IET Control Theory & Applications

Research Article

ISSN 1751-8644
Received on 22nd May 2014
Revised on 8th December 2014
Accepted on 17th December 2014
doi: 10.1049/iet-cta.2014.0549
www.ietdl.org

Fast gradient-based distributed optimisation approach for model predictive control and application in four-tank benchmark

Xiaojun Zhou[1], *Chaojie Li*[2] ✉, *Tingwen Huang*[3], *Mingqing Xiao*[4]

[1]*School of Information Science and Engineering, Central South University, Changsha 410083, People's Republic of China*
[2]*School of Electrical and Computer Engineering, RMIT University, Melbourne VIC 3000, Australia*
[3]*Texas A & M University at Qatar, Doha 5825, Qatar*
[4]*Department of Mathematics, Southern Illinois University, Carbondale, Illinois 62901, USA*
✉ *E-mail: cjlee.cqu@163.com*

Abstract: By taking both control and state vectors as decision variables, the subproblems of model predictive control scheme can be considered as a class of separable convex optimisation problems with coupling linear constraints. A Lagrangian dual method is introduced to deal with the optimisation problem, in which, the primal problem is solved by a parallel coordinate descent method, and a fast dual ascend method is adopted to solve the dual problem iteratively. The proposed approach is applied to the well-known hierarchical and distributed model predictive control four-tank benchmark. Experimental results have testified the effectiveness of the proposed approach and shown that the benchmark problem can be well stabilised.

1 Introduction

Model predictive control (MPC) is a kind of control method, in which, the current control vector is obtained by solving a finite horizon open-loop optimal control problem, namely, a constrained optimisation problem. At each sampling period, using current state of the plant as initial state, the optimisation problem yields an optimal control sequence and the first control in this sequence is applied to the plant [1]. The big advantage of this type of control is its ability to achieve optimal control performance and to cope with hard constraints on both control and state vectors. In recent years, the emergence of large-scale systems, such as power systems, process control systems, multi-agent systems and transportation systems, has challenged the applicability of traditional MPC because of the high computational burden and low control performance [2–6].

There exist three MPC strategies for designing such large-scale systems, namely, distributed MPC, centralised MPC and decentralised MPC, of which, the distributed MPC is arguably the most promising one because it beats the decentralised MPC one in control performance and outperforms the centralised MPC one in computational burden [5]. From a centralised control point of view, all subsystems are optimised with respect to an objective function in a single optimisation problem, making plantwide control difficult to coordinate and maintain. The key feature of a decentralised control framework is that there is no communication among different local controllers. It is well-known that strong interactions among different subsystems may impede achieving stability and desired performance when using decentralised control. In distributed MPC, some level of communication may be established among different controllers to achieve better closed-loop control performance. Distributed MPC can be non-cooperative or cooperative, depending on whether the same global objective function is optimised in each local controller. A controller is considered to have a non-cooperative attitude if it only seeks to minimise its own objective function. On the other hand, if it aims to minimise not only its own local cost but the system-wide (or global) cost, it is called a cooperative controller. A recent review of distributed MPC can be found in [7, 8] and the references therein.

Distributed optimisation is a kind of decomposition method for solving distributed MPC problems, in which, a large optimisation problem is decomposed into a number of smaller and more tractable ones. Decomposition methods can be divided into two main categories: primal and dual decompositions. In primal decomposition, the optimisation problem is solved using the original formulation and variables, while the constraints are handled with methods like interior-point, penalty function, feasible direction, coordinate descent and Newton methods [9–12]. In dual decomposition, the Lagrangian dual is introduced, and the main idea is to solve the primal and dual problems alternatively [13–15]. A limitation of classical gradient-based methods based on Lagrangian duality is the low convergence rate. Fortunately, in [16, 17], it is shown that an accelerated gradient algorithm can be constructed using only the first-order information. Furthermore, when the objective function is separable (a function is called separable if it is a sum of functions of its individual variables), it is able to construct parallel methods for solving the primal problem [18].

The goal of this paper is to solve a benchmark MPC problem using a fast distributed optimisation approach. The contribution of the paper is three-fold: (i) the optimisation formulation which enables parallel implementation is established for the hierarchical and distributed (HD)-MPC benchmark, and to the best of our knowledge, this is the first time, since in traditional optimisation formulation for the benchmark, the objective function is not separable and thus hinders the parallel implementation; (ii) a fast distributed optimisation approach is proposed for the subproblems in the MPC scheme, which combines a fast dual method with a parallel coordinate method for primal problem; (iii) the convergence properties are proven for both the parallel coordinate method and the fast dual ascend method (FDAM). It should be noted that the parallel coordinate descent algorithm in [11] cannot be appropriately applied to the benchmark problem since it only deals with simple box constraints. The accelerated dual gradient-projection algorithm in [13] has the similar computational complexity to this study for dual problem, but the parallel mechanism for primal problem is lost. Although we have adopted the similar parallel algorithm used in [15], the proposed FDAM is much simpler than that dual gradient method, which, moreover, is an inexact dual method, and

© The Institution of Engineering and Technology 2015

cannot reflect the true physical balance relationship between the state and the control input in state equation.

This paper is organised as follows. In Section 2, we give a brief description of the HD-MPC benchmark problem. In Section 3, we show that the subproblems of the MPC scheme can be reformulated as a class of separable convex optimisation problems with coupling linear constraints. Furthermore, a Lagrangian dual method is introduced to deal with the constrained optimisation problem, in which, the primal problem is solved by a parallel coordinate descent method, and a FDAM is adopted to solve the dual problem iteratively. Some experimental results are given to show the effectiveness of the proposed approach in Section 4, followed by the conclusion in Section 5.

2 Problem description

The HD-MPC four-tank benchmark [19, 20] is widely used to test, evaluate and compare different approaches in control system design, because of its interesting properties: (i) there exists strong coupling between subsystems and the degree of coupling can be conveniently manipulated; (ii) there exhibit non-linear dynamics in the plant; (iii) the states can be measured; (iv) the plant is subject to hard states and inputs constraints; and (v) the plant can be safely operated.

As illustrated in Fig. 1, the four-tank plant has been divided into two subsystems which are coupled through the inputs. The tanks at the top (tanks 3 and 4) discharge into the corresponding tanks at the bottom (tanks 1 and 2). The three-way valves are emulated by a proper calculation of the set-points of the flow control loops according to the considered ratio of the three-way valve. Thus, the inlet flows of the three-way valves q_a and q_b can be considered as the manipulated variables of the real plant.

The system is governed by the following differential equations

$$\begin{aligned}
\frac{\mathrm{d}h_1}{\mathrm{d}t} &= -\frac{a_1}{S}\sqrt{2gh_1} + \frac{a_3}{S}\sqrt{2gh_3} + \frac{\gamma_a}{S}q_a \\
\frac{\mathrm{d}h_2}{\mathrm{d}t} &= -\frac{a_2}{S}\sqrt{2gh_2} + \frac{a_4}{S}\sqrt{2gh_4} + \frac{\gamma_b}{S}q_b \\
\frac{\mathrm{d}h_3}{\mathrm{d}t} &= -\frac{a_3}{S}\sqrt{2gh_3} + \frac{1-\gamma_b}{S}q_b \\
\frac{\mathrm{d}h_4}{\mathrm{d}t} &= -\frac{a_4}{S}\sqrt{2gh_4} + \frac{1-\gamma_a}{S}q_a
\end{aligned} \tag{1}$$

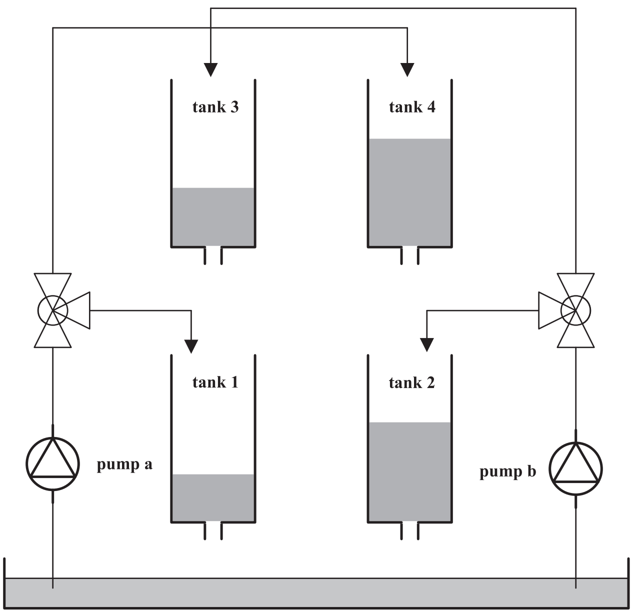

Fig. 1 *Schematic diagram of the four-tank process*

Table 1 Parameters of the benchmark plant

Parameter	Value	Description
$h_{1\max}$	1.36 m	maximum level of tank 1
$h_{2\max}$	1.36 m	maximum level of tank 2
$h_{3\max}$	1.30 m	maximum level of tank 3
$h_{4\max}$	1.30 m	maximum level of tank 4
$h_{\min}$	0.20 m	minimum level of all cases
$q_{a\max}$	3.26 m^3/h	maximum flow of q_a
$q_{b\max}$	4 m^3/h	maximum flow of q_b
$q_{\min}$	0 m^3/h	minimum flow of q_a, q_b
a_1	1.31×10^{-4} m^2	discharge constant of tank 1
a_2	1.51×10^{-4} m^2	discharge constant of tank 2
a_3	9.27×10^{-5} m^2	discharge constant of tank 3
a_4	8.82×10^{-5} m^2	discharge constant of tank 4
S	0.06 m^2	cross section of all tanks
γ_a	0.3	ratio of the three-way valve
γ_b	0.4	ratio of the three-way valve
h_1^0	0.65 m	linearisation level of rank 1
h_2^0	0.66 m	linearisation level of rank 2
h_3^0	0.65 m	linearisation level of rank 3
h_4^0	0.66 m	linearisation level of rank 4
q_a^0	1.63 m^3/h	linearisation flow of q_a
q_b^0	2.00 m^3/h	linearisation flow of q_b

where $h_i, a_i, i = 1, \ldots, 4$, denote the water level and the discharge constant of tank i, respectively; $q_j, \gamma_j, j \in \{a, b\}$ represent the flow and the ratio of the three-way valve of pump j, respectively; S is the cross section of each tank, and g is the gravitational acceleration ($g = 9.8\,\mathrm{m}^2/\mathrm{s}$ in this study).

By linearising the model at an operating point given by the equilibrium levels and flows as shown in Table 1, and defining the deviation variables

$$\begin{aligned}
x_i &= h_i - h_i^0, \quad i = 1, \ldots, 4 \\
u_j &= q_j - q_j^0, \quad j \in \{a, b\}
\end{aligned} \tag{2}$$

then we can obtain the continuous-time linear model as follows

$$\begin{cases} \dot{x} = A_c x + B_c u \\ y = C_c x \end{cases} \tag{3}$$

where $x = (x_1, x_2, x_3, x_4)$, $u = (u_a, u_b)$, $y = (x_1, x_2)$ and

$$A_c = \begin{bmatrix} -\frac{1}{\tau_1} & 0 & \frac{1}{\tau_3} & 0 \\ 0 & -\frac{1}{\tau_2} & 0 & \frac{1}{\tau_4} \\ 0 & 0 & -\frac{1}{\tau_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_4} \end{bmatrix} \tag{4}$$

$$B_c = \begin{bmatrix} \frac{\gamma_a}{S} & 0 \\ 0 & \frac{\gamma_b}{S} \\ 0 & \frac{1-\gamma_b}{S} \\ \frac{1-\gamma_a}{S} & 0 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

where

$$\tau_i = \frac{S}{a_i}\sqrt{\frac{2h_i^0}{g}}, \quad i = 1, \ldots, 4$$

is the time constant of tank i.

Using the ZOH (zero-order hold) method with a sampling period of 5 s, the discrete-time model is obtained in the following

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{6}$$

IET Control Theory Appl., 2015, Vol. 9, Iss. 10, pp. 1579–1586
© The Institution of Engineering and Technology 2015

where

$$A = \begin{bmatrix} 0.9705 & 0 & 0.0207 & 0 \\ 0 & 0.9663 & 0 & 0.0195 \\ 0 & 0 & 0.9790 & 0 \\ 0 & 0 & 0 & 0.9802 \end{bmatrix}$$

$$B = \begin{bmatrix} 24.6291 & 0.5213 \\ -0.1967 & 32.7684 \\ 0 & 49.4735 \\ -19.8011 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

By using an MPC scheme, an online finite horizon open-loop optimal control problem can be described as follows

$$\begin{aligned} \min \quad & J(k, \mathbf{u}) = \sum_{i=0}^{N-1} \left[\frac{1}{2} \| x(k+i+1|k) \|_Q^2 \right. \\ & \qquad \left. + \frac{1}{2} \| u(k+i|k) \|_R^2 \right] \\ \text{s.t.} \quad & x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) \\ & x(k|k) = x(k) \\ & \underline{x} = [h_{\min}, h_{\min}, h_{\min}, h_{\min}]^{\mathrm{T}} \le x(k+i|k) \\ & \quad \le [h_{1\max}, h_{2\max}, h_{3\max}, h_{4\max}]^{\mathrm{T}} = \bar{x} \\ & \underline{u} = [q_{\min}, q_{\min}]^{\mathrm{T}} \le u(k+i|k) \le [q_{a\max}, q_{b\max}]^{\mathrm{T}} \\ & \quad = \bar{u}, i = 0, 1, \ldots, N-1 \end{aligned} \tag{7}$$

where $\mathbf{u} = [u(k|k)^{\mathrm{T}}, u(k+1|k)^{\mathrm{T}}, \ldots, u(k+N-1|k)^{\mathrm{T}}]^{\mathrm{T}}$ are the decision variables; $Q \succ 0, R \succ 0$ are strictly positive definite symmetric weighting matrices; and N is the length of prediction horizon. At time k, $u(k) = u(k|k)$ is implemented and the optimisation problem (7) is repeated at time $k+1$. The goal of the MPC scheme is to achieve stabilisation of the discrete system at the origin by solving the optimal control problem [1].

Considering that

$$\begin{aligned} x(k+i+1|k) &= Ax(k+i|k) + Bu(k+i|k) \\ x(k|k) &= x(k) \end{aligned} \tag{8}$$

we have

$$\begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \cdots \\ x(k+N|k) \end{bmatrix} = \bar{S} \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \cdots \\ u(k+N-1|k) \end{bmatrix} + \bar{T}x(k|k) \tag{9}$$

where

$$\bar{S} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}, \quad \bar{T} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \tag{10}$$

By eliminating state variables $x(k+i+1|k), i = 0, \ldots, N-1$, we can get the traditional optimisation formulation for MPC problem as follows

$$\begin{aligned} \min_{\mathbf{u} \in U} \quad & J(k, \mathbf{u}) = \frac{1}{2} \mathbf{u}^{\mathrm{T}} \bar{H} \mathbf{u} + \bar{g}^{\mathrm{T}} \mathbf{u} \\ \text{s.t.} \quad & \bar{b}_l \le \bar{S}\mathbf{u} + \bar{T}x(k|k) \le \bar{b}_u \end{aligned} \tag{11}$$

here $\bar{H} = \bar{S}^{\mathrm{T}} \bar{Q} \bar{S} + \bar{R}$, $\bar{g} = (\bar{S}^{\mathrm{T}} \bar{Q} \bar{T}) x(k|k)$, $\bar{R} = \mathrm{diag}\{R, \cdots, R\}$, $\bar{Q} = \mathrm{diag}\{Q, \cdots, Q\}$, U indicates the constraints imposed on flow level, $\bar{b}_l$ and $\bar{b}_u$ indicate the constraints imposed on water level.

Remark 1: It should be noted that by the traditional elimination method, the corresponding objective function in the optimisation problem cannot be separable. A function is separable means that it can be written as a sum of functions of its individual variables. Owing to the inseparability, distributed optimisation approaches cannot be applied to this formulation.

3 Distributed optimisation approach

3.1 Dual decomposition

For simplicity and compactness, we use x_{k-1} to represent $x(k+i|k)$ and u_{k-1} to represent $u(k+i|k)$, the constrained finite horizon linear quadratic optimisation problem (7) can be rewritten as

$$\begin{aligned} \min \quad & J(\mathbf{x}, \mathbf{u}) = \sum_{k=1}^{N} \left[\frac{1}{2} x_k^{\mathrm{T}} Q x_k + \frac{1}{2} u_{k-1}^{\mathrm{T}} R u_{k-1} \right] \\ \text{s.t.} \quad & x_k = Ax_{k-1} + Bu_{k-1} \\ & \underline{u} \le u_k \le \bar{u} \\ & \underline{x} \le x_k \le \bar{x} \\ & x_0 = x^0 = x(k|k) \end{aligned} \tag{12}$$

Let $z_k = tx_k^{\mathrm{T}}, u_{k-1}^{\mathrm{T}}]^{\mathrm{T}}$, then (12) can be expressed as

$$\begin{aligned} \min_{z \in \Omega} \quad & f(z) = \sum_{k=1}^{N} \frac{1}{2} z_k^{\mathrm{T}} H z_k \\ \text{s.t.} \quad & Sz = b \end{aligned} \tag{13}$$

where $z = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}, \ldots, z_N^{\mathrm{T}}]^{\mathrm{T}}$, $\Omega = \Omega_1 \times \Omega_2 \cdots \times \Omega_N = \{z | [\underline{x}^{\mathrm{T}}, \underline{u}^{\mathrm{T}}]^{\mathrm{T}} \le z_k \le [\bar{x}^{\mathrm{T}}, \bar{u}^{\mathrm{T}}]^{\mathrm{T}}\}$, $H = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$, $b = \begin{bmatrix} (Ax^0)^{\mathrm{T}} & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}$ and

$$S = \begin{bmatrix} I & -B & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -A & 0 & I & -B & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -A & 0 & I & -B \end{bmatrix}$$

Partitioning the matrix S conformably as

$$S = [S_1, \ldots, S_N]$$

so that $Sz = \sum_{k=1}^{N} S_k z_k$, the Lagrangian associated with the optimisation problem (13) can be expressed as

$$\mathcal{L}(z, \lambda) = \sum_{k=1}^{N} \left(\frac{1}{2} z_k^{\mathrm{T}} H z_k + \lambda^{\mathrm{T}} S_k z_k - (1/N) \lambda^{\mathrm{T}} b \right) \tag{14}$$

where λ are the Lagrange dual (or multiplier) variables.

We use a primal-dual scheme to solve the constrained optimisation problem (13). For a fixed λ, the primal (also called inner) subproblem can be defined by

$$z(\lambda) = \arg\min_{z \in \Omega} \mathcal{L}(z, \lambda) \tag{15}$$

and the dual (also called outer) problem is

$$\max g(\lambda) \tag{16}$$

where $g(\lambda) = \min_{z \in \Omega} \mathcal{L}(z, \lambda)$ is the dual function.

© The Institution of Engineering and Technology 2015

Since the constraints are affine, it is easy to find that strong duality holds, that is

$$\max g(\lambda) = g^* = f^* = \min_{z\in\Omega}\left\{ f(z) = \sum_{k=1}^{N} \frac{1}{2} z_k^{\mathrm{T}} H z_k \,\middle|\, Sz = b \right\} \tag{17}$$

which indicates that the constrained optimisation problem (13) can be solved based on a primal-dual algorithm framework [18] as follows

$$z^{i+1} = \arg\min_{z\in\Omega} \mathcal{L}(z, \lambda^i), \tag{18a}$$

$$\lambda^{i+1} = \arg\max g(\lambda, z^{i+1}) \tag{18b}$$

3.2 Parallel algorithm for primal problem

As shown in (14) and (15), when the dual variables λ are fixed, then the objective function is separable when minimising the inner problem (15) and thus solving the primal problem can be done in parallel. Distributed processors can deal with the N optimisation subproblems simultaneously, and they share the same dual variable. The coordinator is to broadcast the dual variables information to each processor and gather the results obtained by these distributed processors iteratively. In the following, a linearly convergent parallel coordinate descent method is introduced.

Let us partition the identity matrix conformably as

$$I = [E_k^{\mathrm{T}}, \ldots, E_N^{\mathrm{T}}]^{\mathrm{T}}$$

such that $z = \sum_{k=1}^{N} E_k z_k$, and define the partial gradient $\nabla_k \mathcal{L}(z, \lambda)$ of $\nabla \mathcal{L}(z, \lambda)$ with respect to z as $\nabla_k \mathcal{L}(z, \lambda) = E_k^{\mathrm{T}} \nabla \mathcal{L}(z, \lambda)$. It is obvious to find that the gradient of $\mathcal{L}(z, \lambda)$ is coordinate-wise Lipschitz continuous

$$\|\nabla_k \mathcal{L}(z + E_k h_k, \lambda) - \nabla_k \mathcal{L}(z, \lambda)\| \le L_k \|h_k\|$$

where $L_k = \sigma_{\max}(H)$ is the largest singular value of H, and it is easy to deduce that

$$\mathcal{L}(z + E_k h_k, \lambda) \le \mathcal{L}(z, \lambda) + \nabla_k \mathcal{L}^{\mathrm{T}}(z, \lambda) h_k + \frac{L_k}{2} \|h_k\|_2^2 \tag{19}$$

Introduce the following norm

$$\|z\|_1^2 = \sum_{k=1}^{N} L_k \|z_k\|_1^2 \tag{20}$$

and it is not difficult to check that $\mathcal{L}(z, \lambda)$ is strongly convex with respect to z under the above norm, that is

$$\mathcal{L}(z^1, \lambda) \ge \mathcal{L}(z^2, \lambda) + \nabla \mathcal{L}^{\mathrm{T}}(z^2, \lambda)(z^1 - z^2) + \frac{\rho}{2}\|z^1 - z^2\|_1^2 \quad \forall z^1, \quad z^2 \in \Omega \tag{21}$$

where $\rho = \{[\sigma_{\min}(H)]/[\sigma_{\max}(H)]\} \le 1$, here, $\sigma_{\min}(H)$ is the smallest singular value of H.

Considering that for any given λ, the optimal solution of inner problem (15) can be computed separably in parallel via the following coordinate update, which is a parallel version of the coordinate descent method in [21] and further developed in [15].

Parallel coordinate descent method (PCDM)
$\forall\, k = 1, \ldots, N$, compute in parallel:
Step 0. Take $z_k^0 \in \Omega_k$
Step i. ($i > 0$)
repeat

$$\bar{z}_k^i = \arg\min_{z_k\in\Omega_k} \left\{ \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z_k - z_k^i) + \frac{L_k}{2}\|z_k - z_k^i\|_2^2 \right\} \tag{22}$$

$$z_k^{i+1} = \frac{1}{N}\bar{z}_k^i + \frac{N-1}{N} z_k^i \tag{23}$$

until $\|z_k^{i+1} - z_k^i\| \le \epsilon$

Remark 2: The inner problem can be solved in parallel since for a fixed λ, minimising $(\frac{1}{2}z_{k'}^{\mathrm{T}} H z_{k'} + \lambda^{\mathrm{T}} S_{k'} z_{k'} - (1/N)\lambda^{\mathrm{T}} b)$ is independent of minimising $(\frac{1}{2}z_{k''}^{\mathrm{T}} H z_{k''} + \lambda^{\mathrm{T}} S_{k''} z_{k''} - (1/N)\lambda^{\mathrm{T}} b)$, $\forall\, k', k'' \in \{1, 2, \ldots, N\}$, and they can be computed simultaneously.

Introducing the following measure to calculate the distance between current iterative and the optimal solution

$$d_i^2 = \|z^i - z^*\|_1^2 = \sum_{k=1}^{N} L_k \|z_k^i - z_k^*\|_2^2 \tag{24}$$

where z^* is the optimal solution of (15) and $z_k^* = (E^k)^{\mathrm{T}} z^*$, then we have the following lemma.

Lemma 1: If $\mathcal{L}(z, \lambda)$ is coordinate-wise Lipschitz continuous with constant L_k, using the algorithm PCDM for iterative update, with the distance defined in (24), then

$$d_{i+1}^2 + 2\mathcal{L}(z^{i+1}, \lambda) \le d_i^2 + 2\mathcal{L}(z^i, \lambda) + \frac{2}{N}\nabla \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z^* - z^i) \tag{25}$$

Proof:

$$\begin{aligned} d_{i+1}^2 &= \sum_{k=1}^{N} L_k \left\| z_k^{i+1} - z_k^* \right\|_2^2 \\ &= \sum_{k=1}^{N} L_k \left\| \frac{1}{N}\bar{z}_k^i + \frac{N-1}{N} z_k^i - z_k^* \right\|_2^2 \\ &= d_i^2 + \sum_{k=1}^{N} L_k \left[\frac{1}{N^2}\|\bar{z}_k^i - z_k^i\|_2^2 + \frac{2}{N}\left(\bar{z}_k^i - z_k^i\right)^{\mathrm{T}}\left(z_k^i - z_k^*\right) \right] \\ &= d_i^2 + \sum_{k=1}^{N} L_k \left[\left(\frac{1}{N^2} - \frac{2}{N}\right)\|\bar{z}_k^i - z_k^i\|_2^2 + \frac{2}{N}\left(\bar{z}_k^i - z_k^i\right)^{\mathrm{T}}\left(\bar{z}_k^i - z_k^*\right) \right] \\ &\le d_i^2 + \sum_{k=1}^{N} L_k \left[-\frac{1}{N}\|\bar{z}_k^i - z_k^i\|_2^2 + \frac{2}{N}\left(\bar{z}_k^i - z_k^i\right)^{\mathrm{T}}\left(\bar{z}_k^i - z_k^*\right) \right] \end{aligned}$$

The optimality condition for (22) implies that

$$[\nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda) + L_k(\bar{z}_k^i - z_k^i)]^{\mathrm{T}}(z_k - \bar{z}_k^i) \ge 0, \forall z_k \in \Omega_k$$

© The Institution of Engineering and Technology 2015

Let $z_k^* = z_k$ to invoke the above inequality, we have

$$L_k(\bar{z}_k^i - z_k^i)^{\mathrm{T}}(\bar{z}_k^i - z_k^*) \leq \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z_k^* - \bar{z}_k^i)$$

To continue, we can obtain

$$\begin{aligned} d_{i+1}^2 &\leq d_i^2 + \sum_{k=1}^N L_k \bigg[-\frac{1}{N} \|\bar{z}_k^i - z_k^i\|_2^2 \\ &\quad + \frac{2}{N} \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z_k^* - \bar{z}_k^i) \bigg] \\ &= d_i^2 - \frac{2}{N} \sum_{k=1}^N \left[\frac{L_k}{2} \|\bar{z}_k^i - z_k^i\|_2^2 + \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(\bar{z}_k^i - z_k^i) \right] \\ &\quad + \frac{2}{N} \sum_{k=1}^N \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z_k^* - z_k^i) \end{aligned}$$

Due to the fact that

$$\begin{aligned} z^{i+1} &= z^i + \frac{1}{N}(\bar{z}^i - z^i) \\ &= \frac{1}{N} \sum_{k=1}^N [z^i + E_k(\bar{z}_k^i - z_k^i)] \end{aligned}$$

and invoking (19) with $z = z^i$, $h_k = \bar{z}_k^i - z_k^i$, by the convexity of $L(z, \lambda)$ with respect to z, we have

$$\begin{aligned} \mathcal{L}(z^{i+1}, \lambda) &\leq \frac{1}{N} \sum_{k=1}^N \mathcal{L}(z^i + E_k(\bar{z}_k^i - z_k^i), \lambda) \\ &= \frac{1}{N} \sum_{k=1}^N \bigg[\mathcal{L}(z^i, \lambda) + \nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(\bar{z}_k^i - z_k^i) \\ &\quad + \frac{L_k}{2} \|\bar{z}_k^i - z_k^i\|_2^2 \bigg] \\ &= \mathcal{L}(z^i, \lambda) + \frac{1}{N} \sum_{k=1}^N \bigg[\nabla_k \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(\bar{z}_k^i - z_k^i) \\ &\quad + \frac{L_k}{2} \|\bar{z}_k^i - z_k^i\|_2^2 \bigg] \end{aligned}$$

Utilising the above inequality, we can get the one of our main results consequently. □

Theorem 1: Under the conditions of Lemma 1 and the condition expressed in (21), the algorithm PCDM has the following linear convergence

$$\begin{aligned} \mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) &\leq \left(1 - \frac{\beta}{N}\right)^i \\ &\quad \times \left[\frac{1}{2} d_0^2 + \mathcal{L}(z^0, \lambda) - \mathcal{L}(z^*, \lambda) \right] \end{aligned} \quad (26)$$

where $\beta = \frac{2\rho}{1+\rho} \in [0, 1]$.

Proof: Invoking (21) with $z^1 = z^i$, $z^2 = z^*$, we have

$$\mathcal{L}(z^i, \lambda) \geq \mathcal{L}(z^*, \lambda) + \frac{\rho}{2} d_i^2$$

namely

$$\mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) + \frac{\rho}{2} d_i^2 \geq \rho d_i^2$$

Invoking (21) again with $z^1 = z^*$, $z^2 = z^i$, we have

$$\mathcal{L}(z^*, \lambda) \geq \mathcal{L}(z^i, \lambda) + \nabla \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z^* - z^i) + \frac{\rho}{2} d_i^2$$

namely

$$\nabla \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z^i - z^*) \geq \mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) + \frac{\rho}{2} d_i^2 \geq \rho d_i^2$$

or

$$\begin{aligned} -\nabla \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z^i - z^*) &\leq -\mathcal{L}(z^i, \lambda) + \mathcal{L}(z^*, \lambda) - \frac{\rho}{2} d_i^2 \\ &\leq -\rho d_i^2 \end{aligned}$$

Utilising Lemma 1, we can obtain

$$\begin{aligned} &\frac{1}{2} d_{i+1}^2 + \mathcal{L}(z^{i+1}, \lambda) - \mathcal{L}(z^*, \lambda) \\ &\quad \leq \frac{1}{2} d_i^2 + \mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) \\ &\quad\quad - \frac{1}{N} \nabla \mathcal{L}^{\mathrm{T}}(z^i, \lambda)(z^i - z^*) \end{aligned}$$

Defining $\beta = [2\rho/(1 + \rho)] \in [0, 1]$, we have

$$\begin{aligned} &\frac{1}{2} d_{i+1}^2 + \mathcal{L}(z^{i+1}, \lambda) - \mathcal{L}(z^*, \lambda) \\ &\leq \frac{1}{2} d_i^2 + \mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) \\ &\quad - \frac{1}{N} \left\{ \beta[\mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) + \frac{\rho}{2} d_i^2] + (1 - \beta)\rho d_i^2 \right\} \\ &= \left(1 - \frac{\beta}{N}\right) \left[\frac{1}{2} d_i^2 + \mathcal{L}(z^i, \lambda) - \mathcal{L}(z^*, \lambda) \right] \end{aligned}$$

Applying the above inequality iteratively, we can obtain the result consequently. □

3.3 Fast gradient method for dual problem

Now, we need to calculate the gradient of the dual function

$$g(\lambda) = \min_{z \in \Omega} \left\{ \mathcal{L}(z, \lambda) = f(z) + \lambda^{\mathrm{T}} h(z) \right\}$$

where $f(z) = \sum_{k=1}^N z_k^{\mathrm{T}} H z_k$, $h(z) = Sz - b$.

Denote $z(\lambda) = \arg\min_{z \in \Omega} \mathcal{L}(z, \lambda)$, then we have

$$\begin{aligned} \nabla g(\lambda) &= \nabla z^{\mathrm{T}}(\lambda) \nabla f(z(\lambda)) + h(z(\lambda)) \\ &\quad + \nabla z^{\mathrm{T}}(\lambda) \nabla h^{\mathrm{T}}(z(\lambda))\lambda \\ &= \nabla z^{\mathrm{T}}(\lambda) \left[\nabla f(z(\lambda)) + \nabla h^{\mathrm{T}}(z(\lambda))\lambda \right] + h(z(\lambda)) \\ &= h(z(\lambda)) \end{aligned}$$

owing to the following optimality conditions for $z(\lambda)$:

$$\nabla f(z(\lambda)) + \nabla h^{\mathrm{T}}(z(\lambda)) = 0$$

In the following, a FDAM is proposed, which resembles the method in [22], in which, it proposed a fast gradient method which can be used to accelerate the convergence of any method based on traditional gradient approach. However, the fast gradient method in [22] is used to accelerate the convergence of computing primal problem; while the proposed FDAM in this study is used to accelerate the convergence of computing dual problem.

© The Institution of Engineering and Technology 2015

Fast dual ascend method (FDAM)
Step 0. Take some $\eta > 1$, λ^1, and set $\bar{\lambda}^1 = \lambda^1$, $\alpha_1 > 0$, $t_1 = 1$
Step i. ($i \geq 1$),
repeat

$$z^i = \arg\min_{z \in \Omega} \{\mathcal{L}(z, \bar{\lambda}^i)\} \text{ (PCDM)} \tag{27}$$

while(1)

$$\lambda^{i+1} = \bar{\lambda}^i + \alpha_i (Sz^i - b) \tag{28}$$

$$\mathbf{z}^{i+1} = \arg\min_{z \in \Omega} \mathcal{L}(z, \lambda^{i+1}) \text{ (PCDM)} \tag{29}$$

if

$$(\lambda^{i+1} - \bar{\lambda}^i)^{\mathrm{T}} S(z^i - z^{i+1}) \leq \frac{1}{2\alpha_i} \|\lambda^{i+1} - \bar{\lambda}^i\|_2^2$$

break
end if

$$a_i = \eta a_i \tag{30}$$

end while

$$t_{i+1} = \frac{1 + \sqrt{1 + 4t_i^2}}{2} \tag{31}$$

$$\bar{\lambda}^{i+1} = \lambda^{i+1} + \frac{t_i - 1}{t_{i+1}} (\lambda^{i+1} - \lambda^i) \tag{32}$$

until $\|\lambda^{i+1} - \lambda^i\| \leq \epsilon$

Remark 3: In (27) and (29) of FDAM, the PCDM is used to solve the primal problem. The condition $(\lambda^{i+1} - \bar{\lambda}^i)^{\mathrm{T}} S(z^i - z^{i+1}) \leq \frac{1}{2\alpha_i} \|\lambda^{i+1} - \bar{\lambda}^i\|_2^2$ is used to make sure that the dual function is ascending.

Lemma 2: Let sequence $\{z^i, \lambda^i\}$ be generated by FDAM, and if

$$(\lambda^{i+1} - \bar{\lambda}^i)^{\mathrm{T}} S(z^i - z^{i+1}) \leq \frac{1}{2\alpha_i} \|\lambda^{i+1} - \bar{\lambda}^i\|_2^2 \tag{33}$$

then for any dual feasible pair (z, λ), we have

$$\begin{aligned} \mathcal{L}(z^{i+1}, \lambda^{i+1}) - L(z, \lambda) \geq & \frac{1}{2\alpha_i} \|\bar{\lambda}^i - \lambda^{i+1}\|_2^2 \\ & + \frac{1}{\alpha_i} (\lambda - \bar{\lambda}^i)^{\mathrm{T}} (\bar{\lambda}^i - \lambda^{i+1}) \end{aligned} \tag{34}$$

Proof: On the one hand

$$\begin{aligned} \mathcal{L}(z^i, \bar{\lambda}^i) - L(z, \lambda) &= f(z^i) + (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad - f(z) - \lambda^{\mathrm{T}} (Sz - b) \\ &\geq (z^i - z)^{\mathrm{T}} \nabla f(z) + (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad - \lambda^{\mathrm{T}} (Sz - b) \\ &\geq -(z^i - z)^{\mathrm{T}} (S^{\mathrm{T}} \lambda) + (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad - \lambda^{\mathrm{T}} (Sz - b) \\ &= -(\lambda - \bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &= \frac{1}{\alpha_i} (\lambda - \bar{\lambda}^i)^{\mathrm{T}} (\bar{\lambda}^i - \lambda^{i+1}) \end{aligned}$$

On the other hand

$$\begin{aligned} &\mathcal{L}(z^{i+1}, \lambda^{i+1}) - \mathcal{L}(z^i, \bar{\lambda}^i) \\ &\quad = f(z^{i+1}) + (\lambda^{i+1})^{\mathrm{T}} (Sz^{i+1} - b) \\ &\qquad - f(z^i) - (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad \geq (z^{i+1} - z^i)^{\mathrm{T}} \nabla f(z^i) + (\lambda^{i+1})^{\mathrm{T}} (Sz^{i+1} - b) \\ &\qquad - (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad \geq -(z^{i+1} - z^i)^{\mathrm{T}} (S^{\mathrm{T}} \bar{\lambda}^i) + (\lambda^{i+1})^{\mathrm{T}} (Sz^{i+1} - b) \\ &\qquad - (\bar{\lambda}^i)^{\mathrm{T}} (Sz^i - b) \\ &\quad = -(\bar{\lambda}^i - \lambda^{i+1})^{\mathrm{T}} (Sz^{i+1} - b) \\ &\quad = -(\bar{\lambda}^i - \lambda^{i+1})^{\mathrm{T}} [(Sz^i - b) - S(z^i - z^{i+1})] \\ &\quad = \frac{1}{\alpha_i} \|\lambda^{i+1} - \bar{\lambda}^i\|_2^2 + (\bar{\lambda}^i - \lambda^{i+1})^{\mathrm{T}} S(z^i - z^{i+1}) \\ &\quad \geq \frac{1}{2\alpha_i} \|\lambda^{i+1} - \bar{\lambda}^i\|_2^2 \end{aligned}$$

Combing the above two inequalities, we can get the result consequently. □

Lemma 3: The sequence $\{z^i, \lambda^i\}$ generated by FDAM satisfies

$$2\alpha_i t_i^2 v_i - 2\alpha_{i+1} t_{i+1}^2 v_{i+1} \geq \|u_{i+1}\|_2^2 - \|u_i\|_2^2 \tag{35}$$

where $v_i = L(z^*, \lambda^*) - L(z^{i+1}, \lambda^{i+1})$, $u_i = t_i \lambda^{i+1} - (t_i - 1)\lambda^i - \lambda^*$.

Proof: Invoking Lemma 2 with $i = i + 1$, and $\lambda = \lambda^{i+1}$, we obtain

$$\begin{aligned} 2\alpha_{i+1}(v_i - v_{i+1}) \geq & \|\bar{\lambda}^{i+1} - \lambda^{i+2}\|_2^2 \\ & + 2(\lambda^{i+1} - \bar{\lambda}^{i+1})^{\mathrm{T}} (\bar{\lambda}^{i+1} - \lambda^{i+2}) \end{aligned}$$

Invoking Lemma 2 again with $i = i + 1$, and $\lambda = \lambda^*$, we obtain

$$\begin{aligned} -2\alpha_{i+1} v_{i+1} \geq & \|\bar{\lambda}^{i+1} - \lambda^{i+2}\|_2^2 \\ & + 2(\lambda^* - \bar{\lambda}^{i+1})^{\mathrm{T}} (\bar{\lambda}^{i+1} - \lambda^{i+2}) \end{aligned}$$

Multiplying the first inequality by t_i^2 and then adding it to the second inequality by $(t_{i+1}^2 - t_i^2)$, and using $t_{i+1}^2 - t_i^2 = t_{i+1}$, we have

$$\begin{aligned} 2\alpha_{i+1}(t_i^2 v_i - t_{i+1}^2 v_{i+1}) \geq & \|t_{i+1}(\bar{\lambda}^{i+1} - \lambda^{i+2})\|_2^2 \\ & + 2t_{i+1}(\bar{\lambda}^{i+1} - \lambda^{i+2}) \\ & [(t_{i+1} - 1)\lambda^{i+1} - t_{i+1}\bar{\lambda}^{i+1} + \lambda^*] \end{aligned}$$

Applying the relation

$$\|a - b\|_2^2 + 2(a - b)^{\mathrm{T}}(b - c) = \|a - c\|_2^2 - \|b - c\|_2^2$$

with $a = t_{i+1}\lambda^{i+2}$, $b = t_{i+1}\bar{\lambda}^{i+1}$, $c = (t_{i+1} - 1)\lambda^{i+1} + \lambda^*$ and by the fact that $\{a_i\}$ is non-increasing, we have

$$\begin{aligned} & 2\alpha_i t_i^2 v_i - 2\alpha_{i+1} t_{i+1}^2 v_{i+1} \\ & \quad \geq \|t_{i+1}\lambda^{i+2} - (t_{i+1} - 1)\lambda^{i+1} - \lambda^*\|_2^2 \\ & \quad - \|t_{i+1}\bar{\lambda}^{i+1} - (t_{i+1} - 1)\lambda^{i+1} - \lambda^*\|_2^2 \end{aligned}$$

Since $\bar{\lambda}^{i+1} = \lambda^{i+1} + [(t_i - 1)/(t_{i+1})](\lambda^{i+1} - \lambda^i)$, we have

$$t_{i+1}\bar{\lambda}^{i+1} - (t_{i+1} - 1)\lambda^{i+1} - \lambda^* = t_i \lambda^{i+1} - (t_i - 1)\lambda^i - \lambda^*$$

that is to say

$$2\alpha_i t_i^2 v_i - 2\alpha_{i+1} t_{i+1}^2 v_{i+1} \geq \|u_{i+1}\|_2^2 - \|u_i\|_2^2 \qquad \square$$

© The Institution of Engineering and Technology 2015

Lemma 4 [22]: Let $\{a_i, b_i\}$ be positive sequences of reals satisfying

$$a_i - a_{i+1} \geq b_{i+1} - b_i, \quad \forall\, i \geq 1$$

then, $a_i \leq a_1 + b_1$, $\forall\, i \geq 1$.

Lemma 5 [22]: The positive sequence $\{t_i\}$ generated by

$$t_{i+1} = \frac{1 + \sqrt{1 + 4t_i^2}}{2}$$

with $t_1 = 1$ satisfies $t_i \geq (i+1)/2$, $\forall\, i \geq 1$.

Theorem 2: Let sequence $\{z^i, \lambda^i\}$ be generated by FDAM, then for any $i \geq 1$, we have

$$\mathcal{L}(z^*, \lambda^*) - \mathcal{L}(z^{i+1}, \lambda^{i+1}) \leq \frac{2\|\lambda^1 - \lambda^*\|_2^2}{\alpha_i (i+1)^2}$$

Proof: Invoking Lemma 4 with $a_i = 2\alpha_i t_i^2 v_i$, $b_i = \|u_i\|_2^2$ along with Lemma 3, we have

$$\begin{aligned} 2\alpha_i t_i^2 v_i &\leq a_1 + b_1 \\ &= 2\alpha_1[\mathcal{L}(z^*, \lambda^*) - \mathcal{L}(z^2, \lambda^2)] + \|\lambda^2 - \lambda^*\|_2^2 \end{aligned}$$

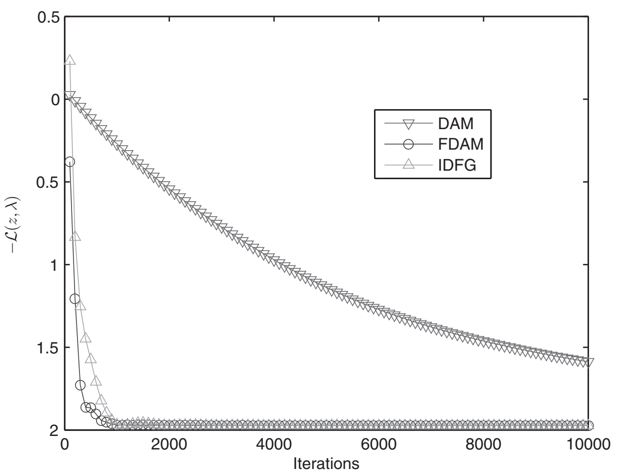

Fig. 2 *Iterative curves of the $-\mathcal{L}(z, \lambda)$ using DAM, FDAM and IDFG*

$$\begin{aligned} &\overset{(34)}{\leq} 2(\bar{\lambda}^1 - \lambda^*)^{\mathrm{T}}(\bar{\lambda}^1 - \lambda^2) \\ &\quad - \|\bar{\lambda}^1 - \lambda^2\|_2^2 + \|\lambda^2 - \lambda^*\|_2^2 \\ &= \|\bar{\lambda}^1 - \lambda^*\|_2^2 - \|\lambda^2 - \lambda^*\|_2^2 + \|\lambda^2 - \lambda^*\|_2^2 \\ &= \|\lambda^1 - \lambda^*\|_2^2 \end{aligned}$$

namely

$$\mathcal{L}(z^*, \lambda^*) - \mathcal{L}(z^{i+1}, \lambda^{i+1}) \leq \frac{\|\lambda^1 - \lambda^*\|_2^2}{2\alpha_i t_i^2} = \frac{2\|\lambda^1 - \lambda^*\|_2^2}{\alpha_i (i+1)^2}$$

□

Remark 4: From Theorem 2. it can be seen that the convergence rate for the proposed FDAM is $\mathcal{O}(1/i^2)$ (i is the iteration counter). Compared with traditional dual ascend method (DAM) (whose convergence rate is $\mathcal{O}(1/i)$) without accelerated strategy, the proposed FDAM has better convergence properties. Anyway, we have to admit that at each iteration, the time consumption in FDAM is a little more than that in DAM because of the additional computation. However, this factor plays much less important role in computing speed than the time consumed in longer iterations.

4 Numerical results

In the experiment, it is assumed that the initial state $x(0) = [0.5, 0.5, 0.5, 0.5]$ and both Q and R are fixed as identity matrices, then at time $k = 1, 2, \ldots,$ the proposed FDAM is used to solve a sequence of constrained optimisation problems. To make sure that (33) is always satisfied, we set α constant at $[\{\sigma_{\min}(H)\}/\{2\sigma_{\max}(S)\}]$.

By simple calculation, it can be found that lower and upper bound for state and control are as follows

$$\begin{aligned} \underline{u} &= \frac{1}{3600}[-1.63; -2], \quad \overline{u} = \frac{1}{3600}[1.63; 2] \\ \underline{x} &= [-0.45; -0.46; -0.45; -0.46] \\ \overline{x} &= [0.71; 0.7; 0.65; 0.64] \end{aligned}$$

For $k = 1$, we will first show that the proposed FDAM has faster convergence rate than the corresponding DAM without accelerated strategy. Fig. 2 illustrates the iterative curves of the $-\mathcal{L}(z, \lambda)$ using DAM, FDAM and IDFG (inexact dual fast gradient method) in [15] (the feasibility tolerance $\epsilon_{\text{in}} = 1 \times 10^{-5}$), respectively, and it is shown that much fewer number of iterations are needed to obtained

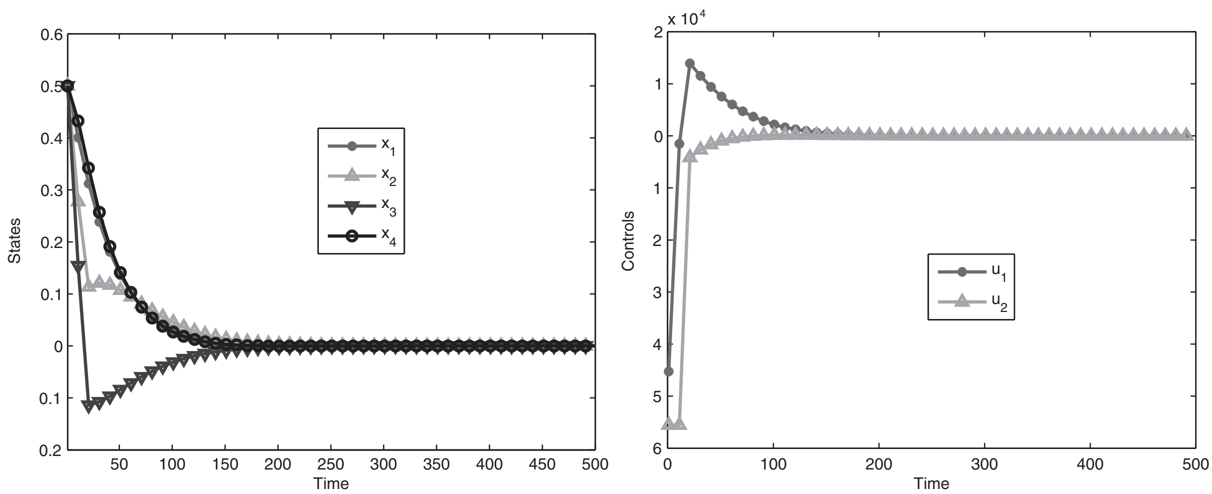

Fig. 3 *Evolution of states and control inputs with the length of prediction horizon N = 10 using FDAM*

© The Institution of Engineering and Technology 2015

the global solution when using FDAM, while the performance of the proposed FDAM is similar to IDFG but with much simper implementation since the updating of current dual variables needs to use accumulated historical information in IDFG [15].

The proposed FDAM is applied to solve the subproblems of MPC scheme sequentially, and only the first control input is used to generate the next state. Repeatedly, the generated control inputs and the states are illustrated in Fig. 3, which shows that the benchmark plant can be well stabilised using the proposed approach.

5 Conclusion

We have studied the HD-MPC four-tank benchmark using an MPC scheme, in which, a fast distributed optimisation approach is proposed to solve the constrained optimisation subproblems. By introducing Lagrangian dual, the primal problem is solved by a parallel coordinate descent method, and a FDAM is adopted to solve the dual problem iteratively. The convergence rate of the proposed FDAM is given and its superiority to classical gradient-based method is illustrated. The experimental results have shown that the proposed approach can stabilise the benchmark plant.

6 Acknowledgments

This publication was made possible by NPRP Grant # NPRP 4-1162-1-181 from the Qatar National Research Fund (a member of Qatar Foundation), and it was also supported by the National Science Found for Distinguished Young Scholars of China (Grant No. 61025015) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61321003). The statements made herein are solely the responsibility of the authors. The authors thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this paper.

7 References

1 Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: 'Constrained model predictive control: stability and optimality', *Automatica*, 2000, **36**, (6), pp. 789–814
2 Venkat, A.N., Hiskens, I.A., Rawlings, J.B., Wright, S.J.: 'Distributed MPC strategies with application to power system automatic generation control', *IEEE Trans. Control Syst. Technol.*, 2008, **16**, (6), pp. 1192–1206
3 Keviczky, T., Borrelli, F., Balas, G.J.: 'Decentralized receding horizon control for large scale dynamically decoupled systems', *Automatica*, 2006, **42**, (12), pp. 2105–2115
4 Baskar, L.D., De, S.B., Hellendoorn, H.: 'Traffic management for automated highway systems using model-based predictive control', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (2), pp. 838–847
5 Li, H.P., Shi, Y.: 'Distributed model predictive control of constrained nonlinear systems with communication delays', *Syst. Control Lett.*, 2013, **62**, (10), pp. 819–826
6 Song, Y., Fang, X.S.: 'Distributed model predictive control for polytopic uncertain systems with randomly occurring actuator saturation and packet loss', *IET Control Theory Appl.*, 2014, **8**, (5), pp. 297–310
7 Christofides, P.D., Scattolini, R., Muñoz de la Peña, D., Liu, J.F.: 'Distributed model predictive control: a tutorial review and future research directions', *Comput. Chem. Eng.*, 2013, **51**, pp. 21–41
8 Negenborn, R., Maestre, J.: 'Distributed model predictive control: an overview and roadmap of future research opportunities', *IEEE Control Syst. Mag.*, 2014, **34**, (4), pp. 87–97
9 Camponogara, E., Scherer, H.F.: 'Distributed optimization for model predictive control of linear dynamic networks with control-input and output constraints', *IEEE Trans. Autom. Sci. Eng.*, 2001, **8**, (1), pp. 233–242
10 Dunbar, W.B.: 'Distributed receding horizon control of dynamically coupled nonlinear systems', *IEEE Trans. Autom. Control*, 2007, **52**, (7), pp. 1249–1263
11 Necoara, I., Clipici, D.: 'Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC', *J. Process Control*, 2013, **23**, (3), pp. 243–253
12 Patrinos, P., Sopasakis, P., Sarimveis, H.: 'A global piecewise smooth Newton method for fast large-scale model predictive control', *Automatica*, 2011, **47**, (9), pp. 2016–2022
13 Giselsson, P., Doan, M.D., Keviczky, T., Schutter, B.D., Rantzer, A.: 'Accelerated gradient methods and dual decomposition in distributed model predictive control', *Automatica*, 2013, **49**, (3), pp. 829–833
14 Patrinos, P., Bemporad, A.: 'An accelerated dual gradient-projection algorithm for embedded linear model predictive control', *IEEE Trans. Autom. Control*, 2014, **59**, (1), pp. 18–33
15 Necoara, I and Nedelcu, V.: 'Rate analysis of inexact dual first-order methods application to dual decomposition', *IEEE Trans. Autom. Control*, 2014, **59**, (5), pp. 1232–1243
16 Nesterov, Y.: 'A method of solving a convex programming problem with convergence rate $O(1/k^2)$', *Sov. Math. Dokl.*, 1983, **27**, (2), pp. 372–376
17 Nesterov, Y.: 'Smooth minimization of non-smooth functions', *Math. Program.*, 2005, **103**, (1), pp. 127–152
18 Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Found. Trends® Mach. Learn.*, 2011, **3**, (1), pp. 1–122
19 Johansson, K.H.: 'The quadruple-tank process: A multivariable laboratory process with an adjustable zero', *IEEE Trans. Control Syst. Technol.*, 2000, **8**, (3), pp. 456–465
20 Alvarado, I., Limon, D., Muñoz de la Peña, D. *et al.*: 'A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark', *J. Process Control*, 2011, **21**, (5), pp. 800–815
21 Nesterov, Y.: 'Efficiency of coordinate descent methods on huge-scale optimization problems', *SIAM J. Optim.*, 2012, **22**, (2), pp. 341–362
22 Beck, A., Teboulle, M.: 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', *SIAM J. Imaging Sci.*, 2009, **2**, (1), pp. 183–202

© The Institution of Engineering and Technology 2015