



# (12)发明专利申请

(10)申请公布号 CN 111708277 A

(43)申请公布日 2020.09.25

(21)申请号 202010479304.6

(22)申请日 2020.05.29

(71)申请人 中国科学技术大学

地址 230026 安徽省合肥市包河区金寨路  
96号

(72)发明人 康宇 王涛 李鹏飞 赵云波  
吕文君

(74)专利代理机构 合肥天明专利事务所(普通  
合伙) 34115

代理人 苗娟

(51)Int.Cl.

G05B 13/04(2006.01)

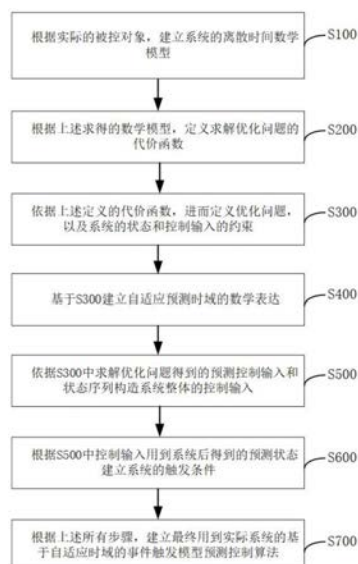
权利要求书3页 说明书8页 附图2页

## (54)发明名称

一种自适应时域的事件触发模型预测控制  
方法

## (57)摘要

本发明的一种自适应时域的事件触发模型预测控制方法,可解决现有的事件触发的模型预测控制方法计算负担较大的技术问题。包括根据实际的被控对象,建立系统的离散时间数学模型;根据上述求得的数学模型,定义求解优化问题的代价函数;定义优化问题,以及系统的状态和控制输入的约束;建立自适应预测时域的数学表达;求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;控制输入用到系统后得到的预测状态建立系统的触发条件;根据上述所有步骤,建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法。本发明预测时域是自适应的,随着状态值越靠近终端约束集,预测时域越小,在线优化计算频率越低,大大降低计算负担。



1. 一种自适应时域的事件触发模型预测控制方法,其特征在于:

包括以下步骤:

S100、根据实际的被控对象,建立系统的离散时间数学模型;

S200、根据上述求得的数学模型,定义求解优化问题的代价函数;

S300、依据上述定义的代价函数,进而定义优化问题,以及系统的状态和控制输入的约束;

S400、基于S300建立自适应预测时域的数学表达;

S500、依据S300中求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;

S600、根据S500中控制输入用到系统后得到的预测状态建立系统的触发条件;

S700、根据上述所有步骤,建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法。

2. 根据权利要求1所述的自适应时域的事件触发模型预测控制方法,其特征在于:所述S100、根据实际的被控对象,建立系统的离散时间数学模型;

具体包括:

建立带有扰动的非线性系统的离散时间模型,其数学描述如式(1)所示:

$$x(k+1) = f(x(k), u(k)) + w(k), k \geq k_0 \quad (1)$$

其中,

$k_0$ 是初始时刻, $f$ 是非线性函数;

$x(k) \in R^n$ 是当前 $k$ 时刻的状态向量;

$u(k) \in R^p$ 是当前 $k$ 时刻的控制输入向量;

$w(k) \in R^m$ 是当前 $k$ 时刻的外部扰动;

设置 $u \in U \subseteq R^p$ ,其中 $U$ 是控制约束的范围。

3. 根据权利要求2所述的自适应时域的事件触发模型预测控制方法,其特征在于:所述S200、根据上述求得的数学模型,定义求解优化问题的代价函数;

其中,所述代价函数数学表达如式(2)所示:

$$V(x(k), \hat{u}_N(k), N) = \sum_{i=0}^{N-1} l(\hat{x}(k+i|k), \hat{u}(k+i|k)) + F(\hat{x}(k+N|k)) \quad (2)$$

其中, $i=0, 1, \dots, N-1$ ,  $k+i$ 表示 $k$ 时刻以后的第 $k+i$ 时刻;

$N$ 为正整数,是模型预测控制所采用的预测步长;

$\hat{u}_N(k)$ 是当前 $k$ 时刻的控制序列:  $\hat{u}_N(k) = \{\hat{u}(k|k), \dots, \hat{u}(k+N-1)\}$ ;

$\hat{x}(k+i|k)$ 是当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量;

$\hat{u}(k+i|k)$ 是当前时刻 $k$ 在 $k+i$ 时刻的预测控制输入向量;

$l(x, u) = x^T Q x + u^T R u$ 是阶段代价,  $x^T, u^T$ 表示状态和控制输入向量的转置,  $Q, R$ 是提前设定的分配给系统状态向量和输入向量的权重正定矩阵;

$F(x) = x^T P x$ 是终端代价,  $P$ 是离线计算出的正定矩阵。

4. 根据权利要求3所述的自适应时域的事件触发模型预测控制方法,其特征在于:所述S300、依据上述定义的代价函数,进而定义优化问题,以及系统的状态和控制输入的约束;

具体包括:

求解使得上述定义的代价函数最小的控制序列 $\hat{\mathbf{u}}_N(k)$ ,同时满足状态,输入约束问题,数学描述如下:

$$\begin{aligned} & \min_{\hat{\mathbf{u}}_N(k)} V(\mathbf{x}(k), \hat{\mathbf{u}}_N(k), N) \\ & \text{s.t. } \hat{\mathbf{x}}(k+i|k) = f(\hat{\mathbf{x}}(k+i-1|k), \hat{\mathbf{u}}(k+i-1|k)) \\ & \hat{\mathbf{x}}(k|k) = \mathbf{x}(k) \\ & \hat{\mathbf{u}}(k+i|k) \in U \\ & \hat{\mathbf{x}}(k+N|k) \in X_f \end{aligned} \quad (3)$$

其中,

当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量 $\hat{\mathbf{x}}(k+i|k)$ 是用系统的标称模型迭代计算的,即式(1)中外部扰动 $\mathbf{w}(k)=0$ ;将触发时刻的预测状态向量值作为优化计算的初始时刻值,即 $\hat{\mathbf{x}}(k|k)=\mathbf{x}(k)$ ;控制输入满足设定的约束 $U$ ;预测的终端时刻 $k+N$ 时的状态满足终端约束 $X_f = \{\mathbf{x}: \|\mathbf{x}\|_P \leq \epsilon_f\}$ ,  $\epsilon_f$ 是正的常数,  $\|\mathbf{x}\|_P$ 是矩阵范数,  $P$ 是正定的权重矩阵,  $\|\mathbf{x}\|_P = \sqrt{\mathbf{x}^T P \mathbf{x}}$ , 终端约束是离线计算得到。

5.根据权利要求4所述的自适应时域的事件触发模型预测控制方法,其特征在于:S400、基于S300建立自适应预测时域的数学表达;

其中数学表达式为:

$$N(k_{i+1}) = \max \{N^*(k_i) - \Delta(k_i) + 1, 0\} \quad (4)$$

其中,

$k_i$ 是某一触发时刻, $k_{i+1}$ 是下一触发时刻;

$\Delta(k_i) = k_{i+1} - k_i$ 表示的是两个相邻触发时刻的差;

$N^*(k_i)$ 是当前 $k_i$ 时刻状态值首次进入到终端约束集的步数,即 $\hat{\mathbf{x}}(k_i + N^*(k_i)|k_i) \in X_f$ ,而在 $k_i + N^*(k_i)$ 时刻之前状态值在终端约束集外;

$N(k_i)$ 是当前 $k_i$ 时刻的预测步长, $N(k_{i+1})$ 则是下一触发时刻 $k_{i+1}$ 的预测步长。

6.根据权利要求5所述的自适应时域的事件触发模型预测控制方法,其特征在于:所述S500、依据S300中求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;

具体包括:

设初始时刻给出一个固定的正整数预测步长记为 $N_{\max}$ ,则:

$$\hat{\mathbf{u}}(k_i + j|k_i) = \begin{cases} \hat{\mathbf{u}}^*(k_i + j|k_i), & j = 0, \dots, N(k_i) - 1 \\ \kappa(\hat{\mathbf{x}}(k_i + j|k_i)), & j = N(k_i), \dots, N_{\max} - 1 \end{cases} \quad (5)$$

那么预测状态值为:

$$\hat{\mathbf{x}}(k_i + j + 1|k_i) = f(\hat{\mathbf{x}}(k_i + j|k_i), \hat{\mathbf{u}}(k_i + j|k_i), j = 0, \dots, N_{\max} - 1 \quad (6)$$

其中,

$\kappa(\mathbf{x})$ 是离线计算得到的局部控制输入;当前 $k_i$ 时刻的在线优化计算的步长是 $N(k_i)$ ,在线优化得到 $N(k_i)$ 个最优控制输入 $\hat{\mathbf{u}}^*(k_i + j|k_i)$ ,剩下 $N_{\max} - N(k_i)$ 个控制输入由离线计算的局

部控制输入 $k(x)$ 得到。

7. 根据权利要求6所述的自适应时域的事件触发模型预测控制方法, 其特征在于:  
S600、根据S500中控制输入用到系统后得到的预测状态建立系统的触发条件;

包括:

如式 (7) 所示:

$$\begin{cases} \|x(k) - \hat{x}(k|k_i)\|_P \leq (\epsilon_f - \epsilon)L_x^{-(N^*(k_i)+1)} - \frac{\|w\|_P}{L_x} \\ k - k_i \leq N_{\max} \end{cases} \quad (7)$$

其中

$\hat{x}(k|k_i)$ 表示从当前时刻 $k_i$ 开始对以后时刻 $k$ 的预测状态;

$\epsilon$ 是定义另一状态约束范围得到, 定义另一状态向量的约束:  $\Phi = \{x: \|x\|_P \leq \epsilon\}$ , 其中  $\epsilon < \epsilon_f$ , 离线计算得到辅助控住律 $\kappa(x)$ , 并使得对于全部的 $x \in X_f$ ,  $f(x, \kappa(x)) \in \Phi$ 都成立, 从而得到约束集 $\Phi$ ;

$L_x$ 是系统的局部Lipschitz常数;

$\|w\|_P$ 是外界扰动的范数最大值且满足:  $\|w\|_P \leq (\epsilon_f - \epsilon)L_x^{-(N_0+1)}$ ;

$N_{\max}$ 是最初设定的总的步长, 为正整数;

$N_0$ 是最初设定的使得给定的初始状态值能够到达终端约束集的预测步长, 满足 $N_0 \leq N_{\max}$ ;

如果在某一时刻 $k$ , (7) 式中任意一个条件不满足, 则重新设定下一触发时刻 $k_{i+1}=k$ 。

8. 根据权利要求7所述的自适应时域的事件触发模型预测控制方法, 其特征在于:

S700、根据上述所有步骤, 建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法;

具体包括以下步骤:

S701: 初始化系统的状态值 $x(k_0)$ , 给定初始的预测步长 $N_0$ , 初始的总的步长 $N_{\max}$ ;

S702: 在每一触发时刻 $k=k_i$ , 初始化状态值:  $\hat{x}(k_i|k_i) = x(k_i)$ , 如果 $i > 0$ , 则根据 (4) 式更新预测步长 $N(k_i)$ ; 否则,  $N(k_0) = N_0$ ;

S703: 判断 $N(k_i)$ 的值, 如果 $N(k_i) = 0$ , 则进入S704; 否则, 根据 (3) 式求解优化问题, 进而得到最优控制输入 $\hat{u}^*(k_i + j|k_i)$ 和最优状态值, 并由定义确定 $N^*(k_i)$ 的值;

S704: 根据 (5) 计算长度为 $N_{\max} - N(k_i)$ 的控制输入, 进而得到总的控制输入 $\hat{u}(k_i + j|k_i)$ ;

S705: 在当前 $k_i$ 时刻, 应用控制输入 $\hat{u}(k|k_i)$ , 根据 (6) 式计算得到相应状态值, 并更新时刻 $k=k+1$ ;

S706: 判断触发条件 (7) 是否违反, 是, 则令 $k_{i+1}=k$ 且 $i=i+1$ , 回到S702; 否, 回到S705。

## 一种自适应时域的事件触发模型预测控制方法

### 技术领域

[0001] 本发明涉及工业过程控制技术领域,具体涉及一种自适应时域的事件触发模型预测控制方法。

### 背景技术

[0002] 模型预测控制在处理多输入多输出和带有约束的系统方面具有优势,因而实际中被广泛的应用。其基本原理是在每个时刻在线求解一个凸优化问题,进而得到一系列最优的控制量,最后将当前最优控制序列的第一个控制量用到被控对象上,此后时刻不断重复此过程。一方面由于每个时刻都需要在线求解凸优化问题,另一方面较大的预测时域,二者都会造成计算方面的负担,所以传统的模型预测控制方法需要很大的计算资源,从而在一定程度上阻碍了其更广泛的应用例如在伺服系统,无人驾驶等一些需要快速计算的系统。有一些方法去解决这样的计算负担问题,其中包括但不仅限于基于事件触发的模型预测控制方法。基于事件触发的模型预测只在一些设定的条件满足或违反时才会重新计算优化问题,因而很大程度的节约了计算资源。较为新颖的基于事件触发的模型预测控制存在以下不足之处:

[0003] 1) 预测时域固定不变,触发条件定义在终端约束集外。这样的方法不足之处在于大的预测时域会导致大的计算的负担且状态进入终端约束集内会导致事件频繁触发;

[0004] 2) 预测时域是自适应的,触发条件定义在终端约束集外。虽然预测时域会随着状态演化而减小,一定程度减少计算资源,但是触发条件在终端约束集内是不适用的,所以一旦状态进入终端约束集内会导致事件频繁地触发即频繁求解优化问题,又在一定程度上增加了计算负担;

[0005] 3) 预测时域是自适应的,触发条件在终端约束集内外各自定义。此方法在很大程度上降低了计算负担,但是需要采取额外的措施解决在终端约束集内外连续触发的问题。

### 发明内容

[0006] 本发明提出的一种自适应时域的事件触发模型预测控制方法,可解决现有的事件触发的模型预测控制方法计算负担较大的技术问题。

[0007] 为实现上述目的,本发明采用了以下技术方案:

[0008] 一种自适应时域的事件触发模型预测控制方法,包括以下步骤:

[0009] S100、根据实际的被控对象,建立系统的离散时间数学模型;

[0010] S200、根据上述求得的数学模型,定义求解优化问题的代价函数;

[0011] S300、依据上述定义的代价函数,进而定义优化问题,以及系统的状态和控制输入的约束;

[0012] S400、基于S300建立自适应预测时域的数学表达;

[0013] S500、依据S300中求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;

[0014] S600、根据S500中控制输入用到系统后得到的预测状态建立系统的触发条件；  
 [0015] S700、根据上述所有步骤，建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法。

[0016] 进一步的，所述S100、根据实际的被控对象，建立系统的离散时间数学模型；

[0017] 具体包括：

[0018] 建立带有扰动的非线性系统的离散时间模型，其数学描述如式 (1) 所示：

$$[0019] \quad x(k+1) = f(x(k), u(k)) + w(k), k \geq k_0 \quad (1)$$

[0020] 其中，

[0021]  $k_0$ 是初始时刻， $f$ 是非线性函数；

[0022]  $x(k) \in R^n$ 是当前 $k$ 时刻的状态向量；

[0023]  $u(k) \in R^p$ 是当前 $k$ 时刻的控制输入向量；

[0024]  $w(k) \in R^m$ 是当前 $k$ 时刻的外部扰动；

[0025] 设置  $u \in U \subseteq R^p$ ，其中 $U$ 是控制约束的范围。

[0026] 进一步的，所述S200、根据上述求得的数学模型，定义求解优化问题的代价函数；

[0027] 其中，所述代价函数数学表达如式 (2) 所示：

$$[0028] \quad V(x(k), \hat{u}_N(k), N) = \sum_{i=0}^{N-1} l(\hat{x}(k+i|k), \hat{u}(k+i|k)) + F(\hat{x}(k+N|k)) \quad (2)$$

[0029] 其中， $i=0, 1, \dots, N-1$ ， $k+i$ 表示 $k$ 时刻以后的第 $k+i$ 时刻；

[0030]  $N$ 为正整数，是模型预测控制所采用的预测步长；

[0031]  $\hat{u}_N(k)$ 是当前 $k$ 时刻的控制序列： $\hat{u}_N(k) = \{\hat{u}(k|k), \dots, \hat{u}(k+N-1)\}$ ；

[0032]  $\hat{x}(k+i|k)$ 是当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量；

[0033]  $\hat{u}(k+i|k)$ 是当前时刻 $k$ 在 $k+i$ 时刻的预测控制输入向量；

[0034]  $l(x, u) = x^T Q x + u^T R u$ 是阶段代价， $x^T, u^T$ 表示状态和控制输入向量的转置， $Q, R$ 是提前设定的分配给系统状态向量和输入向量的权重正定矩阵；

[0035]  $F(x) = x^T P x$ 是终端代价， $P$ 是离线计算出的正定矩阵。

[0036] 进一步的，所述S300、依据上述定义的代价函数，进而定义优化问题，以及系统的状态和控制输入的约束；

[0037] 具体包括：

[0038] 求解使得上述定义的代价函数最小的控制序列  $\hat{u}_N(k)$ ，同时满足状态，输入约束问题，数学描述如下：

$$[0039] \quad \min_{\hat{u}_N(k)} V(x(k), \hat{u}_N(k), N)$$

$$[0040] \quad \text{s.t. } \hat{x}(k+i|k) = f(\hat{x}(k+i-1|k), \hat{u}(k+i-1|k))$$

$$[0041] \quad \hat{x}(k|k) = x(k) \quad (3)$$

$$[0042] \quad \hat{u}(k+i|k) \in U$$

$$[0043] \quad \hat{x}(k+N|k) \in X_f$$

[0044] 其中,

[0045] 当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量 $\hat{x}(k+i|k)$ 是用系统的标称模型迭代计算的,即式(1)中外部扰动 $w(k)=0$ ;将触发时刻的预测状态向量值作为优化计算的初始时刻值,即 $\hat{x}(k|k)=x(k)$ ;控制输入满足设定的约束 $U$ ;预测的终端时刻 $k+N$ 时的状态满足终端约束 $X_f=\{x:\|x\|_P\leq\epsilon_f\}$ , $\epsilon_f$ 是正的常数, $\|x\|_P$ 是矩阵范数, $P$ 是正定的权重矩阵, $\|x\|_P=\sqrt{x^TPx}$ ,终端约束是离线计算得到。

[0046] 进一步的,S400、基于S300建立自适应预测时域的数学表达;

[0047] 其中数学表达式为:

$$[0048] \quad N(k_{i+1})=\max\{N^*(k_i)-\Delta(k_i)+1,0\} \quad (4)$$

[0049] 其中,

[0050]  $k_i$ 是某一触发时刻, $k_{i+1}$ 是下一触发时刻;

[0051]  $\Delta(k_i)=k_{i+1}-k_i$ 表示的是两个相邻触发时刻的差;

[0052]  $N^*(k_i)$ 是当前 $k_i$ 时刻状态值首次进入到终端约束集的步数,即 $\hat{x}(k_i+N^*(k_i)|k_i)\in X_f$ ,而在 $k_i+N^*(k_i)$ 时刻之前状态值在终端约束集外;

[0053]  $N(k_i)$ 是当前 $k_i$ 时刻的预测步长, $N(k_{i+1})$ 则是下一触发时刻 $k_{i+1}$ 的预测步长。

[0054] 进一步的,所述S500、依据S300中求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;

[0055] 具体包括:

[0056] 设初始时刻给出一个固定的正整数预测步长记为 $N_{\max}$ ,则:

$$[0057] \quad \hat{u}(k_i+j|k_i)=\begin{cases} \hat{u}^*(k_i+j|k_i), & j=0,\dots,N(k_i)-1 \\ \kappa(\hat{x}(k_i+j|k_i)), & j=N(k_i),\dots,N_{\max}-1 \end{cases} \quad (5)$$

[0058] 那么预测状态值为:

$$[0059] \quad \hat{x}(k_i+j+1|k_i)=f(\hat{x}(k_i+j|k_i),\hat{u}(k_i+j|k_i),j=0,\dots,N_{\max}-1) \quad (6)$$

[0060] 其中,

[0061]  $\kappa(x)$ 是离线计算得到的局部控制输入;当前 $k_i$ 时刻的在线优化计算的步长是 $N(k_i)$ ,在线优化得到 $N(k_i)$ 个最优控制输入 $\hat{u}^*(k_i+j|k_i)$ ,剩下 $N_{\max}-N(k_i)$ 个控制输入由离线计算的局部控制输入 $\kappa(x)$ 得到。

[0062] 进一步的,S600、根据S500中控制输入用到系统后得到的预测状态建立系统的触发条件;

[0063] 包括:

[0064] 如式(7)所示:

$$[0065] \quad \begin{cases} \|x(k)-\hat{x}(k|k_i)\|_P\leq(\epsilon_f-\epsilon)L_x^{-(N^*(k_i)+1)}-\frac{\|w\|_P}{L_x} \\ k-k_i\leq N_{\max} \end{cases} \quad (7)$$

[0066] 其中

[0067]  $\hat{x}(k|k_i)$ 表示从当前时刻 $k_i$ 开始对以后时刻 $k$ 的预测状态;

[0068]  $\epsilon$ 是定义另一状态约束范围得到,定义另一状态向量的约束: $\Phi=\{x:\|x\|_P\leq\epsilon\}$ ,

其中  $\epsilon < \epsilon_f$ , 离线计算得到辅助控住律  $k(x)$ , 并使得对于全部的  $x \in X_f, f(x, k(x)) \in \Phi$  都成立, 从而得到约束集  $\Phi$ ;

[0069]  $L_x$  是系统的局部 Lipschitz 常数;

[0070]  $\|w\|_P$  是外界扰动的范数最大值且满足:  $\|w\|_P \leq (\epsilon_f - \epsilon)L_x^{-(N_0+1)}$ ;

[0071]  $N_{\max}$  是最初设定的总的步长, 为正整数;

[0072]  $N_0$  是最初设定的使得给定的初始状态值能够到达终端约束集的预测步长, 满足  $N_0 \leq N_{\max}$ ;

[0073] 如果在某一时刻  $k$ , (7) 式中任意一个条件不满足, 则重新设定下一触发时刻  $k_{i+1} = k$ 。

[0074] 进一步的, S700、根据上述所有步骤, 建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法;

[0075] 具体包括以下步骤:

[0076] S701: 初始化系统的状态值  $x(k_0)$ , 给定初始的预测步长  $N_0$ , 初始的总的步长  $N_{\max}$ ;

[0077] S702: 在每一触发时刻  $k = k_i$ , 初始化状态值:  $\hat{x}(k_i|k_i) = x(k_i)$ , 如果  $i > 0$ , 则根据 (4) 式更新预测步长  $N(k_i)$ ; 否则,  $N(k_0) = N_0$ ;

[0078] S703: 判断  $N(k_i)$  的值, 如果  $N(k_i) = 0$ , 则进入 S704; 否则, 根据 (3) 式求解优化问题, 进而得到最优控制输入  $\hat{u}^*(k_i + j|k_i)$  和最优状态值, 并由定义确定  $N^*(k_i)$  的值;

[0079] S704: 根据 (5) 计算长度为  $N_{\max} - N(k_i)$  的控制输入, 进而得到总的控制输入  $\hat{u}(k_i + j|k_i)$ ;

[0080] S705: 在当前  $k_i$  时刻, 应用控制输入  $\hat{u}(k|k_i)$ , 根据 (6) 式计算得到相应状态值, 并更新时刻  $k = k + 1$ ;

[0081] S706: 判断触发条件 (7) 是否违反, 是, 则令  $k_{i+1} = k$  且  $i = i + 1$ , 回到 S702; 否, 回到 S705。

[0082] 由上述技术方案可知, 本发明的自适应时域的事件触发模型预测控制方法具有以下有益效果:

[0083] 1. 预测时域是自适应的, 随着状态值越靠近终端约束集, 预测时域越小, 在线优化计算频率越低, 大大降低计算负担。

[0084] 2. 触发条件在终端约束集内外是统一的, 在终端约束集内触发条件仍然有效, 解决了上述提出的连续触发问题。

[0085] 3. 不需要两个触发时刻之间的间隔必须小于预测时域的条件, 从而更大了降低了计算负担。

## 附图说明

[0086] 图1是本发明的结构示意图。

[0087] 图2是本发明实施例中涉及的总步长  $N_{\max}$ , 在线优化计算自适应步长  $N(k_i)$  以及相邻两触发时刻差  $\Delta(k_i)$  的关系示意图;

[0088] 图3是本发明的预测时域图和触发时刻图;



[0089] 图4是本发明实施例与传统固定时域算法的计算耗时对比。

### 具体实施方式

[0090] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。

[0091] 如图1所示,本实施例所述的自适应时域的事件触发模型预测控制方法,包括以下步骤:

[0092] S100、根据实际的被控对象,建立系统的离散时间数学模型;

[0093] S200、根据上述求得的数学模型,定义求解优化问题的代价函数;

[0094] S300、依据上述定义的代价函数,进而定义优化问题,以及系统的状态和控制输入的约束;

[0095] S400、基于S300建立自适应预测时域的数学表达;

[0096] S500、依据S300中求解优化问题得到的预测控制输入和状态序列构造系统整体的控制输入;

[0097] S600、根据S500中控制输入用到系统后得到的预测状态建立系统的触发条件;

[0098] S700、根据上述所有步骤,建立最终用到实际系统的基于自适应时域的事件触发模型预测控制算法。

[0099] 以下结合图2,对应上述步骤分别展开具体说明:

[0100] 步骤1:建立带有扰动的非线性系统的离散时间模型,其数学描述如式(1)所示:

$$[0101] \quad x(k+1) = f(x(k), u(k)) + w(k), k \geq k_0 \quad (1)$$

[0102] 其中,

[0103]  $k_0$ 是初始时刻, $f$ 是非线性函数;

[0104]  $x(k) \in R^n$ 是当前 $k$ 时刻的状态向量;

[0105]  $u(k) \in R^p$ 是当前 $k$ 时刻的控制输入向量;

[0106]  $w(k) \in R^m$ 是当前 $k$ 时刻的外部扰动。

[0107] 考虑到实际控制过程对控制输入大小有一定的要求,这里设置  $u \in U \subseteq R^p$ ,其中 $U$ 是需要的控制约束的范围。

[0108] 步骤2.建立求解有限时域优化问题所需要的代价函数,其中可以改变代价函数的参数矩阵,进而改变状态变量和控制输入的权重。代价函数数学表达如式(2)所示:

$$[0109] \quad V(x(k), \hat{u}_N(k), N) = \sum_{i=0}^{N-1} l(\hat{x}(k+i|k), \hat{u}(k+i|k)) + F(\hat{x}(k+N|k)) \quad (2)$$

[0110] 其中, $i=0,1,\dots,N-1$ , $k+i$ 表示 $k$ 时刻以后的第 $k+i$ 时刻; $N$ 为正整数,是模型预测控制所采用的预测步长,将通过下述算法获得; $\hat{u}_N(k)$ 是当前 $k$ 时刻的控制序列:

$\hat{u}_N(k) = \{\hat{u}(k|k), \dots, \hat{u}(k+N-1)\}$ ;  $\hat{x}(k+i|k)$ 是当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量;

$\hat{u}(k+i|k)$ 是当前时刻 $k$ 在 $k+i$ 时刻的预测控制输入向量; $l(x, u) = x^T Q x + u^T R u$ 是阶段代价, $x^T, u^T$ 表示状态和控制输入向量的转置, $Q, R$ 是提前设定的分配给系统状态向量和输入向量的权重正定矩阵; $F(x) = x^T P x$ 是终端代价, $P$ 是离线计算出的正定矩阵。

[0111] 步骤3.定义需要在线计算的优化问题,此优化问题具体可以通过matlab中的

fmincon函数来在线求解,即求解使得上述定义的代价函数最小的控制序列 $\hat{\mathbf{u}}_N(k)$ ,同时满足状态,输入约束问题,数学描述如下:

$$\begin{aligned}
 [0112] \quad & \min_{\hat{\mathbf{u}}_N(k)} V(x(k), \hat{\mathbf{u}}_N(k), N) \\
 [0113] \quad & \text{s.t. } \hat{x}(k+i|k) = f(\hat{x}(k+i-1|k), \hat{u}(k+i-1|k)) \\
 [0114] \quad & \hat{x}(k|k) = x(k) \\
 [0115] \quad & \hat{u}(k+i|k) \in U \\
 [0116] \quad & \hat{x}(k+N|k) \in X_f \\
 [0117] \quad & \text{其中,}
 \end{aligned} \tag{3}$$

[0118] 当前 $k$ 时刻对 $k+i$ 时刻的预测状态向量 $\hat{x}(k+i|k)$ 是用系统的标称模型迭代计算的,即式(1)中外部扰动 $w(k)=0$ ;将触发时刻的预测状态向量值作为优化计算的初始时刻值,即 $\hat{x}(k|k)=x(k)$ ;控制输入满足设定的约束 $U$ ;预测的终端时刻 $k+N$ 时的状态满足终端约束 $X_f = \{x: \|x\|_P \leq \epsilon_f\}$ ,  $\epsilon_f$ 是正的常数,  $\|x\|_P$ 是矩阵范数,  $P$ 是正定的权重矩阵,  $\|x\|_P = \sqrt{x^T P x}$ ,终端约束是离线计算得到。

[0119] 步骤4:建立自适应预测时域的数学表达:

$$[0120] \quad N(k_{i+1}) = \max \{N^*(k_i) - \Delta(k_i) + 1, 0\} \tag{4}$$

[0121] 其中,

[0122]  $k_i$ 是某一触发时刻,  $k_{i+1}$ 是下一触发时刻;  $\Delta(k_i) = k_{i+1} - k_i$ 表示的是两个相邻触发时刻的差;  $N^*(k_i)$ 是当前 $k_i$ 时刻状态值首次进入到终端约束集的步数,即 $\hat{x}(k_i + N^*(k_i)|k_i) \in X_f$ ,而在 $k_i + N^*(k_i)$ 时刻之前状态值在终端约束集外;  $N(k_i)$ 是当前 $k_i$ 时刻的预测步长,  $N(k_{i+1})$ 则是下一触发时刻 $k_{i+1}$ 的预测步长。

[0123] 步骤5:建立优化过程中系统需要的控制输入和由此得到的状态值的数学表达。在步骤4中给出了在线求解优化问题的预测步长 $N(k_{i+1})$ ,但是当预测步长减为0后,即状态进入终端约束集后,优化问题将不再求解,则没有控制输入,所以需要额外给出此时的控制输入,初始时刻给出一个固定的正整数预测步长记为 $N_{\max}$ ,则:

$$[0124] \quad \hat{\mathbf{u}}(k_i + j|k_i) = \begin{cases} \hat{\mathbf{u}}^*(k_i + j|k_i), & j = 0, \dots, N(k_i) - 1 \\ \kappa(\hat{x}(k_i + j|k_i)), & j = N(k_i), \dots, N_{\max} - 1 \end{cases} \tag{5}$$

[0125] 那么预测状态值为:

$$[0126] \quad \hat{x}(k_i + j + 1|k_i) = f(\hat{x}(k_i + j|k_i), \hat{u}(k_i + j|k_i), j = 0, \dots, N_{\max} - 1) \tag{6}$$

[0127] 其中,

[0128]  $\kappa(x)$ 是离线计算得到的局部控制输入。当前 $k_i$ 时刻的在线优化计算的步长是 $N(k_i)$ ,在线优化得到 $N(k_i)$ 个最优控制输入 $\hat{\mathbf{u}}^*(k_i + j|k_i)$ ,剩下 $N_{\max} - N(k_i)$ 个控制输入由离线计算的局部控制输入 $\kappa(x)$ 得到。

[0129] 步骤6:建立基于事件触发的模型预测控制的触发条件,如式(7)所示:

$$[0130] \quad \begin{cases} \|x(k) - \hat{x}(k|k_i)\|_P \leq (\epsilon_f - \epsilon)L_x^{-(N^*(k_i)+1)} - \frac{\|w\|_P}{L_x} \\ k - k_i \leq N_{max} \end{cases} \quad (7)$$

[0131] 其中

[0132]  $\hat{x}(k|k_i)$ 表示从当前时刻 $k_i$ 开始对以后时刻 $k$ 的预测状态; $\epsilon$ 是定义另一状态约束范围得到,定义另一状态向量的约束: $\Phi = \{x: \|x\|_P \leq \epsilon\}$ ,其中 $\epsilon < \epsilon_f$ ,离线计算得到辅助控住律 $\kappa(x)$ ,并使得对于全部的 $x \in X_f, f(x, \kappa(x)) \in \Phi$ 都成立,从而得到约束集 $\Phi$ ;  $L_x$ 是系统的局部Lipschitz常数; $\|w\|_P$ 是外界扰动的范数最大值且满足: $\|w\|_P \leq (\epsilon_f - \epsilon)L_x^{-(N_0+1)}$ ;  $N_{max}$ 是最初设定的总的步长,为正整数; $N_0$ 是最初设定的使得给定的初始状态值能够到达终端约束集的预测步长,满足 $N_0 \leq N_{max}$ 。

[0133] 如果在某一时刻 $k$ , (7) 式中任意一个条件不满足,则重新设定下一触发时刻 $k_{i+1} = k$ 。

[0134] 步骤7:根据上述步骤,建立基于自适应时域的事件触发模型预测控制算法,整个算法的具体步骤如下所示:

[0135] S1:初始化系统的状态值 $x(k_0)$ ,给定初始的预测步长 $N_0$ ,初始的总的步长 $N_{max}$ 。

[0136] S2:在每一触发时刻 $k = k_i$ ,初始化状态值: $\hat{x}(k_i|k_i) = x(k_i)$ ,如果 $i > 0$ ,则根据(4)式更新预测步长 $N(k_i)$ ;否则, $N(k_0) = N_0$ 。

[0137] S3:判断 $N(k_i)$ 的值,如果 $N(k_i) = 0$ ,则进入S4;否则,根据(3)式求解优化问题,进而得到最优控制输入 $\hat{u}^*(k_i + j|k_i)$ 和最优状态值,并由定义确定 $N^*(k_i)$ 的值。

[0138] S4:根据(5)计算长度为 $N_{max} - N(k_i)$ 的控制输入,进而得到总的控制输入 $\hat{u}(k_i + j|k_i)$ 。

[0139] S5:在当前 $k_i$ 时刻,应用控制输入 $\hat{u}(k|k_i)$ ,根据(6)式计算得到相应状态值,并更新时刻 $k = k + 1$ 。

[0140] S6:判断触发条件(7)是否违反,是,则令 $k_{i+1} = k$ 且 $i = i + 1$ ,回到S2;否,回到S5。

[0141] 以下举例说明:

[0142] 考虑工业控制中的一个搅拌釜反应器:

[0143] 存在的问题:传统的应用模型预测控制需要不断的在计算机上求解优化问题,进而使得反应器内的浓度和温度达到指定的温度,由于优化问题中一方面预测时域固定,另一方面每个时刻都要求解,导致十分消耗计算机资源,计算压力大。

[0144] 具体应用过程如下:

[0145] 建立搅拌釜反应器的连续时间数学模型为:

$$[0146] \quad \frac{dC_A}{dt} = \frac{q}{V}(C_{Af} - C_A) - k_0 e^{\left(-\frac{E}{RT}\right)} C_A + w_{C_A}$$

$$[0147] \quad \frac{dT}{dt} = \frac{q}{V}(T_f - T) - \frac{\Delta H k_0}{\rho C_p} e^{\left(-\frac{E}{RT}\right)} C_A + \frac{UA}{V \rho C_p}(T_c - T) + w_T$$

[0148] 利用欧拉近似采样时间为0.3s进而离散化得到系统的离散时间模型(1)。其中 $C_A$ 是反应器的浓度, $T, T_c$ 分别是反应器和冷却剂的温度。 $W = \{w \in R^2: \|w\|_P \leq 0.0015\}$ 。控制目

标是使得反应器的浓度维持在  $C_A^e = 0.5 \text{ mol/l}$ ，反应器的温度保持  $T^e = 350 \text{ K}$ ，冷却剂温度维持  $T_C^e = 300 \text{ K}$ 。

[0149] 反应器其他的参数见下表：

[0150] 表1. 搅拌釜反应器系统参数

Symbols	Value	Symbols	Vslue
$\rho$	1000g/l	$C_p$	0.239J/gK
$\Delta H$	$-5 \times 10^4 \text{ J/mol}$	$E/R$	8750K
$k_0$	$7.2 \times 10^{10} \text{ min}^{-1}$	$U.A$	$5.4 \times 10^4 \text{ J/min K}$
$q$	100l/min	$T_f$	350K
$V$	1001	$C_{Af}$	1.0mol/l

[0152] 建立系统的代价函数 (2) 为：

$$[0153] \quad V(x(k), \hat{u}_N(k), N) = \sum_{i=0}^{N-1} l(\hat{x}(k+i|k), \hat{u}(k+i|k)) + F(\hat{x}(k+N|k))$$

[0154] 其中式中阶段代价和终端代价设置为： $l(x, u) = x^T \begin{bmatrix} 1 & 0 \\ 0 & 1/1500 \end{bmatrix} x + u^T \frac{1}{900} u$ ，离线

计算得到  $F(x) = x^T \begin{bmatrix} 116.55 & 0.2245 \\ 0.2245 & 0.6671 \end{bmatrix} x$ 。

[0155] 辅助控制律设计为： $\kappa(x) = [-105.5642 \quad -3.0915]x$ ，依据实际物理模型得到系统的控住约束集为： $U = \{T_c: 280\text{K} \leq T_c \leq 370\text{K}\}$ ，终端约束集为： $X_f = \{x: x^T P x \leq 1.4832\}$ 。初始的预测时域  $N(k_0) = 12$ ，最大预测状态和控制序列长度是  $N_{\max} = 30$ 。

[0156] 通过仿真实验表明本发明实施的方法可以自适应的调整预测时域，预测时域是递减的，并且当系统状态进入终端约束集  $X_f$  后预测时域减为0，如图3所示；另一方面触发优化问题的求解的触发间隔也是非常大的，如图3所示；整体的计算时间(算法1)与传统的模型预测控制(周期MPC)比较如图4所示，可以看出本发明的计算时间大大缩小，可以实现更为高效的计算过程。

[0157] 综上所述，本发明将优化问题的计算中导致消耗计算资源的固定预测时域和固定计算时刻改成了自适应预测时域和事件触发计算，一方面降低了计算难度另一方面降低计算频率，从而使得利用模型预测控制的计算时间大幅度减小，实现更高效的控制。

[0158] 以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

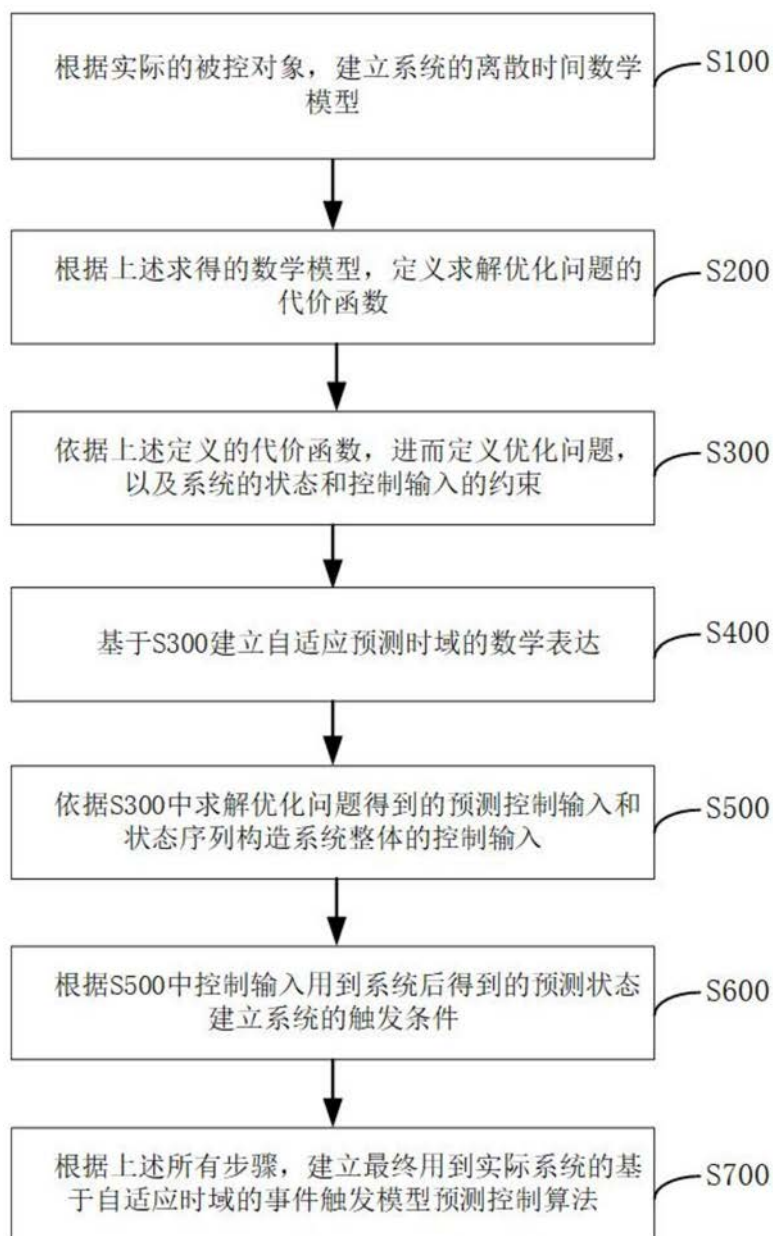


图1

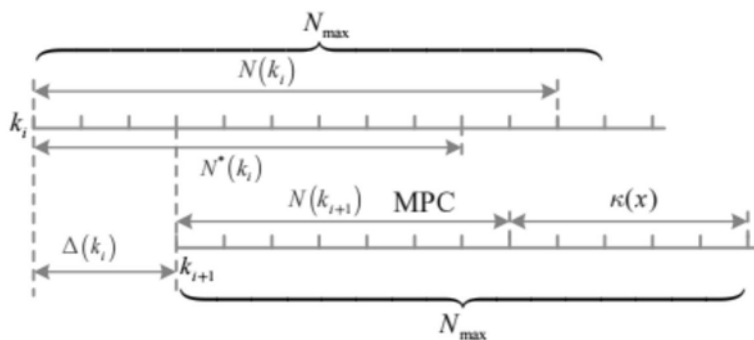


图2

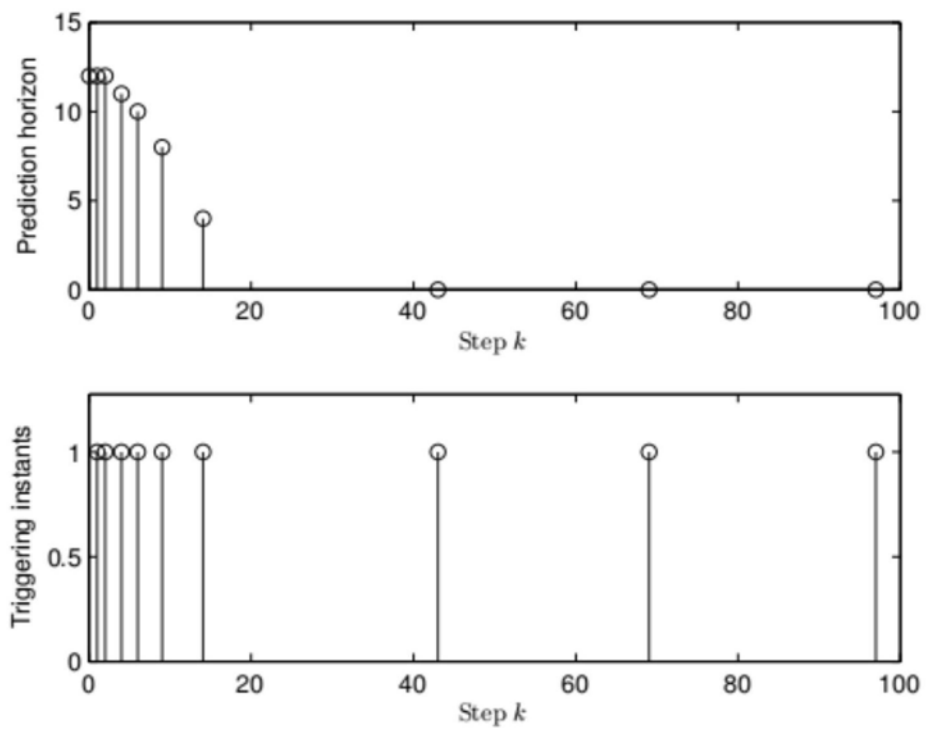


图3

	Total computing time
Periodic MPC	14.7645s
Algorithm 1	2.3499s

图4