



Asadi, E. F., & Richards, A.
(2018).受限系统的可扩展分布式模型预测控制。 *Automatica*, 93, 407-414.
<https://doi.org/10.1016/j.automatica.2018.03.050>,
<https://doi.org/10.1016/j.automatica.2018.03.050>

同行评审版本许可（如果
有的话）。
未说明

链接到发表的版本（如果有的话）。 [10.1016/j.automatica.2018.03.050](https://doi.org/10.1016/j.automatica.2018.03.050) [10.1016/j.automatica.2018.03.050](https://doi.org/10.1016/j.automatica.2018.03.050)

[链接到 "探索布里斯托尔研究
"PDF文件中的出版记录](#)

这是作者接受的手稿（AAM）。最终的出版版本（记录版本）可通过ELSEVIER在线获得，网址是：
<https://www.sciencedirect.com/science/article/pii/S0005109818301377?via%3Dihub>。请参考出版商的任何适用的使用条款。

布里斯托尔大学 - 探索布里斯托尔研究

一般权利

本文件是根据出版商的政策提供的。请仅使用上述参考文献引用已出版的版本。完整的使用条款见：<http://www.bristol.ac.uk/pure/about/ebr-terms>

受限系统的可扩展分布式模型预测控制

Fatemeh Asadi, Arthur Richards

英国布里斯托尔大学皇后楼航空航天工程系, 布里斯托尔, 英国

摘要

为通过约束条件耦合的子系统提出了一个分布式模型预测控制策略。自组织的时分多址被用来协调子系统控制器的顺序, 这样就不会有两个同时重新优化。这种新方法不需要中央协调或预先组织的优化序列。该方案保证了耦合约束的满足, 尽管子系统的动态进入和退出。

关键字。分布式模型预测控制; 受限系统。

1 简介

由于计算规模和协调要求, 以集中的方式控制大规模系统, 如运输网或配电网, 往往是困难的。集中式控制也容易出现单点故障, 促使人们对分布式控制系统产生兴趣。模型预测控制 (MPC) 是一种将约束优化与反馈控制相结合的控制技术 [Maciejowski, 2002, Gha and Pannek, 2011], Scattolini [2009], Christofides 等人 [2013], Negeborn 和 Maestre [2014] 等人已经讨论了分布式 MPC (DMPC) 的方案。本文重点讨论共享有限资源的子系统的 DMPC, 它通过约束条件来耦合系统 [Keviczky 等人, 2006, Kuwata 等人, 2007a, Miler 等人, 2012, Bourdais 等人, 2014, Tedesco 等人, 2014, Lucia 等人, 2015, Li 等人, 2016]。

。这里没有考虑的其他形式的耦合是通过动力学 [Dunbar, 2007, Alessio et al., 2011, Farina and Scattolini, 2012, Hernandez and Trodden, 2016] 或通过整个系统的目标函数 [Borrelli and Keviczky, 2006, Dunbar and Murray, 2006, Wang and Ong, 2010]。

本文采用了一个串行 DMPC 方案, 其中一次只有一个子系统控制器可以优化其计划。由于其他子系统的计划是固定的, 并且通过通信知道, 整个系统的可行性是这样的

确保的。以前关于序列计划的工作已经证明了它的特性, 但前提是假定子系统有一个商定的升级顺序 [Richards 和 How, 2007, Keviczky 等人, 2004b, a, Kuwata 等人, 2007a, Trodden 和 Richards, 2013, Dai 等人, 2015]。

然而, 该序列的确定是一个集中的过程。本文的简单贡献是加入了分布式时隙分配过程, 其灵感来自于通信系统中的多接入信道 (MAC) 共享方法 [Rom and Sidi, 2012]。特别是采用了自组织时分多址 (STDMA) [Gaugel 等人, 2013], 但它不是为通信分配传输槽, 而是为重新规划分配优化槽。

分布式排序 (槽位分配) 的目标与 "即插即用 (PnP) 控制"

"的目标在精神上是一致的, 后者旨在处理子系统数量变化的分布式控制问题。在 PnP 控制中, 通过增加或删除一个子系统, 只需要重新设计被控制的子系统和受其影响的子系统 (邻居) 的本地控制器 [Stoustrup, 2009]。Riverso 等人 [2014] 和 Zeilinger 等人 [2013] 提出的 PnP 控制方法解决了在闭环运行过程中由于增加或重新移动一个或多个子系统而自动适应系统模型不断变化的问题。在这两项工作中, 子系统是物理耦合的, 而在本文中, 子系统是通过其约束条件耦合的。Barreiro-Gomez 等人 [2015] 和 Lucia 等人 [2015] 解决了由于加入和离开具有耦合约束的子系统而执行网络变化的挑战。去除

电子邮件地址: elham.asadi@bristol.ac.uk (Fatemeh Asadi), arthur.richards@bristol.ac.uk (Arthur Richards)。

Barreiro-

Gomez等人[2015]提出的集中式MPC方案只能处理控制信号上的一个单一耦合约束。Lucia等人[2015]提出的基于合同的DMPC通过传输未来可能的轨迹序列保证了并行优化中的约束条件满足。本文提出的方法考虑了与共享有限资源相关的特殊耦合约束。同样与Lucia等人[2015]不同的是，我们的子系统传递精确的轨迹，但采用串行（一次一个）的优化方式，它们实现了分散的排序方式。

2 自组织的测序

考虑一个动态的子系统集合 $P(k)$ 在每个时间步长 k 包含 $n(k)$ 成员。这些子系统的一个子集 $P_c(k) \subseteq P(k)$ 处于共同运作模式，使用串行DMPC来协调他们的

行动，从而使共享资源的限制得到尊重。让 $n_c(k) = |P_c(k)|$ 表示合作子系统的数量。其余的 $P(k) \setminus P_c(k)$ 保持在一个受限的安全模式

资源，因此不需要进行通信。审慎的DMPC要求将子系统唯一地分配给时间步骤，这样，每一个步骤最多关联到

一个子系统， $p_k \in P_c(k) \cup \{0\}$ 。在每一步，位于 a_l 的子系统 p_k （如果有， $p_k \neq 0$ ）解决其局部最优控制问题，并分享所得到的与其他的意图。本节描述了如何以一种动态的、自组织的方式实现分配，使子系统从安全模式进入共同运行模式。由于这个问题类似于通信中的时隙分配，该算法是基于STDMA，这是一种分散的MAC方法。

定义 L_f 为帧长，即时隙分配的重复周期，这样 $p_k = p_{k+L_f}$ ，只要 p_k 保持合作。然后，算法1提出了一个代理的自组织排序的程序

$q \in P(k) \cap P_c(k)$ 希望进入合作模式。由于分配是周期性的，进入只涉及聆听一帧，然后在下一帧中选择一个可用的槽。可能出现的问题是“锁死”，即两个子系统试图在同一步骤中进入，每个都不知道另一个的存在。这一事件被两个子系统检测到，并采用随机的回退时间来避免死锁。

离开合作是通过停止传输任务来实现的，向其他人表明该槽位又可以使用了。因此，与通信情况下槽位分配的有限寿命不同，一个槽位无限期地属于一个子系统，直到该子系统放弃它。与进入和离开相关的控制约束将在第4节中描述。

假设1（帧长度） 帧长度 L_f

算法1 进入合作模式

要求。 子系统ID q ，初始时间 k_1

- 1: 听取 L_f 步，以确定 $\{p_{k_1}, \dots, p_{k_1+L_f}\}$
- 2: 确定空位的偏移量。 $J = \{j \in [0, \dots, L_f] \mid p_{k_1+j} = 0\}$
- 3: 如果没有空位， $J_{free} = \emptyset$ ，则
- 4: 再试一次：转到步骤1
- 5: 否则
- 6: 随机选择空位， $j \in J_{free}$
- 7: 等待下一帧中的插槽 j ， $k = k_1 + L_f + j$
- 8: 发送当前计划 $Y^*(k_1 + L_f + j)$
- 9: 如果没有其他子系统传送，那么
- 10: 保证 $p_{k_1+j} = q \forall n=1, 2, \dots$ 只要 $q \in P_c(k)$
- 11: 返回成功
- 12: 其他
- 13: 碰撞：等待随机的步骤数
- 14: 再试一次：转到步骤1
- 15: 结束，如果
- 16: 结束 如果

是所有子系统控制器都知道的。这构成了子系统通用接口的一部分：接口是标准的，所有代理都知道，这是可扩展性概念的核心。

备注1 由于框架是周期性的[Rom and Sidi, 2012]，所有子系统的连接者没有必要定义框架的共同相位。

备注2 由于不超过 L_f 子系统可以有插槽，那么框架长度 L_f 代表了合作模式下子系统数量的限制： $n_c(k) \leq L_f$ 。因此， L_f 的选择代表了一个重要的去标志选择，因为根据算法1，增加 L_f 意味着为进入的代理提供更多的容量，但进入的等待时间更长。对这种权衡的全面研究超出了本文的范围，读者可参阅Asadi和Richards[2015]进行了更多考虑。

3 控制问题的定义

每个子系统 $p \in P(k)$ 都有自己的动力学。

$$\mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)) \quad k \in \mathbb{N}, \quad \forall p \in P(1)$$
其中 $\mathbf{x}_p \in \mathbb{R}^N$ 和 $\mathbf{u}_p \in \mathbb{R}^N$
 \mathbf{x}_p 和 \mathbf{u}_p 分别是子系统 p 的状态向量和控制输入向量。

备注3 动力学 (1) 不受任何不确定因素的影响。像管子方法[Trodden and Richards, 2010]这样的想法可以用来处理扰动，但为了简单起见，这里省略了这些。

每个子系统 p 都受到状态和输入的局部约束

$$\mathbf{x}_p(k) \in X_p \quad (2)$$

$$\mathbf{u}_p(k) \in U_p \quad (3)$$

并有自己的局部目标函数，以固定水平的MPC形式存在

$$J_p = \sum_{t=0}^{N-1} l_p(\mathbf{x}_p(k+t|k), \mathbf{u}_p(k+t|k), k) + V_p(\mathbf{x}_p(k+N|k), k) \quad (4)$$

其中， N 是预测范围内的时间步数， $l_p: \mathbb{R}^{n_p} \times \mathbb{R}^{m_p} \times \mathbb{R} \rightarrow \mathbb{R}_{0+}$ 表示阶段成本， V_p 是终端成本，双下标符号 $(k+t|k)$ 表示从时间 k 开始提前 t 步预测变量。本文的重点是

准则，因此，成本函数的性质将不会被更详细地说明。然而，读者应该注意到，时间的变化允许在不同的模式下使用不同的成本，这一点将在后面得到利用。

定义一组共享资源 L ，让 $y_p^f \in \mathbb{R}$ 为子系统 p 对特定资源 $f \in L$ 的用量。

$$y_p^f(k) = g_p^f(\mathbf{x}_p(k), \mathbf{u}_p(k)) \quad (5)$$

所有子系统共享有限的资源 L ，因此它们的输出通过以下方式耦合在一起

$$\sum_{p=1}^n y_p^f(k) \leq 1, \forall f \in L. \quad (6)$$

注意每个资源的比例是正好有一个单位的资源。定义 $\mathbf{y}_p(k)$ 为所有资源使用产出的组合。

$$\mathbf{y}_p(k) := \{y_p^f(k)\}_{f \in L}$$

这个符号是用来使资源的集合能够被

连续的，这将出现在即将到来的例子中。在一个简单的离散资源集的情况下， \mathbf{y} 将只是一个矢量。典型的矢量符号将被用在后续， $\mathbf{y}_1 = \mathbf{y}_2$ ，意味着 $y_1^f = y_2^f \forall f \in L$ 和 $\mathbf{y}_1 \leq \mathbf{y}_2$ 意味着 $y_1^f \leq y_2^f \forall f \in L$ 。同时定义 C_n 为 n 个子系统的可行输出的集合。

$$C_n := \left\{ (\mathbf{y}_1, \dots, \mathbf{y}_n) \mid \sum_{p=1}^n \mathbf{y}_p \leq \mathbf{1} \right\} \quad (7)$$

第2节中的插槽分配过程确保

每个时间步骤 k 都被唯一地分配给一个子系统 $p_k \in P_C \cup \{0\}$ 。后面的章节将说明这如何确保约束的满足，然而，这个过程需要一定的时间，有必要在此基础上再做

考虑到子系统已经进入难题但尚未获得一个空位，因此无法与其他代理协调其计划的情况。为此，在子系统不能使用共享资源的情况下，对于进入（和离开）的系统需要一种安全的操作模式。这种模式要求存在一个资源输出等于零的不变集。

命题1 安全模式的可行性

$$C_n \times \mathbf{0}^m \subseteq C_{n+m} \quad \forall m, n \quad (8)$$

证明很简单：加零不会改变输出，所以如果前 n 个满足上限，那么所有 $n+m$ 也必须满足。

正如MPC常见的那样，递归可行性将取决于不变集约束。定义 $F_p(Y_p)$ 为满足输出集约束 Y_p 的子系统 p 的状态内变集。

$$\forall \mathbf{x}_p \in F_p(Y_p) \exists \kappa_p(\mathbf{x}_p) \in U_p. \quad (9)$$

$$f_p(\mathbf{x}_p, \kappa_p(\mathbf{x}_p)) \in F_p(Y_p) \quad (10)$$

$$\mathbf{y}_p(\mathbf{x}_p, \kappa_p(\mathbf{x}_p)) \in Y_p$$

(11) 这些集合 F_p 的选择由设计者决定。可以采用最大不变量集[?]或简单的平衡条件。

假设2 (安全模式不变性) 一个不变的集合存在满足安全模式条件的蚂蚁集。

$$F_p(\{0\}) \neq \emptyset.$$

例1 (电源插座) 对于一组总输出功率限制在 P_{\max} 的电源插座，只形成一个共享资源 $L = \{1\}$ ，并定义一个单一的协同输出 y^1 ，作为每个连接的子系统吸取的可用功率比例。

$$g_p^1(\mathbf{x}_p(k), \mathbf{u}_p(k)) = \frac{P_p(\mathbf{x}_p(k), \mathbf{u}_p(k))}{P_{\max}} \quad \text{最大限度}$$

其中 $P_p(\mathbf{x}_p(k), \mathbf{u}_p(k))$ 是子系统 p 所消耗的功率，取决于其动态状态和输入。

例2 (空中交通) 考虑一组飞机

在 $A \subseteq \mathbb{R}^2$

的空域内以相同的高度飞行。在该空域内，飞机被要求保持最小间隔 R_{\min}

。（其他空域控制器，这里不考虑，提供 A 外的间隔）。

空域中的每个点都被认为是一个共享资源，并且如果没有更多的

多于一架飞机出现在每个点的 R_{\min} 内。因此，资源集 $L=A$ ，协调输出为

$$g_p^f = \begin{cases} 1 & \text{if } \mathbf{r}_p(k) - \mathbf{r}(x) \leq R_{\min} \wedge \mathbf{r}(x) \in A \\ 0 & \text{否则} \end{cases}$$

其中 $\mathbf{r}_p(k)$ 代表飞机 p 的位置。安全模式相当于 $\mathbf{r}_p(k) \in A$ 。对于固定翼飞机，在与所有被占领的空域安全分离的情况下盘旋飞行提供了一个合适的不变集 $F(Y)$ [Kuwata等人, 2007b]。

4 可扩展的DMPC

4.1 概述

在与控制系统的一次互动中，子系统 q 的“生命周期”如下。

1. 不活跃： q 不是系统的一部分，并且 $q \notin P(k)$
2. 安全： $q \in P_C(k) \in P_C$
(k)，即活跃但尚未共同运作。 q 受到约束，使 $\mathbf{y}_q(k) = \mathbf{o}$ 。AI-遵循公式1，直到插槽 $p_k = q$ 得到保障。
3. 合作： $q \in P_C$
(k)，通信被用来维持整个系统的可行性
4. Leaving： $q \in P_C(k)$ and communication ongoing.
 q 回到安全模式 $\mathbf{y}_q(k) \rightarrow \mathbf{o}$
5. 安全： $q \in P_C(k) \in P_C$
(k)，即活跃但不再共同运作。 q 受限，使 $\mathbf{y}_q(k) = \mathbf{o}$
6. 不活跃： q 不是系统的一部分，并且 $q \notin P(k)$ 。

备注4 注意，没有必要将“进入”模式作为“离开”模式的对应物。这是因为所有处于安全模式的轨迹都满足合作模式的要求，所以从安全模式过渡到合作模式无需等待，只要有一个空位就可以实现。然而，反之亦然，并不是所有可行的合作轨迹都处于安全模式，所以需要有一个过渡阶段来从合作到安全。这个过渡阶段就是“离开”模式。

假设 3 (模式转换时间) 每个子系统都有一些方法来决定何时进入和离开合作。

备注5 模式改变的时间决定不是控制协议本身的一部分。对于电源的例子，启用可能是由一个命令触发的，而离开则是由一定时期的零消耗触发的。对于空中交通，启动和离开将由相对于扇区边界的位置来触发。对于其余的，所有的子系统都被假定能够进入（进入第二阶段）或任何时候都可以离开（进入第五阶段）。

4.2 MPC 优化

每一个子项目都有不同类型的约束。系统，取决于其运行模式。幸运的是，MPC的相同表述可以用于所有这些模式。
MPC问题取决于初始子系统状态 \mathbf{x}_p 和一组输出约束 $\mathbf{Y}_p(k) =$

$\mathbf{Y}_p(k|k) \times \mathbf{Y}_p(k+1|k) \times \dots \times \mathbf{Y}_p(k+N|k)$ 取决于的运作模式和通讯方式。
的子系统。它优化一连串的未来条件 $\mathbf{U}_p(k) = (\mathbf{u}_p(k|k), \mathbf{u}_p(k+1|k), \dots, \mathbf{u}_p(k+N-1|k))$ ，每个子系统的一般约束集为 p 被定义为。

受制于 $\forall t \in \{0, \dots, N-1\}$ 。

$$\mathbf{x}_p(k|k) = \mathbf{x}_p(k) \quad (12b)$$

$$\mathbf{x}_p(k+t+1|k) = f_p(\mathbf{x}_p(k+t|k), \mathbf{u}_p(k+t|k)) \quad (12c)$$

$$\mathbf{y}_p(k+t|k) = g_p(\mathbf{x}_p(k+t|k), \mathbf{u}_p(k+t|k)) \quad (12d)$$

$$\mathbf{x}_p(k+t|k) \in X_p \quad (12e)$$

$$\mathbf{u}_p(k+t|k) \in U_p \quad (12f)$$

$$\mathbf{y}_p(k+t|k) \in Y_p(k+t|k) \quad (12g)$$

$$\mathbf{x}_p(k+N|k) \in F_p(Y_p(k+N|k)) \quad (12h)$$

为方便起见，我们还定义了输出序列。

$$\mathbf{Y}_p(k) = (\mathbf{y}_p(k|k), \mathbf{y}_p(k+1|k), \dots, \mathbf{y}_p(k+N|k)) \quad (13)$$

并注意 $\mathbf{Y}_p(k) \in Y(k)$ 等同于 $\mathbf{y}_p(k+t|k) \in Y_p(k+t|k) \forall t \in \{0, \dots, N\}$ 。请注意，这些约束包括终端步骤输出 $t = N$ ，尽管这是

在优化过程中没有明确形成。对终端输出的处理将在后面的第4.6节中解释。在合作模式下，正是这些输出序列将在子系统之间进行交流。

4.3 安全模式的限制条件

安全模式要求 $\mathbf{y}_p(k) = \mathbf{o}$ ，与其他子系统解耦，所以需要的约束是 $Y_p(k) = \{\mathbf{o}\}^{N+1}$ ，因为

$$\mathbf{Y}_p(k) \in \mathbf{o}^{N+1} \Leftrightarrow \mathbf{y}_p(k+t|k) = \mathbf{o} \forall t \in \{0, \dots, N\} \quad (14)$$

4.4 进入和合作模式的制约因素

定义 $P_{\{-p\}}(k)$ 为子系统 p 在时间实例 k 的邻居集合，定义为那些与 p 合作的子系统， $P_{\{-p\}}(k) := P_C(k) \setminus p = \{q_{p,1}, \dots, q_{p,n_C(k)}\}$ 。请注意，只有当 $p \in P_C(k)$ 时才使用邻居的定义。那么在合作模式下，子系统 p 的输出的约束条件是 $Y_p(k) = C_p(Y_{\{-p\}}(k))$ 其中 $Y_{\{-p\}}(k) = \{\mathbf{y}_q(k)\}_{q \in P_{\{-p\}}(k)}$ 是输出序列的集合为 p 的所有邻居，通过通信共享，以及

$$C_p(Y_{\{-p\}}) := \left\{ \begin{array}{l} \mathbf{y}_p(k+t|k), \mathbf{y}_{q_{p,1}}(k+t|k), \dots, \mathbf{y}_{q_{p,n_C(k)-1}}(k+t|k) \\ \in C_{n_C(k)} \forall t \in \{0, \dots, N\} \end{array} \right\} \quad (15)$$

鉴于耦合约束的形式(7)，这就是相当于

$$C_p(Y_{\{-p\}}) := \left\{ \begin{array}{l} \mathbf{y}_p(k+t|k) \leq \mathbf{1} - \bigwedge_{q \in P_{\{-p\}}} \mathbf{y}_q(k+t|k) \forall t \in \{0, \dots, N\} \end{array} \right\} \in \mathbb{R}^{N_y(N+1)} \quad (16)$$

同时定义 C 为 n 个可行输出序列的所有组合的集合。

$$\begin{aligned} \mathbf{P}_p(\mathbf{x}_p(k), \tilde{\mathbf{y}}_p(k)) &: J_p & \mathbb{G}_c^n(k) &:= \mathbb{E}^{RnNy(N+1)} \\ \min_{\mathbf{x}_p(k)} & & \{\{\mathbf{y}_p\}_{p \in \mathcal{P}}\} & \end{aligned}$$

$$(y_1(k+t|k), \dots, y_{n_c}(k+t|k)) \in_{C_{nc}(k)} \forall t \in \{0, \dots, N\} \quad (17)$$

那么根据构造

$$\{Y_p\}_p \in P_{c(k)} \in_{C_{nc}(k)} \Leftrightarrow Y_q(k) \in C_q(Y_{\{-q\}}(k)) \forall q \in P_C(k) \quad (18)$$

4.5 离开模式的制约因素

当离开合作模式时，终端状态被约束为满足安全的要求。

模式 $y_p(k+N|k) = \mathbf{o}$ ，这样子系统状态

转换成满足安全模式， $y_p(k) \rightarrow \mathbf{o}$ 。然而它必须确保对邻居的可行性，然后才是它通过考虑公布的计划达到零产出。

因此，对离开模式的约束是由对合作模式的约束的附加限制给出的。

$$L_p(\{p\}) := C_p(\{p\}) \cap N^u \times \{0\}. \quad (19)$$

$$(Y \quad (Y$$

假设4（离开模式的收敛性） 如果

$P_p(\mathbf{x}_p(k), L_p(Y_{\{-p\}}))$ 是递归可行的，那么 $y_p(k) \rightarrow \mathbf{o}$ 。这有赖于成本条款的适当构建

(4) 和标准的MPC结果[Mayne等人, 2000]。请注意，成本被允许是随时间变化的，因此可以接受在离开模式中使用与其他模式不同的成本，而且这个假设只影响在离开模式中使用的成本。

4.6 沟通的输出序列

在时间步骤 k ，被分配的子系统 p_k

（如果有的话）解决其优化问题并传达其输出序 (13)。条款 $(y_p(k|k), \dots, y_p(k+N-1|k))$ 是通过约束条件 (12d) 可立即获得。该

最后一个术语被用来表达在任何后续步骤中预期消耗的最大资源水平，而不是不变性控制法。

$$y_p(k+N|k) := \max_{t \geq N} g_p(\mathbf{x}_p(k+t|k), \kappa_p(\mathbf{x}_p(k+t|k))) \quad (20)$$

其中，最大是以元素方式进行的，并且 $\forall t \geq N$ 。

$\mathbf{x}_p(k+t+1|k) = \mathbf{f}_p(\mathbf{x}_p(k+t|k), \kappa_p(\mathbf{x}_p(k+t|k)))$ 通过(10)中终端约束(12h)中采用的不变集 $F_p(Y_p)$ 的构造，可以知道 $y_p(k+N|k) \in Y_p(k+N|k)$ ，因此必须是有限的。

4.7 尾部解决方案的构建

在受限的MPC中，“尾部解决方案”是很常见的。是前一个控制序列的延续。

的时间步骤，在可行性证明中发挥作用[Mayne

在(12)中的MPC最优控制问题中，尾巴是被控制的。

结构如下。

$$\hat{U}(U^*(k-1)) = \begin{pmatrix} \mathbf{u}_p^*(k|k-1), \dots, \mathbf{u}_p^*(k+N-2|k-1), \\ \kappa_p(\mathbf{x}^*(k-1+N|k-1)) \end{pmatrix} \quad (21)$$

其中上标 $*$

表示来自前述时间步骤的结果，或者来自优化，或者来自前面的尾部。这是由以下的输出尾巴序列所限定的

$$\hat{Y}_p(Y_p(k-1)) = \begin{pmatrix} y_p(k|k-1), \dots, y_p(k+N-2|k-1), \\ g_p(\mathbf{x}_p(k+N-1|k-1), \kappa_p(\mathbf{x}_p(k+N-1|k-1)) \end{pmatrix} \quad (22)$$

备注6 根据条件 (9)，(10) 和 (11)。状态序列在一个控制不变集上结束，满足——。

ing 输出约束。这使得子系统能够预测他们的邻居的意图。

4.8 可扩展的DMPC算法

算法2根据上述定义，完整地介绍了拟议的可扩展的DMPC方法。它涵盖了第4.1节中描述的生命周期的第2-

5个阶段：第1和第6个阶段被省略了，因为不活动的子系统什么都不做，如果它们存在的话。算法2还编码了分布式控制方案的基本“规则”。

- 一个子系统被限制在安全模式 $y_p \in \{0\}$ ，除非它与其他子系统合作，利用分配的时间分享它的意图 y_p 。
- 一个子系统在离开安全模式之前，不得离开安全模式。争取到了一个被分配的名额。
- 在返回安全模式之前，一个子系统可能不会停止合作并重新获得其分配的插槽。
- 一个合作的子系统被限制在它的尾部——。顺序，除非当前的时间步数是其分配的槽。

在下文中， $*$

表示采用的控制或输入，这些控制或输入可以通过优化或根据算法2使用尾巴来确定，并在子系统之间进行交流。

递归可行性将在下一节中考虑。然而，请注意，不可能预先保证优化的可行性。

模式 $P_q(\mathbf{x}_q(k), L_q(Y_{\{-q\}}^*(k)))$ ，因为它的约束条件比那些合作模式更严格。

$$L_q(\{q\}(k)) \subseteq C_q(Y^*(k)), \quad (23)$$

等人，2000]。在串行的DMPC中，如果没有轮到某些子系统，尾巴就会被明确地使用。

quence $\{p_k\}$ 。这可以被视为一种极端形式的基于合同的DMPC[Lucia等人，2015]，其中合同被简化为一个单一的控制序列。对于

根据 (19)。因此，该算法检查离开模式的可行性，并在不可行的情况下恢复到合作模式问题。

算法2 在时间 k 更新子系统 q 的计划

要求。槽位分配 $\{p_k\}$
 邻居的计划 $Y_{\{-q\}}^*$ ($k-1$)。
 q 的模式 $\{\text{安全、进入、合作、离开}\}$

- 1: 确定状态 $\mathbf{x}_q(k)$
- 2: 如果 q 的模式是"安全", 那么
- 3: 解决 $P_q(\mathbf{x}_q(k), \{0\})^{N+1}$
- 4: 如果 Q 想"进入"这个"合作", 那么
- 5: 按照算法1的下一步进行
- 6: 如果算法1返回成功, 那么
- 7: 将 q 的模式设置为"合作".
- 8: 结束, 如果
- 9: 结束, 如果
- 10: 否则
- 11: 如果 $p_{k-1} \neq 0$, 那么
- 12: 接收邻居更新 $Y_{\{-q\}}^*(k-1)$ 。
- 13: 结束, 如果
- 14: 使用尾巴更新邻居的计划 $Y_{\{-q\}}^*(k)$ 。
 $Y_p^*(k) \leftarrow \hat{Y}(Y^*(k-1)) \forall p \in P_{\{-q\}}(k-1)$
- 15: 如果 $p_k \neq q$ 那么
- 16: 从尾部获得自己的计划 $U^*(k) = U(U^*(k-1))$
- 17: 否则
- 18: 如果 Q 的模式是"进入"或"合作"
- 然后
- 19: 解决 $P_q(\mathbf{x}_q(k), \{0\})^{N+1}$
- 20: 将 $Y^*(k)$ 传递给邻居 $P_{\{-q\}}(k)$ 。
- 21: 否则
- 22: 尝试解决 $P_q(\mathbf{x}_q(k), \{0\})^{N+1}$
- 23: 如果可行, 那么
- 24: 如果 $Y^*(k) \in \{0\}$, 那么 $N+1$ 和 $\mathbf{x}_q^*(k+1)$
- 25: 将 q 的模式设置为
- 26: 否则
- 27: 保持插槽直到回到安全地带
- 28: 结束,
- 29: 否则
- 30: 解决 $P_q(\mathbf{x}_q(k), \{0\})^{N+1}$
- 31: 传送 $Y^*(k)$ 到 $P_{\{-q\}}(k)$
- 32: 结束, 如果
- 33: 结束, 如果
- 34: 结束, 如果
- 35: 结束 如果
- 36: 应用控制 $\mathbf{u}_q(k) = \mathbf{u}^*(k|k)$

5 可扩展的DMPC的可行性

在整个系统范围内, 算法在每个时间步长的进展可以分为四个重要的方面。

罐子的情

4. $p_k \neq 0$, 子系统 p_k 重新优化并离开合作。

同时, 任何数量的子系统可以保持在安全模式, 或从非活动状态进入安全模式, 或离开安全模式成为非活动状态。请注意, 如果两个或更多的子系统试图在同一个空槽 p_k , 根据算法1, 这将被检测为碰撞, 它们都无法进入。本节介绍了整个DMPC方案的递归可行性和约束条件的满足。稳定性超出了本文的讨论范围。证明将以递归为基础, 从一个常见的常设假设开始。

假设5 在时间 k_0 , 所有子系统都有可行的

计划, 其中 $\{Y^*(k_0) \in \mathcal{C}_n(k_0) \text{ 和 } Y^*(k_0) \in \{0\}^{N+1} \forall p \in P(k_0) \setminus P_C(k_0)\}$ 以及满足其局部优化约束的控制序列 $U^*(k_0)_{P_p}(\mathbf{x}_p(k_0), -)$ 。

命题2 (全系统的可行性在 k)。

所有子系统的组合, 在安全, 进入, 合作和离开模式, 满足

$$\{Y_p^*\}_{p \in P(k_0)} \in \mathcal{C}_n(k_0) \quad (24)$$

证明

请注意, 根据(23), 进入模式的约束与合作模式相同, 离开模式的约束比合作模式的约束更严格。因此, 可以假设进入和离开模式的子系统都满足合作模式的约束。

的模式。那么假设5意味着 $\forall t \in \{0, \dots, N\}$ 。

$$(\mathbf{y}^*(k_0 + t|k_0), \dots, \mathbf{y}^*(k_0 + t|k_0)) \in \mathcal{C}_{n(k_0)} \quad (25)$$

和 $\mathbf{y}^*(k_0 + t|k_0) \in \{0\} \forall p \in P(k) \setminus P_C(k)$ 所以由对安全模式的要求(8), 可以看出, $\forall t \in \{0, \dots, N\} : (\mathbf{y}^*(k_0 + t|k_0), \dots, \mathbf{y}^*(k_0 + t|k_0)) \in \mathcal{C}_{n(k_0)}$

这等同于(24)。□

命题3 (安全模式下的递归可行性) 对于安全模式下的任何子系统 $p \in P(k) \setminus P_C(k)$, 优化 $P_p(\mathbf{x}_p(k_0 + 1), \{0\}^{N+1})$ 是可行的。

证明。根据(14), 安全模式是标准的MPC, 具有不变的终端条件和不变的约束。尾部解决方案(21)是可行的, 确保了优化的可行性。□

命题4 (情况1: 没有更新或进入) 如果

$p_k = 0, P_C(k_0 + 1) = P_C(k_0)$, 则

$$\{Y^*(k_0 + 1)\}_{p \in P(k_0 + 1)} \in \mathcal{C}_n(k_0 + 1) \quad (26)$$

和

1. $p_k = 0$, 没有子系统进入
2. $p_k = 0$, 一个子系统利用机会进入合作
3. $p_k \neq 0$, 子系统 p_k 重新优化并保持
在合作中

$$Y_p^*(k_0 + 1) \in \{0\}^{N+1} \quad \forall p \in P(k_0 + 1) \not\in P_C(k_0 + 1) \quad (27)$$

证明。在这种情况下，所有合作的子系统都采用它们的尾部解决方案给出了 $Y_p^*(k_0 + 1) = Y_p(Y_p^*(k_0)) \quad \forall p \in \mathcal{C}$ 。

$$\begin{aligned} & (\mathbf{y}_1^*(k_0 + t|\mathbf{k}_0 + 1), \dots, \mathbf{y}_{n_c}^*(k_0 + t|\mathbf{k}_0 + 1)) = \\ & (\mathbf{y}_1^*(k_0 + t|\mathbf{k}_0), \dots, \mathbf{y}_{n_c}^*(k_0 + t|\mathbf{k}_0))_{\circ} \quad t = \\ & \underset{j_o}{1 \dots N} \qquad n_c(k) \end{aligned} \tag{28a}$$

$$(\mathbf{y}_1^*(k_0 + N + 1 | k_0 + 1), \dots, \mathbf{y}_{n_c(k)}^*(k_0 + N + 1 | k_0 + 1))_{j_0} \\ (g_1^{j_0}(\mathbf{x}^*(k_0 + N | k_0), \kappa_1(\mathbf{x}^*(k_0 + N | k_0))), \dots, g_{n_c(k)}^{j_0}(\mathbf{x}^*(k_0 + N | k_0), \kappa_{n_c(k)}(\mathbf{x}^*(k_0 + N | k_0))))_{j_0} \\ (\mathbf{c}_0)_{j_0} \quad n_c(k) \quad (\mathbf{c}_0)_{j_0} \quad n_c(k) \quad (28b)$$

命题5 (情况2: 子系统进入) 如果 $p_k = 0$, 一个新的子系统进入 $P_C(k_0 + 1) = P_C(k_0) \cup \{q\}$ 。

$$\{\gamma_p^*(k_0 + 1)\}_{p \in P_C(k_0 + 1)} \in \mathcal{C}_{NC}(k_0 + 1) \quad (29)$$

由于已经合作的子系统将采用它们的尾部解决方案，因此，从以下几点可以看出命题4认为 $\{Y^*(k_0 + 1)\}_{p \in P}^{(k)} \in \mathcal{G}_{C_0}^{(k)}$ 即，这些子系统所采用的计划仍然是相互可行的。此外，在这一过程中(算法1的第8步)的计划必须满足 $Y^*(k_0 + 1) \in C_q(Y_{\{-q\}}(k_0 + 1))$ 。由于 q 只能从安全模式进入， $Y^*(k_0) \in \{O\}^{N+1}$ 。所以， q 的尾解是 $\hat{Y}(Y^*(k_0)) \in \{O\}_a^{N+1}$ ，其中

满足 $(Y^*(k_0+1), \{Y^*(k_0+1)\}_{k \in C_0}) \in_{n(k)+1} C_g(Y_{C_0}, (k_0+1))$ 。因此, Fol-
按照命题2相同的推理, 可以得到

$$\bigcup_{q \in P} \{Y^*(k_0+1), \{Y^*(k_0+1)\}_{k \in C_0}\} \in_n (k)+1, \text{ 其中}$$

$$P = \{C_0, C_0\}$$
 等同于 (29)。(30)对于保持在安全模式下的子系统, 从命题3得出。□

pk0 = q / = 0, 则 $P_q(x_q(k_0 + \{-q\}(k_0 + 1)))$ 是 1), $C_q(Y^*)$ 是可行的, 且结果满足

$$\{Y_p^*(k_0 + 1)\}_{p \in P} \in \mathcal{C}_0^{(k+1)} \quad (31)$$

和

$$Y_p^*(k_0 + 1) \in \{0\}^{N+1} \quad \forall p \in P(k_0 + 1) \not\in P_C(k_0 + 1) \quad (32)$$

构成满足全系统的约束条件 C_q 和
从(18)可以看出, $Y(Y^*(k_0+1)) \in C_q(Y^*(k_0+1))$
因为 $Y_{\{-q\}}^*(k_0+1)$ 也必须从re-

$$\{Y^*(k_0 + 1)\}_{p \in P} \quad (k+1) \in \mathfrak{N} \quad (k+1) \quad (33)$$

$$y_p^*(k_0+1) \in \{0\}^{N+1} \quad \forall p \in P(k_0+1) \not\in P_C(k_0+1) \quad (34)$$

可行的产出也将是相互可行的。因此。
 $\{Y_p^*(k_0)\}_{p \in P_C(k)/q} \in \mathcal{C}_{C(k)-1}$ 。那么，从提议-
 根据第4条的规定，所有的分项都采用尾数。
 系统 $p \in P_C(k_0)/q$ 给出 $\{Y_p^*(k_0+1)\}_{p \in P_C(k_0)/q} \in \mathcal{C}_0$
 这相当于 (33)。根据算法2的第22步，子系统 q
 只能如果 $Y^*(k_0+1) \in \{0\}^{N+1}$ 和 $Y^*(k_0+1) \in \{0\}$
 则离开。
 $\{0\}^{N+1} \forall p \in P(k_0) \nexists p \in P_C(k_0)$ 从命题3可以看出
 因此(34)得到满足，因为 $P(k_0+1) \nexists P_C(k_0+1) = (P(k_0) \nexists P_C(k_0)) \cup q$ 。

$$\{Y_p^*(k)\}_{p \in P(k)} \in \mathcal{C}_{\mathbf{p}(k)} \quad (35)$$

$$\{Y_p^*(k)\}_{p \in P(k)} \in \mathbb{C}^{n_k} \quad (35)$$

证明。命题4-
7表明, 条件(33)和(34)在任何情况下都被递归所满足, 命题2表明这些条件意味着(35)。□

在本节中，提出了可扩展的DMPC方案。

每个飞机都有自己的目标和动力。耦合约束已在例2中表达, 集合 $\{O\}$ 已在例4中定义。控制区域内的约束满足度为 ∞ 。

确保没有两架飞机在同一时间重新规划。

面临的挑战是如何处理 "动态 "的性质。
控制区，商定一个重新规划的顺序与
飞机不断地进入和离开；这是通过自我组织的排序
完成的。
ATM问题已经用Dubins的汽车模型进行了模拟，速
度和转弯的曲率作为控制输入，每一个都有限制[Bal
luchi等人, 1996]。所有的飞机
有恒定高度的飞行和每个人的目标
是最小化从初始点到
目的地，尽管非线性优化器将
承认更多种类的费用。所有飞机都需要飞行

蒸发的尾部解决方案。因此， $Y(Y^*(k_0))$ 是已知的一个 pri- q $*$ ori 是 $P_q(\mathbf{x}_q(k_0 + 1), C_q(Y_{\{-q\}}(k_0 + 1))$ 的一个解决方案。) 因此，这被证明是可行的。那么，由于 $Y_q^*(k_0 + 1) \in C_q_{\{-q\}}^*(k_0 + 1)$ 直接由约束条件(12g)强制执行那么等价关系 (18) 证明了 (31)。(32)从命题3。 \square

通过控制区，以到达指定地点。
沟通。对于每架飞机，都有一个无碰撞的徘徊圈。
在水平线结束时发现的裂缝定义了它的不变状态集[Schouwenaars等人, 2004]。为了提供不同的
撤消限制的可行表示法
这与基于梯度的非线性运算兼容。
定时器，排他性区域由极性集合建模

方法[Patel and Goulart, 2011]。
 该程序已经执行了1000秒，80架飞机以随机时间被添加到网络中。时间跨度和框架长度都是10 ($N=10$, $L_f=10$)。图1显示了程序运行期间受控区域的快照，其中代理2正在离开合作，代理9、10和11仍处于安全模式，并试图在网络中获得进入合作的时间槽。图2展示了所有飞机的生成轨迹。图3显示了所有成对的代理人在控制区飞行时的相对距离。尽管该图中的成对分离很难观察到细节，但明显的结论是没有任何轨迹低于安全距离，这就验证了任何一对之间的分离都没有消失。

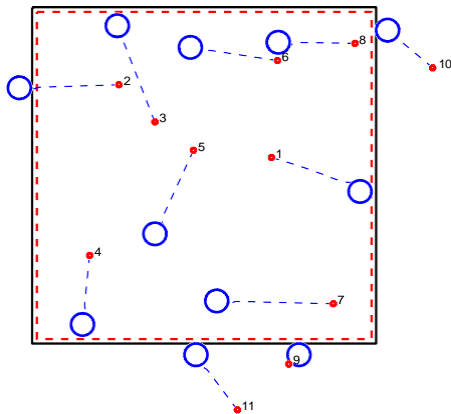


图1.控制区的一张快照

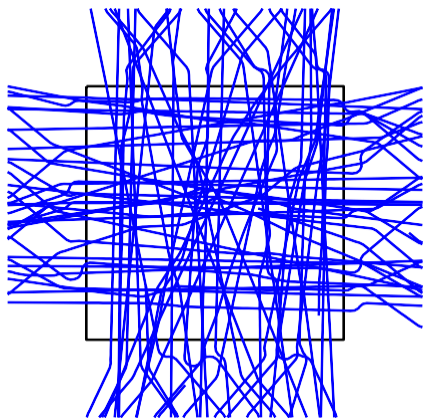


图2.由可扩展的DMPC生成的轨迹

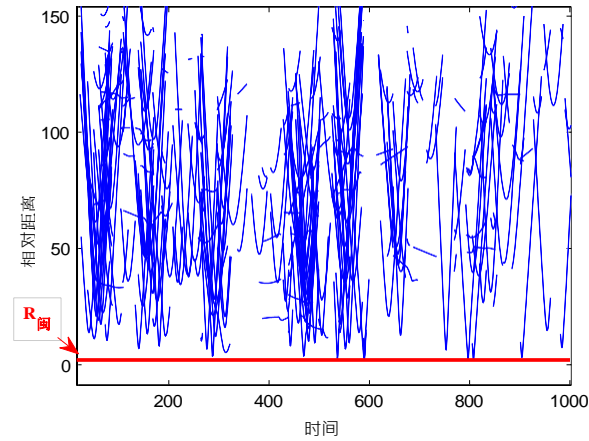


图3.控制区域内特工之间的相对距离

7 总结

在这项工作中，我们提出了一种分布式控制算法，用于不同子系统动态加入或离开的系统。子系统之间的耦合源是输出约束，不允许同时进行重新规划。串行优化可以解决这个问题，但需要对重新规划的顺序进行集中协调，这是不可能扩展的。相反，这项工作建议每个子系统通过遵循基于STDMA通信协议的算法在网络中找到其优化槽。尽管任何基于串行方案的工作方法都有一定程度的保守性，但可扩展的DMPC方法提供了在有限的帧长度内做更多事情的机会，因为子系统可以进入和离开。然而，如果子系统试图进入网络的速度比子系统试图离开的速度快，就会发生记录的堵塞，没有其他子系统可以进入。

未来的工作将探索有几个共同作业组的情况。这是由空中交通管制问题引起的，每个空域部门的飞机被视为一个合作组，飞机可以在不同的空域部门之间旅行。此外，还将研究对不同合作组的子系统进行平行优化的可能性。此外，还将研究对调整参数的敏感性，如规划范围、框架长度和重新规划的速度。

参考文献

- Alessio, A., Barcelli, D., and Bemporad, A. (2011). 动态耦合线性系统的去集中化模型预测控制. *过程控制杂志*, 21 (5) : 705-714.
- Asadi, F. and Richards, A. (2015). 空中交通管理的特设分布式模型预测控制。在

- 第16届{IFAC}会议控制应用CAO研讨会, 第48卷, 第68-73页, 德国Garmisch-Partenkirchen.
- Balluchi, A., Bicchi, A., Balestrino, A., and Casalino, G. (1996). 杜宾汽车的路径跟踪控制。In *Robotics and Automation, 1996.Proceedings., 1996 IEEE 国际会议*, 第4卷, 第3123-3128页。IEEE.
- Barreiro-Gomez, J., Obando, G., Ocampo-Martinez, C., and Quijano, N. (2015). 通过使用分布式史密斯动力学使非集中化的模型预测控制方案。 *IFAC-PapersOnLine*, 48(23):501-506.
- Borrelli, F. and Keviczky, T. (2006). 动态解耦系统的分布式LQR设计。In *Decision and Control, 2006 45th IEEE Conference on*, pages 5639-5644. IEEE.
- Bourdais, R., Buisson, J., Dumur, D., Guéguen, H., and Moro, P. (2014). 基于Dantzig-Wolfe分解的耦合约束下的分布式MPC。In *Distributed Model Predictive Control Made Easy*, pages 101-114. Springer.
- Christofides, P. D., Scattolini, R., de la Haza D. M., and Liu, J. (2013). 分布式模型预测控制。A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21-41.
- Dai, L., Xia, Y., Gao, Y., Kouvaritakis, B., and Cannon, M. (2015). 具有状态估计和耦合概率约束的系统的合作性分布式随机mpc. *Automatica*, 61:89-96.
- Dunbar, W. B. (2007). Distributed receding horizon control of dynamically coupled nonlinear systems. *Automatic Control, IEEE Transactions on*, 52(7): 1249-1263.
- Dunbar, W. B. and Murray, R. M. (2006). 多车辆编队稳定的分布式后退水平线控制. *Automatica*, 42 (4) : 549-558.
- Farina, M. and Scattolini, R. (2012). 分布式预测控制: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48(6):1088-1096.
- Gaugel, T., Mittag, J., Hartenstein, H., Papanastasiou, S., and Strom, E. G. (2013). 深入分析和评估自组织的TDMA。In *Vehicular Networking Conference (VNC), 2013 IEEE*, pages 79-86. IEEE.
- Gian L. and Pannek, J. (2011). *非线性模型预知控制: 理论和算法*. Springer.
- Hernandez, B. and Trodden, P. (2016). 使用管子链的分布式模型预测控制。 *arXiv预印本arXiv:1603.02044*.
- Keviczky, T., Borrelli, F., and Balas, G. J. (2004a). 去耦系统的分散性退避地平线控制器的层次设计. In *Decision and Control, 2004.CDC.43rd IEEE Conference on*, volume 2, pages 1592-1597. IEEE.
- Keviczky, T., Borrelli, F., and Balas, G. J. (2004b). 关于去耦系统的去中心化后退水平线控制的研究。在美国控制会议上。
- 2004.2004年会议记录, 第6卷, 第4921页-4926. IEEE.
- Keviczky, T., Borrelli, F., and Balas, G. J. (2006). 大规模动态解耦系统的去中心化退地平线控制. *Automatica*, 42 (12) : 2105-2115.
- Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. P. (2007a). 多车辆引导的分布式稳健后退水平线控制. *Control Systems Technology, IEEE Transactions on*, 15 (4) : 627-641.
- Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. P. (2007b). 多车辆引导的分布式稳健后退水平线控制. *Control Systems Technology, IEEE Transactions on*, 15 (4) : 627-641.
- Li, H., Shi, Y., and Yan, W. (2016). 有保证的 γ -增益稳定性的分布式后退水平线控制受限非线性车辆for-mations. *Automatica*, 68:148-154.
- Lucia, S., Keviczky, M., and Findeisen, R. (2015). 具有即插即用能力的分布式系统的基于合同的预测性控制。 *IFAC-PapersOnLine*, 48(23):205-211.
- Maciejowski, J. M. (2002). *预测性控制: 与约束。 培生教育*.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Sckaert, P. O. (2000). 受限模型预测控制。稳定性和最优性。 *Automatica*, 36 (6) : 789-814. Miller, M. A., Reble, M., and Allgower, F. (2012). 通过分布式模型预测控制实现动态解耦系统的合作控制。In *International Journal of Robust and Nonlinear Control*, 22(12):1376-1397.
- Negenborn, R. R. and Maestre, J. (2014). 分布式模型预测控制。概述和未来研究机会的路线图。 *控制系统, IEEE*, 34 (4) : 87-97.
- Patel, R. B. and Goulart, P. J. (2011). 使用在线优化的飞机避障机动的轨迹生成。 *Journal of guidance, control, and dynamics*, 34(1):218-230.
- Richards, A. and How, J. P. (2007). 稳健的分布式模型预测控制。 *国际控制杂志*, 80 (9) : 1517-1531.
- Riverso, S., Farina, M., and Ferrari-Trecate, G. (2014). 基于robust控制不变集的即插即用模型预测控制。 *Automatica*, 50 (8) : 2179-2186.
- Rom, R. and Sidi, M. (2012). *多重访问协议: 性能和分析*. Springer Science & Business Media.
- Scattolini, R. (2009). 分布式和分层式模型预测控制的架构-回顾。 *Journal of Process Control*, 19 (5) : 723-731.
- Schouwenaars, T., How, J., and Feron, E. (2004). 带有隐性安全保证的重现水平线路径规划。In *American Control Conference, 2004.Proceedings of the 2004*, volume 6, pages 5576-5581. IEEE.
- Stoustrup, J. (2009). *Plug & Play Control: 迎接新挑战的控制技术*. 欧洲杂志

- 的控制, 15 (3-4) : 311-330。
- Tedesco, F., Raimondo, D. M., and Casavola, A. (2014).多车辆无人驾驶系统的防撞指令调节器。In *International Journal of Robust and Nonlinear Control*, 24(16):2309-2330.
- Trodden, P. and Richards, A. (2010).具有持续干扰的线性系统的分布式模型预测控制。*International Journal of Control*, 83(8):1653-1663.
- Trodden, P. and Richards, A. (2013).具有耦合约束的线性系统的合作性分布式移动通信。 *Automatica*, 49 (2) : 479-487。
- Wang, C. and Ong, C.-J. (2010). 分布式模型具有耦合成本的动态解耦系统的预测控制。*Automatica*, 46(12):2053-2058.
- Zeilinger, M., Pu, Y., Rivero, S., Ferrari-Trecate, G., and Jones, C. (2013).基于分布式不变性和优化的即插即用分布式模型预测控制。第52届IEEE决策与控制会议, 第5770-5776页。