Distributed Model Predictive Control of Linear Systems with Coupled Constraints Based on Collective Neurodynamic Optimization[*]

Zheng Yan[1], Jie Lu[1], and Guangquan Zhang[1]

Centre for Artificial Intelligence, Faculty of Engineering and IT,
University of Technology Sydney, Broadway, NSW 2007, Australia.
{yan.zheng, jie.lu, guangquan.zhang}@uts.edu.au

Abstract. Distributed model predictive control explores an array of local predictive controllers that synthesize the control of subsystems independently yet they communicate to efficiently cooperate in achieving the closed-loop control performance. Distributed model predictive control problems naturally result in sequential distributed optimization problems that require real-time solution. This paper presents a collective neurodynamic approach to design and implement the distributed model predictive control of linear systems in the presence of globally coupled constraints. For each subsystem, a neurodynamic model minimizes its cost function using local information only. According to the communication topology of the network, neurodynamic models share information to their neighbours to reach consensus on the optimal control actions to be carried out. The collective neurodynamic models are proven to guarantee the global optimality of the model predictive control system.

Keywords: Collective neurodynamic optimization · Recurrent neural networks · Distributed optimization · Model predictive control.

1 Introduction

Model predictive control (MPC) is a popular optimization-based control technique. It iteratively predicts and optimizes control performances based on a system model. Dynamic feedback control actions are computed by solving online sequential optimization problems. As MPC can naturally deal with multivariable control problems and can explicitly take account of system constraints, it has been attracting much attention in many areas in recent years [13, 22].

Many real-world control plants such as waste-to-energy facilities [8] and smart grid [18] usually consist of linked units that can be grouped into subsystems. These subsystems are connected through a network and each subsystem can transmit certain local information to the others. As a result, the control technology can be implemented in a distributed fashion which takes the advantage

---

of the plant structure to improve reliability and reduce cost. In distributed control, each subsystem computes its control action by considering its local plant information as well as the effect of local control actions on all subsystems in the network [19]. Correspondingly, a distributed optimization problem arises in the setting of distributed MPC [4]. A challenging issue is distributed MPC lies in constraint satisfaction of coupled subsystems where local constraints are affected by the network topology.

Real-time optimization plays an increasingly important role in a range of control problems [7]. The success and performance of an MPC system are largely determined by the deployed optimization algorithm. Conventional optimization methods may not be sufficiently efficient for real time MPC implementation for problems with very large dimensions and fast sampling frequency. In the past two decades, neurodynamic optimization using recurrent neural networks (RNNs) emerged as a promising computational approach to real time optimization [21]. The essence of neural optimization lies in its inherent nature of parallel and distributed information processing and the availability of hardware implementation. Various RNN models have been presented for solving constrained optimization problems, such as the one-layer neural network with a hard-limiting activation function [15], the improved dual network [9], the finite-time convergent neural network [16], the neural network for nonsmooth optimization [17], the neural network for pesudoconvex optimization [6], the neural network for invex optimization [14], the collective neural networks for global optimization [24], and the neural network for distributed optimization [12]. These RNNs have shown good performance in terms of global convergence and low model complexity.

Neural networks have demonstrated advantages to the design and analysis of MPC methods. Due to their capabilities to approximate any continuous function mapping, many studies on incorporating neural networks with MPC synthesis have been carried out. Generally speaking, the use of neural networks fall into three categories: (1) using neural networks for system identification and modeling [23], (2) using neural networks for real time optimization [3], (3) using neural networks for off-line control law approximation [1]. In these works, distinct advantages of neural networks are exploited in MPC design.

In this paper, a distributed MPC scheme is proposed for linear systems with coupled constraints. The distributed MPC problem is formulated to distributed convex optimization with globally coupled constraints. The overall performance index to be minimized is the summation of local convex objectives. Cooperative neurodynamic models are applied to collectively solve the distributed optimization problems in real-time. One salient feature this work is that the optimization algorithm is designed by exploring the characteristics of the control problem, which greatly improves the scalability and reduces the computational cost. The rest of this paper is organized as follows. Section 2 discusses some preliminaries. Section 3 describes the distributed MPC formulation. Section 4 presents a collective neurodynamic optimization approach. Section 5 provides simulation results. Finally, Section 6 concludes this paper.

2   Preliminaries

In this section, some basic concepts and results from the algebraic graph theory are introduced [5].

Let a triplet $\mathcal{G} = (\mathcal{V}, \xi, \mathcal{A})$ denote a graph, where $\mathcal{V} = \{\nu_1, ..., \nu_m\}$ denotes a set of vertexes of order m. Each vertex corresponds to an agent, $\xi \subseteq \mathcal{V} \times \mathcal{V}$ denotes a set of edges, and $\mathcal{A} = \{a_{ij}\}$ is a nonnegative $m \times m$ matrix called the adjacency matrix satisfying $a_{ij} > 0$ if and only if ν_i and ν_j are connected, i.e., $(\nu_i, \nu_j) \in \xi$. If $a_{ij} > 0$, then it indicates that the two corresponding agents can exchange information. The graph $\mathcal{G}$ is undirected if

$$\forall \nu_i, \nu_j \in \mathcal{V} : (\nu_i, \nu_j) \in \xi \leftrightarrow (\nu_j, \nu_i) \in \xi.$$

Correspondingly, $\mathcal{A}$ becomes a symmetric matrix. Moreover, an undirected graph $\mathcal{G}$ is connected if for any pair of vertexes ν_i and ν_j, $i, j = 1, ..., m$, there is a path. In this paper, the following assumption holds.

Assumption 1 *The graph $\mathcal{G} = (\mathcal{V}, \xi, \mathcal{A})$ is undirected and connected. Moreover, no self-connection exists in the graph; i.e., $a_{ii} = 0, i = 1, ..., m$.*

Given a graph $\mathcal{G} = (\mathcal{V}, \xi, \mathcal{A})$, a diagonal matrix $D = \{\deg(\nu_1), ..., \deg(\nu_m)\} \in \Re^{m \times m}$ is called the degree matrix where $\deg(\nu_i) = \sum_{j=1, j \neq i}^{m} a_{ij} (i = 1, ..., m)$. In view of it, the Laplacian matrix of the graph is defined as $L = D - \mathcal{A}$. According to the Assumption 1, the Laplacian matrix has the following properties [18]:

1. L is positive semidefinite and symmetric.
2. 0 is a simple eigenvalue of L.

3   Problem Formulation

Consider a network of M discrete-time subsystems where each subsystem is described as follows:

$$x^i(k+1) = f^i(x^i(k), u^i(k)) + g^i(u), \tag{1}$$

where $x^i \in \Re^n$ is the state vector of the ith subsystem, $x = [x^1; \cdots ; x^m]$ is the state vector of the network, $u^i \in \Re^m$ is the input vector of the ith subsystem, $u = [u^1; \cdots ; u^m]$ is the input vector of the network, f^i is the model of the ith subsystem, and g^i denotes the coupling effects on the subsystem i caused by inputs of its neighbouring systems in the network.

In (1), $x^i(k)$, $u^i(k)$ are required to fulfill the following constraints:

$$x^i(k) \in \mathcal{X}^i, u^i(k) \in \mathcal{U}^i, \forall k \geq 0, \tag{2}$$

where $\mathcal{X}^i$ and $\mathcal{U}^i$ are closed compact convex sets. It is assumed that both $\mathcal{X}^i$ and $\mathcal{U}^i$ contain the origin as an interior point. In addition, the control system

(1) is sometimes subject to coupling constraints which denote communication structures and requirements among subsystems [19]:

$$\phi^i(x^i, x^j, u^i, u^j) \leq 0. \tag{3}$$

An MPC law is supposed to optimize a performance index iteratively over a predicted future horizon via the explicit use of the system model (1). In MPC, the control inputs are obtained by solving a constrained optimization problem during each sampling interval, using the current state as an initial state. For each subsystem i, the following performance index is considered:

$$J^i(u^i(k)) = \sum_{q=0}^{N-1} x^i(k+q|k)^T Q^i x^i(k+q|k)$$

$$+ \sum_{q=0}^{N-1} u^i(k+q|k)^T R^i u^i(k+q|k)$$

$$+ x^i(k+N|k)^T P^i x^i(k+N|k) \tag{4}$$

where $x^i(k+q|k)$ denotes the predicted state vector, $u^i(k+q|k)$ denotes the predicted input vector, N is the prediction horizon, Q^i, R^i are weighting matrices with compatible dimensions, and P^i is designed for closed-loop stability.

In many distributed MPC settings, each subsystem i independently minimizes the performance index (4) subject to its local constraints (1)-(3) to obtain the optimal control input $u^{i*}(k)$. In this paper, we consider a performance index of the overall control system as follows:

$$J(u(k)) = \sum_{i=1}^{M} J^i(u^i(k)), \tag{5}$$

where $J^i(u^i(k))$ is defined as (4). Correspondingly, the MPC problem is formulated as follows:

$$\min_{u^1(k), \cdots, u^M(k)} J = \sum_{i=1}^{M} J^i(u^i(k))$$

$$= \sum_{i=1}^{M} \sum_{q=0}^{N-1} x^i(k+q|k)^T Q^i x^i(k+q|k)$$

$$+ \sum_{i=1}^{M} \sum_{q=0}^{N-1} u^i(k+q|k)^T R^i u^i(k+q|k)$$

$$+ \sum_{i=1}^{M} x^i(k+N|k)^T P^i x^i(k+N|k)$$

$$\text{s.t.} \quad x^i(k+1) = f^i(x^i(k), u^i(k)) + g^i(u), i = 1, \cdots, M$$

$$x^i(k) \in \mathcal{X}^i, u^i(k) \in \mathcal{U}^i, \forall k \geq 0, i = 1, \cdots, M,$$

$$\phi^i(x^i, x^j, u^i, u^j) \leq 0, i = 1, \cdots, M. \tag{6}$$

The optimization problem (6) offers a framework for subsystems to cooperatively solve the MPC problem in a distributed manner. It differs from centralized MPC in that all parameters in (6) are designed based on the structures and characteristics of the corresponding subsystems. As a result, the objective functions $J^1, \cdots, J^M$ in optimization problem (6) are separable, which makes it suitable to be tackled by distributed optimization methods. However, it is worth noting that the presence of coupling effects $g(u)$ and $\phi(x, u)$ result in globally coupling constraints, which posts challenges for the design and implementation of distributed optimization algorithms. In this paper, we focused our attention on a special case of (6).

3.1   Linear systems with coupled constraints

For each subsystem i it is assumed that its state space model independent of other subsystem j in the network, however, the states and control inputs are required to satisfy coupled constraints [20].

$$x^i(k + 1) = A^i x^i(k) + B^i u^i(k),$$
$$\phi^i(x, u) \leq 0. \tag{7}$$

Assumption 2 *The constraint $\phi^i(x, u)$ is convex and linearly separable, i.e.,* $\phi^i(x, u) = \sum \phi^i_j(x^j, u^j)$.

This assumption is valid in many scenarios, especially when $\phi^i(x, u)$ is linear, i.e., $\phi^i(x, u) = C^i x + D^i u \leq 0$, it is convex and linearly separable.

Denote the following vectors as the predicted system information:

$$\bar{x}^i(k) = [x^i(k); x^i(k + 1); \cdots ; x^i(k + N)];$$
$$\bar{u}^i(k) = [u^i(k); u^i(k + 1); \cdots ; u^i(k + N)];$$
$$\Delta\bar{u}^i(k) = [\Delta u^i(k); \Delta u^i(k + 1); \cdots ; \Delta u^i(k + N)];$$

where $u^i(k) = u^i(k - 1) + \Delta u^i(k)$

Using (7) as the prediction model, the predicted states and control inputs of the subsystem i can be obtained

$$\bar{x}^i(k + 1) = S^i x^i(k) + M^i \Delta\bar{u}^i(k) + V^i u(k - 1), \tag{8}$$

where

$$S^i = \begin{bmatrix} A^i; \\ A^{i^2} \\ \vdots \\ A^{i^N} \end{bmatrix} \in \Re^{Nn \times n}, \; V = \begin{bmatrix} B^i \\ (A^i + I)B^i \\ \vdots \\ (A^{i^{N-1}} + \ldots + I)B \end{bmatrix} \in \Re^{Nn \times m},$$

$$M = \begin{bmatrix} B^i & \ldots & 0 \\ (A^i + I)B^i & \ldots & 0 \\ \vdots & \ddots & \vdots \\ (A^{i^{N-1}} + \ldots + I)B^i & \ldots & B^i) \end{bmatrix} \in \Re^{Nn \times Nm},$$

The corresponding distributed MPC problem can be correspondingly formulated as follows

$$
\min_{\Delta\bar{u}(k)} J = \sum_{i=1}^{M} \|S^i x^i(k) + M^i \Delta\bar{u}^i(k) + V^i u(k-1)\|_{Q_i}^2 + \|\Delta\bar{u}^i(k)\|_{R_i}^2
$$

$$
\text{s.t. } u_{\min}^i \leq M^i(\Delta\bar{u}^i(k) + V^i u(k-1)) \leq u_{\max}^i,
$$

$$
\sum_{i}^{M} C^i(S^i x^i(k) + M^i \Delta\bar{u}^i(k) + V^i u(k-1)) + D^i(\Delta\bar{u}^i(k) + V^i u(k-1)) \leq 0
$$

$$
\Delta u_{\min}^i \leq \Delta\bar{u}^i(k) \leq \Delta u_{\max}^i \tag{9}
$$

The optimization problem (9) is a distributed convex program, whose solution provides optimal control increments for all subsystems. Equivalently, (9) can be put into a compact form as follows:

$$
\min_{\Delta\bar{u}} J = \sum_{i=1}^{M} \frac{1}{2} \Delta\bar{u}^{i^T} W^i \Delta\bar{u}^i + p^{i^T} \Delta\bar{u}^i
$$

$$
\text{s.t. } \sum_{i}^{M} E^i \Delta\bar{u}^i + b^i \leq 0,
$$

$$
\sum_{i}^{M} H^i \Delta\bar{u}^i + q^i \leq 0, \ H^{j\neq i} = 0, \ q^{j\neq i} = 0, \ j = 1,...i,...,M,
$$

$$
\Delta u_{\min}^i \leq \Delta\bar{u}^i \leq \Delta u_{\max}^i, \tag{10}
$$

where $W^i = 2M^{i^T} Q M^i + R^i$, $p^i = 2M^{i^T} Q^i(S^i x^i(k) + +V^i u(k-1))$, $E^i = C^i M^i + D^i$, $b^i = C^i(S^i x^i(k) + V^i u(k-1)) + D^i(V^i u(k-1))$, $H^i = [M^i; -M^i]$, $q^i = [M^i V^i u(k-1) - u_{\max}^i; u_{\min}^i - M^i V^i u(k-1)]$.

4 Collective Neurodynamic Optimization Model

In this section, we propose a collective neurodynamic optimization model described by cooperative recurrent neural networks to solve the optimization problem (10) in a fully distributed fashion. Each recurrent neural network is employed by a subsystem to minimize its local cost function. Their collective efforts, guided by the topology of the network, enforce satisfaction of coupled constraints. The recurrent neural networks share information if and only if the two subsystems are connected. For each subsystem i, the corresponding recurrent neural network

is modeled as follows:

$$\frac{d}{dt}\beta_i = \text{Prog}_{\Omega^i}(\beta^i - W^i\beta^i - p^i - [E^i; H^i]^T\lambda^i) - \beta^i$$

$$\frac{d}{dt}\lambda^i = (\lambda^i + [E^i\beta^i + b^i; H^i\beta^i + q^i] - \sum_{j\in\mathscr{E}^i}(\gamma^i - \gamma^j + \lambda^i - \lambda^j))^+ - \lambda^i$$

$$\frac{d}{dt}\gamma^i = \sum_{j\in\mathscr{E}^i}(\lambda^i - \lambda^j) \tag{11}$$

At each time instant k, the output of the neurodynamic model in its equilibrium state β^{i*} is equal to the optimal control increment vector $\Delta\bar{u}^i(k)$. The optimal control input at k is obtained by implementing the first control action of the predicted vector: $u^i(k) = u^i(k-1) + \Delta u^i(k)$.

Intuitively, the neurodynamic model (11) exploits local information of each subsystem to reach consensus with the help of information sharing over the network. Denote $\mathscr{E}^i$ as the vertex set of the neighbors of the subsystem i. It can be viewed that λ^i is driven toward the average of λ^j by a proportional-integral controller $\sum_{j\in\mathscr{E}^i}\int(\lambda^i - \lambda^j)$. Thereafter, $(\lambda^i - \lambda^j)$ is expected to converge to 0. Next, we proceed to show that λ^i is equivalent to the Lagrange multiplier vector of (10).

Let β denote $[\Delta\bar{u}^1; \cdots; \Delta\bar{u}^M]$ and $\phi(\beta) = [\sum_i^M E^i\Delta\bar{u}^i + b^i; \sum_i^M H^i\Delta\bar{u}^i + q^i]$ for simplicity. The following lemma can be obtained.

Lemma 1. β^* *is an optimal solution to* (10) *if and only if there exists* λ^* *such that*

$$\beta^* = \text{Prog}_\Omega(\beta^* - (W\beta^* + p) - [E; H]^T\lambda^*)$$
$$\lambda^* = (\lambda^* + [E\beta^* + b; H\beta^* + q])^+.$$

Proof. It can be seen that the optimization (10) is convex since the objective is a convex function and the feasible domain is a convex set. According to the KKT conditions, variational equality conditions, and projection theorems [11], under the complementary conditions [2], β^* is an optimal solution to (10) if and only if there exist (β^*, λ^*) such that

$$\beta^* = \text{Prog}_\Omega(\beta^* - (\nabla J(\beta^*) + \nabla\phi^T(\beta^*)\lambda^*)), \ \beta^* \in \Omega \tag{12}$$

$$\phi^T(\beta^*) \le 0, \ \lambda^* \ge 0, \ {\lambda^*}^T\phi(\beta^*)\lambda^* = 0. \tag{13}$$

where $\Omega = \{\mu \in \Re^n : l_k \le \mu_k \le h_h\}$ and

$$\text{Prog}_\Omega(\mu) = \begin{cases} l_k, \ \mu_k < l_k; \\ \mu_k, l_k \le \mu_k \le h_k; \\ h_k, \mu_k > h_k; \end{cases}$$

Moreover, (13) can be equivalently put into $\lambda^* = (\lambda^* + \phi(\beta^*))^+$, where

$$(\mu_k)^+ = \begin{cases} 0, \ \mu_k \le 0; \\ \mu_k, \mu_k > 0. \end{cases}$$

Therefore, it is shown that λ in (11) is equal to the Lagrange multiplier. $\square$

The distributed MPC approach based on the collective neurodynamic optimization models is summarized as follows:

1. Let $k = 1$. Set MPC parameters including the control time terminal T, prediction horizon N, sampling period t, weight matrices Q and R.
2. Compute parameters of the optimization model including W, p, E, b, H, q.
3. Solve the distributed optimization problem using the proposed neurodynamic models to obtain the optimal control increment vector $\Delta\bar{u}(k)$.
4. Compute and implement the optimal control action $u(k)$.
5. If $k < T$, let $k = k + 1$, go to Step 2; otherwise terminate.

5 Simulation Results

In this section, the formation control of flying robots which aim to form and maintain desired relative position and orientation is considered. The coupled constraints arise due to the considerations for collision and obstacles avoidance [10]. For each mobile robot i, its state space model is

$$\begin{bmatrix} x^i(k+1) \\ y^i(k+1) \\ \dot{x}^i(k+1) \\ \dot{y}^i(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^i(k) \\ y^i(k) \\ \dot{x}^i(k) \\ \dot{y}^i(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \begin{bmatrix} u^i_x(k) \\ u^i_y(k) \end{bmatrix} \tag{14}$$

where $[x^i, y^i]$ denotes the position coordinates of the robot i, $[\dot{x}^i, \dot{y}^i]$ denotes a vector of velocity components along x-axis and y-axis, and $[u^i_x, u^i_y]$ denotes a vector of acceleration components along x-axis and y-axis.

The distributed MPC of the flying mobiles seeks the solution to the problem (6) via the neurodynamic model (11) based on the formulation (9). The linear constraints on states and inputs of every flying robots are $|x| \leq [100; 100; 24; 24]$ and $|u| \leq [2; 2]$. The coupled constraints are introduced to ensure flying robots cannot enter protection zones of each other, and they are represented as $\|(x^{i^2} + y^{i^2}) - (x^{j^2} + y^{j^2})\|_\infty \leq d^2_{\min}$.

We consider a scenario of three flying robots formed a formation in a structure shown in Fig. 1. The initial conditions of the three robots are $x^1(0) = [1; -3; 0; 0]$,
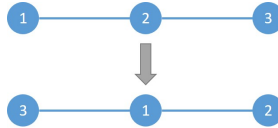

Fig. 1. Formation structure of three robots

$x^2(0) = [10; -3; 0; 0]$, $x^3(0) = [15; -3; 0; 0]$. The final conditions are $x^1(T) =$

$[6; 5; 0; 0]$, $x^2(T) = [11; 5; 0; 0]$, $x^3(T) = [3; 5; 0; 0]$. The protection zone of each robot is $0.3m$. The controlled result is depicted in Fig. 2. The control inputs are shown in Figs. 3-4. It is shown that flying robots can effectively form the desired formation with guaranteed input and safety constraints satisfaction.
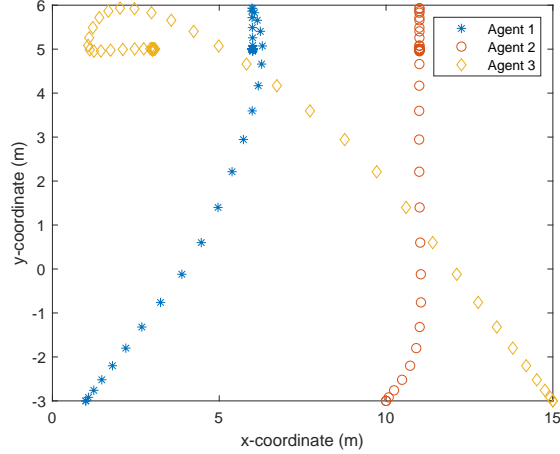

Fig. 2. Formation trajectory

6   Conclusion

This paper presented a model predictive control approach to linear systems with coupled constraints in a fully distributed fashion. The global cost function took an additive form of each local cost functions. The model predictive controllers of each subsystem were designed based on local information only, and were computed by using a neurodynamic model in real time. The collective efforts of neurodynamic models forced the local controllers to reach consensus at the global optimal control with theoretically guaranteed optimality. optimality were given. An illustrative example on the formation control of flying robots was provided to demonstrate the effectiveness of the approach. Future research will be directed to uncertain systems involving more complex network topologies.

References

1. Akesson, B., Toivonen, H.: A neural network predictive controller. J. Process Contr. **16**, 937–946 (2006)
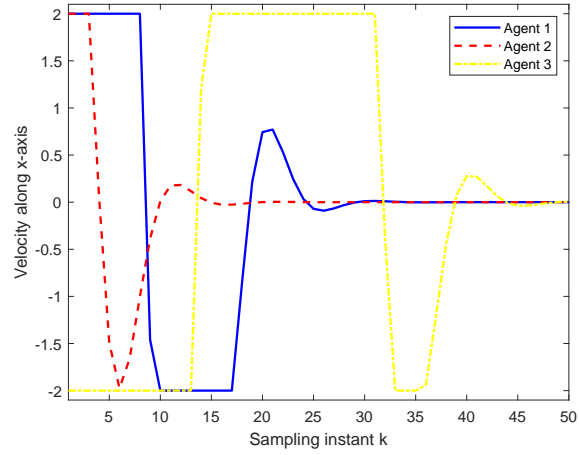2. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear programming: theory and algorithms. John Wiley & Sons (2013)
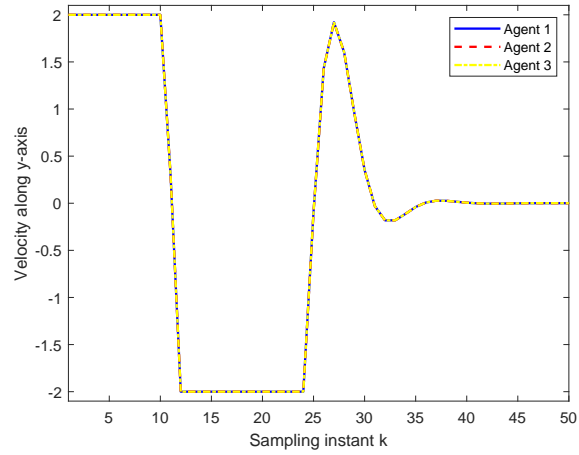
Fig. 3. Velocity along x-axis

Fig. 4. Velocity along y-axis

3. Cheng, L., Hou, Z., Tan, M.: Constrained multi-variable generalized predictive control using a dual neural network. Neural Comput. Appl. **16**(6), 505–512 (2007)
4. Christofides, P.D., Scattolini, R., de la Pena, D.M., Liu, J.: Distributed model predictive control: A tutorial review and future research directions. Computers & Chemical Engineering **51**, 21–41 (2013)
5. Godsil, C., Royle, G.F.: Algebraic Graph Theory, vol. 207. Springer Science & Business Media (2013)
6. Guo, Z., Liu, Q., Wang, J.: A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints. IEEE Trans. Neural Netw. **22**(12), 1892–1900 (2011)
7. Hu, C., Lu, J., Liu, X., Zhang, G.: Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. Computers & Operations Research **94**, 139–153 (2018)
8. Hu, C., Liu, X., Lu, J.: A bi-objective two-stage robust location model for waste-to-energy facilities under uncertainty. Decision Support Systems **99**, 37–50 (2017)
9. Hu, X., Wang, J.: An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application. IEEE Trans. Neural Netw. **19**(12), 2022–2031 (2008)
10. Keviczky, T., Borrelli, F., Balas, G.J.: A study on decentralized receding horizon control for decoupled systems. In: Proc. American Contr. Conf. pp. 4921–4926 (2004)
11. Kinderlehrer, D., Stampacchia, G.: An introduction to variational inequalities and their applications, vol. 31. Siam (2000)
12. Le, X., Chen, S., Yan, Z., Xi, J.: A neurodynamic approach to distributed optimization with globally coupled constraints. IEEE Transactions on Cybernetics (2018), doi:10.1109/TCYB.2017.2760908
13. Lezana, P., Aguilera, R., Quevedo, D.: Model predictive control of an asymmetric flying capacitor converter. IEEE Trans. Ind. Electron. **56**(6), 1894–1905 (2009)
14. Li, G., Yan, Z., Wang, J.: A one-layer recurrent neural network for constrained nonsmooth invex optimization. Neural Networks **50**, 79–89 (2014)
15. Liu, Q., Wang, J.: A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. IEEE Trans. Neural Netw. **19**(4), 558–570 (2008)
16. Liu, Q., Wang, J.: Finite-time convergent current neural network with hard-limiting activation function for constrained optimization with piecewise-linear objective functions. IEEE Trans. Neural Netw. **22**(4), 601–613 (2011)
17. Liu, Q., Wang, J.: A one-layer recurrent neural network for constrained nonsmooth optimization. IEEE Trans. Syst., Man Cyberne. B Cybern. **40**(5), 1323–1333 (2011)
18. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE **95**(1), 215–233 (2007)
19. Richards, A., How, J.P.: Robust distributed model predictive control. International Journal of Control **80**(9), 1517–1531 (2007)
20. Stewart, B.T., Venkat, A.N., Rawlings, J.B., Wright, S.J., Pannocchia, G.: Co-operative distributed model predictive control. Systems & Control Letters **59**(8), 460–469 (2010)
21. Xia, Y., Feng, G., Wang, J.: A novel neural network for solving nonlinear optimization problems with inequality constraints. IEEE Trans. Neural Netw. **19**(8), 1340–1353 (2008)
22. Yan, J., Bitmead, R.: Incorporating state estimation into model predictive control and its application to network traffic control. Automatica **41**, 595–604 (2005)

23. Yan, Z., Wang, J.: Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks. IEEE Trans. Ind. Informat. **8**(4), 746–756 (2012)
24. Yan, Z., Fan, J., Wang, J.: A collective neurodynamic approach to constrained global optimization. IEEE Transactions on Neural Networks and Learning Systems **28**(5), 1206–1215 (2017)