

## 采用双速率框架的快速预测控制算法

王曦, 李德伟<sup>†</sup>, 席裕庚

(上海交通大学自动化系, 上海 200240; 系统控制与信息处理教育部重点实验室, 上海 200240)

**摘要:** 约束模型预测控制(model predictive control, MPC)在实际应用中优化计算复杂度高, 无法在采样周期内完成优化以保证系统实时性. 本文针对这一问题, 提出采用双速率框架的快速预测控制算法(DSF-MPC). 该算法将实时控制量的求解分解到两个时间尺度上进行, 即双速率框架: 每隔数个采样周期, 慢速率层负责完成一次对完整MPC优化问题的求解; 而在每个采样周期, 快速率层负责根据系统反馈信息和慢速率层算法预测信息的差值, 朝着使目标函数值下降的负梯度方向, 修正慢速率层的优化结果来获取实际控制量, 以满足控制的实时性要求. 该算法不要求在每个采样周期内都完成MPC中的在线优化, 故能在继承MPC优点的同时, 满足快速系统的控制实时性要求. 针对直流电动机和倒立摆组合系统的仿真结果, 验证了该算法的有效性, 反映了其在快速系统中的应用潜力.

**关键词:** 模型预测控制; 双速率框架; 实时修正; 快速算法

中图分类号: TP273 文献标识码: A

## Double-speed-frame-based fast predictive control algorithm

WANG Xi, LI De-wei<sup>†</sup>, XI Yu-geng

(Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China;

Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, China)

**Abstract:** The constrained model predictive control (MPC) is often with high computational complexity, there may not be enough time to solve the optimization problem in a sampling period. To solve this problem, we propose a double-speed-frame-based fast predictive control algorithm (DSF-MPC), in which the solving process of real-time control is decomposed into two parts with high and low speeds. In every finite number of sampling periods, the low-speed part solves once the MPC optimization problem to obtain the control value. In each sampling period, the fast part determines the real control value by modifying the control value obtained from the low-speed part in the negative gradient direction of the objective function, based on the difference between the feed-back information from the system and the predictive information from the low speed part. Since this algorithm doesn't solve the optimization problem in each sampling period, it not only meets the real-time requirement of fast-speed system, but also inherits advantages of MPC. Simulation results of the system model of DC motor with inverted pendulum show the effectiveness of this proposed algorithm and reflect its application potentiality for fast systems.

**Key words:** model predictive control; double speed frame; real-time correct; fast algorithm

### 1 引言(Introduction)

模型预测控制(model predictive control, MPC)因具备优化系统性能, 较方便地处理约束等能力, 近几十年来在工业领域得到广泛应用<sup>[1]</sup>. MPC控制器在每个采样时刻都要在线求解有限时域的优化问题, 并将第一个优化控制量施加给被控对象, 以实现滚动优化. 相比于其他控制方法, 滚动优化使MPC的控制效果有一定优势, 但同时也引入了较大计算负担<sup>[2]</sup>. 而在实际系统中, 为改进控制性能, 往往会增加控制时域长

度, 即优化的控制量数目, 这使控制器的在线计算量急剧增加<sup>[3]</sup>. 因此, 国内外学者为降低MPC的在线计算负担进行了大量研究.

Pannocchia等人提出了结合存储表和在线优化的局部查记法(partial enumeration)<sup>[4]</sup>, 该方法加快了大规模MPC问题的求解, 但在高频采样系统中的应用有局限. 对于具有输入输出约束的线性系统, MPC滚动优化需求解二次规划(quadratic programming, QP)问题, Wang等人受其特殊结构启发, 提出了对优化变

收稿日期: 2015-04-24; 录用日期: 2015-07-07.

<sup>†</sup>通信作者. E-mail: dwli@sjtu.edu.cn; Tel.: +86 13917551024.

本文责任编辑: 陈增强.

国家自然科学基金项目(61221003, 61374110, 61333009), 国家高新技术研究发展计划("863"计划)项目(2015AA043102)资助.

Supported by National Natural Science Foundation of China (61221003, 61374110, 61333009) and National High-tech R&D Program ("863" Program) (2015AA043102).

量进行再排序来实现快速在线优化的MPC算法<sup>[5]</sup>,该方法在处理稀疏问题及小规模问题时效果明显. Cagienard等人的移动窗口块策略(moving window blocking)<sup>[6-7]</sup>及Lu等人的基于集结策略(aggregation strategy)的快速算法<sup>[8-9]</sup>将几个采样周期内的控制量设定为相同值以减少优化变量的自由度,从而减少计算量,前者主要针对大规模系统,而后者可能导致闭环系统不稳定. Kouramas等人将动态规划和显示预测控制方法相结合,把预测控制的优化问题分解为小规模问题<sup>[10]</sup>,但其前提是原有预测控制器设计方法可进行分解. Ferreau和Giselsson等人提出利用多核CPU并行实现快速QP求解<sup>[11-12]</sup>,该方法对大规模稀疏问题提速明显,但不适用于所有快速系统. 尽管上述方法一定程度上解决了MPC的实时性问题,但它们要求在一个采样周期内完成一次在线优化,而在线计算量和MPC设计有很大关系,因此,这限制了MPC在实际应用中的灵活性.

本文针对MPC在线优化耗时大,在某些快速系统中会出现单个采样周期内无法完成一次QP求解,但使用上一次QP求解结果进行控制又会导致控制不准确的问题,提出了采用双速率框架的快速预测控制算法(DFS-MPC). 该算法通过双速率策略将实时控制量的求取分解到两个时间尺度上进行,慢速率层MPC控制器以较低频率进行滚动优化,在完成QP求解后将优化结果送给快速率层,同时获取状态实测值滚动进行新的QP求解;快速率层则以慢速率层算出的控制量为基准,根据系统的实测状态去实时修正控制量,并将修正后的当前控制量输送给被控对象. 文中给出了该算法控制直流电动机<sup>[13-14]</sup>和倒立摆<sup>[15-16]</sup>的MATLAB仿真结果,结果表明,对于时间常数小的快速系统,本文算法修正控制量的耗时比完成一次QP求解降低了1至2个数量级,从而可解决因QP求解耗时而导致MPC算法难以应用于快速系统的问题.

本文结构如下:第2节介绍了MPC在线优化的求解方法;第3节描述了双速率框架的核心思想;第4节说明了在一般输入/输出约束及盒子约束下的实时修正模块设计;第5节给出了将该算法用于控制电枢直流电机和倒立摆的MATLAB仿真结果并进行了讨论;第6节对全文进行总结并提出若干可能的研究方向.

## 2 系统描述(System description)

考虑离散时间线性时不变系统:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \end{cases} \quad (1)$$

其中:  $\mathbf{u}(k) \in \mathbb{R}^{n_i}$  为控制输入,  $\mathbf{x}(k) \in \mathbb{R}^n$  为系统可测状态,  $\mathbf{y}(k) \in \mathbb{R}^{n_o}$  是系统输出.

在任意时刻  $k$ , 考虑对系统施加  $M$  步控制后预测

系统未来  $P$  步输出, 为简化计, 令控制时域与优化时域一致, 即  $P = M$ , 且记为  $N$ , 系统未来输出预测如下<sup>[17]</sup>:

$$\mathbf{Y}_k = \mathbf{F}_y \mathbf{x}(k) + \mathbf{G}_y \mathbf{U}_k, \quad (2)$$

其中:

$$\begin{aligned} \mathbf{Y}_k &= (\mathbf{y}^T(k+1|k), \dots, \mathbf{y}^T(k+N|k))^T, \\ \mathbf{U}_k &= (\mathbf{u}^T(k|k), \dots, \mathbf{u}^T(k+N-1|k))^T, \\ \mathbf{F}_y &= \begin{bmatrix} \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^N \end{bmatrix}, \\ \mathbf{G}_y &= \begin{bmatrix} \mathbf{C}\mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \dots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{N-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{C}\mathbf{B} \end{bmatrix}. \end{aligned}$$

式(2)中  $\mathbf{u}(k|k), \dots, \mathbf{u}(k+N-1|k)$  是当前和未来时刻施加于系统的控制量,  $\mathbf{y}(k+i|k)$ ,  $i = 1, \dots, N$  表示  $k$  时刻对  $k+i$  时刻的输出预测值. 在  $k$  时刻, MPC需在线求解以下优化问题:

$$\begin{aligned} \min_u J(k) &= \frac{1}{2} \sum_{i=1}^N \|\mathbf{y}(k+i|k) - \mathbf{w}(k+i)\|_{\mathbf{Q}_0}^2 + \\ &\quad \frac{1}{2} \sum_{i=1}^N \|\mathbf{u}(k+i-1|k)\|_{\mathbf{R}_0}^2 + \\ &\quad \frac{1}{2} \|\mathbf{y}(k+N|k) - \mathbf{w}(k+N)\|_{\Psi}^2 = \\ &\quad \frac{1}{2} \|\mathbf{Y}_k - \mathbf{W}_k\|_{\mathbf{Q}}^2 + \frac{1}{2} \|\mathbf{U}_k\|_{\mathbf{R}}^2, \end{aligned} \quad (3)$$

$$\text{s.t. } |\mathbf{y}(k+i|k)_j| \leq \mathbf{y}_{j,\max}, \quad j = 1, \dots, n_o,$$

$$|u(k+i|k)_l| \leq u_{l,\max}, \quad l = 1, \dots, n_i,$$

$$\mathbf{x}(k+i|k) = \mathbf{A}\mathbf{x}(k+i-1|k) + \mathbf{B}\mathbf{u}(k+i-1|k),$$

$$\mathbf{y}(k+i|k) = \mathbf{C}\mathbf{x}(k+i|k), \quad i = 1, \dots, N,$$

$$\mathbf{x}(k+N|k) \in \mathbf{X}_f = \{\mathbf{x} | \mathbf{E}\mathbf{x} \leq \mathbf{f}\},$$

其中:  $\mathbf{w}(k+i)$  是  $k+i$  时刻的期望输出,  $\mathbf{W}_k = [\mathbf{w}^T(k+1), \dots, \mathbf{w}^T(k+N)]^T$ ,  $\mathbf{y}_{j,\max}$  为第  $j$  个输出约束,  $u_{l,\max}$  为第  $l$  个输入约束,  $\mathbf{Q} = \text{diag}\{\mathbf{Q}_0, \dots, \mathbf{Q}_0, \Psi + \mathbf{Q}_0\}$  为误差权矩阵,  $\mathbf{R} = \text{diag}\{\mathbf{R}_0, \dots, \mathbf{R}_0\}$  为控制权矩阵,  $\mathbf{X}_f$  为预先定义的多面体终端不变集. 在预测状态进入终端集后, 存在反馈控制率  $\mathbf{u} = \mathbf{H}\mathbf{x}$  保证状态继续留在不变集内, 并满足输入输出约束.

传统MPC把在线求解问题(3)得到的第1个控制量(当前控制量)  $\mathbf{u}(k|k)$  作用于对象, 到下一时刻通过反馈校正, 重新定义  $k+1$  时刻的QP问题, 如此滚动优化, 实施MPC控制.

### 3 双速率框架(Double speed framework)

对于时间常数小、采样频率高的快速系统(采样频率高于MPC实施频率), MPC的在线优化问题可能需要多个采样周期 $mT$ ( $m$ 为正整数)才能完成一次求解.

如图1所示, 由于计算耗时 $mT$ , 所以 $k$ 时刻开始的滚动优化到 $k+m$ 时刻才能完成. 此时,  $k$ 时刻开始计算得到的 $k$ 到 $k+m$ 时刻的 $m$ 个控制量已成为过去而无实施意义, 系统实际实施的控制量为 $k-m$ 时刻开始计算 $k$ 时刻求出的相关控制序列.

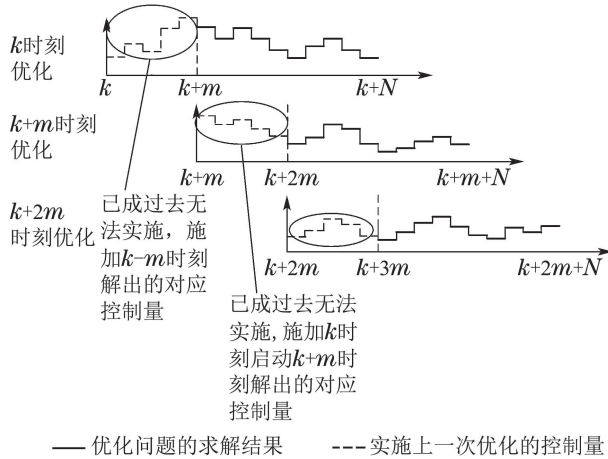


图 1 DSF-MPC算法慢时标层控制时序图  
Fig. 1 Control timing diagram of DSF-MPC algorithm in slow time scale

按照传统MPC, 在 $k$ 时刻, 根据实测状态 $\bar{x}(k)$ 和已给定的 $k, \dots, k+m-1$ 时刻的控制序列 $\bar{u}(k|k-m), \dots, \bar{u}(k+m-1|k-m)$ 预测 $k+m$ 时刻的状态 $\bar{x}(k+m|k)$ ; 然后基于 $\bar{x}(k+m|k)$ 优化从 $k+m$ 到 $k+N$ 时刻的优化问题(3), 得到 $u(k+m|k), \dots, u(k+N-1|k)$ . 在 $k+m$ 时刻, 上述步骤重复进行.

然而, 按上述实施方法, 在 $k, k+m, \dots$ 时刻, 系统实际是开环的, 它只利用了滚动优化起始时刻的系统实时信息, 并没有顾及到起始时刻到实施时刻间系统可能受到的干扰及模型失配等因素造成的预测误差, 简单地实施开环优化的结果, 将导致控制品质下降.

为应对上述问题, 受分层控制思想启发<sup>[18-19]</sup>, 作者考虑增加一个高密度、快速率的反馈来改变滚动优化层(慢速率层)已求得的控制量, 即增加一个快速率层来实时修正控制量.

快速率层以慢速率层开环优化得到的控制量为基准, 以采样周期 $T$ 根据实时检测信息修正它们并付诸实施, 图2即为 $k+m$ 时刻到 $k+2m-1$ 时刻的修正过程. 这种控制结构把MPC的在线优化计算根据复杂度的不同置于两个不同的时间尺度, 根据不同的反馈信息来进行. 慢速率层以 $mT$ 为周期根据系统的实际状态进行滚动优化, 在解出的控制序列中, 后 $N-m$ 个

为有效控制量, 送入快速率层, 快速率层则以 $T$ 为周期检测系统实际状态, 并基于该状态对慢速率层计算出的有效控制量快速修正, 然后实施到控制对象.

$k$ 时刻, 慢速率层只需求解 $k+m$ 到 $k+N$ 时刻的QP问题, 因为可以基于 $\bar{x}(k)$ 和 $k$ 时刻求得的 $k$ 到 $k+m$ 的控制量预估 $\bar{x}(k+m|k)$ . 快速率层继承了MPC中滚动优化和反馈校正的思想, 以周期 $T$ 修正慢速率层下发的控制序列, 但与传统MPC不同, 为在时间 $T$ 内快速计算修正量, 快速率层的滚动优化不求解QP, 而采用固定方向搜索步长的方法, 基于系统的实测状态和约束条件寻找从当前步到第 $k+N$ 步的控制修正量, 见图2.

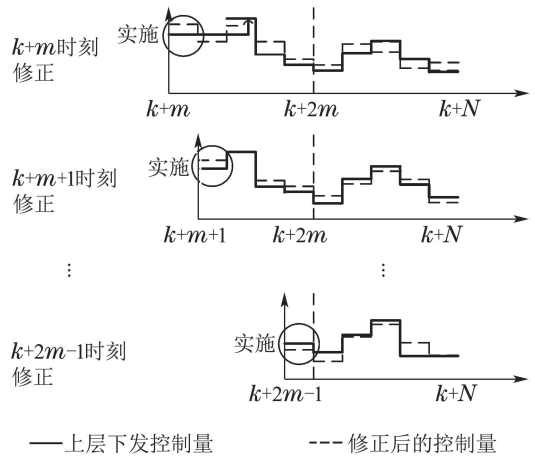


图 2 DSF-MPC算法的快时标层控制时序图  
Fig. 2 Controlling timing diagram of DSF-MPC algorithm in fast time scale

由于慢速率层的MPC求解的是完整的QP问题, 使用通用算法(当然也可以采用其他算法)求解, 不是本文的研究重点, 因此, 下面主要介绍快速率层实时修正环节的设计.

### 4 实时修正环节的设计(Design the real-time correction part)

#### 4.1 DSF-MPC算法(DSF-MPC algorithm)

根据QP求解器的计算能力及快速率层实时修正的采样周期 $T$ , 可确定慢速率层MPC完成一次滚动优化所需的步数 $m$ , 慢速率层每隔 $mT$ 时间将优化结果下发给快速率层, 完成控制量的实时修正. 实时修正的基本思路是: 在每一时刻以慢速率层下发的控制量为基准寻找修正量, 从系统实测状态出发, 选择目标函数下降方向及指向无约束最优解的方向作为修正控制量的可能方向, 分别搜索最优步长并比较, 快速确定控制量修正值并实施控制.

现考虑实时修正的具体实施, 在 $k+m+i$ 时刻(表示当前时刻与慢速率层MPC最近一次下发控制量时的步数间隔,  $i$ 为正整数且 $0 \leq i \leq m-1$ )系统检测

到的实时状态为 $\bar{\mathbf{x}}(k+m+i)$ , 其实 $k+m$ 时刻下发的信息已经包括了从 $k+m$ 到 $k+N$ 的控制量, 之所以还要实时修正, 是因为实测状态 $\bar{\mathbf{x}}(k+m+i)$ 可能不等于基于模型计算出的预测状态 $\mathbf{x}(k+m+i|k)$ . 若在该时刻根据 $\bar{\mathbf{x}}(k+m+i)$ 重新启动一个 $k+m+i$ 时刻到 $k+N$ 时刻的预测控制在线优化问题, 则表示为(为简化记, 将各约束表示为输出约束集 $\Omega_y$ 和输入约束集 $\Omega_u$ ):

$$\begin{aligned} \min_u J(k+m+i) = & \\ & \frac{1}{2} \|\bar{\mathbf{Y}}_{k+m+i} - \mathbf{W}_{k+m+i}\|_{\mathbf{Q}_{k+m+i}} + \\ & \frac{1}{2} \|\bar{\mathbf{U}}_{k+m+i}\|_{\mathbf{R}_{k+m+i}} = \\ & \frac{1}{2} \|\bar{\mathbf{U}}_{k+m+i}\|_{\mathbf{R}_{k+m+i}} + \\ & \frac{1}{2} \|\mathbf{F}_{k+m+i}\bar{\mathbf{x}}(k+m+i) + \\ & \mathbf{G}_{k+m+i}\bar{\mathbf{U}}_{k+m+i} - \mathbf{W}_{k+m+i}\|_{\mathbf{Q}_{k+m+i}}, \quad (4) \\ \text{s.t. } & \bar{\mathbf{Y}}_{k+m+i} \in \Omega_y, \bar{\mathbf{U}}_{k+m+i} \in \Omega_u. \end{aligned}$$

其中:  $\bar{\mathbf{U}}_{k+m+i}$ 为问题(4)的最优解,  $\bar{\mathbf{Y}}_{k+m+i} = \mathbf{F}_{k+m+i}\bar{\mathbf{x}}(k+m+i) + \mathbf{G}_{k+m+i}\bar{\mathbf{U}}_{k+m+i}$ 是基于 $\bar{\mathbf{x}}(k+m+i)$ 预测的 $k+m+i$ 到 $k+N$ 步的系统输出.

$$\begin{aligned} \bar{\mathbf{U}}_{k+m+i} &= (\bar{u}^T(k+m+i|k+m+i), \\ & \quad \dots, \bar{u}^T(k+N-1|k+m+i))^T, \\ \bar{\mathbf{Y}}_{k+m+i} &= (\bar{y}^T(k+m+i+1|k+m+i), \\ & \quad \dots, \bar{y}^T(k+N|k+m+i))^T, \\ \mathbf{F}_{k+m+i} &= [(\mathbf{CA})^T \quad \dots \quad (\mathbf{CA}^{N-m-i})^T]^T, \\ \mathbf{G}_{k+m+i} &= \\ & \begin{pmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \vdots \\ \vdots & \vdots & \dots & \mathbf{0} \\ \mathbf{CA}^{N-m-i-1}\mathbf{B} & \mathbf{CA}^{N-m-i-2}\mathbf{B} & \dots & \mathbf{CB} \end{pmatrix}. \end{aligned}$$

根据最优性原理<sup>[20-21]</sup>, 若模型准确且无扰动, 实测状态等于预测状态, 优化问题(4)得到的最优解就等于在 $k+m$ 时刻解优化问题(3)所得的相应最优控制量(即已下发的控制量). 因此, 若 $\bar{\mathbf{x}}(k+m+i) = \mathbf{x}(k+m+i|k)$ , 则可实施下传的 $u(k+m+i)$ , 不必修正. 但若两者不等, 则优化问题的解 $\bar{\mathbf{U}}_{k+m+i}$ 与 $\mathbf{U}_k$ 中相应的控制量(记为 $\mathbf{U}_{k+m+i}$ )存在误差, 并且记为 $\Delta\mathbf{U}_{k+m+i} = \bar{\mathbf{U}}_{k+m+i} - \mathbf{U}_{k+m+i}$ , 要求出 $\Delta\mathbf{U}_{k+m+i}$ 对控制进行校正.

为了进行实时修正, 首先计算该时刻优化问题(4)的无约束最优解:

$$\begin{aligned} \mathbf{U}_{k+m+i}^* &= \\ & (\mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{G}_{k+m+i} + \mathbf{R}_{k+m+i})^{-1}. \end{aligned}$$

$$\begin{aligned} & \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \cdot \\ & (\mathbf{W}_{k+m+i} - \mathbf{F}_{k+m+i}\bar{\mathbf{x}}(k+m+i)). \quad (5) \end{aligned}$$

若 $\mathbf{U}_{k+m+i}^*$ 满足全部约束, 则它就是 $\bar{\mathbf{U}}_{k+m+i}$ . 然而, 若 $\mathbf{U}_{k+m+i}^*$ 不在可行域内, 则需要考虑两个因素来确认修正量 $\Delta\mathbf{U}_{k+m+i}$ .

1)  $\Delta\mathbf{U}_{k+m+i}$ 的修正应向改善性能指标的方向进行. 为此, 计算 $J(k+m+i)$ 对 $\Delta\mathbf{U}_{k+m+i}$ 的梯度方向:

$$\begin{aligned} \frac{\partial J(k+m+i)}{\partial \Delta\mathbf{U}_{k+m+i}} &= \\ & \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{F}_{k+m+i} \bar{\mathbf{x}}(k+m+i) - \\ & \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{W}_{k+m+i} + \\ & (\mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{G}_{k+m+i} + \mathbf{R}_{k+m+i}) \cdot \\ & (\mathbf{U}_{k+m+i} + \Delta\mathbf{U}_{k+m+i}). \quad (6) \end{aligned}$$

$\Delta\mathbf{U}_{k+m+i}$ 可按下式迭代求解:

$$\begin{aligned} \Delta\mathbf{U}_{k+m+i}^l &= \Delta\mathbf{U}_{k+m+i}^{l-1} + \alpha_{k+m+i}^l \cdot \mathbf{d}_{k+m+i}^l = \\ \Delta\mathbf{U}_{k+m+i}^{l-1} &- \frac{\partial J(k+m+i)}{\partial \Delta\mathbf{U}_{k+m+i}^l} \alpha_{k+m+i}^l. \quad (7) \end{aligned}$$

2) 步长选择应满足可行约束, 考虑式(7)计算一次.

取 $\Delta\mathbf{U}_{k+m+i}^0 = 0$ , 沿负梯度方向的搜索一次, 记步长为 $\alpha_{k+m+i}$ , 即

$$\begin{aligned} \Delta\mathbf{U}_{k+m+i} &= \\ & \alpha_{k+m+i} \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{W}_{k+m+i} - \\ & \alpha_{k+m+i} \mathbf{R}_{k+m+i} \mathbf{U}_{k+m+i} - \\ & \alpha_{k+m+i} \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{G}_{k+m+i} \mathbf{U}_{k+m+i} - \\ & \alpha_{k+m+i} \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{F}_{k+m+i} \bar{\mathbf{x}}(k+m+i). \quad (8) \end{aligned}$$

令 $k+m+i$ 时刻修正方向为 $\mathbf{d}_{k+m+i}$ , 且令

$$\Gamma_{k+m+i} = \mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{G}_{k+m+i} + \mathbf{R}_{k+m+i},$$

首先 $\alpha_{k+m+i}$ 应使 $J(k+m+i)$ 下降最大<sup>[22]</sup>:

$$\alpha_{k+m+i,1} = \frac{\|\mathbf{d}_{k+m+i}\|}{\mathbf{d}_{k+m+i}^T (\Gamma_{k+m+i}) \mathbf{d}_{k+m+i}}. \quad (9)$$

但基于该步长得到的 $\bar{\mathbf{U}}_{k+m+i}$ 可能不满足约束, 为此须考虑使 $\bar{\mathbf{U}}_{k+m+i}$ 满足式(4)中的所有约束, 即

$$\begin{cases} \mathbf{G}_{k+m+i} \bar{\mathbf{U}}_{k+m+i} \geq \mathbf{Y}_{kl} - \mathbf{F}_{k+m+i} \bar{\mathbf{x}}(k+m+i), \\ \mathbf{G}_{k+m+i} \bar{\mathbf{U}}_{k+m+i} \leq \mathbf{Y}_{ku} - \mathbf{F}_{k+m+i} \bar{\mathbf{x}}(k+m+i), \\ \mathbf{U}_{kl} \leq \bar{\mathbf{U}}_{k+m+i} \leq \mathbf{U}_{ku}, \end{cases} \quad (10)$$

$\mathbf{Y}_{ku}$ 和 $\mathbf{Y}_{kl}$ 是系统输出上、下约束集,  $\mathbf{U}_{ku}$ 和 $\mathbf{U}_{kl}$ 是系统输入上、下约束集. 式(10)整理后表示为

$$\Phi_{k+m+i} \Delta\mathbf{U}_{k+m+i} \leq \Psi_{k+m+i} - \Phi_{k+m+i} \mathbf{U}_{k+m+i}. \quad (11)$$

因慢时标层进行QP求解所得结果满足约束, 所以

在 $k+m+i$ 时刻,若系统所受扰动或模型偏差不大, $\Psi_{k+m+i} - \Phi_{k+m+i}U_{k+m+i} \geq \mathbf{0}$ 成立,式(11)一般有解,若因实测状态和预测状态的差值过大而导致无解,则通过将部分约束放松来获得可行解.记 $\Omega_{k+m+i} = \Psi_{k+m+i} - \Phi_{k+m+i}U_{k+m+i}$ ,于是

$$\alpha_{k+m+i,2} = \min\left\{\frac{\Omega_{k+m+i}(j)}{\Phi_{k+m+i}d'_{k+m+i}(j)} \mid \Phi_{k+m+i}d'_{k+m+i}(j) > \mathbf{0}\right\}, \quad (12)$$

式(12)中 $j = 1, 2, \dots, L$ ,  $L$ 是向量 $\Omega_{k+m+i}$ 的维数.

综合满足约束且使目标函数值下降最多的要求,取 $\alpha_{k+m+i} = \min\{\alpha_{k+m+i,1}, \alpha_{k+m+i,2}\}$ .

在每个控制时刻,快速率层实时计算第 $k+m+i$ 步到第 $k+N$ 步经修正后的控制向量,并将第1个控制量施加给对象,然后将经修正的控制序列保存在存储器中并覆盖原有控制序列,以实现修正的逐层叠加.由于每次迭代都使已有控制量更靠近实际最优控制,因此经本文算法修正后的解比慢速率层下发的解更优.

#### 4.2 盒子约束下的DSF-MPC(Box constraints in DSF-MPC)

具有如下表述形式的问题称之为盒子约束(box constraints)问题<sup>[23]</sup>:

$$\min\{f(\mathbf{x}) \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}, \quad (13)$$

其中: $f$ 为定义在该约束集上的连续可微函数, $\mathbf{u}$ , $\mathbf{l}$ 分别为上、下边界约束向量.包括障碍问题、弹塑性扭转及优化设计等很多实际生产过程都是求解盒子约束下的优化问题.

对于只有输入约束,且各时刻输入约束间无耦合关系的系统,使用MPC进行控制时所求解的在线优化问题即为盒子约束问题<sup>[24]</sup>.针对该问题,当按第4.1节寻找的修正方向不可行时,可利用盒子约束的约束边界特性,将求得的修正方向在可行域中投影,并将投影方向作为新的修正方向,快速率层可继续完成对下发控制量的实时修正.

1) 寻找修正方向 $d'_{k+m+i}$ .

在 $k+m+i$ 时刻,若 $d_{k+m+i}$ 不可行,对于 $\forall \varepsilon > 0$ , $\exists j \in 1, 2, \dots, L$ ,使 $\varepsilon \Phi_{k+m+i}d_{k+m+i}(j) > \Omega_{k+m+i}(j)$ .因只有输入约束, $\Phi_{k+m+i} = (\mathbf{I}, -\mathbf{I})^T$ .于是,按照图3所示,在可行域内寻找 $d_{k+m+i}$ 的映射 $d'_{k+m+i}$ ,并将其作为新的修正方向.

选取一个足够小的 $\varepsilon$ 计算出 $\varepsilon \Phi_{k+m+i}d_{k+m+i}$ ,并将其各分量与 $\Omega_{k+m+i}$ 各分量比较.若 $\Omega_{k+m+i}(j) \geq \varepsilon \Phi_{k+m+i}d_{k+m+i}(j)$ ,则令 $d'_{k+m+i}(j) = d_{k+m+i}(j)$ ;若 $\Omega_{k+m+i}(j) < \varepsilon \Phi_{k+m+i}d_{k+m+i}(j)$ ,则令 $d'_{k+m+i}(j)$ 满足 $\varepsilon \Phi_{k+m+i}d'_{k+m+i}(j) = \Omega_{k+m+i}(j)$ .

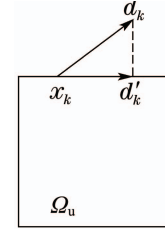


图3 负梯度方向在可行域上的投影

Fig. 3 Negative gradient direction projected on feasible region

2) 寻找修正步长 $\alpha_{k+m+i}$ .

与第4.1节类似, $\alpha_{k+m+i}$ 应使 $J_{k+m+i}$ 下降最大,故

$$\alpha_{k+m+i,1} = \frac{d'_{k+m+i}d_{k+m+i}}{d_{k+m+i}^T \Gamma_{k+m+i} d'_{k+m+i}}. \quad (14)$$

然后再寻找基于约束边界获得的最佳步长.若 $\Phi_{k+m+i}d'_{k+m+i}(j) \leq \mathbf{0}$ ,则对于该分量,不需对步长做修改,否则有

$$\alpha_{k+m+i,2} = \min\left\{\frac{\Omega_{k+m+i}(j)}{\Phi_{k+m+i}d'_{k+m+i}(j)} \mid \Phi_{k+m+i}d'_{k+m+i}(j) > \mathbf{0}\right\}, \quad j = 1, 2, \dots, L. \quad (15)$$

最后,取 $\alpha_{k+m+i} = \min\{\alpha_{k+m+i,1}, \alpha_{k+m+i,2}\}$ .

本节方法增强了在盒子约束下实时修正模块利用实测状态作为反馈来修正控制量的能力.

#### 4.3 算法实现(Algorithm implementation)

因每次都修正第 $k+m+i$ 时刻到第 $k+N$ 时刻的控制量,如图2,故不同采样时刻的 $\mathbf{G}_{k+m+i}$ , $\mathbf{F}_{k+m+i}$ , $\mathbf{Q}_{k+m+i}$ , $\mathbf{R}_{k+m+i}$ 不同.对于 $n_i$ 个输入, $n$ 个状态和 $n_o$ 个输出的系统, $\mathbf{F}_{k+m+i}$ 由 $\mathbf{F}_{k+m}$ 的前 $n_o(N-m-i)$ 行构成, $\mathbf{G}_{k+m+i}$ 由 $\mathbf{G}_{k+m}$ 的前 $n_i n_o(N-m-i)$ 行和前 $n_i n_o(N-m-i)$ 列构成,而 $\mathbf{Q}_{k+m+i}$ 和 $\mathbf{R}_{k+m+i}$ 分别是 $\mathbf{Q}_{k+m}$ 和 $\mathbf{R}_{k+m}$ 的后 $N-m-i$ 个对角元素构成.从存储器中容易直接读得,不需重新求取.

综上所述,可归纳DSF-MPC算法如下:

离线计算部分:

- 1) 求 $\mathbf{F}_y$ ,  $\mathbf{G}_y$ ,  $\mathbf{F}_{k+m}$ ,  $\mathbf{G}_{k+m}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$ ,  $\mathbf{Q}_{k+m}$ ,  $\mathbf{R}_{k+m}$ ;
- 2) 求 $\mathbf{G}_y^T \mathbf{Q} \mathbf{F}_y$ ,  $\mathbf{G}_y^T \mathbf{Q} \mathbf{G}_y + \mathbf{R}$ ,  $\mathbf{G}_y^T \mathbf{Q} \mathbf{G}_y$ 和 $\mathbf{G}_y^T \mathbf{Q} \mathbf{W}_k$

以完成QP求解;

- 3) 求 $\mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{G}_{k+m+i} + \mathbf{R}_{k+m+i}$ ,  $\Phi_{k+m+i}$ ,  $\mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{F}_{k+m+i}$ 和 $\mathbf{G}_{k+m+i}^T \mathbf{Q}_{k+m+i} \mathbf{W}_{k+m+i}(i = 0, \dots, m-1)$ 以求取控制量修正值.

在线计算部分:

快时标层实时修正:

- 1) 获取实测状态 $\bar{\mathbf{x}}(k+m+i)$ ,将其与预测状态 $\mathbf{x}(k+m+i|k)$ 比较,若相等,直接将存储器中控制向量的第 $i$ 个分量施加给对象,令 $i \leftarrow i+1$ ,转1),若二者不相等,转2);

2) 计算无约束最优解 $U_{k+m+i}^*$ , 检查该解是否在可行域内, 若是则将它的一个控制量施加给被控对象, 令 $U_{k+m+i} \leftarrow \bar{U}_{k+m+i}$ , 转4), 否则转3);

3) 从存储器中获取 $U_{k+m+i}$ , 按照式(9)迭代求取控制修正量 $\Delta U_{k+m+i}$ (若为盒子约束问题, 则按第4.2节中的描述求取), 然后将修正后的控制量施加给对象, 令 $U_{k+m+i} \leftarrow \bar{U}_{k+m+i}$ , 转4);

4) 若 $i \% m = 0$ (取模为0), 表示慢时标层MPC完成对控制量的优化, 转5)且令 $i = 0$ , 否则 $i \leftarrow i + 1$ , 转1);

慢时标层求解QP问题:

5) 令 $k \leftarrow k + m$ , 将优化所得的控制向量及第 $k + m$ 步的预测状态送入快时标层, 快时标层的实时修正模块转1), 慢时标层转6);

6) 获取实测值 $\bar{x}(k)$ , 根据 $\Delta x(k) = \bar{x}(k) - x(k|k-m)$ 按1)至4)步修正已有控制序列 $u(k|k-m), \dots, u(k+m-1|k-m)$ , 基于修正后的控制序列 $\bar{u}(k|k-m), \dots, \bar{u}(k+m-1|k-m)$ 求取预测状态 $x(k+m|k)$ , 再基于该预测状态和相关MPC参数开始求解 $k+m$ 时刻到 $k+N$ 时刻的优化问题。

**说明:** 考虑到实际系统中可能存在扰动等因素, 慢速率层的MPC在开始新的QP求解前先修正所需要采用的预估控制序列(第6步), 其目的在于保证在慢速率层得到反馈的时刻( $k, k+m, \dots$ ), 快速率层的实际控制量与慢速率层估计的控制量一致. 而在没有扰动影响时, 该修正并不影响之前计算所得的控制输入。

#### 4.4 性能分析(Performance analysis)

DSF-MPC算法的实时性主要考虑快时标层的实时修正. 在 $k+m+i$ 时刻, 算法考虑 $U_{k+m+i}^*$ 不在可行域内的最坏情况, 则获取 $\Delta U_{k+m+i}$ 所需的计算量见表1.

表1 DSF-MPC实时修正的计算量

Table 1 Calculation complexity of real-time correction in DSF-MPC

加法次数	$2(N-m-i)^2 n_i^2 + 2(N-m-i)n_i - 1$
乘法次数	$2(N-m-i)n_i n + 3(N-m-i)^2 n_i^2$
除法次数	$(N-m-i)^2 n_i^2 - 1$

此时实时修正的计算复杂度为 $O(N^2)$ , 而QP问题求解的计算复杂度为 $O(N^3)$ . 若 $U_{k+m+i}^*$ 在可行域内, 则实时修正的计算复杂度降为 $O(N)$ . 可见, 相比于QP求解, 本文算法确实可快速修正控制量。

尽管本文算法无法保证每次修正都能达到实际最优控制量, 但由于每个时刻都对控制量进行修正, 且在文中所选的步长下, 修正后的控制量一定不比慢时标层下发的控制量差, 因此, 快速率层能够一定程度上弥补慢速率层开环求解优化问题产生的不准确, 并

能快速利用系统的实测输出进行反馈校正, 实现MPC对快速系统的控制。

**注1** 关于闭环系统的稳定性问题, 因没有对噪声干扰及模型失配的具体描述, 只能对名义系统进行讨论. 若无噪声干扰和模型失配, 且将系统控制目标设为状态空间的原点, 则根据第4.1节的描述及最优性原理可知实时修正环节求得的 $\Delta U = 0$ , 施加给被控对象的控制量即为慢速率层求解问题(3)所得的控制量, 系统预测状态 $x(k+i|k)$ 与实际状态 $\bar{x}(k+i)$ 一致( $i = 1, \dots, m$ ), 闭环系统的名义稳定性可以保证<sup>[25-26]</sup>.

#### 5 仿真结果(Simulation results)

为验证本文算法的有效性, 本节分别将电枢控制直流电机和倒立摆作为被控对象, 对它们数学建模, 得到式(16)描述的连续时间动态微分方程。

$$\begin{cases} \dot{x}(t) = \tilde{A}x(t) + \tilde{B}u(t), \\ y(t) = \tilde{C}x(t). \end{cases} \quad (16)$$

对于直流电动机, 式(16)中参数如下<sup>[13-14]</sup>:

$$\tilde{A} = \begin{bmatrix} -\frac{1}{T_s} & 0 & 0 \\ \frac{1}{T_1 R_c} & -\frac{1}{T_1} & -\frac{1}{T_1 R_c} \\ 0 & \frac{R_c}{T_m} & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} \frac{K_s}{T_s} \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{C_e} \end{bmatrix}^T,$$

其中:  $U_c$ 为控制电压,  $U_{d0}$ 为整流器平均输出直流电压,  $E$ 为电机的反电动势, 其结构见图4.

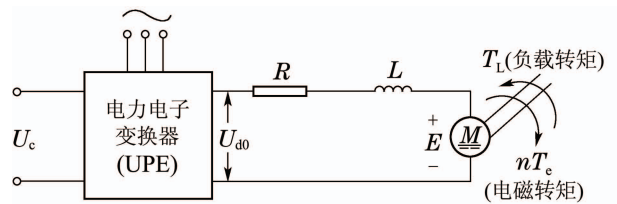


图4 电枢控制直流电动机结构图

Fig. 4 Armature control DC motor structure diagram

系统状态向量为 $x(t) = (U_{d0}, I_d, E)^T$ ,  $I_d$ 为直流电机的电枢电流, 控制量 $u(t)$ 为调节触发装置的控制电压. 矩阵中的各参数分别为电枢回路总电阻 $R_c = 0.5 \Omega$ , 整流装置中的晶闸管放大系数 $K_s = 40$ , 整流装置中三相桥式电路的平均失控时间 $T_s = 0.0017 \text{ s}$ , 直流电机的电磁时间常数 $T_1 = 0.03 \text{ s}$ , 机电时间常数 $T_m = 0.18 \text{ s}$ , 电势系数 $C_e = 0.132 \text{ V} \cdot \text{min/r}$ .

取

$$T = 0.002 \text{ s}, \quad x(0) = (0, 0, 0)^T, \quad u_{\max} = 90,$$

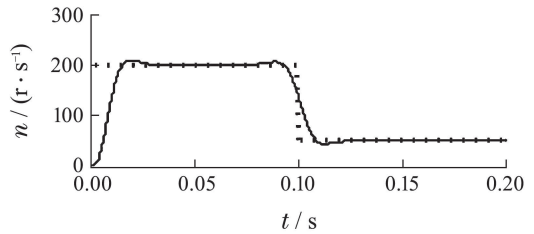
$$P = M = N = 60, \quad Q = \text{diag}\{1, \dots, 1, 20\},$$

$$\mathbf{R} = \text{diag}\{0.1, \dots, 0.1\}.$$

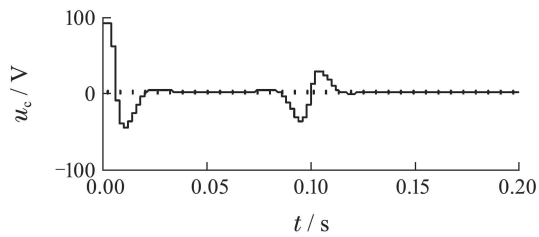
此时无输出约束, 属盒子约束问题.

经多次仿真统计, 在上述参数下, QP求解器完成一次计算的平均时间为0.013706 s, 而慢速率层获取修正控制量的平均时间为0.000981 s. 实时修正可在周期 $T$ 内完成, 而一次优化求解需要5-10个周期. 考虑一定余量, 取 $m = 10$ . 同时采用一步修正. 设控制目标为使直流电机的转速跟踪设定轨迹, 该设定轨迹为在 $t = 0\text{ s}$ 时,  $n = 200\text{ (r/s)}$ ;  $t = 0.1\text{ s}$ 时, 降为 $n = 50\text{ (r/s)}$ .

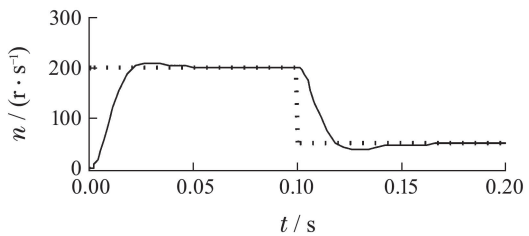
分别采用传统MPC(因QP求解器效率限制, 控制器只能每隔 $m$ 个采样周期将优化所得的第 $m + 1$ 至第 $m + 2m$ 个控制量输送给被控对象), PID, DSF-MPC控制, 且假设响应过程无噪声干扰及模型失配等因素, 则控制效果见图5.



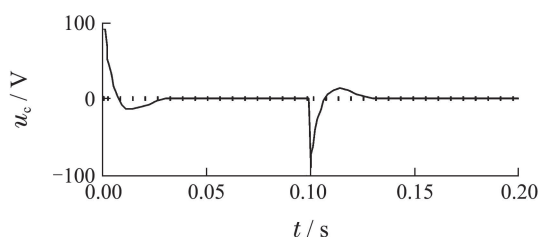
(a) 传统MPC输出



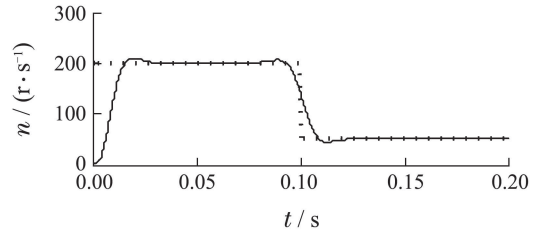
(b) 传统MPC输入



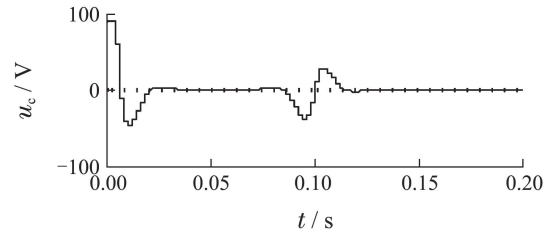
(c) PID输出



(d) PID输入



(e) DSF-MPC输出



(f) DSF-MPC输入

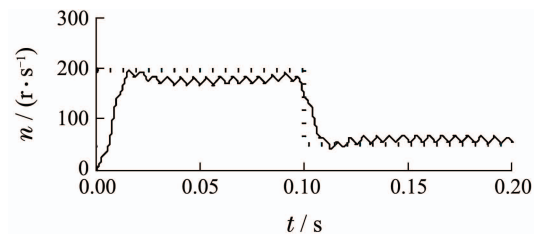
图 5 无噪声及模型失配时的控制效果图

Fig. 5 Control results without noise and model mismatch

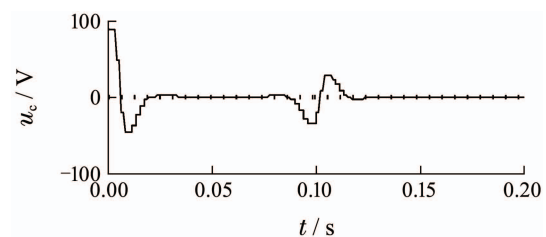
相比常用于控制快速系统的PID算法, MPC算法具备直接处理约束、提前反应等优势. 从图5可见, 该仿真中MPC的调节时间优于PID, 且在设定值发生改变时会提前反应. 另外, 该情况下系统的预测状态等于实测状态, 实时修正模块未起作用, 故DSF-MPC和传统MPC的响应曲线基本一致.

若系统受到来自电网电压的形如 $\omega = \sin t$ 的噪声干扰, 且系统建模存在1%~5%的参数误差, 则分别使用3种控制器得到的控制结果见图6.

由图6可见, PID的调节时间和超调量都明显增加, 但DSF-MPC继承了传统MPC优点的同时能快速反应, 控制效果优于PID, 解决了传统MPC难以在快速系统中应用的问题.



(a) 传统MPC输出



(b) 传统MPC输入

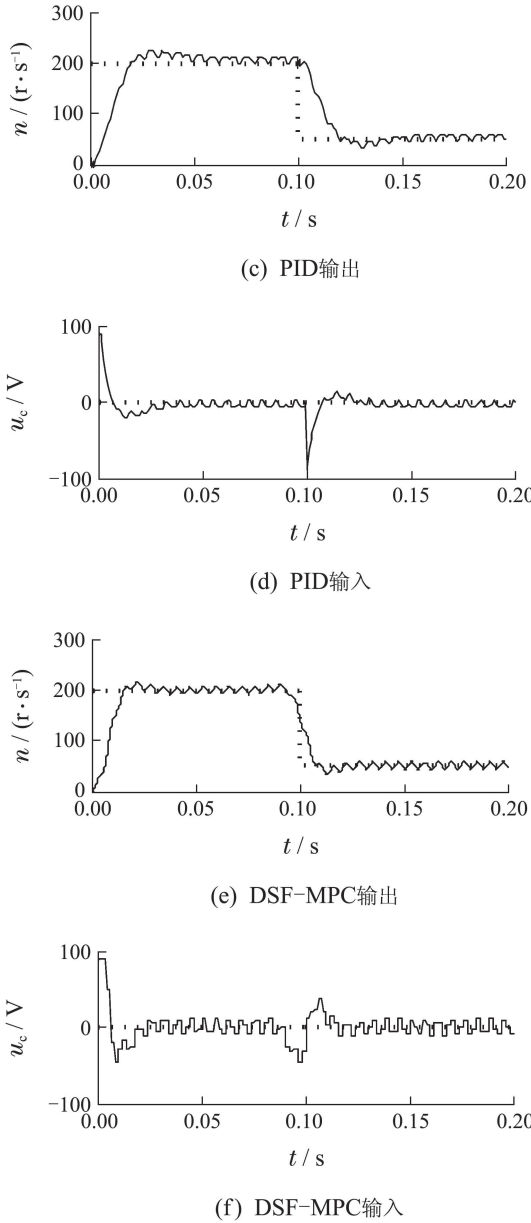


图6 有噪声及模型失配时的控制效果图

Fig. 6 Control results with noise and model mismatch

对于倒立摆系统, 式(16)中参数如下<sup>[15-16]</sup>:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m_p^2 L^2 g}{I(m_p + m_c) + m_p m_c L^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m_p^2 L^2 g}{I(m_p + m_c) + m_p m_c L^2} & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} 0 & -\frac{m_p L}{I(m_p + m_c) + m_p m_c L^2} \\ 0 & \frac{I + m_p L^2}{I(m_p + m_c) + m_p m_c L^2} \end{bmatrix}^T.$$

4个状态分别为摆杆偏角、摆杆角速度、小车位移、小车速度. 系统输入为对摆杆的控制力. GIP-100-L单轴倒立摆实验系统的实际参数为:  $m_p = 0.07$  kg (摆杆质量);  $m_c = 1.32$  kg (小车质量);  $2L = 0.4$  m (摆杆长度);  $g = 10$  m/s<sup>2</sup>.

取

$$\begin{aligned} T &= 0.008 \text{ s}, \mathbf{x}(0) = (0, 0, 0, 0.1)^T, \\ P &= M = N = 40, \mathbf{y}_{\max} = (0.52, 0.1)^T, \\ u_{\max} &= 6, \mathbf{Q}_1 = \text{diag}\{300, 1, 200, 20\}, \\ \mathbf{Q}_2 &= 0.5\mathbf{Q}_1, \mathbf{Q}_3 = 0.01\mathbf{Q}_1, \\ \mathbf{R} &= \text{diag}\{0.02, \dots, 0.02\}, \\ \mathbf{Q} &= \text{diag}\{\underbrace{\mathbf{Q}_3, \dots, \mathbf{Q}_3}_{20}, \underbrace{\mathbf{Q}_2, \dots, \mathbf{Q}_2}_{10}, \underbrace{\mathbf{Q}_1, \dots, \mathbf{Q}_1}_{10}\}. \end{aligned}$$

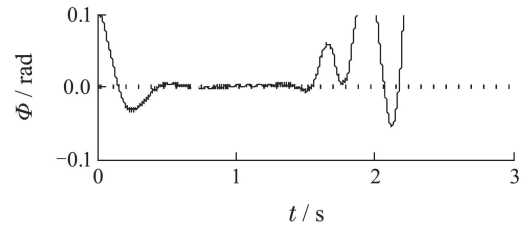
在上述参数下, QP计算平均耗时为0.012127 s, 实时修正平均耗时为0.000806 s, 故取  $m = 3$ . 无噪声干扰及模型失配时传统MPC与DSF-MPC的控制效果一致, 此处不再展示.

若系统受到噪声  $\omega = 0.001 \cdot \sin t$  的干扰且系统建模存在1%~5%的参数误差, 则3种控制方法所得控制结果见图7. 此时传统MPC已经无法保使系统稳定, 在保证所有约束的前提下, 与PID的控制结果相比, DSF-MPC的超调更小、调节时间更短.

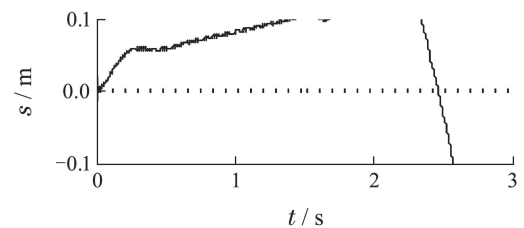
进一步, 令

$$u_{\max} = 10, \mathbf{x}(0) = (0, 0, 0, 0.4)^T,$$

则PID和DSF-MPC的控制效果见图8. 若摆杆受到较大扰动(幅值大于0.4 rad), PID将无法让摆杆角度和小车位移都控制在约束内, 会出现摆杆调节稳定、小车匀速直线运动的情况, 这在实际应用中是不允许的. 但上述情况下, DSF-MPC仍然可以保持较好的控制效果且控制量的波动更轻微, 这有利于保护执行器的电机.



(a) 传统MPC摆角



(b) 传统MPC位移



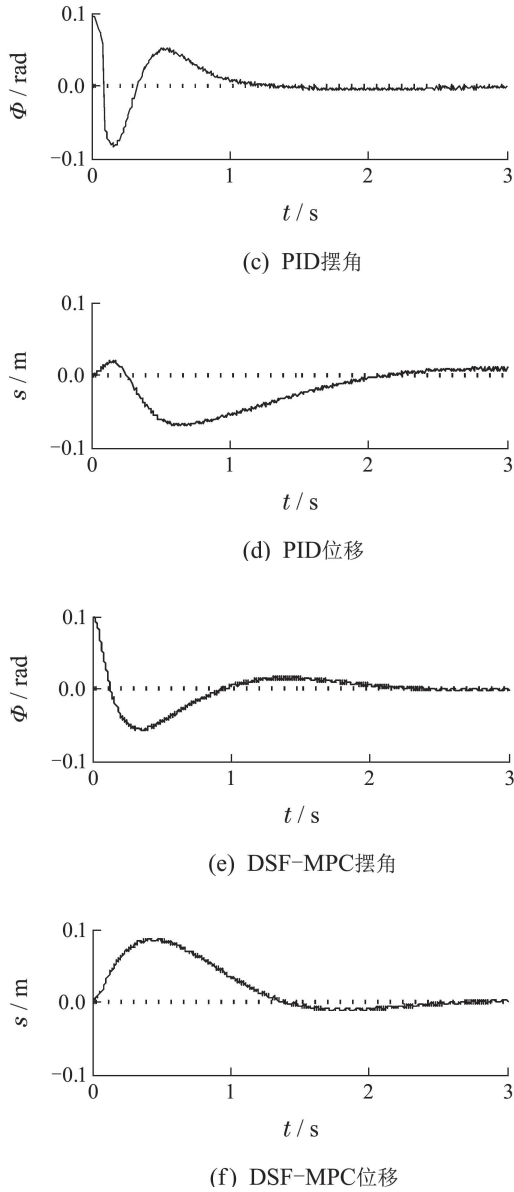


图 7 小偏角下有噪声及模型失配时的控制效果图  
 Fig. 7 Control results with noise and model mismatch when facing small deflection

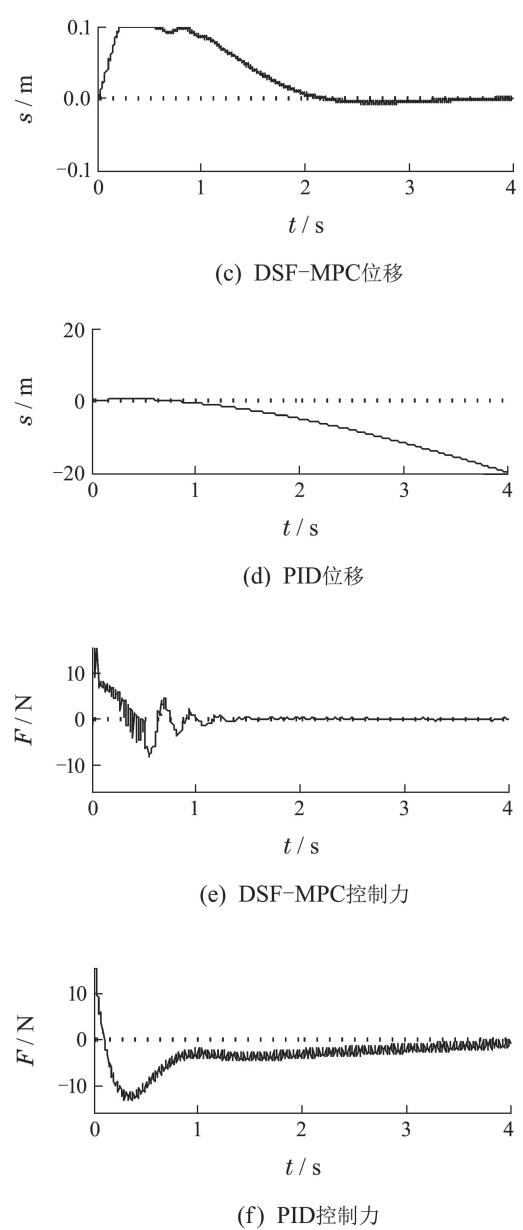
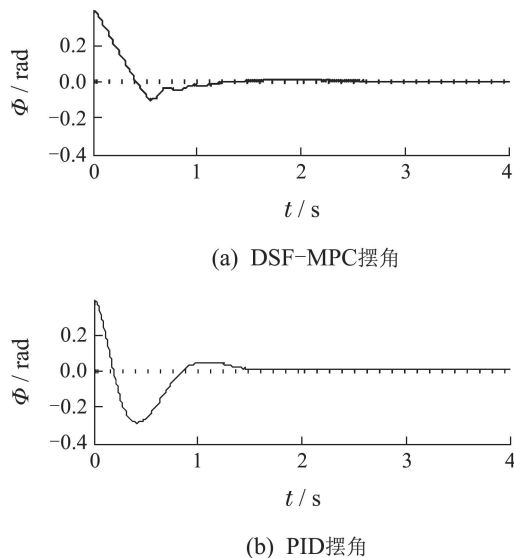


图 8 大偏角下有噪声及模型失配时的控制效果图  
 Fig. 8 Control results with noise and model mismatch when facing big deflection

## 6 结论(Conclusions)

本文针对因QP求解器计算效率有限而导致在快速系统中MPC算法可能会在单个采样周期内无法完成一次控制量在线优化问题, 提出了DSF-MPC算法. 该算法采用双速率框架将实时控制量的求取放到两个时间尺度上完成, 慢速率层每隔 $mT$ 时间完成一次滚动优化, 然后将结果下发给快速率层并基于系统实测状态开始新一轮优化求解; 快速率层则根据每 $T$ 时刻的系统实测信息, 在慢速率层下发控制量的基础上, 对控制量实时修正, 然后施加给对象. 此外, 对DSF-MPC算法计算量分析及MATLAB仿真结果验证了它在修正控制量时的快速性及控制的有效性.

## 参考文献(References):

- [1] XI Yugeng, LI Dewei, LIN Shu. Model predictive control — status and challenges [J]. *Acta Automatica Sinica*, 2013, 39(3): 222 – 236. (席裕庚, 李德伟, 林姝. 模型预测控制—现状与挑战 [J]. *自动化学报*, 2013, 39(3): 222 – 236.)
- [2] LI D W, XI Y G. Quality guaranteed aggregation based model predictive control and stability analysis [J]. *Science in China Series F: Information Science*, 2009, 52(7): 1145 – 1156.
- [3] LI D W, XI Y G, LIN Z L. An improved design of aggregation-based model predictive control [J]. *Systems & Control Letters*, 2013, 62(11): 1082 – 1089.
- [4] PANNOCCHIA G, RAWLINGS J B, WRIGHT S J. Fast large-scale model predictive control by partial enumeration [J]. *Automatica*, 2007, 43(5): 852 – 860.
- [5] WANG Y, BOYD S. Fast model predictive control using online optimization [J]. *IEEE Transactions on Control Systems Technology*, 2010, 18(2): 267 – 278.
- [6] CAGIENARD R, GRIEDER P, KERRIGAN E C, et al. Move blocking strategies in receding horizon control [J]. *Journal of Process Control*, 2007, 17(6): 563 – 570.
- [7] GONDHALEKAR R, IMURA J. Least-restrictive move-blocking model predictive control [J]. *Automatica*, 2010, 46(7): 1234 – 1240.
- [8] LI D W, XI Y G. Aggregation based on closed-loop MPC with guaranteed performance [C] // *The 48th Joint IEEE Conference on Decision and Control and the 28th Chinese Control Conference*. Shanghai: IEEE, 2009: 7400 – 7405.
- [9] LU C S, LI D W, XI Y G. A fast algorithm for MPC based on aggregation strategy [C] // *The 25th Chinese Control and Decision Conference*. Guiyang: IEEE, 2013: 738 – 743.
- [10] KOURAMAS K I, FAISCA N P, PANOS C, et al. Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming [J]. *Automatica*, 2011, 47(8): 1638 – 1645.
- [11] FERREAU H J, KOZMA A, DIEHL M. A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC [C] // *The 4th IFAC Nonlinear Model Predictive Control Conference*. Netherlands: ELSEVIER, 2012, 4(1): 74 – 79.
- [12] GISELSSON P, DOAN M D, KEVICZKY T. Accelerated gradient method and dual decomposition in distributed model predictive control [J]. *Automatica*, 2013, 49(3): 829 – 833.
- [13] RUAN Yi, CHEN Boshi. *Control Systems of Electric Drives — Motion Control Systems* [M]. The 4th Edition. Beijing: China Machine Press, 2010. (阮毅, 陈伯时. 电力拖动自动控制系统: 运动控制系统 [M]. 第4版. 北京: 机械工业出版社, 2010.)
- [14] SIRA-RAMIREZ H, OLIVER-SALAZAR M A. On the robust control of buck-converter DC-motor combinations [J]. *IEEE Transactions on Power Electronics*, 2013, 28(8): 3912 – 3922.
- [15] CUI Ping, WENG Zhengxin. Balance control for single inverted pendulum system based on pole placement method [J]. *Research and Exploration Laboratory*, 2003, 22(2): 70 – 73. (崔平, 翁正新. 基于状态空间极点配置的倒立摆平衡控制 [J]. *实验室研究与探索*, 2003, 22(2): 70 – 73.)
- [16] QI G Y, CHEN Z Q, YUAN Z Z. Adaptive high order differential feedback control for inverted pendulum system [J]. *Control Theory & Applications*, 2004, 21(5): 676 – 683.
- [17] XI Yugeng. *Predictive Control* [M]. 2nd Edition. Beijing: National Defense Industry Press, 2013. (席裕庚. 预测控制 [M]. 第2版. 北京: 国防工业出版社, 2013.)
- [18] ZOU Tao, PAN Hao, DING Baocang, et al. Research development of two-layered predictive control [J]. *Control Theory & Applications*, 2014, 31(10): 1327 – 1337. (邹涛, 潘昊, 丁宝苍, 等. 双层结构预测控制研究进展 [J]. *控制理论与应用*, 2014, 31(10): 1327 – 1337.)
- [19] LU X N, GUERRERO J M, SUN K. Hierarchical control parallel AC-DC converter interfaces for hybrid microgrids [J]. *IEEE Transactions on Smart Grid*, 2014, 5(2): 683 – 692.
- [20] LI Chuanjiang, MA Guangfu. *Optimal Control* [M]. Beijing: Science Press, 2011. (李传江, 马广富. 最优控制 [M]. 北京: 科学出版社, 2011.)
- [21] LIBERZON D. *Calculus of Variations and Optimal Control Theory* [M]. Princeton: Princeton University Press, 2013.
- [22] WANG Yiju, XIU Naihua. *Nonlinear Optimization Theory and Methods* [M]. Beijing: Science Press, 2012. (王宜举, 修乃华. 非线性最优化理论与方法 [M]. 北京: 科学出版社, 2012.)
- [23] HAGER W W, ZHANG H. A new active set algorithm for box constrained optimization [J]. *SIAM Journal on Optimization*, 2006, 17(2): 526 – 557.
- [24] SUN J, FU L, HUANG Z, et al. The method of deterministic optimal control with box constraints [C] // *International Conference of the 3rd Intelligent Networks and Intelligent Systems*. Shenyang: IEEE, 2010: 237 – 240.
- [25] MAYNE D Q, RAWLING J B, RAO C V. Constrained model predictive control: stability and optimality [J]. *Automatica*, 2000, 36(6): 798 – 814.
- [26] LEE J W, KWON W H, CHOI J H. On stability of constrained receding horizon control with finite terminal weighting matrix [J]. *Automatica*, 1998, 34(12): 1607 – 1612.

## 作者简介:

王曦 (1991–), 女, 硕士研究生, 目前研究方向为高效预测控制策略及快速实现, E-mail: salila@sjtu.edu.cn;

李德伟 (1971–), 男, 副研究员, 目前研究方向为预测控制理论与应用, E-mail: dwli@sjtu.edu.cn;

席裕庚 (1946–), 男, 教授, 目前研究方向为预测控制理论与应用、大系统理论与应用, E-mail: ygxi@sjtu.edu.cn.