

班级 1802012

学号 18020100198

本科毕业设计（论文）

# 外文资料翻译

毕业设计题目 基于无人机的自主航点飞行与双  
光热成像目标监测算法研究

外文资料题目 基于在线优化的快速模型预测控  
制

学 院 电子工程学院

专 业 电子信息工程

学 生 姓 名 吕瑞涛

指导教师姓名 李隐峰

# 基于在线优化的快速模型预测控制

**摘要——**模型预测控制(MPC)的一个广泛公认的缺点是，它通常只能用于慢动态的应用程序，其中样本时间以秒或分钟为单位测量。实现快速MPC的一个著名技术是离线计算整个控制律，在这种情况下，在线控制器可以实现为一个查找表。这种方法适用于状态小和输入维数（例如不超过5）、约束少和时间范围短的系统。在本文中，我们描述了一组方法来提高MPC的速度，使用在线优化。这些利用MPC问题的特定结构的定制方法可以进行计算控制动作的速度比一种方法快100倍

它使用了一个通用的优化器。作为一个例子，我们的方法计算了一个在大约5个ms内有12个状态、3个控制和30个时间步长（需要解决一个有450个变量和1284个约束的二次程序）的问题的控制操作，允许MPC在200Hz下进行。

**指标项——**模型预测控制(MPC)，实时凸优化。

## I. 介绍

通过一个经典模型预测控制(MPC)，通过求解一个在线优化问题得到每个时间步长的控制动作。利用线性模型、多面体约束和二次代价，得到的优化问题是一个二次规划(QP)。使用通用方法解决QP可能很慢，传统上将MPC限制在具有慢动态的应用中，只使用以秒或分钟为单位测量样本时间。实现快速MPC的一种方法是显式地计算QP作为初始状态[1]，[2]的函数；然后，控制操作以查找表的形式在线实现。这里的主要缺点是，表中的条目数量可以随着视界、状态和输入维数呈指数级增长，因此“显式MPC”只能可靠地应用于小问题（其中状态维数不超过5个左右）。

在本文中，我们描述了一组方法，可以用来大大加快控制动作的计算，使用在线优化。其中一些观点已经在文献中被注意到了，在这里我们将证明当它们结合使用，它们允许MPC比通用优化器快几个数量级。

我们的主要策略是利用MPC[3]，[4]中出现的qp的结构。我们已经注意到，通过适当的变量重排序，通过求解一个线性方程组，可以找到每步的内点搜索方向。利用这种特殊的结构，需要一个关于状态维度 $n$ 、输入维度 $m$ 和视界 $T$ 的问题内点方法中的每一步操作 $O(T(n+m)^3)$ ，而不是没有利用特殊结构 $O(T(n+m)^3)$ 。由于内点方法只需要一个常数（和适度的）步数，因此MPC的复杂性在问题视界中是线性的，而不是三次的。

我们在这里应该提到的是，降低MPCQP复杂性的一种流行方法是通过完全根据控制输入来重新制定QP。在这种情况下，QP变得密集（问题的结构丢失了），并且需要 $O(T^3m^3)$ 内点方法中每个步骤的操作。这种技术通常与一种被称为移动阻塞的策略

相结合，其中假设视野中固定部分的输入是恒定的，因此优化变量的数量会进一步减少(见，e. g., [5] 和[6])。当视界很小时，这可以很好地工作，但我们将看到，即使对于适度的值，这种方法也会比充分利用问题结构的方法更慢。

另一个可以用于在线MPC的重要技术是热启动[7], [8]，其中每一步的计算都使用上一步所做的预测进行初始化。暖启动技术通常不用于一般的中间点方法（部分原因是这些方法已经非常有效），但它们可以选择适当的内点方法，将步骤数减少5倍或更多。

我们介绍的最后一种方法是（非常）提前终止一个适当的内点方法。即使相关的QP没有完全精确地解决，获得高质量的控制也不足为奇；毕竟，在每个MPC步骤中解决的优化问题实际上是一个计划练习，旨在确保当前的行动不会忽视未来。然而，我们发现，仅经过令人惊讶的少量迭代（通常在3到5之间），获得的控制质量非常高，即使获得的QP解很差。

我们将通过几个例子来说明我们的方法：一个机械控制系统、一个供应链问题和一个随机生成的例子。机械控制系统的高性能性能样本有12个州，3个控制，和一个周期  $T = 30$ ；每个MPC步骤都需要解决一个包含450个变量、924个不等式约束和360个等式约束的QP。我们的方法的一个简单的、非优化的C实现允许每个MPC步骤在大约5ms，3GHz的PC上执行，这将允许MPC在大约200Hz下执行。对于一个更大的例子，有  $n = 30$  个状态， $m = 8$  个控制和周期  $T = 30$ ，每一步需要解决一个超过1000个变量和几千个约束的QP；我们的方法可以在大约25ms内计算每个MPC步骤，允许40Hz的更新速率。（所获得的业绩详情将于稍后提供。）

MPC可用于几种类型的控制（以及估计）问题，包括跟踪问题、调节器问题和随机控制问题；此外，这些问题的时变和有限水平版本都可以被处理。在本文中，我们将重点关注无限水平时变随机控制问题，并在本文的最后提到这些方法如何扩展或应用于其他类型的问题。

## A. 相关工作

模型预测控制还有其他几个名称，如滚动水平规划、后退水平控制、动态矩阵控制和动态线性规划。它已被应用于广泛的应用，包括化学过程和工业控制[5], [9]–[12]，控制的排队系统[13]，供应链管理[14], [15]，随机控制问题在经济和金融[16], [17]，动态对冲[18]，和收入管理[19], [20]。

从最优控制的角度对MPC的讨论，见[21]–[23]；关于稳定性和最优性结果的调查，请参见[24]。有关封闭形式（脱机）方法，请参见[1]、[2]和[25]。有关使用优化方法的动态轨迹优化，请参见[26]。

先前关于在MPC中出现的QPs的有效解决方案的工作包括[27]–[29]和[63]；关于在化学过程控制中出现的大规模MPC问题的有效方法，请参见[30]–[32]。通过[33]–[35]解决了鲁棒MPC的有效方法。将牛顿方法应用于光滑的非二次代价函数的思想可以在[36], [37]中找到。

## II. 微光定常随机控制

### A. 系统动力学和控制

在本节中，我们将描述一个具有线性动力学的基本定不变随机控制问题。状态动力学是由

$$x(t+1) = Ax(t) + Bu(t) + w(t), t = 0, 1 \quad (1)$$

其中 $t$ 表示时间， $x(t) \in \mathbf{R}^n$ 是状态， $u(t) \in \mathbf{R}^m$ 是输入或控制， $w(t) \in \mathbf{R}^n$ 是扰动。矩阵 $A \in \mathbf{R}^{n \times n}$ 以及 $B \in \mathbf{R}^{n \times m}$ 为已知的数据。我们假设，对于不同的值，是已知分布的独立同分布(i.i.d)。我们让 $\bar{w} = \mathbf{E}w(t)$ 表示 $w(t)$ 的平均值（独立于 $t$ ）。

控制策略必须确定来自当前状态和以前状态 $x(0), \dots, x(t)$ ，i.e.中的当前输入 $u(t)$ ，我们将有一个因果状态反馈政策。我们让 $\phi_t : \mathbf{R}^{(t+1)n} \rightarrow \mathbf{R}^m$ 来表示控制策略，因此

$$u(t) = \phi_t(x(0), \dots, x(t))$$

这相当于随机变量 $(w(0), \dots, w(t-1))$ 在随机变量上是可测量的陈述。一个重要的特殊情况是时不变的静态状态反馈控制，当 $u(t) = \psi(x(t))$ ，其中 $\psi : \mathbf{R}^n \rightarrow \mathbf{R}^m$ 称为控制函数。

### B. 目标和约束条件

我们定义了以下目标：

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbf{E} \sum_{t=0}^{T-1} \ell(x(t), u(t)) \quad (2)$$

其中 $\ell : \mathbf{R}^{n+m} \rightarrow \mathbf{R}$ 是一个凸的二次阶段代价函数，在表格中

$$\ell(x(t), u(t)) = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + q^T x(t) + r^T u(t)$$

这里， $Q = Q^T \in \mathbf{R}^{n \times n}$ ， $S \in \mathbf{R}^{n \times m}$ ， $R = R^T \in \mathbf{R}^{m \times m}$ ， $q \in \mathbf{R}^n$ ，和 $r \in \mathbf{R}^m$ 为参数，我们假设

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0$$

式中 $\geq$ 为矩阵不等式。

我们也有状态约束和控制约束，定义为一组线性不等式

$$F_x x(t) + F_u u(t) \leq f, \quad t = 0, 1, \dots \quad (3)$$

其中， $F_x \in \mathbf{R}^{l \times n}$ ,  $F_u \in \mathbf{R}^{l \times m}$ ，和  $f \in \mathbf{R}^l$  为问题数据， $\leq$  表示向量（分量）不等式。

在许多问题中，目标和约束在状态和控制中是可分离的。这意味着  $S = 0$ ，并且状态约束和控制约束可以分别写为

$$F_x x(t) \leq f_x, \quad F_u u(t) \leq f_u, \quad t = 0, 1, \dots$$

在这里， $F_x \in \mathbf{R}^{l_1 \times n}$ ,  $f_x \in \mathbf{R}^{l_1}$ ,  $F_u \in \mathbf{R}^{l_2 \times m}$ ,  $f_u \in \mathbf{R}^{l_2}$ 。

进一步的专门化是， $Q$  和  $R$  是对角的，状态和控制约束包括下界和上界

$$x_{\min} \leq x(t) \leq x_{\max}, \quad u_{\min} \leq u(t) \leq u_{\max}, \quad t = 0, 1, \dots$$

在这里， $x_{\min}, x_{\max} \in \mathbf{R}^n$ ,  $u_{\min}, u_{\max} \in \mathbf{R}^m$ ，我们将这些约束称为盒子约束。

### C. 随机控制问题

随机控制问题是找到一个满足状态约束和控制约束 (3) 的控制策略，并使目标 (2) 最小化。可能会发生几种病理。随机控制问题可能是不可行的：不存在因果关系（概率为1），并且  $J$  是有限的。我们的阶段代价可以在下面是无界的（因为它有线性项，而二次部分不需要是正定的），所以随机控制问题也可以在下面是无界的，i.e., 有一个策略  $J = -\infty$ 。本文的兴趣是描述一种方法有效计算MPC控制策略（本身只是一个启发式次优策略），而不是随机控制的技术细节，所以我们不会进一步考虑这些问题，只是假设这个问题是可行的，有限的最优成本。见，e. g., [43]为更多关于线性随机控制的技术方面。

可以证明存在一个具有时不变静态状态反馈控制形式的最优策略，i.e.,  $u(t) = \psi_{\text{opt}}(x(t))$ 。不幸的是，状态反馈函数  $\psi_{\text{opt}}$  只能在少数特殊情况下有效计算，如当没有约束和  $\omega(t)$  是高斯的。

### D. 稳态确定性等效问题

为了将来的参考，我们描述了稳态确定基价问题，其中我们假设和具有常数值和，并且过程噪声的常数值等于其平均值。动力学方程变成

$$\bar{x} = A\bar{x} + B\bar{u} + \bar{w}$$

约束变成了  $F_x \bar{x} + F_u \bar{u} \leq f$ ，目标也变成了  $\ell(\bar{x}, \bar{u})$ 。

稳态确定性等价问题是

最小化  $\ell(\bar{x}, \bar{u})$

受制于  $\bar{x} = A\bar{x} + B\bar{u} + \bar{w} \quad F_x \bar{x} + F_u \bar{u} \leq f$

与变量  $\bar{x} \in \mathbf{R}^n$  和  $\bar{u} \in \mathbf{R}^m$ 。这个问题是一个凸QP，很容易解决。我们将让  $\bar{x}^*$  来表示确定性等价问题中的一个最优值  $\bar{x}$ 。

在许多情况下，稳态确定性等价问题有一个简单的解析解。例如，当  $\bar{w} = 0$ ，目标是纯二次的(i. e.,  $q$  和  $r$  都为零)，和  $f \geq 0$ （这意味  $\bar{x} = 0, \bar{u} = 0$  满足约束条件），我们有  $\bar{x}^* = 0$ 。

## E. 模型预测控制

MPC是一种启发式方法，用于寻找一个好的，如果不是最优的控制策略。在MPC中，通过首先求解优化来找到每一步的控制问题。

$$\begin{aligned} & \text{最小化 } \ell_f(x(t+T)) + \sum_{\tau=t}^{t+T-1} \ell(x(\tau), u(\tau)) \\ & \quad F_f x(t+T) \leq f_f \\ & \text{受制于 } \begin{aligned} & F_x x(\tau) + F_u u(\tau) \leq f, \quad \tau = t, \dots, t+T-1 \\ & x(\tau+1) = Ax(\tau) + Bu(\tau) + \bar{w} \\ & \tau = t, \dots, t+T-1 \end{aligned} \end{aligned} \quad (4)$$

与变量  $x(t+1), \dots, x(t+T)$  和  $u(t), \dots, u(t+T-1)$ 。

这里， $T$  是（规划）视界，这个函数  $\ell_f : \mathbf{R}^n \rightarrow \mathbf{R}$  是终端成本函数，我们假设它是二次的

$$\ell_f(x(t+T)) = x(t+T)^T Q_f x(t+T) + q_f^T x(t+T)$$

$Q_f \geq 0$  和  $F_f x(t+T) \leq f_f$  是终端状态的约束。选择MPC参数  $T, Q_f, q_f, F_f$  和  $f_f$  的方法有很多种，看e.g., [44]-[48]。

问题（4）是一个带有问题数据的凸QP

$$x(t), A, B, Q, Q_f, S, R, q, q_f, r, F_x, F_u, F_f, f, f_f, \bar{w}$$

设  $u^*(t), \dots, u^*(t+T-1), x^*(t+1), \dots, x^*(t+T)$  是最优的QP(4)。MPC策略采取了  $u(t) = u^*(t)$ 。MPC输入  $u(t)$  显然是当前状态的（复杂）函数，因此MPC策略具有静态状态反馈形式  $u(t) = \psi_{\text{mpc}}(x(t))$ 。可以证明，当二次代价项  $\psi_{\text{mpc}}$  为正定时（例如，见[1]），它是分段仿射函数。

我们可以对MPC QP(4)给出简单的解释。在这个QP中，我们将真实的代价函数截断到未来的水平步长，并用它们的平均值替换当前和未来的扰动(这些扰动尚不知道)。我们用一个近似值函数表示成本函数的截断部分。然后，我们计算了这个简化问题的最优轨迹 $\ell_f$ 。如果对下一步的干扰达到其平均值，我们可以将 $u^*(t), \dots, u^*(t + T - 1)$ 视为我们将采取的行动计划。我们只使用此计划中的第一个控制操作 $u^*(t)$ ，作为我们的实际控制。在 $t + 1$ 时刻，我们可以获得的实际价值 $x(t + 1)$ ，因此我们在一段时间内再次执行计划过程。对于每个时间步长 $t + 1, \dots, t + T + 1$ ，都会重复此操作。

## F. 显式MPC

在显式模型预测控制中，分段仿射控制律的显式形式 $\psi_{mpc}$ 是离线计算的。我们离线计算控制是仿射的多面体区域，以及每个区域[1]、[2]、[49]的偏移量和控制增益。然后将在线控制算法简化为查找表：首先确定与当前状态相关联的区域，然后应用与该区域相关联的控制律。

这种方法对于具有少量状态和控制、简单约束和适度范围的问题非常有吸引力，对于这些问题，区域数目是可控的。对于其他问题（例如，对于 $n \geq 5, m \geq 3$ 和 $l \geq 12$ ），区域的数量可能非常大，因此显式MPC在实际中不再可行。（有希望通过简化分段仿射函数 $\phi_{mpc}$ ，例如使用[50]-[52]中描述的方法，将其替换为具有可管理区域数目的分段仿射函数，从而获得良好的控制律。）

即使当区域的数量是可管理的（比如几千个）时，使用本文描述的方法仍然可以更快地求解QP(4)，而不是实现显式MPC中所需的查找表。此外，显式MPC不能处理系统、成本函数或约束是时变的情况。另一方面，对于矩阵 $A, B, Q, R, S, F_x, F_u, d, r, f$ 和 $\omega$ 随时间变化的情况，可以很容易地修改QP(4)（我们只需要跟踪这些矩阵的时间指数）。

## III. 原屏障内点法

在这一部分中，我们描述了一个利用其特殊结构来求解QP(4)的基本原始障碍内点。这些材料中的许多都在其他地方被报道过，可能是以不同的形式或背景(但似乎在使用或研究MPC的人中没有得到足够广泛的欣赏)；我们在这里收集它，使用统一的符号。在第四节中，我们描述了这里描述的方法的变体，这些变体提供了更快的方法。

我们首先以更紧凑的形式重写QP(4)。我们定义了一个全局优化变量

$$z = (u(t), x(t + 1), \dots, u(t + T - 1), x(t + T)) \in \mathbf{R}^{T(m+n)}$$

并将QP(4)表示为

$$\text{最小化 } z^T H z + g^T z$$

$$\text{受制于 } P_z \leq h, \quad Cz = b \quad (5)$$

$$H = \begin{bmatrix} R & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & Q & S & \cdots & 0 & 0 & 0 \\ 0 & S^T & R & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Q & S & 0 \\ 0 & 0 & 0 & \cdots & S^T & R & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & Q_f \end{bmatrix}$$

$$P = \begin{bmatrix} F_u & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & F_x & F_u & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & F_x & F_{tw} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & F_f \end{bmatrix}$$

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



$$\begin{aligned}
g &= \begin{bmatrix} r + 2S^T x(t) \\ q \\ r \\ \vdots \\ q \\ r \\ q_f \end{bmatrix} \\
h &= \begin{bmatrix} f - F_x x(t) \\ f \\ \vdots \\ f \\ f_f \end{bmatrix} \\
b &= \begin{bmatrix} Ax(l) + \bar{w} \\ \bar{w} \\ \bar{w} \\ \vdots \\ \bar{w} \\ \bar{w} \end{bmatrix}
\end{aligned}$$

显然，这些矩阵具有相当大的结构；例如， $H$ 是块三对角线。在特殊情况下，我们可以有更多的结构。例如，当问题是状态控制可分时， $H$ 是块对角线。

## A. 原始势垒法

我们将使用一种不可行的起始原始势垒方法来求解QP[3, CH.11], [53]。我们将QP(5)中的不等式约束替换为目标中的障碍项，以得到近似问题

$$\text{最小化 } z^T H z + g^T z + \kappa \phi(z)$$

$$\text{受制于 } Cz = b \quad (5)$$

其中是 $\kappa > 0$ 障碍参数，并且 $\phi$ 是与不等约束相关联的对数障碍，定义如下

$$\phi(z) = \sum_{i=1}^{lT+k} -\log(h_i - p_i^T z)$$

其中 $p_1^T, \dots, p_{lT+k}^T$ 为 $P$ 的行数。（如果 $Pz \not\leq h$ 我们取 $\phi(z) = \infty$ ）问题(6)是一个具有光滑目标和线性等式约束的凸优化问题，例如可以用牛顿方法来求解。

当 $\kappa$ 的解趋于零时，(6)的解收敛到QP(5)的解：可以证明(6)的解 $\kappa(lT + K)$ 对于QP(5)是次优的(例如，见[3，第11.2.2节])。在基本的原始障碍法中，对于递减的值 $\kappa$ 序列，我们使用牛顿方法从先前计算点开始，求解形式(6)的一系列问题。一种典型的方法是在每次计算(6)的解时(在一定的精度范围内)减去 $\kappa$ 的10倍。这种方法只需50步左右的牛顿步长即可得到原QP的精确解。(利用自协调理论[3，Sect.11.5]，[54]，人们可以得到在给定精度下求解QP所需牛顿步数的严格上界。但这一界限远远大于实践中经常观察到的步数。)

## B. 不可行的开始牛顿法

现在我们专注于使用不可行的开始牛顿方法来解决(6)[3，Sect.10.3.2]。我们将对偶变量 $\nu \in \mathbf{R}^{Tn}$ 与等式约束 $Cz = b$ 相关联。则(6)的最优性条件是

$$\begin{aligned} r_d &= 2Hz + g + \kappa P^T d + C^T \nu = 0 \\ r_p &= Cz - b = 0 \end{aligned} \quad (7)$$

其中 $d_i = 1/(h_i - p_i^T z)$ ，和 $p_i^T$ 表示 $P$ 中的第 $i$ 行。这一术语 $\kappa P^T d$ 是 $\kappa\phi(z)$ 的梯度。我们在这里还有一个隐含的约束 $Pz < h$ 。我们称 $r_p$ 为原始残数， $r_d$ 为对偶残数。叠加后的向量 $r = (r_d, r_p)$ 称为残差；(6)的最优性条件可以表示为 $r = 0$ 。

在不可行初始牛顿法中，算法是用一个严格满足不等式约束( $Pz^0 < h$ )但不满足等式约束的点 $z^0$ 来初始化的(因此，初始值 $z^0$ 可能是不可行的，这就是该方法的由来)。我们可以从任何一个 $\nu^0$ 开始。

我们在每一步都保持大致的 $z$  (与 $Pz < h$ ) 和 $\nu$ 。如果残差足够小，我们就退出；否则，我们通过线性化最优性条件(7)并计算原始步骤和对偶步骤 $\Delta z, \Delta \nu$ ，其中 $z + \Delta z, \nu + \Delta \nu$ 来改进我们的估计，对于这些步骤，在线性化的近似下，给出的残差为零。

原始搜索步骤和对偶搜索步骤 $\Delta z$ 和 $\Delta \nu$ 是通过解线性方程组找到的

$$\begin{bmatrix} 2H + \kappa P^T \text{diag}(d)^2 P & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix} \quad (8)$$

(这个词 $\kappa P^T \text{diag}(d)^2 P$ 是 $\kappa\phi(z)$ 的黑森矩阵。)一旦计算了 $\Delta z$ 和 $\Delta \nu$ ，我们就使用基于残差范数的回溯线 $r$ 搜索来找到步长 $s \in (0, 1]$ ，确保对于更新点是成立的(例如，见[3，第9.2节])。然后，我们将原始变量和对偶变量更新为AND。重复这一过程，直到残差的标准低于可接受的阈值。

可以证明，假设问题(6)是严格可行的，则原始可行性 (即 $Cz = b$ ) 将在有限数量的步骤中实现。一旦我们有 $r_p = 0$ ，它将在所有进一步的迭代中保持为零。此外， $z$ 和

$\nu$ 将收敛到最优点。计算精确解所需的牛顿步数取决于初始点 $z^0$  (和 $\nu^0$ )。这个步数可以使用自和谐理论进行形式上的限定,但这个界限比通常所需的步数大得多。

### C. 牛顿步长的快速计算

如果我们不利用(8)的结构,并使用稠密因式 $LDL^T$ 分解来求解它,例如,代价是 $(1/3)T^3(2n+m)^3$ 浮点。但通过利用我们问题中的结构,我们可以做得更好。

我们将使用块消除[3, App.C]。在我们继续之前,让我们定义一下 $\phi = 2H + \kappa P^T \text{diag}(d)^2 P$ ,它是块对角线,其中第一个块 $m \times m$ 、最后一个块 $n \times n$ 和其余的 $T-1$ 块 $(n+m) \times (n+m)$ 。它的逆也是块对角线;我们写成

$$\Phi^{-1} = \begin{bmatrix} \tilde{R}_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \tilde{Q}_1 & \tilde{S}_1 & \cdots & 0 & 0 & 0 \\ 0 & \tilde{S}_1^T & \tilde{R}_1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{Q}_{T-1} & \tilde{S}_{T-1} & 0 \\ 0 & 0 & 0 & \cdots & \tilde{S}_{T-1}^T & \tilde{R}_{T-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \tilde{Q}_T \end{bmatrix}$$

用块消去法求解(8)涉及以下一系列步骤。

- 1) 构成舒尔补 $Y = C\Phi^{-1}C^T$ 和 $\beta = -r_p + C\Phi^{-1}r_d$ 。
- 2) 通过求解 $Y\Delta\nu = -\beta$ 确定 $\Delta\nu$
- 3) 通过求解 $\Phi\Delta z = -r_d - C^T v$ 确定 $\Delta z$

舒尔补 $Y$ 具有块三对角线形式

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & 0 & \cdots & 0 & 0 \\ Y_{21} & Y_{22} & Y_{23} & \cdots & 0 & 0 \\ 0 & Y_{32} & Y_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Y_{T-1,T-1} & Y_{T-1,T} \\ 0 & 0 & 0 & \cdots & Y_{T,T-1} & Y_{TT} \end{bmatrix}$$

其中

$$\begin{aligned}
Y_{11} &= B\tilde{R}_0B^T + \dot{Q}_1, \\
Y_{ii} &= A\tilde{Q}_{i-1}A^T + A\tilde{S}_{i-1}B^T \\
&\quad + B\tilde{S}_{i-1}^T A^T + B\tilde{R}_{i-1}B^T + \tilde{Q}_i, \quad i = 2, \dots, T \\
Y_{i,i+1} &= Y_{i+1,i}^T = -\tilde{Q}_iA^T - \tilde{S}_iB^T, \quad i = 1, \dots, T-1.
\end{aligned}$$

我们可以形成 $Y$ 如下形式。首先，我们计算中每个块中 $\phi$ 的Cholesky因式分解，这需要 $(1/3)m^3 + (T-1)(1/3)(m+n)^3$ 触发器，这是 $T(m+n)^3$ 顺序的。然后，通过使用从 $A$ 和 $B$ 中获取的列的向后和向前替换，然后乘以中的相关块 $C$ 来完成形成 $Y$ ；这需要提供 $T(n^3 + n^2m)$ 触发器。因此，步骤1需要提供 $T(n+m)^3$ 触发器。

第二步是通过对 $Y$ 的Cholesky因式分解，然后是向后和向前替换。 $Y$ 矩阵是块三对角线，有 $T$ （块）行，有 $n \times n$ 块。对于块三对角矩阵(这与控制理论中的Riccati递归有关)[27]-[29]、[55]、[56]，可以使用下面描述的用于块三对角矩阵的专门方法来有效地分解它，或者通过将其视为带状矩阵、具有 $3n$ 带宽来进行分解。这两种方法都需要提供 $Tn^3$ 触发器([3]、[4]、[29]-[32]、[55]、[56])。因此，步骤2需要进行翻转。

$Y = LL^T$ 的Cholesky因式分解，其中 $L$ 是下三角的，如下所示。Cholesky因子 $L$ 具有较低的双对角块结构

$$L = \begin{bmatrix} L_{11} & 0 & 0 & \cdots & 0 & 0 \\ L_{21} & L_{22} & 0 & \cdots & 0 & 0 \\ 0 & L_{32} & L_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & L_{T-1,T-1} & 0 \\ 0 & 0 & 0 & \cdots & L_{T,T-1} & L_{TT} \end{bmatrix}$$

其中 $L_{ii}$ 是具有正对角线项的下三角，并且 $L_{i+1}$ 是一般的 $n \times n$ 矩阵。直接从 $LL^T = Y$ 中我们发现

$$\begin{aligned}
L_{11}L_{11}^T &= Y_{11} \\
L_{ii}L_{i+1,i}^T &= Y_{i,i+1}, \quad i = 1, \dots, T-1, \\
L_{ii}L_{ii}^T &= Y_{ii} - L_{i,i-1}L_{i,i-1}^T, \quad i = 2, \dots, T.
\end{aligned}$$

因此，我们可以通过对 $Y_{11}$ 的Cholesky因式分解求出 $L_{11}$ ，然后我们通过正向代换求解 $L_{11}L_{21}^T = Y_{12}$ 找到 $L_{12}$ ，然后我们可以通过对 $Y_{22} - L_{21}L_{11}^T$ 的Cholesky因式分解找到 $L_{22}$ ，以此类推。这些步骤中的每一步都需要 $n^3$ 触发。

步骤3的成本主要由其他步骤决定，因为在步骤1中已经计算了 $\Phi$ 的Cholesky因式分解。

计算搜索方向 $\Delta z$ 和 $\Delta v$ 使用块消去法所需的总工作量是顺序的，是 $T(m + n)^3$ 。请注意，如果使用泛型方法求解(8)，则此顺序与时间 $T$ 呈线性关系，而不是立方关系。

对于特殊情况，可以进一步降低复杂性。例如，当问题是状态控制可分的，具有对角 $Q$ 和框约束 $R$ 时，矩阵 $\Phi$ 是对角的。在这种情况下，最大的复杂性是顺序 $T(n^3 + n^2m)$ 触发器，它在 $T$ 和 $m$ 中都呈线性增长。

## D. 热启动

在一个典型的原始障碍法中，我们解(6)的值 $\kappa$ 的递减序列。当我们减少 $\kappa$ 并再次求解(6)时，对于新值 $\kappa$ ，我们可以简单地使用先前计算的 $z$ 作为第一个牛顿步长的初始值 $z$ 。这种热启动方法在减少每个值所需的牛顿步数方面非常有效。(事实上，这是原始屏障方法的关键。)利用自协调理论可以证明，当减去任何常数时，从先前计算的解开始，求解(6)到给定精度所需的牛顿步数是由问题维度的多项式所限定的。虽然这样一个界限的存在是令人放心的，但它的价值远远大于实际所需的牛顿步数，牛顿步数通常很小（例如，在5到10之间）[3，第10.3.3节]。

我们也可以使用这个热启动的想法来初始化我们在时间步长 $t$ 上第一次求解的 $z(6)$ ，使用先前计算的时间步长 $t - 1$ 的轨迹。粗略地说，在MPC中，我们通过为下一个时间步长 $T$ 制定完整的计划来计算当前的行动。我们可以使用先前计算的计划，适当地及时移动，作为当前计划的良好起点。

我们用前一个时间步的轨迹和计算的轨迹来初始化求解时间步长 $t(6)$ 的原始障碍法，并在最后附加 $\hat{u}$ 和 $\hat{x}$ 。对于 $\hat{u}$ 和 $\hat{x}$ 一个简单选择是 $\bar{u}^*$ 和 $\bar{x}^*$ 。假设 $t - 1$ 时刻计算出的轨迹是

$$\tilde{z} = (\tilde{u}(t - 1), \tilde{x}(t - 1), \dots, \tilde{u}(t + T - 2), \tilde{x}(t + T - 1))$$

我们可以为时间步长 $t$ 初始化原始屏障方法，用

$$z^{\text{init}} = (\tilde{u}(t), \tilde{x}(t + 1), \dots, \tilde{x}(t + T - 1), \bar{u}^*, \bar{x}^*)$$

假设 $\tilde{z}$ 满足等式约束，则 $z^{\text{init}}$ 也满足，除第一个和最后一个时间步长外，也将满足等式约束。如果 $\tilde{z}$ 严格满足不等约束，那么 $z^{\text{init}}$ 也将严格满足，除了可能在第一个和最后一个时间步长。在这种情况下，我们可以修改 $u(t)$ 和 $\hat{u}$ ，使不等式约束得到严格满足。

在特殊情况下，还可以使用其他几种热启动方法。例如，让我们考虑具有框约束的情况。我们可以使用相关的LQR问题，忽略约束，或使用[44]中描述的技术来计算获得的线性控制，然后将该轨迹投影到可行集(即方框)中。这种热启动初始化具有仅依赖于 $x(t)$ 而不依赖于先前状态或任何控制器状态的(可能)优势。

## IV. 一种预离散化二进制数算法

在这一部分中，我们描述了上述基本不可行起始原始势垒方法的一些简单变体。这些变化只产生基本MPC QP(5)的良好近似解，但不会显著降低MPC控制律的质量（按目标 $J$ 衡量）。然而，这些变分的计算速度比前面描述的原始势垒法快得多。

### A. 固定的 $\kappa$

我们的第一个变种实际上是对势垒方法的简化。我们建议使用一个不变的固定值，而不是求解 $\kappa$ 的递减的值序列(6)。此外，我们建议这个固定值不能太小，这意味着次最优界 $\kappa(lT + k)$ 不会太小。

对于一般的QP求解器，使用单个固定值 $\kappa$ 将导致非常糟糕的算法，这很可能需要非常多的步骤，具体取决于问题数据，此外，还只能计算近似解。我们建议使用这里 $\kappa$ 的固定值有几个原因。首先，我们必须记住，我们的目标是计算出一个给出良好目标值的控制，测量为 $J$ ，而不是准确地求解QP(5)。(实际上，QP只不过是计算良好控制的启发式方法。)在大量的数值实验中，我们发现通过求解近似的QP(6)而不是精确的QP(5)来获得的闭环控制品质是非常好的，即使在求解QP的次最优界不小的情况下也是如此。威尔斯和希思[57]也观察到了这一点，他们对这一现象的解释如下。当我们把问题(6)替换为(5)时，我们可以将其解释为精确地解决MPC问题，其中我们将势垒解释为附加的非二次状态和控制成本函数项。

事实证明，使用的 $\kappa$ 特定值并不重要（就所需的牛顿迭代次数而言，或者就所需的最终值 $J$ 而言）；在非常宽的范围内的任何值（例如系数 $10^3$ ）似乎都能给出很好的结果。对于任何特定的应用，可以通过增加它来找到一个很好的值，可能每次增加10倍，直到模拟显示所实现的控制质量下降。然后即可使用之前的值。

在[58]中描述了通过固定屏障参数来加快凸问题的近似解的思想，其中作者使用固定的屏障参数来极快地计算出一组接近最优的抓取力。

我们通过固定 $\kappa$ 获得的第二个优势是从先前计算的轨迹开始变暖。通过固定 $\kappa$ ，每一次预测控制迭代只不过是一个牛顿过程。在这种情况下，从先前计算的轨迹热开始可靠地在所需牛顿步数方面提供了非常好的优势。相比之下，完全原始障碍法的热启动提供的优势有限且不稳定。

通过修正 $\kappa$ ，我们可以将每次MPC迭代所需的牛顿步数从原始势垒方法的典型值50步左右减少到5阶值。(确切的数字取决于应用；在某些情况下可能更小。)

### B. 固定迭代限制

我们对原始势垒方法的第二种变形也是一种简化。通过固定 $\kappa$ ，我们将每一次MPC计算简化为进行牛顿法求解(6)。在标准牛顿法中，只有当残差的范数变小或达到某个迭代极限时，才停止迭代。当在残差范数足够小之前达到迭代极限 $K^{max}$ 时，它被认为是算法失败。(例如，当问题没有严格的可行点时，就会发生这种情况。)在MPC中，

控制的计算以固定的周期周期性地运行，并且具有硬运行时间约束，因此我们不得不处理计算控制所需的最坏情况的时间，即每牛顿步长的时间的 $K^{max}$ 倍数。

现在我们来查看我们的第二个简化。我们只选择一个非常小的值 $K^{max}$ ，通常在3和10之间。(但是，当MPC控件启动时，我们没有以前的轨迹，所以我们可能会使用较大的 $K^{max}$ 值。)事实上，简单地在每次MPC迭代中运行固定数量 $K^{max}$ 的牛顿步数并不会有什么坏处，无论残差是大是小。

当 $K^{max}$ 有一些较小的值时，如5，牛顿过程可能终止于一个甚至不是原始可行的点 $z$ 。因此，计算的计划甚至不满足动力学方程。然而，它确实尊重约束；特别是 $u(t)$ 满足当前的时间约束 $F_x(t) + F_u(t) \leq f$ （假设这些约束是严格可行的）。此外，当然，对偶残差不必很小。

人们可能会认为，通过这样一个粗暴的QP解决方案获得这样的控制，可能是相当糟糕的。但大量的数值实验表明，所得到的控制具有非常高的质量，与精确的预测控制相比，仅有很小的(或没有) $J$ 的增加。事实上，我们观察到， $K^{max}$ 在低至1的情况下，在某些情况下获得的控制并不太差。我们不推荐使用 $K^{max} = 1$ ；我们只是将其作为MPC的一个有趣的变体来提及，它需要的计算量大大减少。

我们不完全理解为什么这种控制效果这么好，即使在许多情况下，优化在实现最初的可行性之前就被终止了。一种看似合理的解释可以追溯到对MPC的基本解释：在每一步，我们都为后面的 $T$ 步制定出完整的计划，但只使用当前的控制。计划的其余部分并未真正使用；进行计划是为了确保当前控制不会对系统未来的行为产生不良影响。因此，即使计划不是非常仔细地执行，目前的控制将是良好的似乎是合理的。

有趣的是，当算法在 $K^{max}$ 次迭代中不完全收敛时，所得到的控制律不像(精确的)预测控制那样是静态反馈。事实上，控制器状态就是计划 $z$ ，而控制确实（在某种程度上）依赖于 $x(t-1)$ 。

## C. 总结和实施结果

我们开发了一个用C语言编写的近似原始屏障方法的简单实现，使用LAPACK库[59]、[60]来执行数值线性代数计算。我们当前的C实现，可从[http://www.stanford.edu/~boyd/fast\\_mpc.html](http://www.stanford.edu/~boyd/fast_mpc.html)，处理可分离的纯二次目标（即 $q = 0, r = 0$ ）和框约束的情况。我们报告了在运行Linux的3 GHz AMD Athlon上，使用V-C部分中描述的方法构造的12个不同维度的问题的计时结果。

$n$	$m$	$T$	QP size	$K^{\max}$	Time (ms)	SDPT3 (ms)
4	2	10	60/168	3	0.94	150
4	2	20	120/328	3	1.87	250
4	2	30	180/488	3	2.79	400
10	3	10	130/380	3	1.84	450
10	3	20	260/740	3	3.68	800
10	3	30	390/1100	3	5.52	1400
16	4	10	200/592	3	2.50	500
16	4	20	400/1152	3	5.04	1300
16	4	30	600/1712	3	7.58	2600
30	8	10	380/1120	5	7.63	1400
30	8	20	760/2180	5	16.42	4800
30	8	30	1140/3240	5	25.95	3400

表1：为随机生成的例子求解每个qp所需的时间

为了确保我们的方法计算的输入提供的控制性能基本上等同于精确的MPC，我们用精确的MPC模拟每个例子，使用由CVX[62]调用的通用优化求解器SDPT3[61]精确地求解QP。（不过，报告的时间仅包括SDPT3 CPU时间。）SDPT3是一个最先进的原始-对偶内点求解器，它利用了稀疏性。我们的近似原始障碍方法中的参数 $K^{\max}$ 和 $\kappa$ 选择是为了给出控制性能，通过蒙特卡罗模拟来估计平均阶段成本，本质上与通过精确求解QP获得的控制性能相同；在任何情况下，都不会超过几个百分点的差（在某些情况下，更好）。

表一列出了12个问题的结果。列出QP SIZE的列提供了变量总数（第一个数字）和约束总数(第二个数字)。我们可以看到，小问题（然而，对于显式MPC方法来说，这将被认为是大问题）在毫秒内就得到了解决，这使得千赫的控制率成为可能。最大的问题，涉及一个不小的QP，有一千多个变量和数千个约束，在26ms左右就解决了，允许控制几十赫兹的速率。我们也解决了更大的问题；我们的近似原始势垒方法所需的时间如预期的那样增长，甚至更慢。

我们还可以看到，近似势垒方法的性能远远优于通用的(并且非常有效的)求解器SDPT3，后者只利用稀疏性。当然，这种比较并不完全公平；在每次调用SDPT3时，都必须检测稀疏模式，并确定良好的消除顺序；相反，在我们的代码中，已经进行了稀疏利用。（事实上，我们可以看到这样的效果：对于 $n = 30, m = 8$ 和 $T = 30$ 的问题，SDPT3所需的时间实际上比 $T = 20$ 少，因为SDPT3选择将前者视为稀疏问题，而将后者视为稠密问题。）。以下数字构成了我们的观点：一般的凸优化求解器，即使是非常高效的求解器，也不能很快地重复解决小型和中等规模的问题。



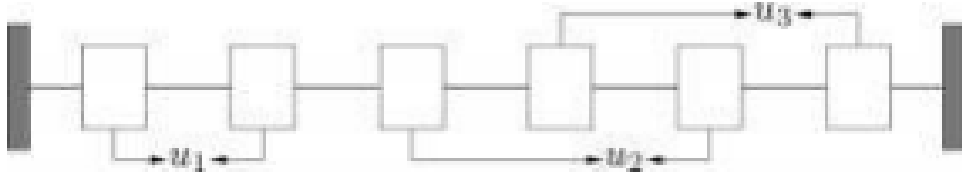


图1：振荡质量模型。粗线代表弹簧，两边的黑暗区域代表墙壁。

## V. 例子

在下一节中，我们将提供三个具体的例子：机械控制系统、供应链和随机生成系统。我们比较了我们的快速近似MPC方法和精确MPC方法的性能，并给出了我们目前的C实现可以处理的两种情况的时序规范。

### A. 摆动质量

如图1所示，第一个例子由六个质量块组成，它们通过弹簧相互连接，并与两侧的墙壁相连。有三个致动器，它们在不同的质量块之间施加张力。质量值为1，所有弹簧的弹性常数均为1，没有阻尼。控制件最大施力为 $\pm 0.5$ ，质点位移不能超过 $\pm 4$ 。

我们使用一阶保持模型对这个连续时间系统进行采样，其周期为0.5(这比开环系统最快振荡模式的周期快约3倍)。状态向量 $x(t) \in \mathbf{R}^{12}$ 是质量的位移和速度。扰动 $\omega(t) \in \mathbf{R}^{12}$ 是作用在每个质量上的随机作用力，其 $[-0.5, 0.5]$ 上的分布是均匀的。

(因此，扰动与执行力的最大允许值一样大，但作用于所有六个质量。)对于MPC，我们选择一个时间 $T = 30$ ，和可分离的二次目标 $Q = I, R = I, S = 0, q = 0, q_f = 0, r = 0$ 。我们选择使用[44]中描述的后发式方法。问题的维度是 $n = 12, m = 3, l = 18$ 和 $k = 24$ 。当然，等价的最优状态和控制的稳态确定性为零。

所有的模拟都是在1100个时间步长内进行的，去掉前100个时间步长，使用相同的随机力干扰实现。初始状态 $x(0)$ 被设置为稳态确定性等效值，在这种情况下为零。我们首先将精确MPC(使用CVX[62]计算，依赖于求解器SDPT3[61])与近似MPC进行比较，后者使用固定的正值 $\kappa$ 计算，但没有迭代限制。精确的MPC达到目标值 $J = 5.93$

(按1000个时间步长的平均阶段成本计算)。图2显示了精确预测预测和近似预测预测的阶段成本分布 $\kappa = 1, 10^{-1}, 10^{-2}$ 。对于 $\kappa \leq 10^{-2}$ ，控制性能与精确预测控制基本相同，因为目标比精确预测控制的目标大不到3%，而对于 $\kappa = 10^{-1}$ 精确预测控制，目标大约比精确预测控制大18%。

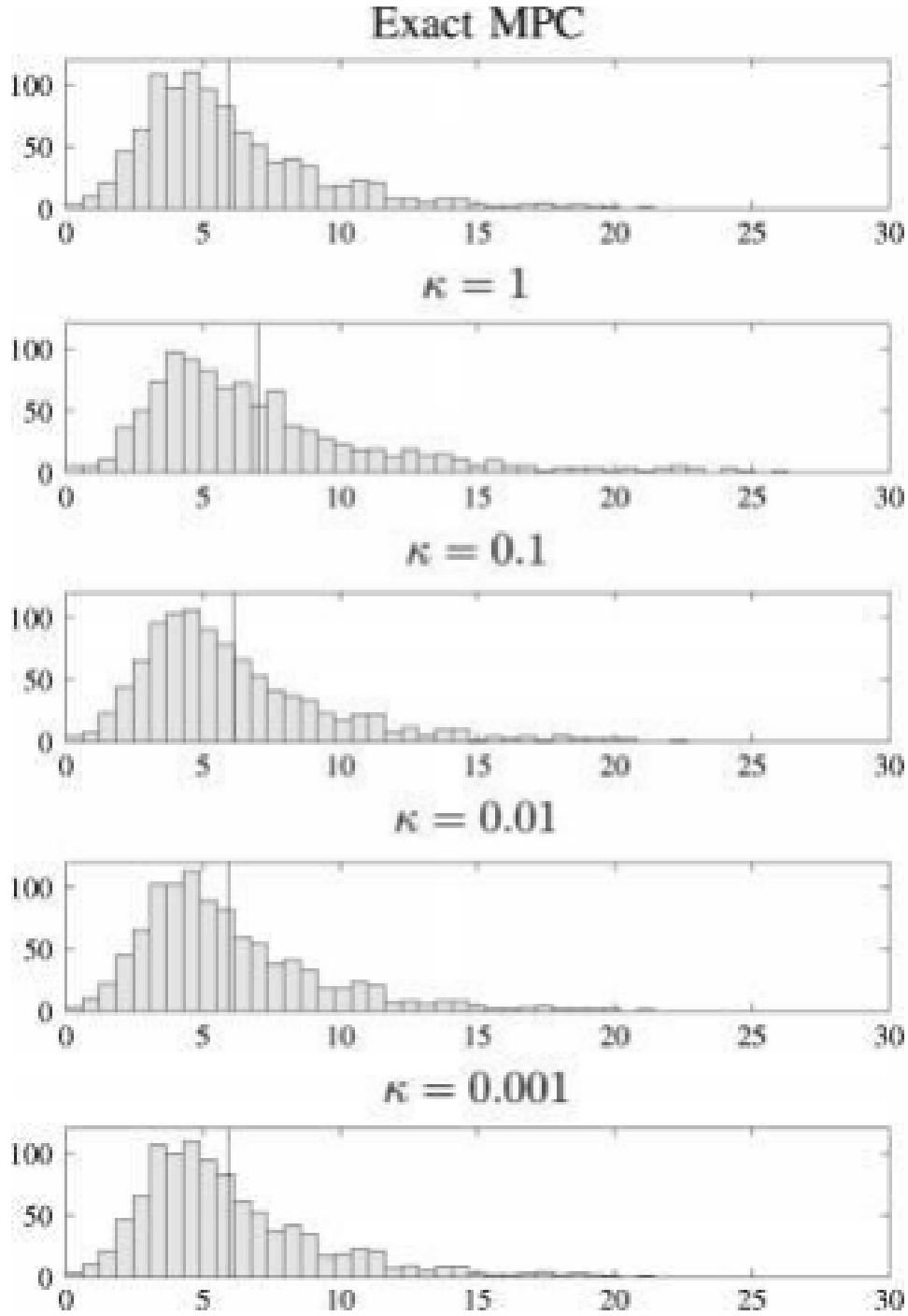


图2：阶段成本的直方图，对于不同的kappa值，对于振荡质量的例子。实心竖线表示每个分布的平均值。

为了研究迭代极限 $K^{max}$ 的影响，我们令 $\kappa = 10^{-2}$ ，对 $K^{max} = 1, K^{max} = 3$ 和 $K^{max} = 5$ 进行了固定和模拟。阶段成本的分布如图3所示。对于 $K^{max} = 3$ 和 $K^{max} = 5$ ，获得的控制质量（以平均阶段成本衡量）基本上与精确的MPC相同。因为，在每次迭代中只采取一个牛顿步长，控制质量明显比精确的MPC差，但对我们来说，这种控制竟然如此之好似乎令人惊讶。对于 $K^{max} = 1$ ，5%的步骤达到原始可行性，而对于 $K^{max} = 3$ 只有68%的步骤达到原始可行性，并且对于 $K^{max} = 5$ ，95%

的步骤达到原始可行性。很明显（有点令人震惊），原始的可行性并不是获得令人满意的控制性能的必要条件。

对于这个例子，合理的参数选择应该是 $\kappa = 10^{-2}$ 和 $K^{max} = 5$ ，其产生的平均阶段成本基本上与精确MPC相同，具有10倍或更大的加速比（基于QP精确解的50次迭代）。由 $\kappa = 10^{-2}$ 和 $K^{max} = 5$ 产生的控件与Exact MPC非常相似，但并不相同。图4显示了 $x(t)$ ，精确MPC和快速MPC的第一质量和 $u_1(t)$ ，第一控制的位移。状态轨迹几乎难以区分，而控制轨迹则显示出一些微小的偏差。

我们简单的C语言实现可以在大约1ms的时间内完成振荡质量问题的一个牛顿步骤。当 $K^{max} = 5$ ，我们的近似MPC控制可以用5ms的采样时间来实现。因此，MPC控制可以在高达200Hz的速率下执行。

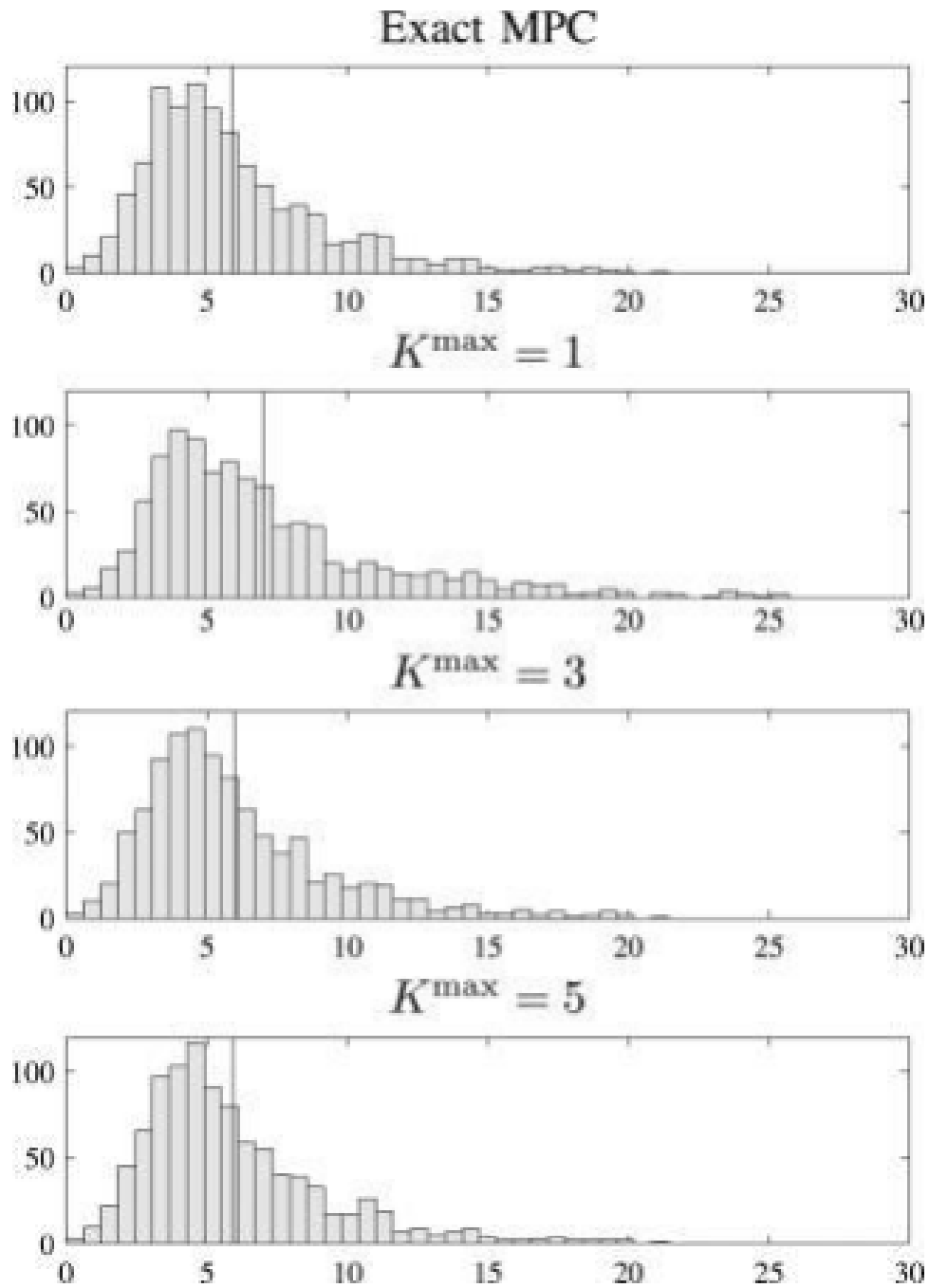


图3：阶段成本的直方图， $\sigma=10$ ，振荡质量的例。实心竖线表示每个分布的平均值。

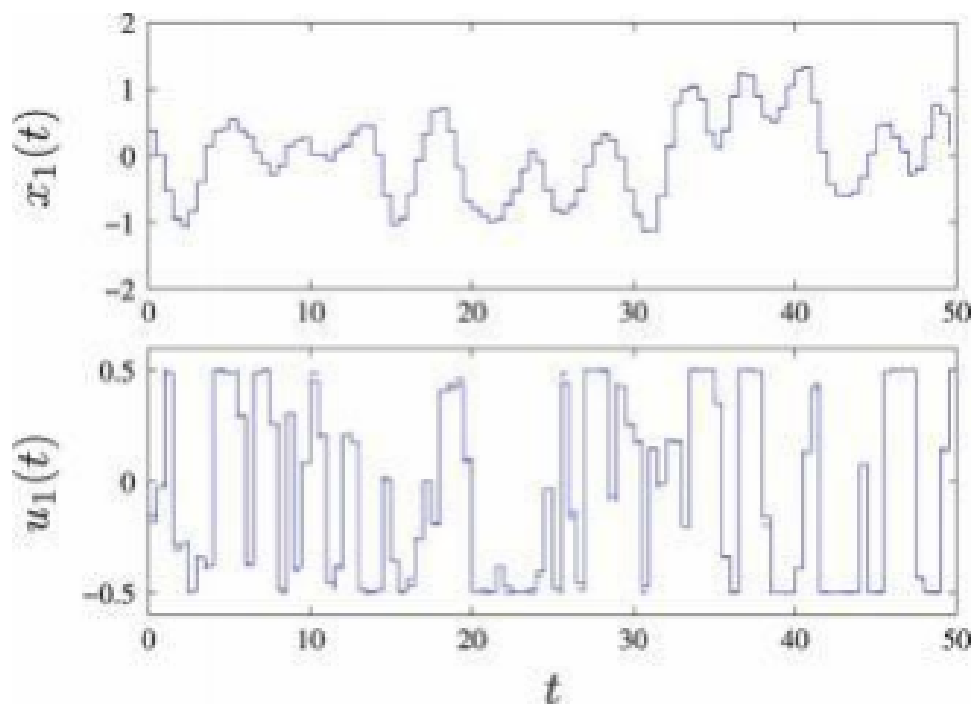


图4：阶段成本的直方图，kappa=10，Kmax=5，振荡质量的例。实心竖线表示快的MPC，虚线曲线表示具体的MPC。

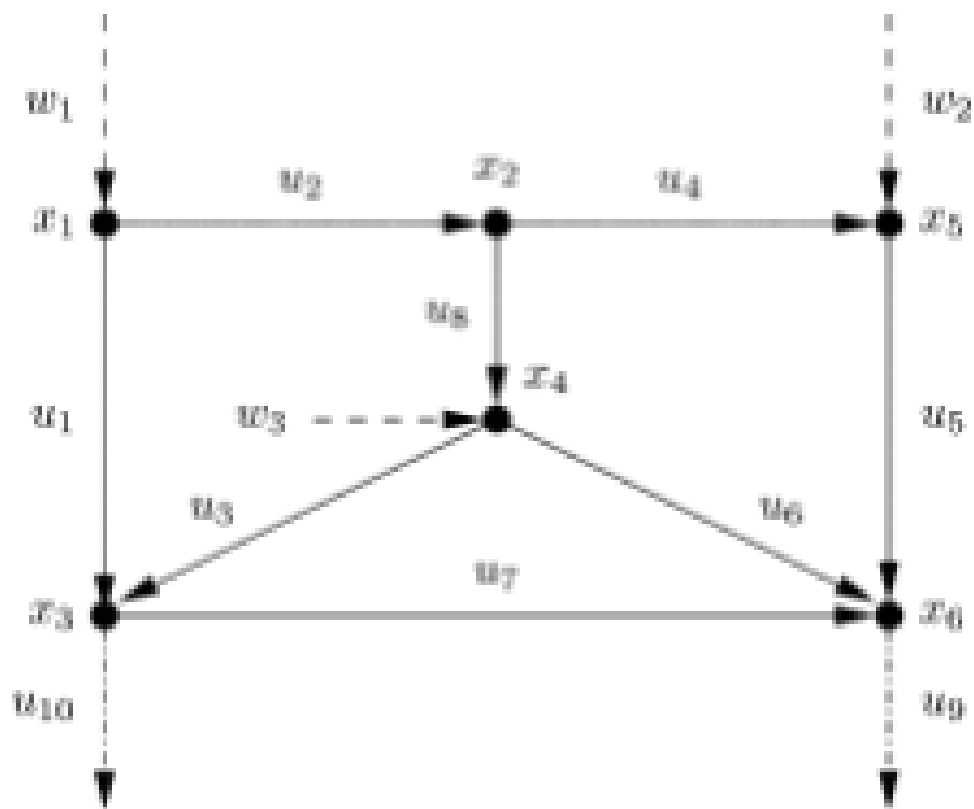


图5：供应链模型。点表示节点或仓库。箭头表示链接或商品流。虚线箭头是流入，虚号箭头是流出。

## B. 供应链

这个例子是一个单一的商品供应链，由6个节点组成，代表仓库或缓冲区，与13个单向链路互连，代表商品的某种运输机制或流，如图5所示。虚线箭头表示的流中有三个是流入的，超出了我们的控制；然而，剩下的10个流是控制的。这些控件中有两个是流出，用虚线表示。

在每个时间步长内，不确定数量的商品沿着三条流入链路进入网络；控制律选择沿其他10条边的流量，包括两条流出链路。系统状态是存在于每个节点的商品数量，因此  $x(t) \in \mathbf{R}^6$ 。该状态被约束为满足  $x(t) \geq 0$ 。控制是  $u(t) \in \mathbf{R}^6$ ，它的每个分量都被约束在区间  $[0, 2.5]$  内。（因此，每个链路的容量为2.5。）扰动  $\omega(t)$  就是流入，所以  $\omega(t) \in \mathbf{R}^3$ 。IID的成分为指数分布，均值为1。还有一个将控件和状态链接在一起的约束：流出任何节点的总流量在任何时候都不能超过该节点上可用的商品数量。与摆动质量例子中的约束不同，这个问题在状态和控制中是不可分离的。问题的维度是  $n = 6, m = 10$ ，和  $l = 32$ 。

客观参数是  $Q = I, R = 0, S = 0, q = 0, r = 1, Q_f = 0, q_f = 0$ 。（此处表示所有条目均为1的向量。）这意味着每个节点都有价值的存储成本，值为  $x_i(t) + x_i(t)^2$ ，以及与每个边缘上的流量相等的代价。存储成本促使商品通过流出链路和从网络中路由出去。对于这个问题，稳态确定性等价问题不是微不足道的；它必须通过求解一个QP来计算。

对于供应链的MPC控制，我们使用了时间  $T = 10$ 。在我们的模拟中，我们使用初始状态  $x(0) = \bar{x}^*$ 。在本例中，我们发现当  $\kappa = 0.01, K^{max} = 10$  时，与精确的MPC相比，我们的近似MPC提供了基本上相同的控制质量。图6显示了精确MPC(虚线)和近似MPC(实线)的  $x_1(t)$  和  $u_1(t)$ 。显然，两者的控件和状态几乎难以区分。

我们当前的快速MPC方法的C实现不处理耦合的状态输入约束，这些约束出现在这个例子，所以我们不能报告时间。（此示例中显示的所有模拟都是使用快速MPC方法的MATLAB实现来找到的，由于许多原因，该方法速度较慢。）然而，我们可以基于上面给出的复杂性分析并从其他示例中推断，形成对将获得的性能的合理估计。我们估计，快速的MPC实现每牛顿一步大约需要1.2毫秒，因此每一次求解大约需要12毫秒。

### C. 随机系统

我们的第三个例子是随机生成的系统，其中  $A$  和  $B$  的条目是零均值单位方差正态随机变量。然后我们进行缩放  $A$ ，使其光谱半径为1，因此系统是中性稳定的。我们已经在表I中给出了这类问题的时序结果；在这里我们考虑一个具体的问题，以检验QP的近似解对所获得的控制性能几乎没有影响。

我们考虑的特定示例具有  $n = 30$  状态和  $m = 8$  控制。这些约束是框约束： $|x_i| \leq 5, |u_i| \leq 0.1$ 。扰动  $\omega(t) \in \mathbf{R}^{30}$  有IID条目，均匀分布在上面。（选择约束和干扰大

小时，控制通常是饱和的。) 我们使用[44]中描述的方法来选择成本参数。对于MPC，我们使用的时间为 $T = 30$ ，具有随机生成的起点 $x(0)$ 。

与前面的例子一样，我们发现带有参数 $\kappa = 0.01$ 和 $K^{max} = 5$ 的近似原始屏障MPC产生的控制本质上与精确MPC相同。样本轨迹如图7所示。

使用我们的C实现，我们可以在大约5毫秒内执行本例的一个牛顿步骤。通过 $K^{max} = 5$ ，我们的方法允许在25ms的采样时间内实现预测控制，因此可以在40Hz的速率下进行控制。

## VI. 扩展和证明

到目前为止，我们的讨论集中在时不变的无限水平随机控制问题上。这里，我们描述了如何将同样的方法应用于这种特殊的MPC配方的许多变体中。首先，应该清楚的是，当系统、成本函数或约束是时变的时，可以使用完全相同的方法；我们只需要跟踪矩阵 $A, B, Q, R, S, F_x$ ，和 $F_u$ ，和矢量 $q, r, f, \bar{w}$ 中的时间指数。可以使用相同的不可行开始牛顿法，以及快速计算牛顿步长的相同方法。所有的变化是这些数据矩阵（可能）在每个时间步改变。其他简单的扩展包括多速率问题，或者具有分段线性或分段常量输入的问题，或者添加最终状态约束（而不是我们的最终状态成本项）。

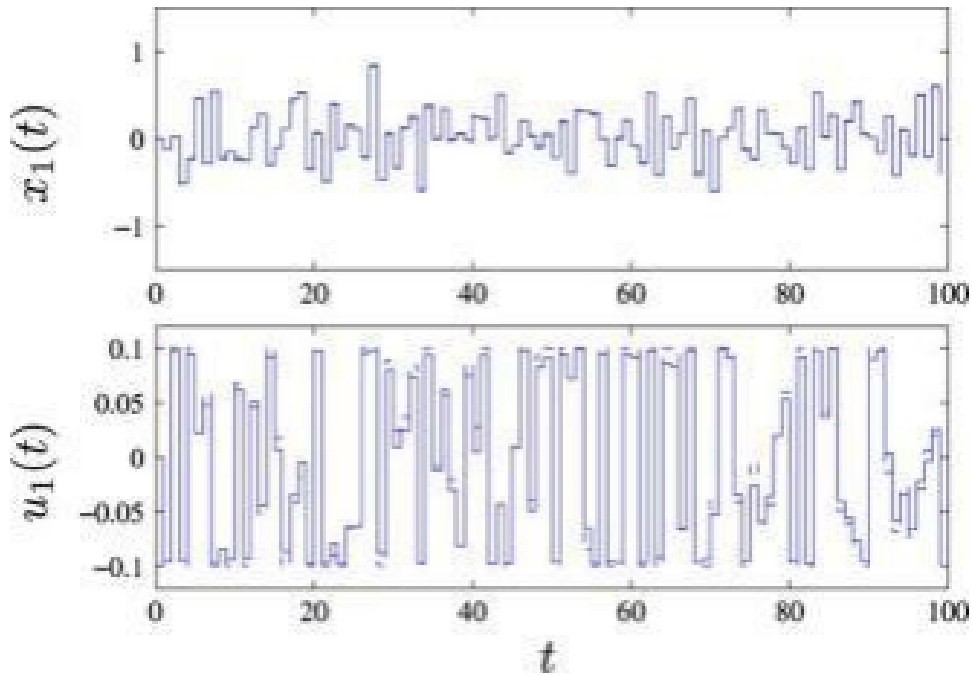


图7：基于 $\kappa=0.01$ ， $K_{max}=5$ 的随机系统仿真。实线：快速MPC；虚线：精确的MPC。

该方法也适用于采用标准技术的非线性系统。在每个牛顿步长，我们简单地线性化动力学(使用雅可比或粒子滤波方法)，在当前值 $z$ ，并使用该线性化模型计算步长。(然而，我们将使用具有非线性动力学的真实原始残差而不是线性化动力学中的原始残差来执行线搜索。)在这种情况下，数据矩阵改变每个牛顿步长(随着线性化的更新)。我

们将这些方法应用于非线性动力学问题的实验并不多；我们预计需要更多的迭代才能获得良好的控制性能。我们快速计算每个牛顿步长的能力在这里也将是有用的。

我们可以使用任何光滑的凸阶段费用函数，变化很小。我们可以通过引入产生(更大的)光滑问题的局部变量来结合非光滑的凸阶段成本函数(例如，见[3，CH.4])。这些添加的变量是“本地的”，即只与 $x(t)$ ,  $u(t)$ 交互，因此它们对黑森的贡献也将是本地的，并且可以应用相同的方法。一个例子是使用Huber成本函数的滚动水平估计，它给出了一个平滑过滤器，它对过程干扰或噪声中的异常值非常稳健[3，6.1节]。我们还可以添加任何在时间上是“局部”的凸约束，即在固定数量的时间步长上的链接状态或控制；这样的约束导致KKT矩阵具有相同的带状形式。

## VII. 结论及应用

通过结合几个想法，其中一些已经在预测控制或其他上下文中已知，我们可以极大地提高在线计算预测控制控制律的速度。我们所描述的方法是对离线方法的补充，为问题规模较小时的快速控制计算提供了一种方法。结合不断增长的计算机能力，MPC控制律的快速在线计算的可能性向我们表明，MPC现在或很快就可以用于许多以前认为不可行的应用中。

在快速在线预测方法领域，仍有许多实际和理论上的工作要做。虽然我们的大量模拟表明FAST在线预测控制效果很好，但正式的稳定性和(或者更好的性能保证)将是一个受欢迎的进步。

## 确认

作者要感谢M.Morari、S.Meyn、A.Bemporad和P.Parrilo进行了非常有帮助的讨论。他们还想感谢Koh和A.Mutapcic就C实现提供的建议，以及D.Gorneski在示例方面的帮助。

## 参考

- [1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.
- [2] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MFC solutions," in *Proc. IEEE Conf. Dec. Control*, 2001, pp. 1199–1204.
- [3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.



- [4] S. J. Wright, "Applying new optimization algorithms to model predictive control," *Chemical Process Control-V.*, vol. 93, no. 316, pp. 147–155, 1997.
- [5] J. M. Maciejowski, *Predictive Control with Constraints*. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [6] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," in *Proc. 43rd IEEE Conf. Dec. Control*, Dec. 2004, pp. 2023–2028.
- [7] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, no. 1–2, pp. 281–302, 2000.
- [8] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM J. Opt.*, vol. 12, no. 3, pp. 782–810, 2002.
- [9] R. Soeterboek, *Predictive Control: A Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [10] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [11] E. Camacho and C. Bordons, *Model Predictive Control*. New York: Springer-Verlag, 2004.
- [12] W. Wang, D. E. Rivera, and K. Kempf, "Model predictive control strategies for supply chain management in semiconductor manufacturing," *Int. J. Production Economics*, vol. 107, pp. 57–77, 2007.
- [13] S. P. Meyn, *Control Techniques for Complex Networks*. Cambridge, U.K.: Cambridge University Press, 2006.
- [14] E. G. Cho, K. A. Thoney, T. J. Hodgson, and R. E. King, "Supply chain planning: Rolling horizon scheduling of multi-factory supply chains," in *Proc. 35th Conf. Winter Simulation: Driving Innovation*, 2003, pp. 1409–1416.
- [15] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley, 2007.
- [16] H. Dawid, "Long horizon versus short horizon planning in dynamic optimization problems with incomplete information," *Economic Theory*, vol. 25, no. 3, pp. 575–597, Apr. 2005.
- [17] F. Herzog, "Strategic portfolio management for long-term investments: An optimal control approach," Ph.D. dissertation, ETH, Zurich, The Netherlands, 2005.
- [18] J. Primbs, "Dynamic hedging of basket options under proportional transaction costs using receding horizon control," in *Int. J. Control*, 2009. [Online]. Available:

<http://www.Stanford.edu/japrimbs/AutoSubmit20070813.pdf>

- [19] K. T. Talluri and G. J. V. Ryzin, *The Theory and Practice of Revenue Management*. New York: Springer, 2004.
- [20] D. Bertsimas and I. Popescu, "Revenue management in a dynamic network environment," *Transportation Sci.*, vol. 37, no. 3, pp. 257–277, 2003.
- [21] W. H. Kwon and S. Han, *Receding Horizon Control*. New York: Springer-Verlag, 2005.
- [22] P. Whittle, *Optimization Over Time*. New York: Wiley, 1982.
- [23] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Control and Estimation*. New York: Springer, 2005.
- [24] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.
- [25] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Fast, large-scale model predictive control by partial enumeration," *Automatica*, vol. 43, no. 5, pp. 852–860, May 2006.
- [26] J. T. Belts, *Practical Methods for Optimal Control Using Nonlinear Programming*. Warrendale, PA: SIAM, 2001.
- [27] A. Hansson and S. Boyd, "Robust optimal control of linear discrete-time systems using primal-dual interior-point methods," in *Proc. Amer. Control Conf.*, 1998, vol. 1, pp. 183–187.
- [28] A. Hansson, "A primal-dual interior-point method for robust optimal control of linear discrete-time systems," *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1639–1655, Sep. 2000.
- [29] L. Vandenberghe, S. Boyd, and M. Nouralishahi, "Robust linear programming and optimal control," presented at the 15th IFAC World Congr. Autom. Control, Barcelona, Spain, Jul. 2002.
- [30] R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal, "Quadratic programming algorithms for large-scale model predictive control," *J. Process Control*, vol. 12, no. 7, pp. 775–795, 2002.
- [31] L. T. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds. Cambridge, MA: Birkhauser, 2000, pp. 119–243.

- [32] J. Albuquerque, V. Gopal, G. Staus, L. T. Biegler, and B. E. Ydstie, "Interior point SQP strategies for large-scale, structured process optimization problems," *Comput. Chem. Eng.*, vol. 23, no. 4–5, pp. 543–554, 1999.
- [33] P. J. Goulart, E. C. Kerrigan, and D. Ralph, "Efficient robust optimization for robust control with constraints," *Math. Program.*, vol. Series A, pp. 1–33, 2007.
- [34] M. Cannon, W. Liao, and B. Kouvaritakis, "Efficient MFC optimization using pontryagin's minimum principle," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 831–844, 2008.
- [35] V. M. Savala, C. D. Laird, and L. T. Biegler, "Fast implementations and rigorous models: Can both be accommodated in NMPC?," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 800–815, 2008.
- [36] C. G. Economou, M. Morari, and B. O. Palsson, "Internal model control: Extension to nonlinear system," *Ind. Eng. Chem. Process Des. Development*, vol. 25, no. 2, pp. 403–411, 1986.
- [37] W. C. Li, L. T. Biegler, C. G. Economou, and M. Morari, "A constrained Pseudo-Newton control strategy for nonlinear systems," *Comput. Chem. Eng.*, vol. 14, no. 4, pp. 451–468, 1990.
- [38] R. Bitmead, V. Wertz, and M. Gevers, *Adaptive Optimal Control: The Thinking Man's GPC.* Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [39] A. Bemporad, "Model predictive control design: New trends and tools," in *Proc. 45th IEEE Conf. Dec. Control*, 2006, pp. 6678–6683.
- [40] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [41] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, 1990.
- [42] S. M. Estill, "Real-time receding horizon control: Application programmer interface employing LSSOL," *Dept. Mech. Eng., Univ. California, Berkeley*, Dec. 2003.
- [43] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2005.
- [44] Y. Wang and S. Boyd, "Performance bounds for linear stochastic control," *Syst. Control Lett.*, vol. 53, no. 3, pp. 178–182, Mar. 2009.
- [45] M. Corless and G. Leitmann, "Controller design for uncertain system via Lyapunov functions," in *Proc. Amer. Control Conf.*, 1988, vol. 3, pp. 2019–2025.

- [46] R. A. Freeman and J. A. Primbs, "Control Lyapunov functions, new ideas from an old source," in Proc. 35th IEEE Conf. Dec. Control, 1996, vol. 4, pp. 3926–3931.
- [47] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," SIAM J. Control Opt., vol. 21, no. 3, pp. 462–471, 1983.
- [48] M. Sznaier, R. Suarez, and J. Cloutier, "Suboptimal control of constrained nonlinear systems via receding horizon constrained control Lyapunov functions," Int. J. Robust Nonlinear Control, vol. 13, no. 3–4, pp. 247–259, 2003.
- [49] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," Int. J. Robust Nonlinear Control, vol. 18, no. 8, pp. 816–830, 2008.
- [50] P. Tøndel and T. A. Johansen, "Complexity reduction in explicit linear model predictive control," presented at the 15th IFAC World Congr. Autom. Control, Barcelona, Spain, Jul. 2002.
- [51] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," J. Opt. Theory Appl., vol. 117, no. 1, pp. 9–38, Nov. 2004.
- [52] A. Magnani and S. Boyd, "Convex piecewise-linear fitting," Opt. Eng. Mar. 2008. [Online]. Available: [http://www.stanford.edu/~boyd/cvx\\_pwl\\_fitting.html](http://www.stanford.edu/~boyd/cvx_pwl_fitting.html)
- [53] J. Nocedal and S. J. Wright, Numerical Optimization. New York: Springer, 1999.
- [54] Y. Nesterov and A. Nemirovsky, Interior-Point Polynomial Methods in Convex Programming. Warrendale, PA: SIAM, 1994.
- [55] M. A. Kerblad and A. Hansson, "Efficient solution of second order cone program for model predictive control," Int. J. Control, vol. 77, no. 1, pp. 55–77, Jan. 2004.
- [56] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior point methods to model predictive control," J. Opt. Theory Appl., vol. 99, no. 3, pp. 723–757, Nov. 2004.
- [57] A. G. Wills and W. P. Heath, "Barrier function based model predictive control," Automatica, vol. 40, no. 8, pp. 1415–1422, Aug. 2004.
- [58] S. Boyd and B. Wegbreit, "Fast computation of optimal contact forces," IEEE Trans. Robot., vol. 23, no. 6, pp. 1117–1132, Dec. 2007.
- [59] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. D. Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen, "LAPACK: A portable linear algebra library for high-performance computers," in Proc. Supercomput., 1990, pp. 2–11.

- [60] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK Users' Guide. Warrendale, PA: SIAM, 1999.
- [61] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3: A matlab software package for semidefinite programming," *Opt. Methods Softw.*, vol. 11, no. 12, pp. 545–581, 1999.
- [62] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2006. [Online]. Available: <http://www.stanford.edu/~boyd/cvx>
- [63] K. Ling, B. Wu, and J. Maciejowski, "Embedded model predictive control (MPC) using a FPGA," in *Proc. 17th IFAC World Congr.*, Jul. 2008, pp. 15250–15255.