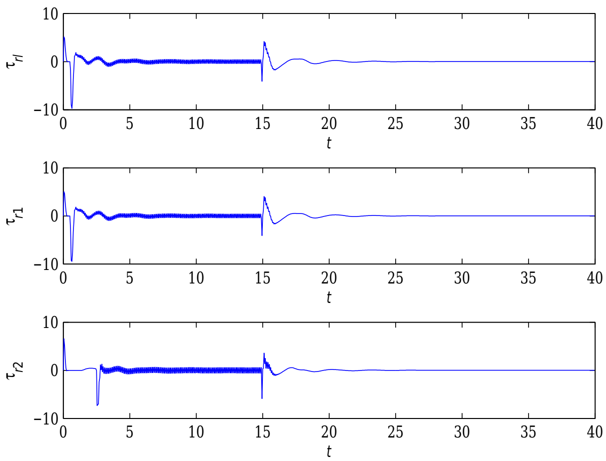

Fig. 3. Torques of the right wheel

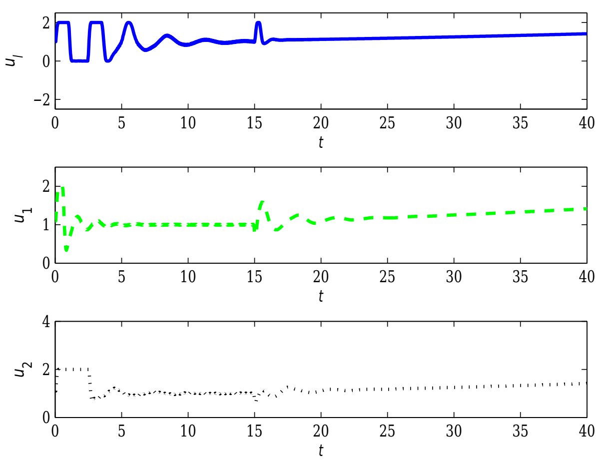

Fig. 6. Linear velocities of the agents

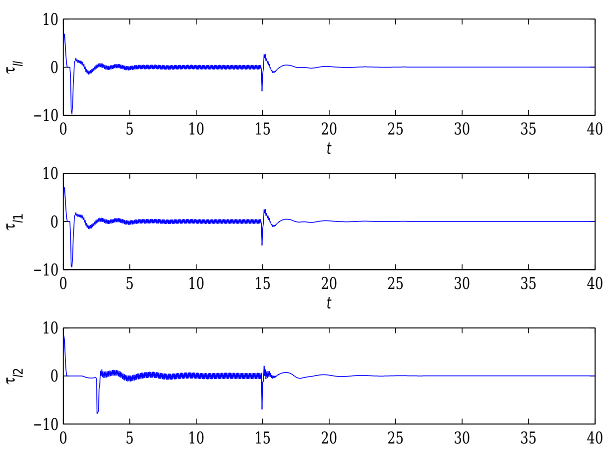

Fig. 4. Torques of the left wheel

[2] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader–follower formations," in *Proc. IEEE Eur. Control Conf. Decision Control*, Dec. 2005, pp. 635–640.

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multi robot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[4] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 837–846, Oct. 2002.

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

[6] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 527–538, May 2008.

[7] I. F. Ihle, M. Arcak, and T. I. Fossen, "Passivity-based designs for synchronized path following," *Automatica*, vol. 43, no. 9, pp. 1508–1518, Dec. 2007.

[8] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[9] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, "Decen-

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.

tralized control of vehicle formations," *Syst. Control Lett.*, vol. 54, no. 9, pp. 899–910, Sep. 2005.
[10] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory Appl.*, vol. 1, no. 2, pp. 505–512, Mar. 2007.
[11] D. Sun, C. Wang, W. Shang, and G. Feng, "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations," *IEEE Trans. Robotics*, vol. 25, no. 5, pp. 1074–1086, Oct. 2009.
[12] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader–follower formations," in *Proc. IEEE Eur. Control Conf. Decis. Control*, Dec. 2005, pp. 635–640.
[13] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.
[14] X. Li, J. Xiao and Z. Cai, "Backstepping-based multiple-mobilerobots formation control," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 887–892.
[15] H. Hsu and A. Liu, "Multiagent-based formation control using a simple representation," in Proc. *IEEE Int. Conf. Netw., Sens. Control*, Taipei, Taiwan, Mar. 2004, pp. 276–281.
[16] J. Shao, G. Xie, J. Yu, and L. Wang, "A tracking controller for motion coordination of multiple mobile robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 783–788.
[17] Y. Li and X. Chen, "Dynamic control of multi-robot formation," in *Proc. IEEE Int. Conf. Mechatronics*, Jul. 2005, pp. 352–357.
[18] M. Breivik, M. Subbotin, and T. Fossen, "Guided formation control for wheeled mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, Dec. 2006, pp. 1–7.
[19] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile robots," in *Proc IEEE Conf. Ind. Electron.*, Nov. 2006, pp. 3880–3885.
[20] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 527–538, May 2008.
[21] T. Dierks and S. Jaggannathan, "Neural network control of mobile robot formations using RISE feedback," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 332–347, Apr. 2009.
[22] T. Dierks, S. Jagannathan, "Neural network output feedback control of robot formations" *IEEE Trans. Syst., Man, Cybern. B, Cybern.*vol.40, no. 2, pp. 383-399, 2010.
[23] B. S. Park, J. B. Park, Y. H. Choi, "Adaptive formation control of electrically driven nonholonomic mobile robots with limited information" *IEEE Trans. Syst., Man, Cybern. B, Cybern.*vol.41, no.4, pp. 1061-1075, Aug. 2011.
[24] C. Luca, M. Fabio, P. Domenico, and T. Mario, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, May 2008.
[25] K. Choi, S. J.Yoo, J. B. Park, andY.H. Choi, "Adaptive formation control in absence of leader's velocity information," *IET Control Theory Appl.*, vol. 4, no. 4, pp. 521–528, Apr. 2010.
[26] J. C. Ryu, and S. K. Agrawal, "Differential flatness-based robust control of mobile robots in the presence of slip," *The International Journal of Robotics Research.*, vol. 30 no. 4, pp. 463-475 2011.
[27] Y. Xia, G. Feng, and J. Wang, "A recurrent neural network with exponential convergence for solving convex quadratic program and linear piecewise equations," *Neural Networks*, vol. 17, no. 7, pp. 1003-1015, 2004.
[28] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWTA application," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1500-1510, 2006.
[29] Q. Liu, J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Transactions on Neural Networks*, vol. 19, no. 4, pp. 558-570, 2008.
[30] X. Hu, J. Wang, "An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application," *IEEE Transactions on Neural networks*, vol. 19, no. 12, pp. 2022-2031, 2008.
[31] X. Hu and B. Zhang, "A new recurrent neural network for solving convex quadratic programming problems with an application to the kwinners-take-all problem," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 654-664, 2009.
[32] Y. Xia and J. Wang, "A general projection neural network for solving optimization and related problems," IEEE Transactions on Neural Networks, vol. 15, pp. 318-328, 2004.
[33] Z. Yan and J. Wang, "Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 746-756, 2012.

Authorized licensed use limited to: Universitas Brawijaya. Downloaded on May 05,2021 at 02:11:04 UTC from IEEE Xplore. Restrictions apply.