

Predictive Control

预测控制

第6章 显式预测控制

控制科学与工程

2019年3月

前言

终端不等式约束预测控制

Find $\min_{u^{(*)}} \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N)\|_P^2$

s.t. $x(k) = Ax(k-1) + Bu(k-1)$

$$y(k) = Cx(k)$$

$$x(k) \in X, u(k) \in U, k = 1, 2, \dots, N-1$$

$$x(k+N) \in \Omega$$

当 $x(k+m) \in \Omega$, 取 $u(k) = Lx(k)$

$$P = (A + BL)^T P (A + BL) + Q + L^T R L$$

只要 $A + BL$ 稳定, 则 (A, B) 可控, P 一定存在!

终端不等式约束预测控制

$$\text{Find} \quad \min_{u^{(*)}} \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N)\|_P^2$$

$$\text{s.t.} \quad x(k) = Ax(k-1) + Bu(k-1)$$

$$y(k) = Cx(k)$$

$$x(k) \in X, \quad u(k) \in U, \quad k = 1, 2, \dots, N-1$$

$$x(k+N) \in \Omega$$

通过求解上述约束优化问题，获得最优控制序列

$$U_N^*(k) = \begin{bmatrix} u^*(k) & u^*(k+1) & \cdots & u^*(k+N-1) \end{bmatrix}^T$$

约束预测控制实际应用时面临以下问题：

需要**反复在线优化计算**，只能适应系统动态变化较慢的系统，如过程控制系统，难以适应采样速率较高的系统，如动态变化较快的机电系统、汽车电子控制等。

近年来在减少预测控制在线计算时间、扩大预测控制适用范围方面的研究取得了很多成果，如**显式预测控制**。

第6章 显式预测控制

6.1 问题描述

6.2 多参数规划

6.3 约束线性系统的显式预测控制

6.1 问题描述

约束预测控制

优化问题的决策变量

$$u(k+i), i=0,1,\dots,N-1$$

Find $\min_{u^{(*)}} \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N)\|_P^2$

s.t. $x(k) = Ax(k-1) + Bu(k-1)$

$$y(k) = Cx(k)$$

$$x(k) \in X, u(k) \in U, k=1,2,\dots,N-1$$

$$x(k+N) \in \Omega$$

$$x(k+i), i=0,1,\dots,N$$

定义 $U_N(k) = [u(k) \quad u(k+1) \quad \dots \quad u(k+N-1)]^T$

6.1 问题描述

根据线性系统的状态方程：

$$x(k+1) = Ax(k) + Bu(k)$$

可得状态的递推方程

$$x(k+m) = A^m x(k) + \sum_{j=0}^{m-1} A^j Bu(k+m-1-j), m=1,2,\dots,N$$

则约束预测控制的优化问题可转化为

$$\begin{aligned} \text{Find} \quad & \min_{U_N(k)} \left(\frac{1}{2} (U_N(k))^T H U_N(k) + x^T(k) F U_N(k) \right) + \frac{1}{2} x^T(k) Y x(k) \\ \text{s.t.} \quad & G U_N(k) \leq W + E x(k) \end{aligned}$$

其中， H, F, Y, G, W, E 可根据系统矩阵 A, B 和 Q, R 计算获得。

6.1 问题描述

约束预测控制

$$\text{Find} \quad \min_{u^{(*)}} \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N)\|_P^2$$

$$\text{s.t.} \quad x(k) = Ax(k-1) + Bu(k-1)$$

$$y(k) = Cx(k)$$

$$x(k) \in X, \quad u(k) \in U, \quad k = 1, 2, \dots, N-1$$

$$x(k+N) \in \Omega$$

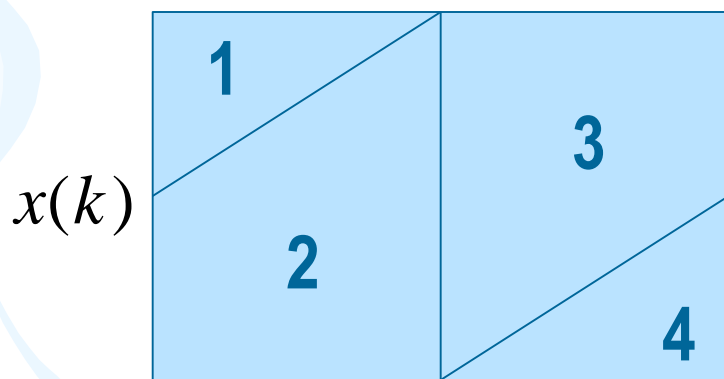
优化问题的解依赖系统当前时刻的状态
 $x(k)$

通过求解上述约束优化问题，获得最优控制序列

$$U_N^*(k) = \begin{bmatrix} u^*(k) & u^*(k+1) & \cdots & u^*(k+N-1) \end{bmatrix}^T$$

6.1 问题描述

对于约束预测控制，如果将系统当前时刻的状态 $x(k)$ 看做优化问题的参数，设法求得 $x(k)$ 在某一区域内时优化问题解的变化规律，即 $u^*(k)$ 与 $x(k)$ 之间的显式函数关系，则



$$u_1 = K_1 x + b_1$$

$$u_2 = K_2 x + b_2$$

$$u_3 = K_3 x + b_3$$

$$u_4 = K_4 x + b_4$$

约束预测控制

通过求解优化问题
获得最优控制序列



◆ 离线划分状态区域

◆ 离线计算决策向量与状态之间的关系

◆ 在线判断系统状态所在区域并计算控制量

6.1 问题描述

对于优化问题

$$\text{Find} \quad \min_{U_N(k)} \left(\frac{1}{2} (U_N(k))^T H U_N(k) + x^T(k) F U_N(k) \right) + \frac{1}{2} x^T(k) Y x(k)$$

$$\text{s.t.} \quad G U_N(k) \leq W + E x(k)$$

$$\text{定义} \quad z = U_N(k) + H^{-1} F^T x(k)$$

则优化问题转化为

$$\text{Find} \quad \min_z \frac{1}{2} z^T H z + \frac{1}{2} x^T(k) \left(Y - F (F H^{-1})^T \right) x(k)$$

$$\text{s.t.} \quad G z \leq W + S x(k)$$

$$\text{其中} \quad S = E + G H^{-1} F^T$$

6.1 问题描述

对于优化问题

$$\text{Find} \quad \min_{U_N(k)} \left(\frac{1}{2} (U_N(k))^T H U_N(k) + x^T(k) F U_N(k) \right) + \frac{1}{2} x^T(k) Y x(k)$$

$$\text{s.t.} \quad G U_N(k) \leq W + E x(k)$$

$$\text{定义} \quad z = U_N(k) + H^{-1} F^T x(k)$$

则优化问题转化为

$$\text{Find} \quad \min_z \frac{1}{2} z^T H z + \frac{1}{2} x^T(k) \left(Y - F (F H^{-1})^T \right) x(k)$$

$$\text{s.t.} \quad G z \leq W + S x(k)$$

$$\text{其中} \quad S = E + G H^{-1} F^T$$

z 为决策变量
 x 为参数向量

6.1 问题描述

最终将约束预测控制的在线优化问题转化为

$$\begin{array}{ll} \text{Find} & \min_z \frac{1}{2} z^T H z \\ \text{s.t.} & Gz \leq W + Sx \end{array}$$

优化问题中参数向量只出现在约束表达式的右侧

即为**多参数规划问题**。

通过求解上述多参数规划问题，可以获得

- 参数分区 $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\} i = 1, 2, \dots, r$, r 为参数分区数量
- 每个参数分区对应的优化问题解与参数之间的**显式**表达式

$$z_i^* = K_i x + b_i$$

6.1 问题描述

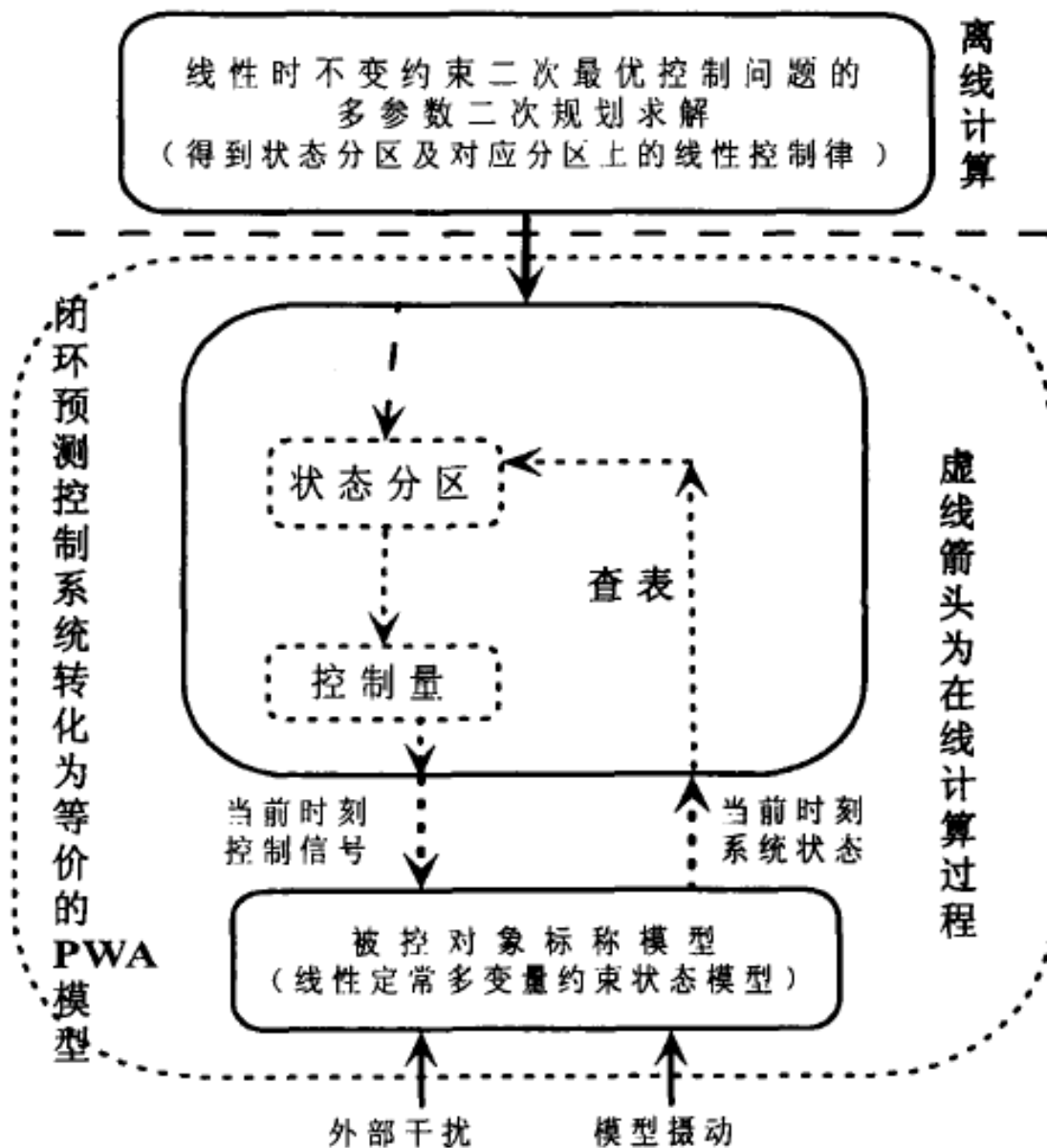
显式预测控制

- 1) 其主要思想是把**多参数规划**理论引入到线性时不变对象的约束二次优化控制问题的求解中；
- 2) 对系统的状态区域(即参数区域)进行划分，离线计算得到对应每个状态分区上的状态反馈**显式控制律**；
- 3) 在线计算就只需要查找当前状态所在的分区就可以获得相应的控制律，从而获得当前所需要的控制量。

6.1 问题描述

显式预测控制

算法步骤:



6.1 问题描述

显式预测控制

显式预测控制算法的实现步骤：

1) 离线计算

- ◆ 应用 **多参数规划** 方法对系统的状态区域进行划分，计算得到每个状态分区上的状态反馈最优控制律

2) 在线计算

- ◆ 确定当前时刻的系统状态所处分区，按照该分区上的控制律计算当前时刻的控制量

6.1 问题描述

显式预测控制的意义：

- 1) 降低了在线计算时间，有利于应用于采样速度高的系统并且控制律的实现简单，有望在嵌入式系统实现。
- 2) 获得控制律的解析表达式有助于进行控制器性能分析，如稳定性分析、可行性分析、鲁棒性分析等。

因此，扩大预测控制的应用范围到新的领域，如汽车、航空电子、机电一体化等，以及其它需要在线快速计算的领域。

第6章 显式预测控制

6.1 问题描述

6.2 多参数规划

6.3 约束线性系统的显式预测控制

6.2 多参数规划

通常在两个层面分析参数变化对系统的影响：

1) 灵敏度分析 (Sensitivity Analysis)

分析系统状态或输出对参数摄动的敏感程度

2) 参数规划 (Parametric Programming)

参数在所有可能的范围内变化时，问题的解随之变化表现出来的规律

多参数规划 (Mutliparametric Programming)

分割参数区域，在每一个参数子区域内建立优化问题的解与参数之间的函数关系；一旦确定参数值，即可获得问题的解。

6.2 多参数规划

6.2.1 多参数线性规划

考虑如下形式的多参数线性规划（Mutliparametric Linear Programming, mpLP）问题

$$\begin{aligned} \text{Find } J(z, x) &= \min_z c^T z \\ \text{s.t. } &Gz \leq W + Sx \end{aligned}$$

其中 $z \in \mathbb{R}^s$ 为优化变量, $x \in \mathbb{R}^n$ 为参数向量, $J(z, x) \in \mathbb{R}$ 为目标函数, 此外 $c \in \mathbb{R}^s$, $G \in \mathbb{R}^{m \times s}$, $W \in \mathbb{R}^m$, $S \in \mathbb{R}^{m \times n}$

6.2 多参数规划

给定参数集

$$K = \{x \in \mathbb{R}^n : Tx \leq Z\}$$

可行域：满足约束条件的解组成的集合

用 $K^* \subseteq K$ 表示多参数线性规划问题的可行参数域

目标是通过求解多参数线性规划问题得到：

- 在参数集 K 内找到参数的可行域 K^*
- 计算目标函数 $J(z, x)$
- 获得优化问题解的表达式 $z^*(x) = \operatorname{argmin}_z J(z, x)$

6.2 多参数规划

临界域的定义

有效约束：对于优化问题

$$\begin{array}{ll}\min & h(x) \\ \text{s.t.} & f(x) \leq 0\end{array}$$

如果点 x 是可行的，且 $f(x) = 0$ ，则称不等式约束 $f(x) \leq 0$ 在点 x 上是有效（Active）的，如果 $f(x) < 0$ ，则该约束在点 x 上是非有效（Inactive）的。

6.2 多参数规划

临界域的定义

最优划分： 对于多参数线性规划问题

$$\begin{aligned} \text{Find } J(z, x) &= \min_z c^T z \\ \text{s.t. } Gz &\leq W + Sx \end{aligned}$$

$$G \in \mathbb{R}^{m \times s}$$

$$W \in \mathbb{R}^m$$

$$S \in \mathbb{R}^{m \times n}$$

如果用 P 表示由下标 $\{1, 2, \dots, m\}$ 索引的约束集合，则 P 的最优划分（**Optimal Partition**）为 $(A(x), NA(x))$

$$A(x) = \left\{ j \in \{1, 2, \dots, m\} : G_j z - S_j x = W_j, \forall z^*(x) \in Z^*(x) \right\}$$

$$NA(x) = \left\{ j \in \{1, 2, \dots, m\} : G_j z - S_j x < W_j, \exists z^*(x) \in Z^*(x) \right\}$$

其中 G_j, W_j, S_j 分别表示矩阵 G, W, S 的第 j 行。

6.2 多参数规划

临界域的定义

对于给定的 $x^* \in K^*$ ，令

$$(A, NA) = (A(x^*), NA(x^*))$$

定义

$$CR_A = \{x \in K : A(x) = A\}$$

有效约束形成的集合

则 CR_A 是有效约束集相关的临界域（**Critical Region, CR**）

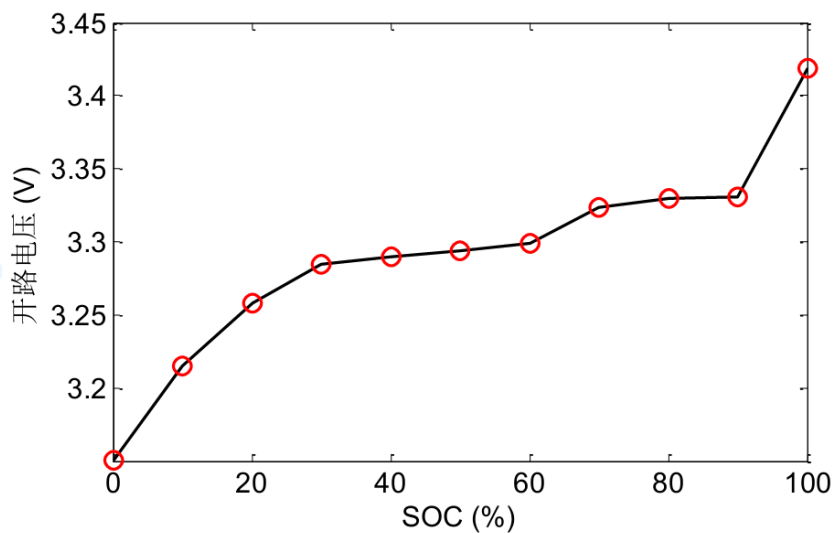
6.2 多参数规划

分段仿射函数(Piecewise Affine Function)

又称为PWA函数

$$f(x) = Kx + b$$

常用于近似描述非线性系统



6.2 多参数规划

仿射空间

仿射空间是一个点集，它的定义是：

- ◆ 设 A 为一个点集， A 中任意两个有序点 P 、 Q 对应于 n 维向量空间中的一个向量 a ；
- ◆ 设 P 、 Q 、 R 为 A 中任意三点， P 、 Q 对应于向量 a ， Q 、 R 对应于向量 b ，则 P 、 R 对应于向量 $a+b$ 。

具有上面两个性质的点集 A 就称为一个仿射空间。

6.2 多参数规划

多参数线性规划算法的主要内容：

□ 计算可行参数集 K^* 的最小仿射子空间----减少参数数量

为降低计算复杂度，希望能够对最小维数的参数向量求解多参数规划问题，需找到包含可行参数的最小仿射子空间

□ 划分可行参数集并求取最优解 $z^* = Kx + b$

得到可行参数集 K^* 的临界域划分，并对每一个临界域建立优化问题的解与参数之间的函数关系

6.2 多参数规划

计算可行参数集 K^* 的最小仿射子空间

对于约束 $Gz \leq W + Sx$

记矩阵 S 的秩为 r_s ，则

$$G \in \mathbb{R}^{m \times s}$$

$$W \in \mathbb{R}^m$$

$$S \in \mathbb{R}^{m \times n}$$

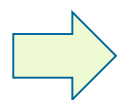
- 1) 如果 $r_s < n$ ，则可通过坐标变换消除 $n - r_s$ 个参数
- 2) 即使 S 是列满秩 ($r_s = n$) 的也可能存在维数为 h ($h \leq n$) 且包含可行参数集 K^* 的最小仿射子空间

6.2 多参数规划

例子：考虑如下的mpLP问题

$$\text{Find } J(z, x) = \min_z 2z_1 + 3z_2$$

$$\text{s.t. } \begin{cases} z_1 + z_2 \leq 9 - x_1 - x_2 \\ z_1 - z_2 \leq 1 - x_1 - x_2 \\ z_1 + z_2 \leq 7 + x_1 + x_2 \\ z_1 - z_2 \leq -1 + x_1 + x_2 \\ -z_1 \leq -4 \\ -z_2 \leq -4 \\ z_1 \leq 20 - x_2 \end{cases}$$



$$Gz \leq W + Sx \quad S = \begin{bmatrix} -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

问题的解 $z_1^* = 4, z_2^* = 4 \quad \forall (x_1, x_2) \in CR_1$

临界域 $CR_1 \quad \begin{cases} x_1 + x_2 = 1, \\ -100 \leq x_1 \leq 100, -100 \leq x_2 \leq 100 \end{cases}$

其中 $K^* = \{x : -100 \leq x_1 \leq 100, -100 \leq x_2 \leq 100\}$

6.2 多参数规划

计算可行参数集 K^* 的最小仿射子空间

对于约束 $Gz \leq W + Sx$

记矩阵 S 的秩为 r_s ，则

$$G \in \mathbb{R}^{m \times s}$$

$$W \in \mathbb{R}^m$$

$$S \in \mathbb{R}^{m \times n}$$

- 1) 如果 $r_s < n$ ，则可通过坐标变换消除 $n - r_s$ 个参数
- 2) 即使 S 是列满秩 ($r_s = n$) 的也可能存在维数为 h ($h \leq n$) 且包含可行参数集 K^* 的最小仿射子空间

具体计算方法详见张聚的《显式模型预测控制理论与应用》

6.2 多参数规划

计算临界域

根据最优划分 $(A(x), NA(x))$

$$A(x) = \{j \in \{1, 2, \dots, m\} : G_j z - S_j x = W_j, \forall z^*(x) \in Z^*(x)\}$$

$$NA(x) = \{j \in \{1, 2, \dots, m\} : G_j z - S_j x < W_j, \exists z^*(x) \in Z^*(x)\}$$

得

$$G_A z^*(x) = W_A + S_A x$$

$$G_{AN} z^*(x) < W_{AN} + S_{AN} x$$

进一步

$$z^*(x) = G_A^{-1} W_A + G_A^{-1} S_A x = Q + Ex$$

$$G_{AN} (Q + Ex) < W_{AN} + S_{AN} x$$

6.2 多参数规划

计算临界域

因此对于多参数规划问题

$$\begin{aligned} \text{Find } J(z, x) &= \min_z c^T z \\ \text{s.t. } Gz &\leq W + Sx \end{aligned}$$

可得临界域 CR_0

$$G_{AN}(Q + Ex) < W_{AN} + S_{AN}x \xrightarrow{\text{闭包}} G_{AN}(Q + Ex) \leq W_{AN} + S_{AN}x$$

6.2 多参数规划

剩余空间的划分方法

令 $Y \subseteq \mathbb{R}^n$ 为一个多面体, $CR_0 \triangleq \{x \in Y: Ax \leq b\}$ 是 Y 的多面体子集, 且不为空集, 其中 $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, 令

$$R_i = \left\{ x \in Y: \begin{array}{l} A^i x > b^i \\ A^j x \leq b^j, \forall j < i \end{array} \right\}, \quad i = 1, \dots, m,$$

$$CR^{\text{rest}} \triangleq \bigcup_{i=1}^m R_i$$

则有如下结论

- 1) $CR^{\text{rest}} \cup CR_0 = Y$
- 2) $CR_0 \cap R_i = \emptyset, R_i \cap R_j = \emptyset \quad \forall i \neq j$
- 3) $\{CR_0, R_1, \dots, R_m\}$ 是 Y 的一个划分

6.2 多参数规划

剩余空间划分的例子

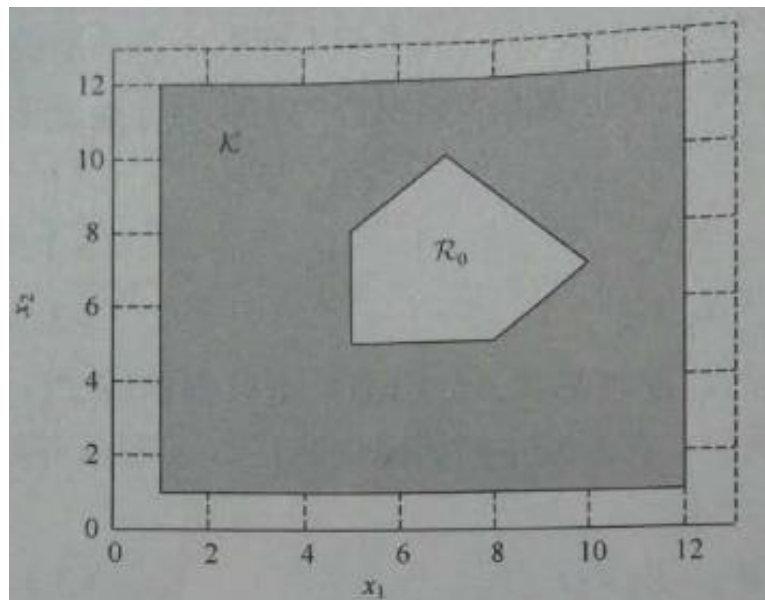
定义参数空间为 $K = \{x : 1 \leq x_1 \leq 12, 1 \leq x_2 \leq 12\}$, 初始域

$\mathfrak{R}_0 = \{x : C_1 \leq 0, C_2 \leq 0, C_3 \leq 0, C_4 \leq 0, C_5 \leq 0\}$, 其中

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

下面给出剩余空间 $\mathfrak{R}^{rest} = K - \mathfrak{R}_0$ 的划分过程。

6.2 多参数规划



参数空间 $K = \{x : 1 \leq x_1 \leq 12, 1 \leq x_2 \leq 12\}$

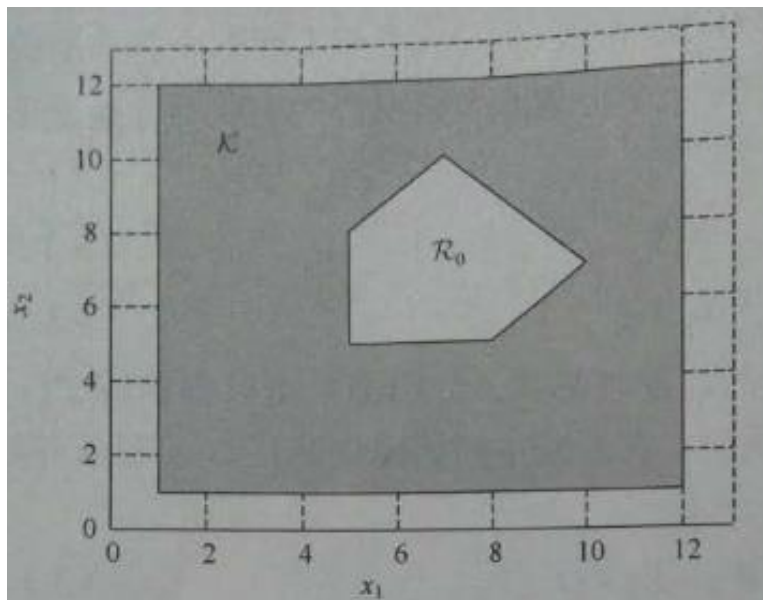
初始 $\mathfrak{R}_0 = \{x : C_1 \leq 0, C_2 \leq 0, C_3 \leq 0, C_4 \leq 0, C_5 \leq 0\}$

$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$

6.2 多参数规划

多面体的定义

有限数量的线性等式方程和线性不等式方程的解集



注意

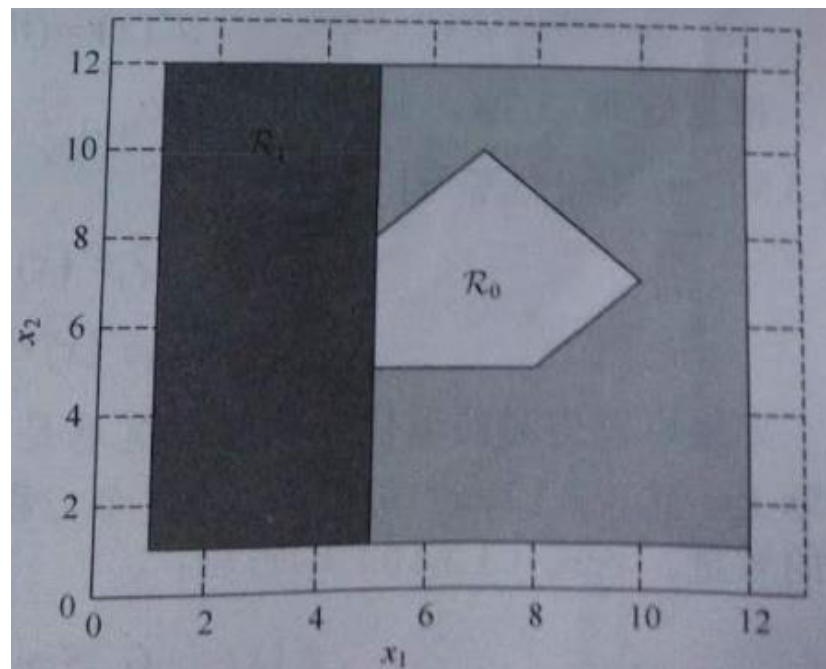
多面体的边数与约束个数之间的关系

参数空间 $K = \{x : 1 \leq x_1 \leq 12, 1 \leq x_2 \leq 12\}$

初始 $\mathcal{R}_0 = \{x : C_1 \leq 0, C_2 \leq 0, C_3 \leq 0, C_4 \leq 0, C_5 \leq 0\}$

$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$

6.2 多参数规划

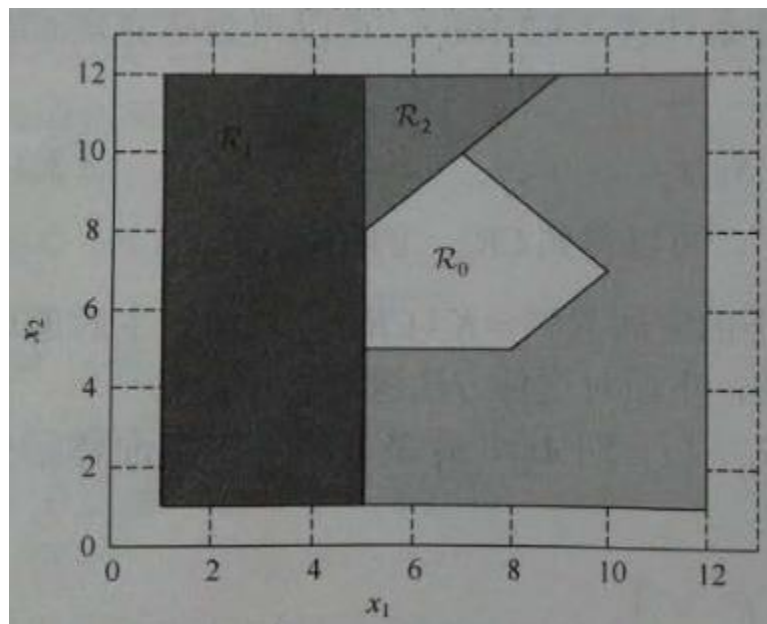


第一步划分 $\mathfrak{R}_1 = \{x : C_1 \geq 0, x_1 \geq 1, 1 \leq x_2 \leq 12\}$

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

通过不等式 $C_1 \leq 0$ 取反，并消除在 K 中的冗余约束得到。

6.2 多参数规划

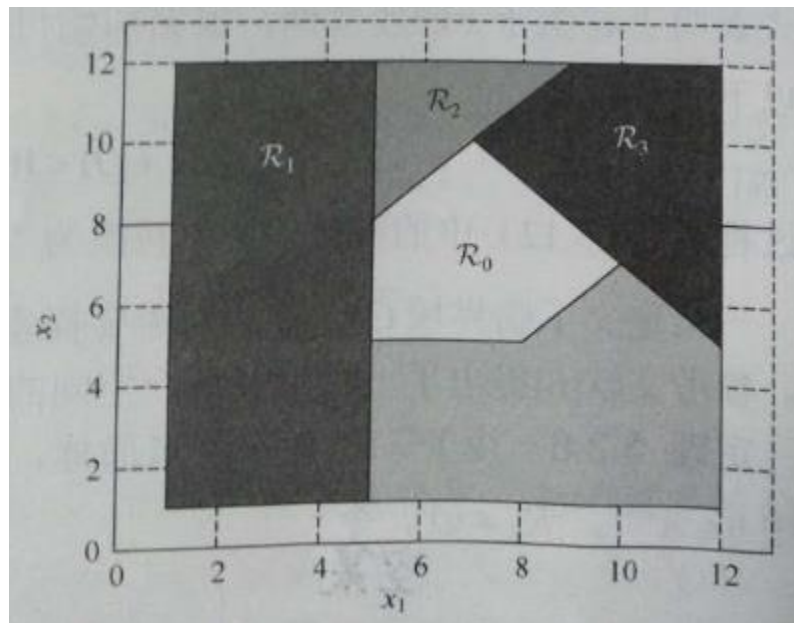


第二步划分 $\mathcal{R}_2 = \{x : C_2 \geq 0, C_1 \leq 0, x_2 \leq 12\}$

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

通过不等式 $C_2 \leq 0$ 取反，并消除在 K 中的冗余约束得到。

6.2 多参数规划

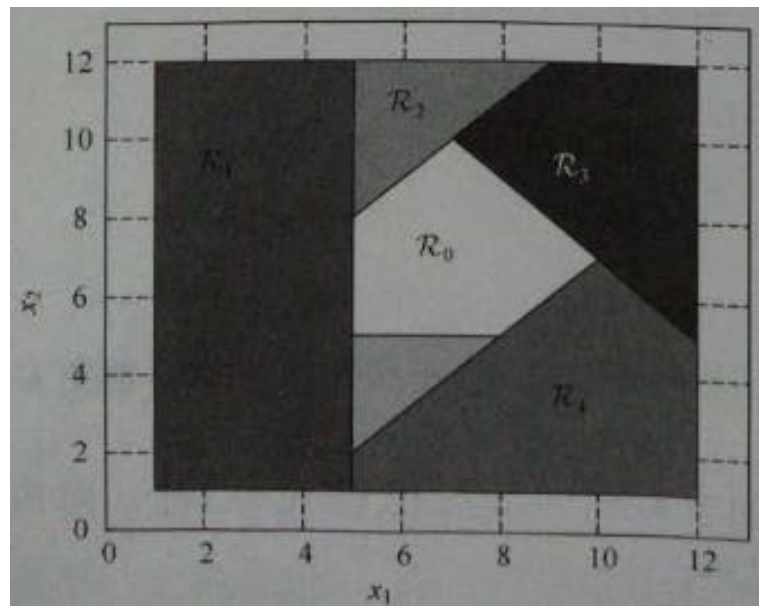


第三步划分 $\mathcal{R}_3 = \{x : C_3 \geq 0, C_2 \leq 0, x_1 \leq 12, x_2 \leq 12\}$

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

通过不等式 $C_3 \leq 0$ 取反，并消除在 K 中的冗余约束得到。

6.2 多参数规划

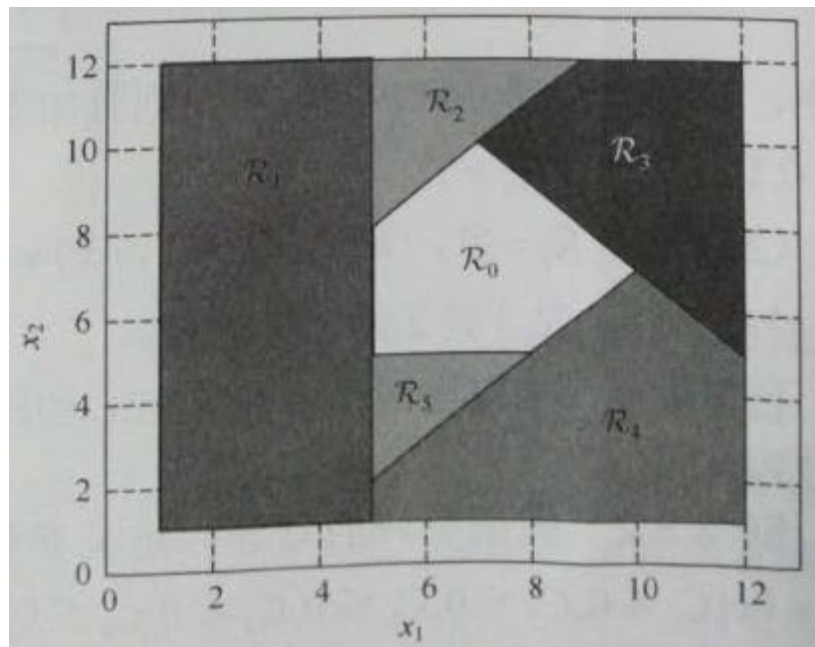


第四步划分 $\mathcal{R}_4 = \{x : C_4 \geq 0, C_1 \leq 0, C_3 \leq 0, x_1 \leq 12, x_2 \geq 1\}$

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

通过不等式 $C_4 \leq 0$ 取反，并消除在 K 中的冗余约束得到。

6.2 多参数规划



最后一步划分 $\mathcal{R}_5 = \{x : C_5 \geq 0, C_1 \leq 0, C_4 \leq 0\}$

$$C_1 : -x_1 + 5, C_2 : -x_1 + x_2 - 3, C_3 : x_1 + x_2 - 17, C_4 : x_1 - x_2 - 3, C_5 : -x_2 + 5$$

通过不等式 $C_5 \leq 0$ 取反，并消除在 K 中的冗余约束得到。

6.2 多参数规划

多参数线性规划的Matlab求解函数

[Pn, Fi, Gi, activeConstraints, Phard, details] = mpt_mplp(Matrices, Options)

MPT工具箱

DESCRIPTION

Multiparametric linear programming

Solves the problem

$$V(x) = \min_U H U + F x$$

$$\begin{aligned} \text{s.t.} \quad & G U \leq W + E x \\ & \text{bndA} * x \leq \text{bndb} \end{aligned}$$

Find $J(z, x) = \min_z c^T z$

s.t. $Gz \leq W + Sx$

As a solution we get 'nR' regions

$$Pn(i) = \{x : H x \leq K\}$$

域的划分

with the optimal control law

$$U = Fi\{i\} x + Gi\{i\}$$

最优解表达式

and the corresponding cost function expression

$$V(x) = Bi\{i\} x + Ci\{i\}$$

目标函数

6.2 多参数规划

多参数线性规划的Matlab求解函数

```
[Pr, Fi, Gi, activeConstraints, Phard, details]=mpt_mplp(Matrices, Options)
```

DESCRIPTION

Multiparametric linear programming

Solves the problem

$$V(x) = \min_U H U + F x$$

$$\begin{aligned} \text{s.t.} \quad & G U \leq W + E x \\ & \text{bndA} * x \leq \text{bndb} \end{aligned}$$

INPUT

Matrices - a struct with all the parameters which are needed.
See description above for explanation.

```
Matrices.H=H;  
Matrices.G=G;  
Matrices.W=W;  
Matrices.E=E;  
Matrices.bndA=bndA;  
Matrices.bndb=bndb;
```

6.2 多参数规划

多参数线性规划的Matlab求解函数

```
[Pr, Fi, Gi, activeConstraints, Phard, details]=mpt_mplp (Matrices, Options)
```

DESCRIPTION

Multiparametric linear programming

调用的求解器，注意选用original mpLP

Solves the problem

$$\begin{aligned} V(x) = \min & H U + F x \\ & U \\ \text{s.t.} \quad & G U \leq W + E x \\ & \text{bndA} * x \leq \text{bndb} \end{aligned}$$

Options.mplpver	- <u>version of mpLP solver to use (3 is default)</u>
Options.verbose	- level of verbosity
Options.lpsolver	- which LP solver to use (help mpt_solveLP)
Options.max_iter	- maximum number of iterations of the algorithm
Options.step_size	- length of step over a facet
Options.f_perturb	- Perturbation of the optimization direction
Options.nu	- How many elements to extract from the optimizer (to deal with slacks)
Options.debug_level	

Due to numerical problems tiny regions are sometimes difficult to calculate, i.e. are not identified at all. This may create "gaps" in the computed control law. For the exploration, these will be jumped over and the exploration in the state space will continue. "debug_level" can have three values:

6.2 多参数规划

多参数线性规划的Matlab求解函数

[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mplp(Matrices,Options)

DESCRIPTION

Multiparametric linear programming

Solves the problem

$$\begin{aligned} V(x) = \min & \quad H U + F x \\ & \quad U \\ \text{s.t.} \quad & \quad G U \leq W + E x \\ & \quad \text{bndA} * x \leq \text{bndb} \end{aligned}$$

OUTPUT

- | | |
|-------------------|---|
| Pn,Fi,Gi | - for region Pn(i).H*x <= Pn(i).K computed input is U=Fi{i}*x+Gi{i} |
| activeConstraints | - Cell Array which stores the active constraints of the optimizer in each region. |
| Phard | - Defines the feasible state space partition (i.e. all regions) as Phard.H*x<=Phard.K |
| details | - a structure with the following fields: |

6.2 多参数规划

多参数线性规划的Matlab求解函数

```
[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mplp(Matrices,Options)
```

DESCRIPTION

Multiparametric linear programming

	details	- a structure with the following fields:
Solves the problem	nR	number of regions
$V(x) = \min_U H U + F x$	Pn	polyhedral partition
	Fi	control law
s.t. $G U \leq W + E x$	Gi	control law
$bndA*x \leq bndb$	BC	connection list
	Bi	value function
	Ci	value function
	nHard	number of hard constraints
	Phard	polytope given by hard constraints
	nb	number of constraints for each region
	LISTa	list of active constraints

6.2 多参数规划

多参数线性规划例子

考虑如下例子

$$\begin{array}{ll}\min_{z(x)} & -3z_1 - 8z_2 \\ \text{subj. to} & z_1 + z_2 \leq 13 + x_1 \\ & 5z_1 - 4z_2 \leq 20 \\ & -8z_1 + 22z_2 \leq 121 + x_2 \\ & -4z_1 - z_2 \leq -8 \\ & -z_1 \leq 0 \\ & -z_2 \leq 0 \\ & -11 \leq x_1 \leq 20 \\ & -20 \leq x_2 \leq 20\end{array}$$

$$\begin{array}{ll}\text{Find} & J(z, x) = \min_z c^T z \\ \text{s.t.} & Gz \leq W + Sx\end{array}$$



6.2 多参数规划

多参数线性规划例子

考虑如下例子

$$\begin{array}{ll}\min_{z(x)} & -3z_1 - 8z_2 \\ \text{subj. to} & z_1 + z_2 \leq 13 + x_1 \\ & 5z_1 - 4z_2 \leq 20 \\ & -8z_1 + 22z_2 \leq 121 + x_2 \\ & -4z_1 - z_2 \leq -8 \\ & -z_1 \leq 0 \\ & -z_2 \leq 0 \\ & -11 \leq x_1 \leq 20 \\ & -20 \leq x_2 \leq 20\end{array}$$

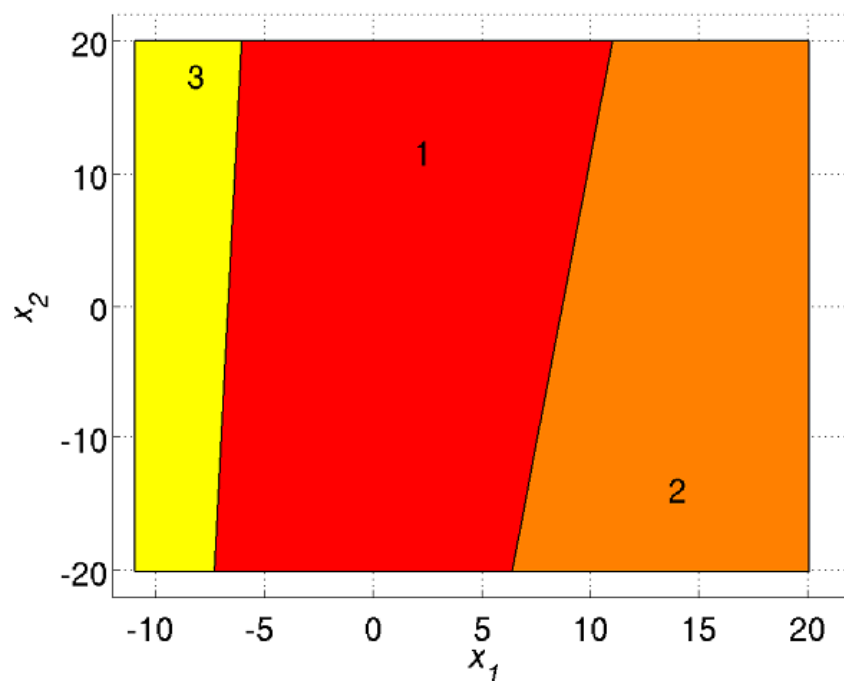
$$\begin{array}{ll}\text{Find} & J(z, x) = \min_z c^T z \\ \text{s.t.} & Gz \leq W + Sx\end{array}$$



6.2 多参数规划

多参数线性规划例子

应用函数mpt_mplp求解，可得

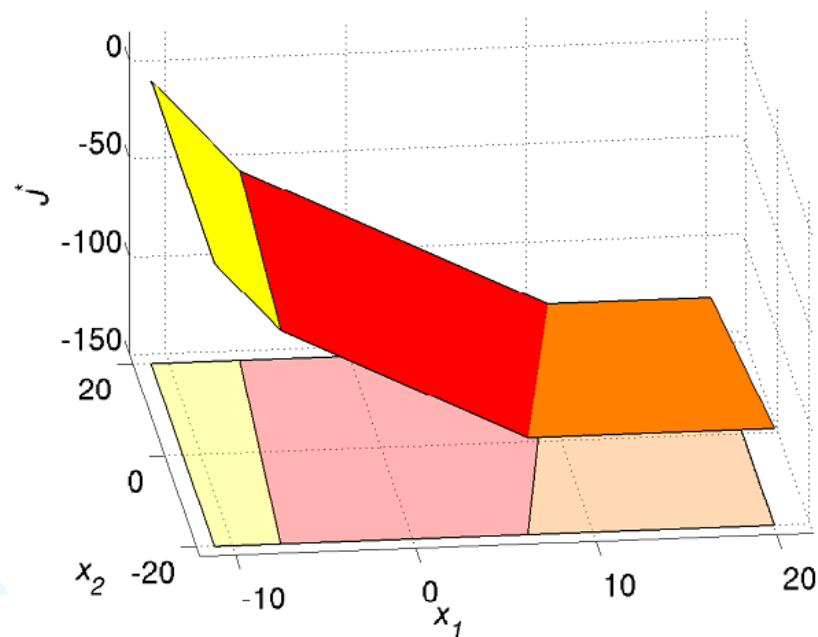


临界域 $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\} \quad i = 1, \dots, 3$

6.2 多参数规划

多参数线性规划例子

应用函数mpt_mplp求解，可得



每个域对应的优化目标函数 $J^*(x)$

6.2 多参数规划

多参数线性规划例子

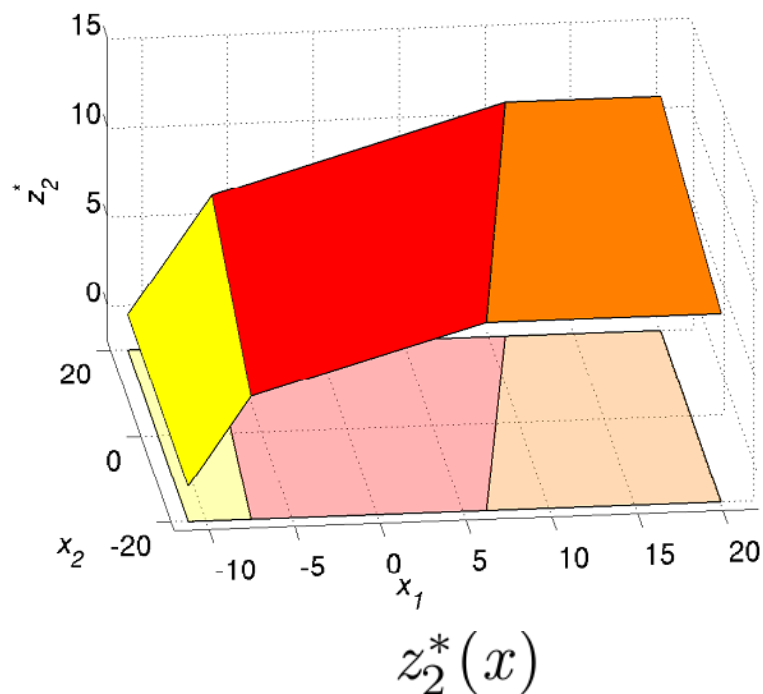
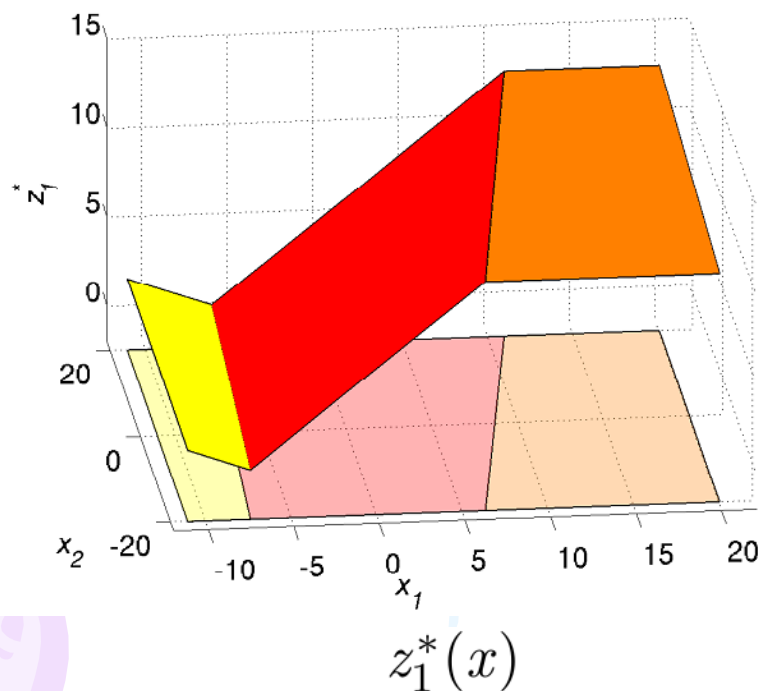
应用函数mpt_mplp求解，可得每个分区的最优解表达式

$$z^*(x) = \begin{cases} \begin{pmatrix} 0.733 & -0.033 \\ 0.267 & 0.033 \end{pmatrix} x + \begin{pmatrix} 5.5 \\ 7.5 \end{pmatrix} & \text{if } x \in \mathcal{P}_1 \\ \begin{pmatrix} 0 & 0.051 \\ 0 & 0.064 \end{pmatrix} x + \begin{pmatrix} 11.846 \\ 9.808 \end{pmatrix} & \text{if } x \in \mathcal{P}_2 \\ \begin{pmatrix} -0.333 & 0 \\ 1.333 & 0 \end{pmatrix} x + \begin{pmatrix} -1.667 \\ 14.667 \end{pmatrix} & \text{if } x \in \mathcal{P}_3 \end{cases}$$

6.2 多参数规划

多参数线性规划例子

应用函数mpt_mplp求解，可得每个分区的最优解表达式



6.2 多参数规划

6.2.2 多参数二次规划

考虑如下形式的多参数二次规划（Mutliparametric quadratic Programming, mpQP）问题

$$\begin{aligned} \text{Find } J(z, x) &= \min_z \frac{1}{2} z^T H z \\ \text{s.t. } Gz &\leq W + Sx \end{aligned}$$

其中 $z \in \mathbb{R}^s$ 为优化变量, $x \in \mathbb{R}^n$ 为参数向量, $J(z, x) \in \mathbb{R}$ 为目标函数, $G \in \mathbb{R}^{m \times s}$, $W \in \mathbb{R}^m$, $S \in \mathbb{R}^{m \times n}$, 此外 $H \in \mathbb{R}^{s \times s}$, 且 $H > 0$

6.2 多参数规划

给定参数集

$$K = \{x \in \mathbb{R}^n : Tx \leq Z\}$$

用 $K^* \subseteq K$ 表示多参数二次规划问题的可行参数域

目标是通过求解多参数二次规划问题得到：

- 在参数集 K 内找到参数的可行域 K^*
- 计算目标函数 $J(z, x)$
- 获得优化问题解的表达式 $z^*(x) = \operatorname{argmin}_z J(z, x)$

6.2 多参数规划

多参数二次规划算法的主要内容：

□ 计算可行参数集 K^* 的最小仿射子空间----减少参数数量

为降低计算复杂度，希望能够对最小维数的参数向量求解多参数规划问题，需找到包含可行参数的最小仿射子空间

□ 划分可行参数集并求取最优解 $z^* = Kx + b$

得到可行参数集 K^* 的临界域划分，并对每一个临界域建立优化问题的解与参数之间的函数关系

6.2 多参数规划

mpQP问题一阶Karush-Kuhn-Tucker (KKT) 条件

$$Hz^* + G^T \lambda = 0, G \in \mathbb{R}^{m \times s}, \lambda \in \mathbb{R}^m$$

$$\lambda_i (G_i z^* - W_i - S_i x) = 0, i = 1, 2, \dots, m$$

$$\lambda \geq 0$$

$$Gz^* \leq W + Sx$$

根据 $Hz^* + G^T \lambda = 0, G \in \mathbb{R}^{m \times s}, \lambda \in \mathbb{R}^m$

得 $z^* = -H^{-1}G^T \lambda$, 并代入 $\lambda_i (G_i z^* - W_i - S_i x) = 0, i = 1, 2, \dots, m$

得互补松弛条件 $\lambda(-GH^{-1}G^T \lambda - W - Sx) = 0$

6.2 多参数规划

对于互补松弛条件

$$\lambda(-GH^{-1}G^T\lambda - W - Sx) = 0$$

令 λ_{NA_0} 和 λ_{A_0} 分别表示对应的非有效约束和有效约束

则对非有效约束 $\lambda_{NA_0} = 0$

对有效约束，有

$$\lambda_{A_0}(-G_{A_0}H^{-1}G_{A_0}^T\lambda_{A_0} - W_{A_0} - S_{A_0}x) = 0$$

故 $\lambda_{A_0} = -(G_{A_0}H^{-1}G_{A_0}^T)^{-1}(W_{A_0} + S_{A_0}x)$

6.2 多参数规划

由 $\lambda_{A_0} = -(G_{A_0} H^{-1} G_{A_0}^T)^{-1} (W_{A_0} + S_{A_0} x)$ 和 $z^* = -H^{-1} G^T \lambda$

可得

$$z^* = H^{-1} G^T (G_{A_0} H^{-1} G_{A_0}^T)^{-1} (W_{A_0} + S_{A_0} x)$$

可见, z^* 是 x 的分段仿射函数。

6.2 多参数规划

构造 CR_0

根据条件 $\lambda \geq 0, Gz^* \leq W + Sx$

$$\text{及 } \lambda_{A_0} = -(G_{A_0} H^{-1} G_{A_0}^T)^{-1} (W_{A_0} + S_{A_0} x)$$

$$z^* = H^{-1} G^T (G_{A_0} H^{-1} G_{A_0}^T)^{-1} (W_{A_0} + S_{A_0} x)$$

$$\text{得 } \begin{cases} GH^{-1}G^T(G_{A_0}H^{-1}G_{A_0}^T)^{-1}(W_{A_0}+S_{A_0}x) \leq W+Sx \\ -(G_{A_0}H^{-1}G_{A_0}^T)^{-1}(W+Sx) \geq 0 \end{cases}$$

通过消除上述不等式中的冗余不等式，可获得 CR_0

剩余空间的划分可参考多参数线性规划中的方法。

6.2 多参数规划

多参数二次规划的Matlab求解函数

```
[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mqp(Matrices,Options)
```

MPT工具箱

DESCRIPTION

Solves the following QP as a multiparametric program:

$\min_U \quad 0.5 U' H U + \cancel{x(0)' F + C_f} U + \cancel{x(0)' Y x(0) + C_x x + C_c}$

subj. to $GU \leq W + Ex$ (constraints)

$bndA*x \leq bndb$ (bound exploration space)



$$\text{Find } J(z, x) = \min_z \frac{1}{2} z^T H z$$

$$\text{s.t. } Gz \leq W + Sx$$

6.2 多参数规划

多参数二次规划的Matlab求解函数

```
[Pn, Fi, Gi, activeConstraints, Phard, details]=mpt_mqpq(Matrices, Options)
```

DESCRIPTION

Solves the following QP
 $\min_U \quad 0.5 U' H U + (x' G U + x' E)$
subj. to $GU \leq W + Ex$
 $bndA * x \leq bndb$

INPUT

Matrices - a struct with all the parameters which are needed.
See description above for explanation.

```
Matrices.G=G;  
Matrices.E=E;  
Matrices.W=W;  
Matrices.H=H;  
Matrices.F=F;  
Matrices.Y=Y;  
Matrices.Cf=Cf;  
Matrices.Cx=Cx;  
Matrices.Cc=Cc;  
Matrices.bndA=bndA;  
Matrices.bndb=bndb;
```

6.2 多参数规划

多参数二次规划的Matlab求解函数

```
[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mqpq(Matrices,Options)
```

DESCRIPTION

Solves the following QP as a multiparametric program:

$$\min_U \quad 0.5 U' H U + (x(0)' F + C_f) U + x(0)' Y x(0) + C_x x + C_c$$

subj. to $GU \leq W + Ex$ (constraints)

$bndA*x \leq bndb$ (bound exploration space)

Options.verbose	- Optional: level of verbosity
Options.lpsolver	- Optional: which LP solver to use (help mpt_solveLP)
Options.qpsolver	- Optional: which QP solver to use (help mpt_solveQP)
Options.step_size	- Optional: length of step over a facet; Making this value larger often mitigates numerical problems but may produce small gaps in the partition. Default is 1e-4;
Options.debug_level	

6.2 多参数规划

多参数二次规划的Matlab求解函数

```
[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mqpq(Matrices,Options)
```

DESCRIPTION

Solves the following QP as a multiparametric program:

$$\min_U \quad 0.5 U' H U + (x(0)' F + C_f) U + x(0)' Y x(0) + C_x x + C_c$$

subj. to $GU \leq W + Ex$

$bndA*x \leq bndb$

OUTPUT

- | | |
|-------------------|--|
| Pn,Fi,Gi | - for region Pn(i).H*x <= Pn(i).K computed input is $U=Fi\{i\}*x+Gi\{i\}$ |
| activeConstraints | - Cell Array which stores the active constraints of the optimizer in each region. |
| Phard | - Defines the feasible state space partition (i.e. union of all regions) as $Phard.H*x \leq Phard.K$ |
| details | - a structure with the following fields: |

6.2 多参数规划

多参数二次规划的Matlab求解函数

```
[Pn,Fi,Gi,activeConstraints,Phard,details]=mpt_mppq(Matrices,Options)
```

DESCRIPTION

Solves the following QP as a multiparametric program:

$$\min_U \quad 0.5 U' H U + (x(0)' F + C_f) U + x(0)' Y x(0) + C_x x + C_c$$

subj. to $GU \leq W + Ex$

$bndA*x \leq bndb$

details - a structure with the following fields:

nR	number of regions
Pn	polyhedral partition
Fi	control law
Gi	control law
BC	connection list
Bi	value function
Ci	value function
nHard	number of hard constraints
Phard	polytope given by hard constraints
nb	number of constraints for each region
LISTa	list of active constraints

6.2 多参数规划

多参数二次规划例子

考虑如下例子

$$\min_{z(x)} \quad \frac{1}{2}(z_1^2 + z_2^2)$$

subj. to

$$z_1 \leq 1 + x_1 + x_2$$

$$-z_1 \leq 1 - x_1 - x_2$$

$$z_2 \leq 1 + x_1 - x_2$$

$$-z_2 \leq 1 - x_1 + x_2$$

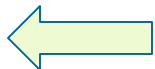
$$z_1 - z_2 \leq x_1 + 3x_2$$

$$-z_1 + z_2 \leq -2x_1 - x_2$$

$$-1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1$$

Find $J(z, x) = \min_z \frac{1}{2} z^T H z$

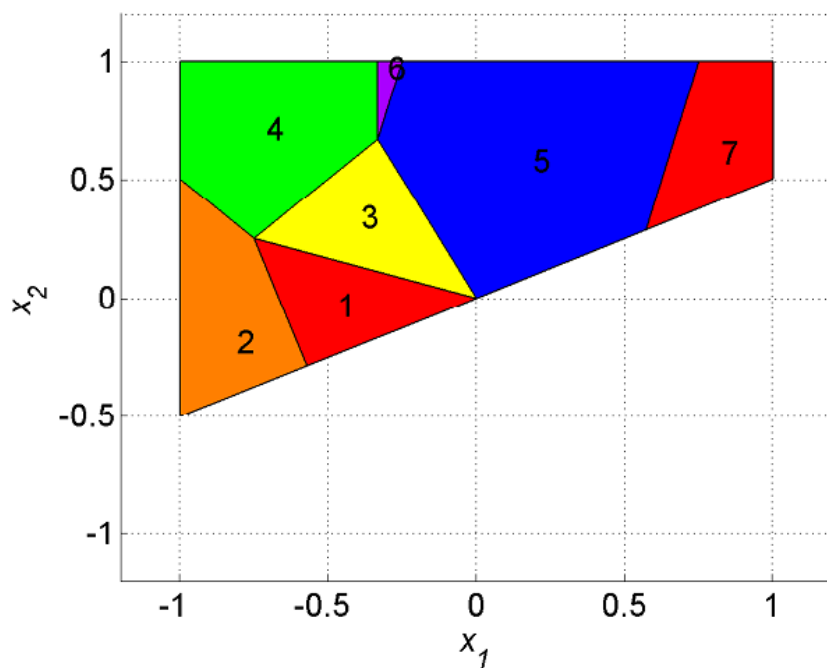
s.t. $Gz \leq W + Sx$



6.2 多参数规划

多参数二次规划例子

应用函数mpt_mpqp求解，可得

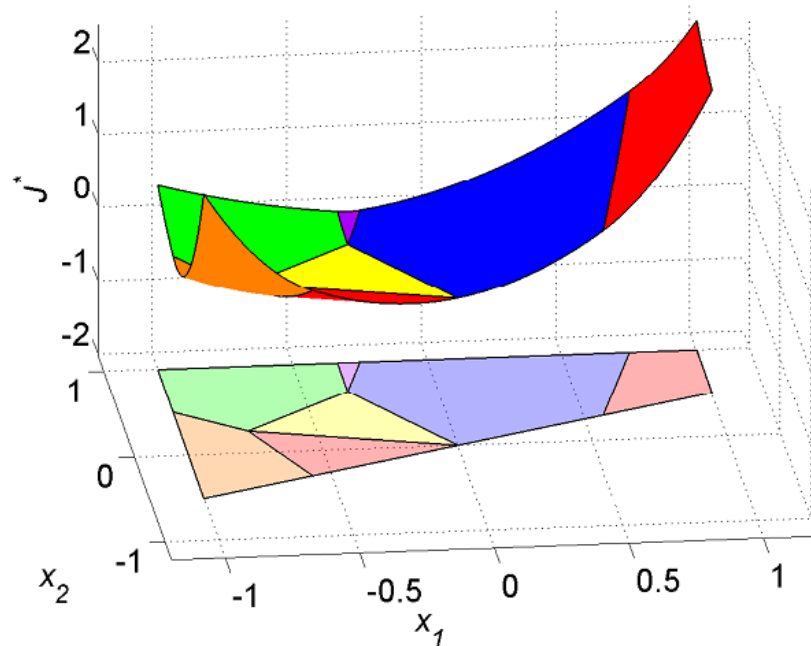


临界域 $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\} \ i = 1, \dots, 7$

6.2 多参数规划

多参数二次规划例子

应用函数mpt_mpqp求解，可得



每个域对应的优化目标函数 $J^*(x)$

6.2 多参数规划

多参数二次规划例子

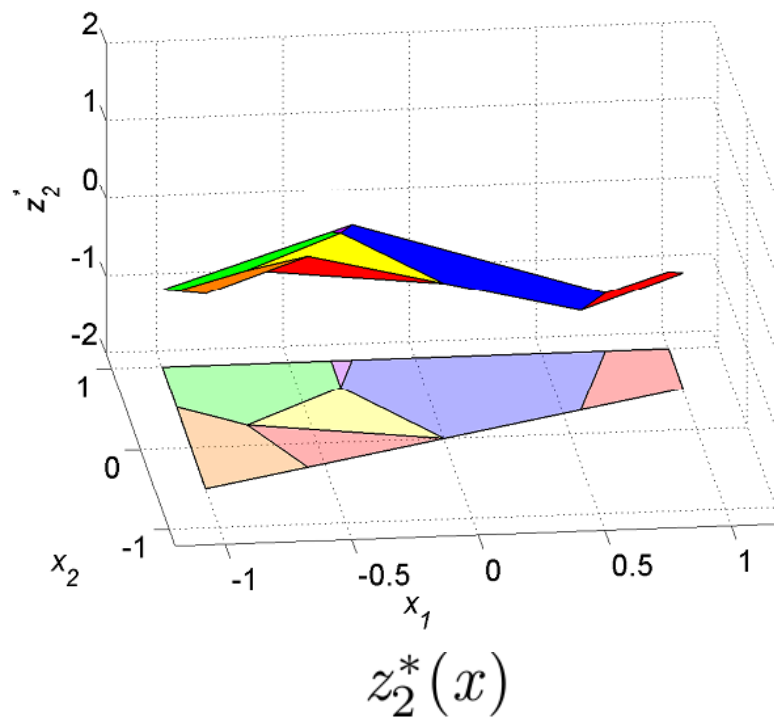
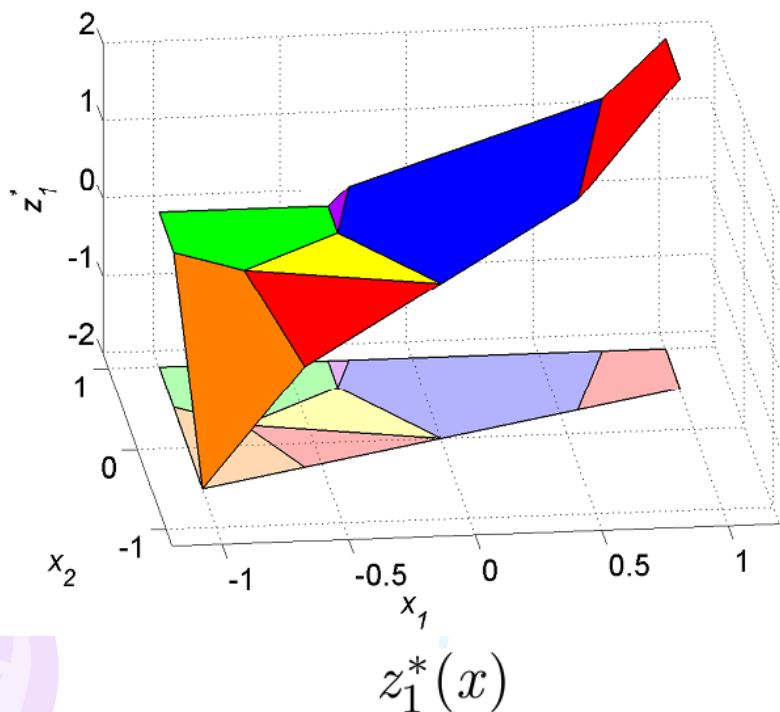
应用函数mpt_mpqp求解，可得每个分区的最优解表达式

$$z^*(x) = \begin{cases} \begin{pmatrix} 0.5 & 1.5 \\ -0.5 & -1.5 \end{pmatrix} x & \text{if } x \in \mathcal{P}_1 \\ \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \text{if } x \in \mathcal{P}_2 \\ \vdots & \\ \vdots & \end{cases}$$

6.2 多参数规划

多参数二次规划例子

应用函数mpt_mpqp求解，可得每个分区的最优解表达式



第6章 显式预测控制

6.1 问题描述

6.2 多参数规划

6.3 约束线性系统的显式预测控制

6.3 约束线性系统的显式预测控制

考虑如下线性时不变系统：

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

满足约束条件 $x(k) \in X$, $u(k) \in U$

根据预测控制算法，每个采样时刻需求解以下优化问题

$$\text{Find} \quad \min_{u^{(*)}} \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N)\|_P^2$$

$$\text{s.t.} \quad x(k) = Ax(k-1) + Bu(k-1)$$

$$y(k) = Cx(k)$$

$$x(k) \in X, \quad u(k) \in U, \quad k = 1, 2, \dots, N-1$$

$$x(k+N) \in \Omega$$

6.3 约束线性系统的显式预测控制

根据线性系统的状态方程：

$$x(k+1) = Ax(k) + Bu(k)$$

可得状态的递推方程

$$x(k+m) = A^m x(k) + \sum_{j=0}^{m-1} A^j Bu(k+m-1-j), m=1, 2, \dots, N$$

则约束预测控制的优化问题可转化为

$$\begin{aligned} \text{Find} \quad & \min_{U_N(k)} \left(\frac{1}{2} (U_N(k))^T H U_N(k) + x^T(k) F U_N(k) \right) + \frac{1}{2} x^T(k) Y x(k) \\ \text{s.t.} \quad & G U_N(k) \leq W + E x(k) \end{aligned}$$

$x(k)$

其中， H, F, Y, G, W, E 可根据系统矩阵 A, B 和 Q, R 计算获得。

6.3 约束线性系统的显式预测控制

对于优化问题

$$\text{Find} \quad \min_{U_N(k)} \left(\frac{1}{2} (U_N(k))^T H U_N(k) + x^T(k) F U_N(k) \right) + \frac{1}{2} x^T(k) Y x(k)$$

$$\text{s.t.} \quad G U_N(k) \leq W + E x(k)$$

$$\text{定义} \quad z = U_N(k) + H^{-1} F^T x(k)$$

则优化问题转化为

$$\text{Find} \quad \min_z \frac{1}{2} z^T H z + \frac{1}{2} x^T(k) (Y - F H^{-1} F^T) x(k)$$

$$\text{s.t.} \quad G z \leq W + S x(k)$$

$$\text{其中} \quad S = E + G H^{-1} F^T$$

6.3 约束线性系统的显式预测控制

最终将约束预测控制的在线优化问题转化为

$$\begin{aligned} \text{Find} \quad & \min_z \frac{1}{2} z^T H z \\ \text{s.t.} \quad & Gz \leq W + Sx \end{aligned}$$

求解方法

1) 每个采样时刻求解优化问题获得最优控制序列

在线 \longrightarrow Quadratic Program (QP)

2) 通过离线求解多参数规划问题，获得

□ 参数分区及对应的优化问题解与参数之间的显式表达式

离线 \longrightarrow multi-parametric Quadratic Program (mp-QP)

□ 在线确定参数所在分区并计算最优解

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器

1) 求解多参数二次规划问题

$$\text{Find} \quad \min_z \frac{1}{2} z^T H z$$

$$\text{s.t.} \quad Gz \leq W + Sx$$

获得控制器分区 $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\}$ 及每个分区对应的控制律 $z_i^* = K_i x + b_i$, $i = 1, 2, \dots, r$, r 为分区数量

2) 根据 $z = U_m(k) + H^{-1} F^T x(k)$ 获得原约束系统的控制器及对应的分区

$$U_m(k) = (K_i - H^{-1} F^T) x(k) + b_i = F_i x(k) + G_i \quad \text{PWA形式的控制器}$$

$$\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\} \quad i = 1, 2, \dots, r \quad \text{控制器对应的分区}$$

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

考虑以下系统 $x(k+1) = Ax(k) + Bu(k)$

$$A = \begin{bmatrix} 0.5 & -1 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

满足约束 $-1 \leq u(k) \leq 1$ $\begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

选取二次型目标函数

$$J_0(x(0), U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

其中 $P \succeq 0$, $Q \succeq 0$, $R \succ 0$

优化问题的一般描述形式

$$\begin{aligned} J_0^*(x(0)) = & \min_{U_0} J_0(x(0), U_0) \\ \text{subj. to} & \quad x_{k+1} = A x_k + B u_k, \quad k = 0, \dots, N-1 \\ & \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad k = 0, \dots, N-1 \\ & \quad x_N \in \mathcal{X}_f \\ & \quad x_0 = x(0) \end{aligned}$$

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

将优化问题

$$\begin{aligned} J_0^*(x(0)) = & \min_{U_0} J_0(x(0), U_0) \\ \text{subj. to} & \quad x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N-1 \\ & \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad k = 0, \dots, N-1 \\ & \quad x_N \in \mathcal{X}_f \\ & \quad x_0 = x(0) \end{aligned}$$

转化为多参数规划问题

$$\begin{aligned} \text{Find} \quad & \min_z \frac{1}{2} z^T H z \\ \text{s.t.} \quad & Gz \leq W + Sx \end{aligned}$$

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

将一般形式描述的约束

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N-1$$

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad k = 0, \dots, N-1, \quad \mathcal{X} = \{x \mid A_x x \leq b_x\} \quad \mathcal{U} = \{u \mid A_u u \leq b_u\}$$

$$x_N \in \mathcal{X}_f, \quad \mathcal{X}_f = \{x \mid A_f x \leq b_f\}$$

转化为 $G_0 U_0 \leq w_0 + E_0 x(0)$

$$G_0 = \begin{bmatrix} A_u & 0 & \dots & 0 \\ 0 & A_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_u \\ 0 & 0 & \dots & 0 \\ A_x B & 0 & \dots & 0 \\ A_x A B & A_x B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \dots & A_f B \end{bmatrix}, \quad E_0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_f A^N \end{bmatrix}, \quad w_0 = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{bmatrix}$$

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

将优化问题

$$\begin{aligned} J_0^*(x(0)) = & \min_{U_0} J_0(x(0), U_0) \\ \text{subj. to} & \quad x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N-1 \\ & \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad k = 0, \dots, N-1 \\ & \quad x_N \in \mathcal{X}_f \\ & \quad x_0 = x(0) \end{aligned}$$

转化为多参数规划问题

$$\begin{aligned} \text{Find} \quad & \min_z \frac{1}{2} z^T H z \\ \text{s.t.} \quad & Gz \leq W + Sx \end{aligned}$$

求解mpQP问题

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

因此，对于系统 $x(k+1) = Ax(k) + Bu(k)$

$$A = \begin{bmatrix} 0.5 & -1 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

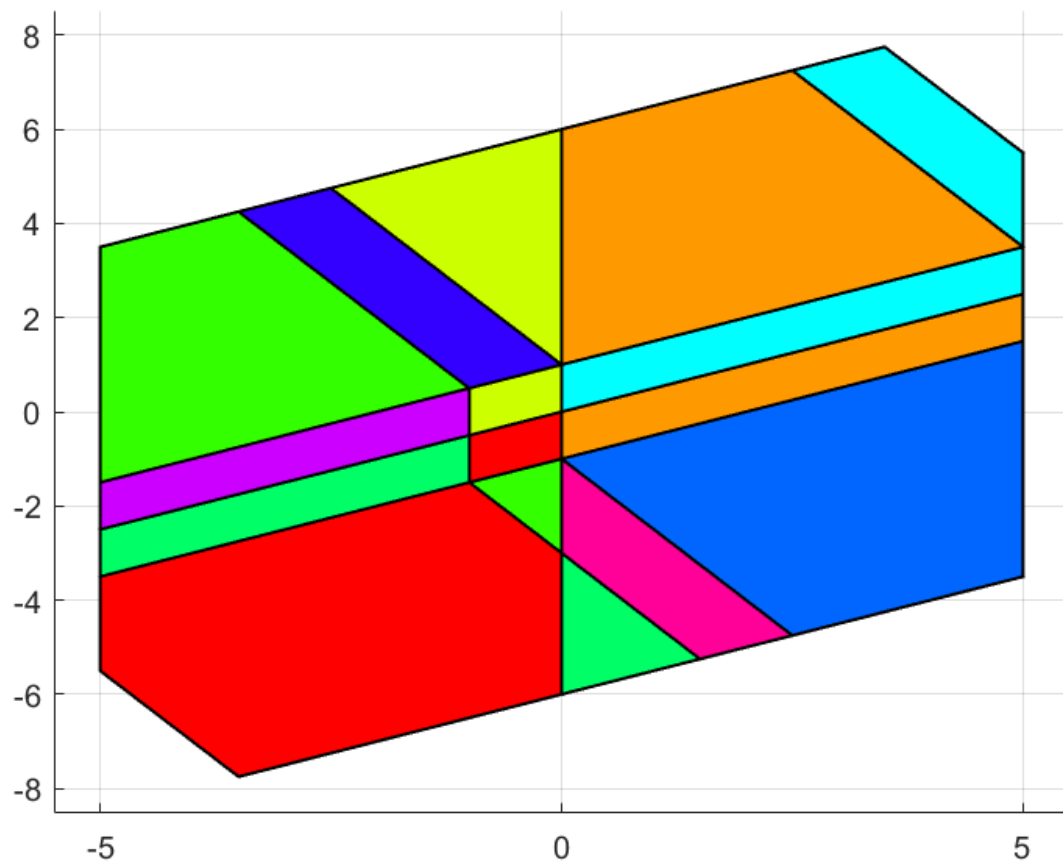
满足约束 $-1 \leq u(k) \leq 1$ $\begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

和目标函数 $J = \min_{u(*)} \sum_{i=0}^N \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right)$

取 $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1$

6.3 约束线性系统的显式预测控制

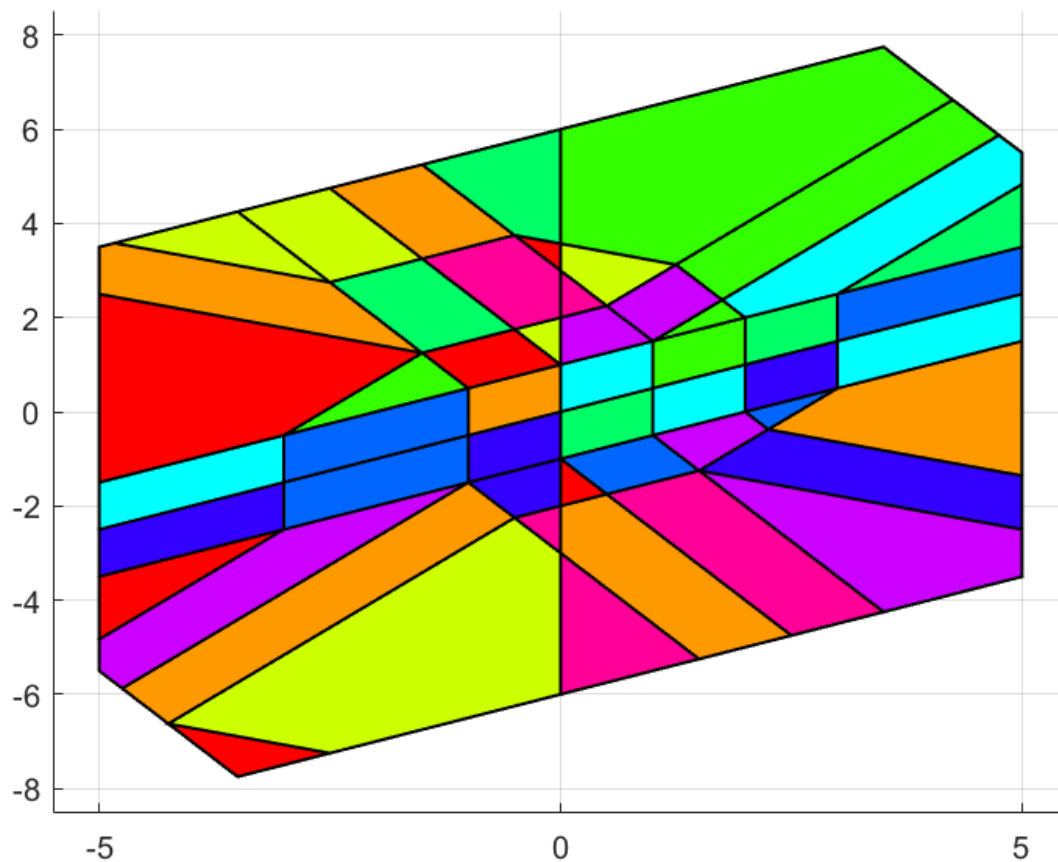
显式预测控制：离线求解控制器例子



$N=2$, 16个控制器分区

6.3 约束线性系统的显式预测控制

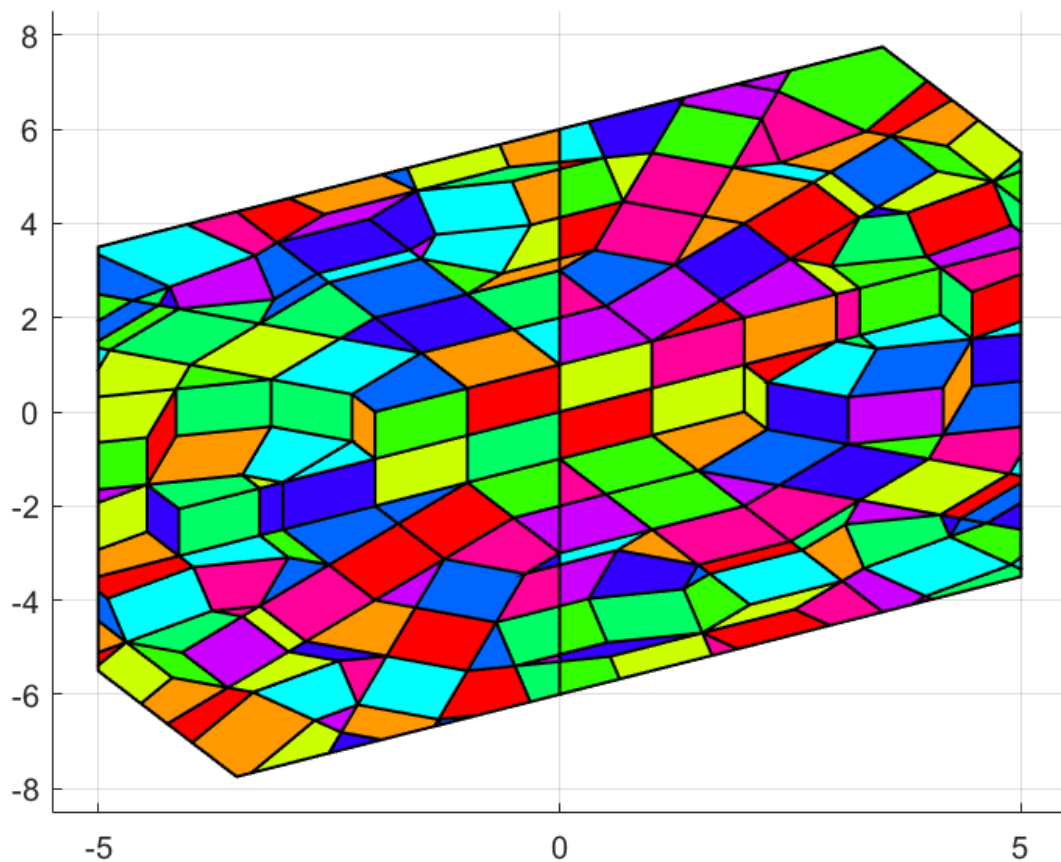
显式预测控制：离线求解控制器例子



$N=4$, 52个控制器分区

6.3 约束线性系统的显式预测控制

显式预测控制：离线求解控制器例子

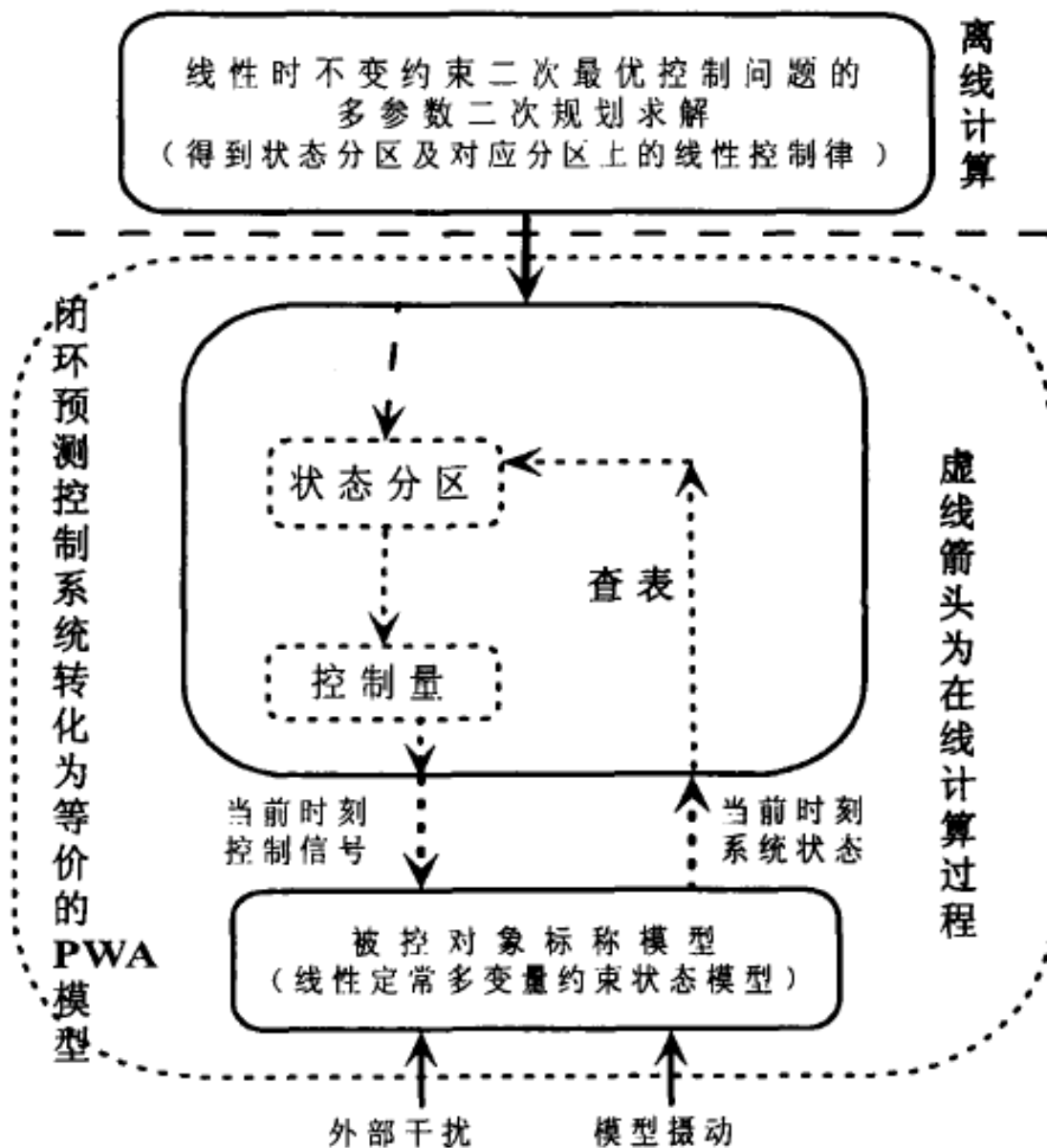


$N=10$, 226个控制器分区

6.3 约束线性系统的显式预测控制

显式预测控制

算法步骤:



6.3 约束线性系统的显式预测控制

线性约束系统显式模型预测控制算法的复杂度主要体现在以下三个方面

- 1) 预处理：在给定的约束集中消除冗余约束
 - 2) 控制器求解：合理构造控制问题，减少分区数量
 - 3) 点定位：减少查找控制器分区的时间
- 1) 和 2) 为离线处理，3) 为在线处理。

下面讨论降低显式模型预测控制算法复杂度的方法

6.3 约束线性系统的显式预测控制

线性约束系统显式模型预测控制算法的复杂度主要体现在以下三个方面

- 1) 预处理：在给定的约束集中消除冗余约束
 - 2) 控制器求解：合理构造控制问题，减少分区数量
 - 3) 点定位：减少查找控制器分区的时间
- 1) 和 2) 为离线处理，3) 为在线处理。

6.3 约束线性系统的显式预测控制

多参数规划求解的一般步骤（离线）

- 1) 得到给定状态 x 的有效约束 $A_N(x)$ ，并得到最优值关于 x 的显式表达式 $U_N^*(x)$
- 2) 计算状态的多面体集 P_r ，且 $U_N^*(x)$ 是最优并满足约束的
- 3) 不断迭代，直至搜索完整个可行状态空间

多参数规划问题的求解时间：

总计算时间=控制器分区数量*每个区计算时间+额外开销

6.3 约束线性系统的显式预测控制

影响控制器分区数量的因素

- 1) 有效约束的数量 c
- 2) 系统控制输入的数量 m
- 3) 预测时域长度 N

在给定约束集中消除冗余约束可有效减少控制器分区数量

→ 目的是为了获得系统的有效约束集 $A_N(x)$

6.3 约束线性系统的显式预测控制

对于约束集

$$P = \{x : Hx \leq K\}$$

其中 $H = [h_1 \quad h_2 \quad \cdots \quad h_m]^T$, $K = [k_1 \quad k_2 \quad \cdots \quad k_m]^T$

且 $x \in \mathbb{R}^n$, $h_i \in \mathbb{R}^n$, $k_i \in \mathbb{R}$, $i = 1, 2, \dots, m$

下面给出两种消除约束集中的冗余约束的方法

6.3 约束线性系统的显式预测控制

方法1：循环判断约束集 P 中第 j 个约束是否为冗余约束

在约束集 P 中去掉第 j 个约束，构成新的约束集

$$\{x: \tilde{H}x \leq \tilde{K}\}$$

$$\tilde{H} = \begin{bmatrix} h_1 & \cdots & h_{j-1} & h_{j+1} & \cdots & h_m \end{bmatrix}^T,$$

$$\tilde{K} = \begin{bmatrix} k_1 & \cdots & k_{j-1} & k_{j+1} & \cdots & k_m \end{bmatrix}^T$$

求解

$$\max_x h_j x$$

$$\text{s.t. } \tilde{H}x \leq \tilde{K}$$

最坏情况下需要求解 m 次
带 $m-1$ 个约束和 n 个变量的
线性规划问题

如果该问题的最优目标值 $\leq k_j$ ，那么第 j 个约束是冗余约束。

6.3 约束线性系统的显式预测控制

为了提高方法1的计算效率，需减少待求解线性规划的数量

方法2：首先在约束集

$$P = \{x : Hx \leq K\}$$

用简单的方法确定部分冗余约束集和非冗余约束集，再针对
剩余约束集采用方法1去除冗余约束

6.3 约束线性系统的显式预测控制

方法2

1) 针对约束集 P ，获得参数的边界条件 $L \leq x \leq U$

对所有 $i \in \{1 \ 2 \ \dots \ n\}$ ，求解以下线性规划问题

$$\begin{aligned} \min_x \quad & \pm x_i \\ \text{s.t.} \quad & Hx \leq K \end{aligned}$$

需要求解 $2n$ 个线性规划问题

其中 x_i 表示向量 $x \in \mathbb{R}^n$ 的第 i 个元素。

如果冗余约束比较多，且 n 相对于约束数量 m 又充分小，则求解 $2n$ 次线性规划问题是合理的

6.3 约束线性系统的显式预测控制

方法2

2) 根据边界条件计算冗余约束 P_R

根据 $L \leq x \leq U$ 计算 $h_i x, i \in \{1 \ 2 \ \dots \ m\}$ 的一个上界

$$h_i x \leq \sum_{j \in \{j: h_{ij} > 0\}} h_{ij} U_j + \sum_{j \in \{j: h_{ij} < 0\}} h_{ij} L_j$$

如果上式右边部分小于 k_i ，则该约束是冗余的。

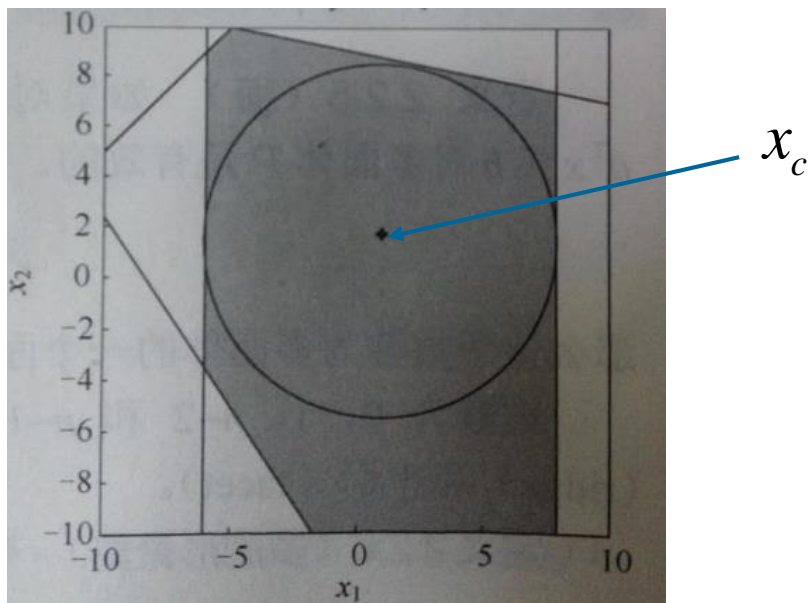
上述检查出的冗余约束集记为

$$P_R = \left\{ i \in \{1 \ 2 \ \dots \ m\} : \left(\sum_{j \in \{j: h_{ij} > 0\}} h_{ij} U_j + \sum_{j \in \{j: h_{ij} < 0\}} h_{ij} L_j \right) \leq k_i \right\}$$

6.3 约束线性系统的显式预测控制

方法2

3) 计算Chebychev球，找到内点和非冗余约束的一个子集 P_c



切比雪夫 (Chebychev) 球 $B(x_c, R)$

6.3 约束线性系统的显式预测控制

方法2

3) 计算Chebychev球，获得非冗余约束的一个子集 P_C

对于给定的约束集

$$P = \{x : Hx \leq K\}$$

可通过求解以下线性规划，找到Chebychev球的中心和半径

$$\begin{aligned} & \max_{x_c, R} R \\ & \text{s.t.} \quad Hx_c + R\|H\|_2 \leq K \end{aligned}$$

由此获得非冗余约束集 P_C 和球的中心 x_c

6.3 约束线性系统的显式预测控制

方法2

4) 利用在约束 P / P_R 上的射线找到非冗余约束的一个子集 P_N

① 初始化非冗余约束集 P_N 为空集，内点 $x_{\text{int}} = x_c$

② 随机产生 d ，得直线 $x_{\text{int}} + t_i d, d \in \mathbb{R}^n, t_i \in \mathbb{R}$

③ 计算直线与 $h_i x = k_i$ 之间的交集，得 $t_i = \frac{k_i - h_i x_{\text{int}}}{h_i d}$

④ 对 P / P_R 中所有约束的索引计算 t_i 的最小正值和最大负值，对应约束分别记为 i_p, i_n ，如果 $P_N = P_N \cup i_p \cup i_n$ 则约束是非冗余的

⑤ 令 $x_{\text{int}} = x_{\text{int}} + \frac{t_{i_p} + t_{i_n}}{2} d$ ，从步骤2开始重复

6.3 约束线性系统的显式预测控制

方法2

5) 利用方法1检查余下的约束集

$$P / (P_R \cup P_C \cup P_N)$$

6.3 约束线性系统的显式预测控制

总结方法2

- 1) 针对约束集 P 获得参数的边界条件 $L \leq x \leq U$
- 2) 根据边界条件计算冗余约束 P_R
- 3) 计算Chebychev球，获得非冗余约束的一个子集 P_C
- 4) 利用在约束 P / P_R 上的射线找到非冗余约束的一个子集 P_N
- 5) 利用方法1检查余下的约束集 $P / (P_R \cup P_C \cup P_N)$

与方法1的区别：通过前面四步相对简单的计算减少待求解线性规划问题的数量，提高计算效率

6.3 约束线性系统的显式预测控制

线性约束系统显式模型预测控制算法的复杂度主要体现在以下三个方面

- 1) 预处理：在给定的约束集中消除冗余约束
 - 2) 控制器求解：合理构造控制问题，减少分区数量
 - 3) 点定位：减少查找控制器分区的时间
- 1) 和 2) 为离线处理，3) 为在线处理。

6.3 约束线性系统的显式预测控制

控制器的求解（离线）

- 1) 对于给定状态 x 的有效约束 $A_N(x)$ ，并得到最优值关于 x 的显式表达式 $U_N^*(x)$
- 2) 计算状态的多面体集 P_r ，且 $U_N^*(x)$ 是最优并满足约束的

解决状态空间的划分和控制律的求解问题

可通过以下两个方面提高计算效率

- ◆ 改进控制器求解方法 减少控制器分区数量及每个区计算的时间
- ◆ 控制器求解完成之后的后处理（合并控制域等）
进步一步减少控制器分区数量

6.3 约束线性系统的显式预测控制

控制器的求解方法

- 1) 直接求解标准多参数规划问题获得控制器
- 2) 低复杂度控制器的计算方法，如最小时间控制器、N步控制器等，同时保证闭环系统是收敛的。

低复杂度：PWA控制器分区少

具体计算方法详见张聚的《显式模型预测控制理论与应用》

6.3 约束线性系统的显式预测控制

控制器分区后处理方法

目的：通过减少控制器分区的数量，减少数据的存储空间及算法的在线执行时间

□ 插值多参数二次规划（Interpolation mpQP）算法

去掉某些控制器分区，并对消失的控制器分区应用插值方法获得控制律（离线处理与在线计算相结合）

具体计算方法详见张聚的《显式模型预测控制理论与应用》

6.3 约束线性系统的显式预测控制

线性约束系统显式模型预测控制算法的复杂度主要体现在以下三个方面

- 1) 预处理：在给定的约束集中消除冗余约束
- 2) 控制器求解：合理构造控制问题，减少分区数量
- 3) 点定位：减少查找控制器分区的时间

1) 和 2) 为离线处理，3) 为在线处理。

6.3 约束线性系统的显式预测控制

点定位（**在线**）

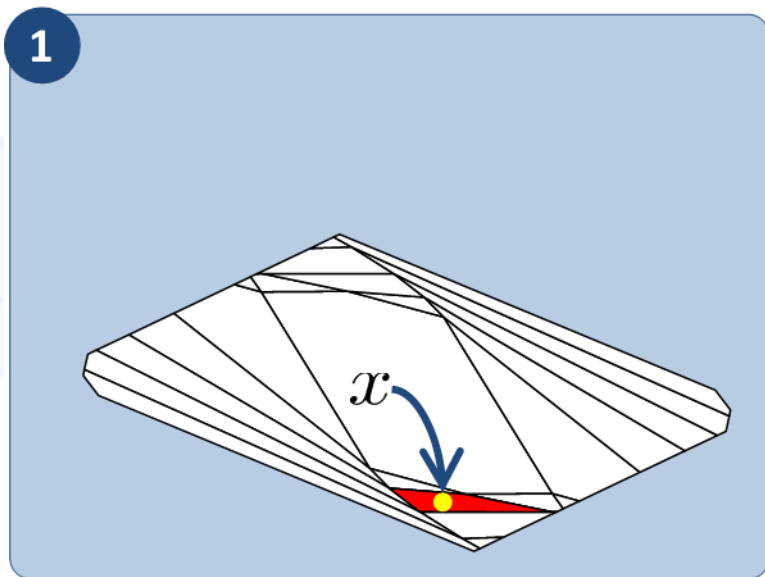
显式预测控制在线计算的核心任务是：判断系统当前的状态处于哪个分区，被称为点定位问题。

点定位算法的性能要求

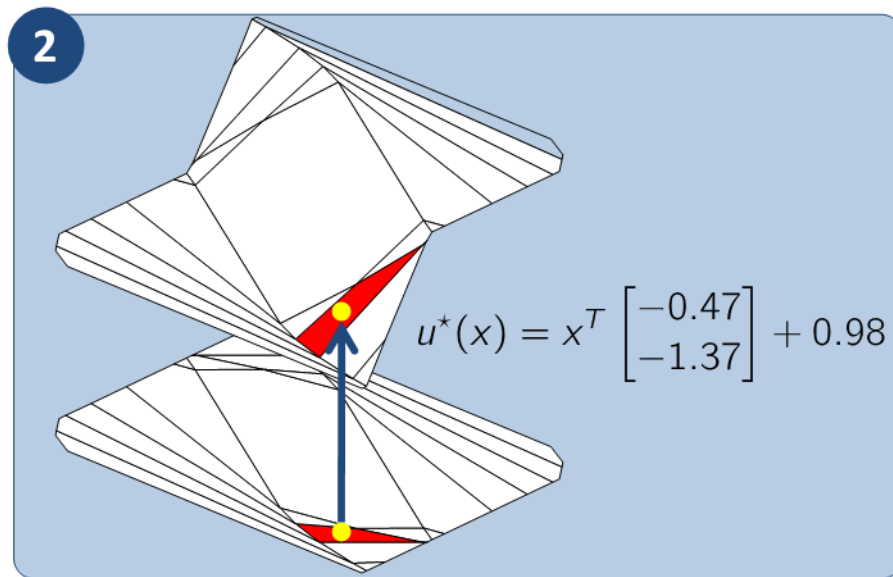
- ◆分区和特征值等数据对存储空间的需求
- ◆在线解决点定位问题的效率
- ◆根据算法要求预处理数据所需时间

6.3 约束线性系统的显式预测控制

点定位 \rightarrow PWA控制律



定位当前状态所在的分区



获取该分区对应的PWA控制律

6.3 约束线性系统的显式预测控制

状态分区的描述方法

以二维分区为例

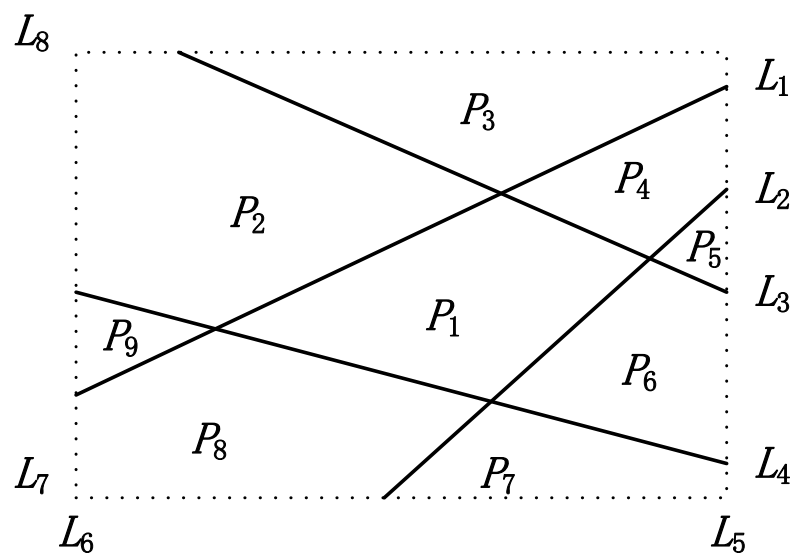
分区 P_1 的边界由8条直线确定

$$L_1 : h_{11}x_1 + h_{12}x_2 = k_1 \quad L_5 : x_1 = k_5$$

$$L_2 : h_{21}x_1 + h_{22}x_2 = k_2 \quad L_6 : x_1 = k_6$$

$$L_3 : h_{31}x_1 + h_{32}x_2 = k_3 \quad L_7 : x_2 = k_7$$

$$L_4 : h_{41}x_1 + h_{42}x_2 = k_4 \quad L_8 : x_2 = k_8$$



分区 P_i 的描述方法 $P_i = \{x : H_i x \leq K_i\}$

所谓的点定位，就是对于给定的 x ，找到其所在的分区。

6.3 约束线性系统的显式预测控制

常用点定位算法

◆ 直接查找法

◆ 可达分区法

◆ 哈希表法

◆ 二叉树法

6.3 约束线性系统的显式预测控制

直接查找法

原理：从编号为1的第一个分区开始，依次利用下式

$$P_i = \{x : H_i x \leq K_i\}$$

判断所要定位的是否在当前分区内，直到找到正确的分区或判断完所有分区为止。

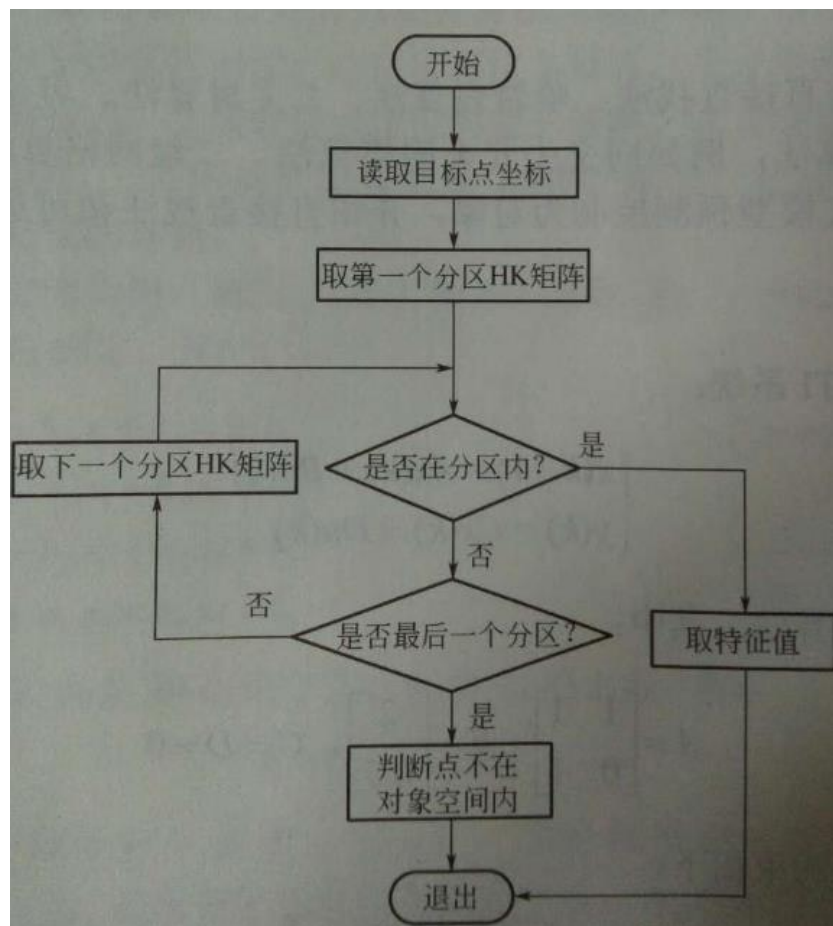
特点：

- 不需要对多参数二次规划得到的数据执行预处理
- 最坏情况需要遍历所有分区

6.3 约束线性系统的显式预测控制

直接查找法

在线计算流程



6.3 约束线性系统的显式预测控制

可达分区法

考虑系统

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

经mpQP求解后，获得系统在任一分区 P_i 上对应的控制律

$$u(k) = F_i x(k) + G_i$$

将控制律代入状态方程可获得下一时刻系统的状态

$$x(k+1) = (A + BF_i)x(k) + BG_i$$

6.3 约束线性系统的显式预测控制

可达分区法

根据 $x(k+1) = (A + BF_i)x(k) + BG_i$

可获得分区 P_i 上所有状态经过一个采样周期的状态转移后可达的状态空间区域，即为**可达域**，与该可达域的交集不为空的所有分区称为 P_i 的**可达分区**。

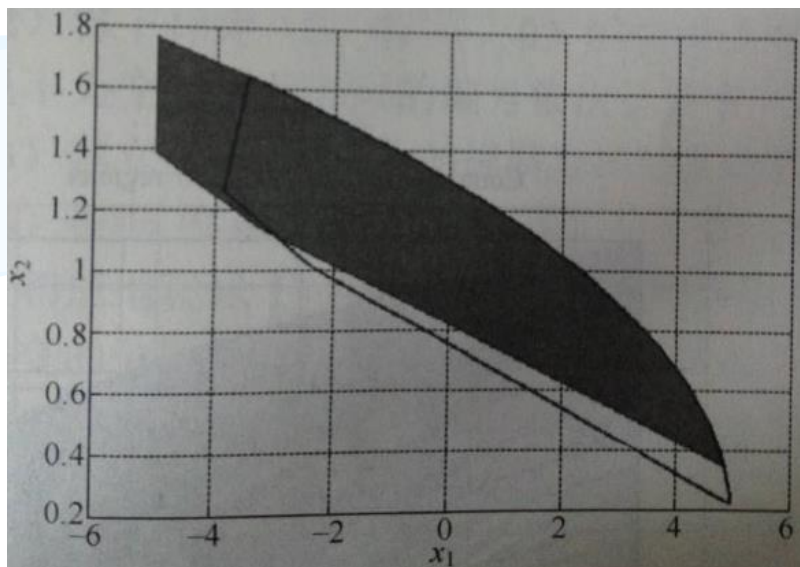
特点：

- 通过预处理建立每个分区对应的可达分区
- 相对于直接查找法，增加了预处理和数据存储所需空间
- 最差情况下（系统每个分区的可达分区都包含系统的所有分区）可达分区算法相当于直接查找法

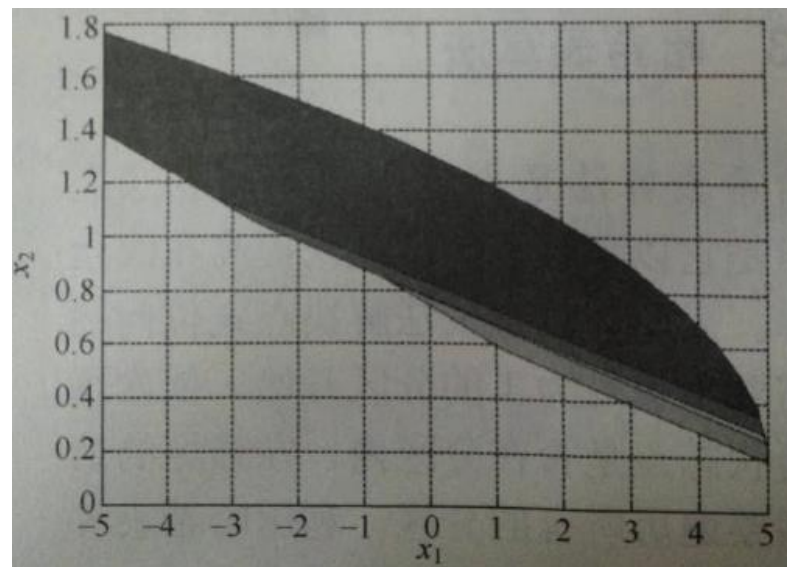
6.3 约束线性系统的显式预测控制

可达分区法

可达域与可达分区



黑框包围的区域称为可达域

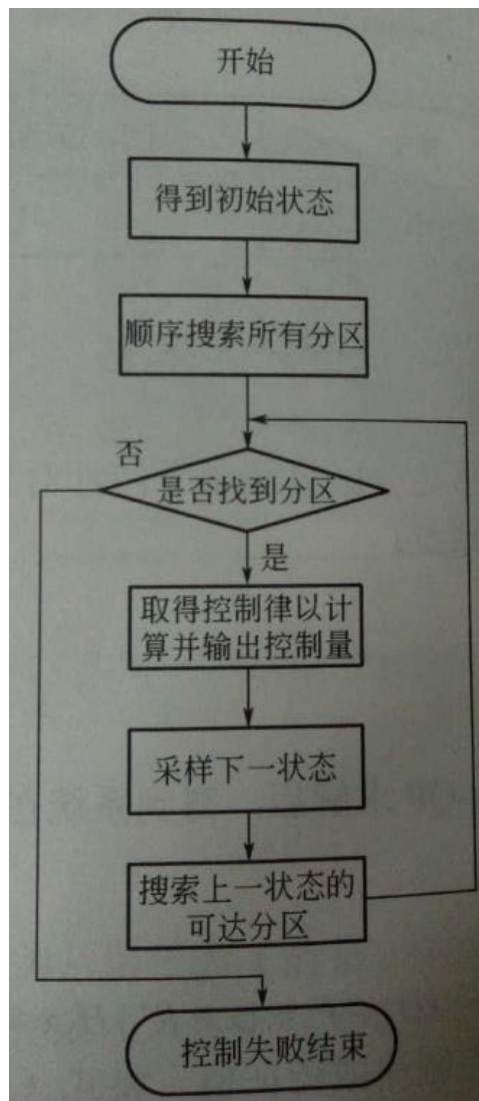


可达分区 (4个)

6.3 约束线性系统的显式预测控制

可达分区法

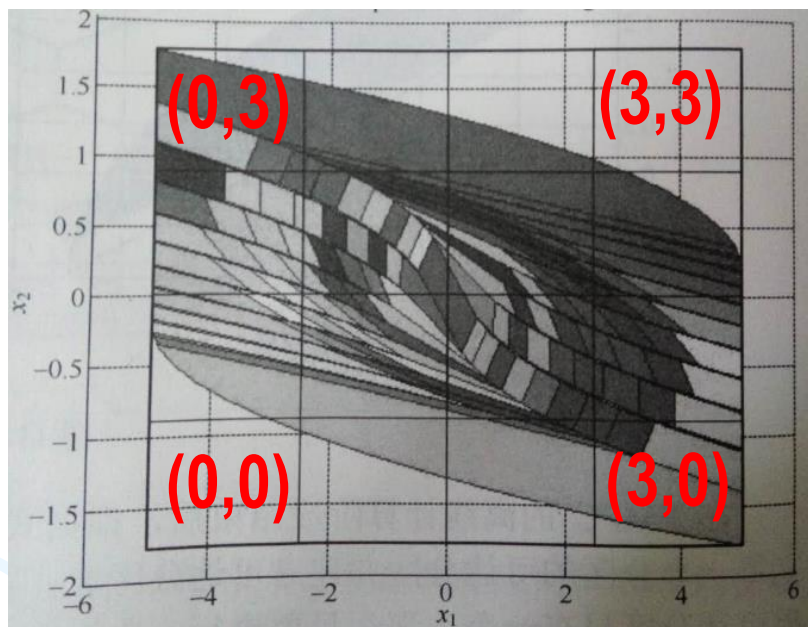
在线计算流程



6.3 约束线性系统的显式预测控制

哈希表法

对于给定的状态分区，在每个维度上平均划分为 N 个部分



将每个网格分别以 $(0, 0), (0, 1), \dots, (3, 3)$ 编号，称为区域下标。

6.3 约束线性系统的显式预测控制

哈希表法

查找与每个网格有交集的分区并记录。

在线计算时，首先判断当前系统状态所在的网格，即可缩小需要查找的分区数量。

利用哈希函数判断当前系统状态所在的网格

$$n_x = \text{floor} \left((x - a) \times \frac{N}{b - a} \right)$$

$$n_y = \text{floor} \left((y - c) \times \frac{N}{d - c} \right)$$

6.3 约束线性系统的显式预测控制

哈希表法

哈希函数的参数选取与计算

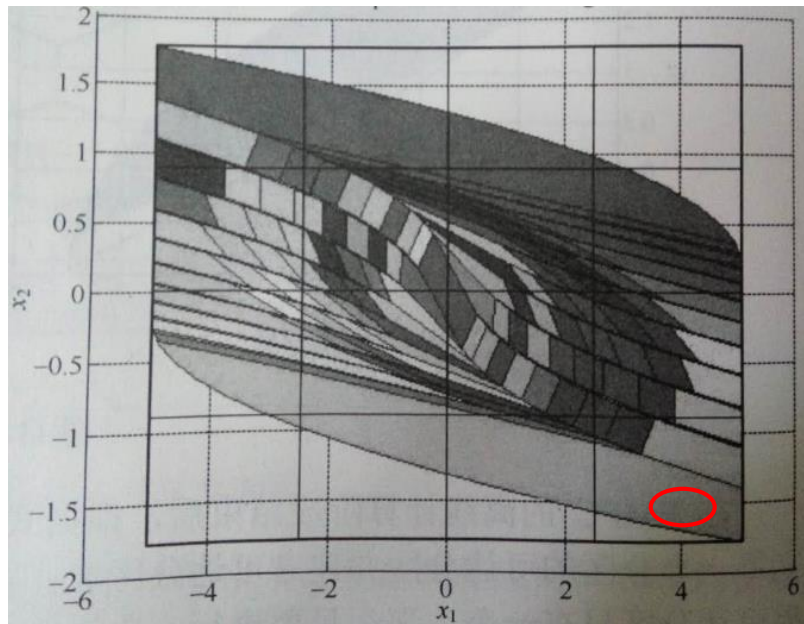
$$n_x = \text{floor}\left((x - a) \times \frac{N}{b - a}\right)$$

$$n_y = \text{floor}\left((y - c) \times \frac{N}{d - c}\right)$$

$$N = 4$$

$$a = -6, \quad b = 6$$

$$c = -2, \quad d = 2$$



$(4, 1.5) \rightarrow (3, 0)$

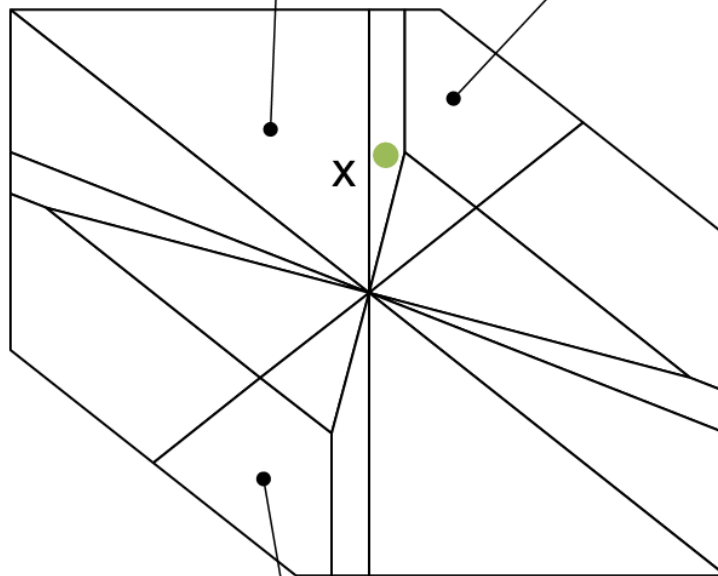
6.3 约束线性系统的显式预测控制

二叉树法

控制域分区

$$CR(B_1) = \{x \mid A_1x + b_1 \leq 0\}$$

$$CR(B_2) = \{x \mid A_2x + b_2 \leq 0\}$$



$$CR(B_3) = \{x \mid A_3x + b_3 \leq 0\}$$

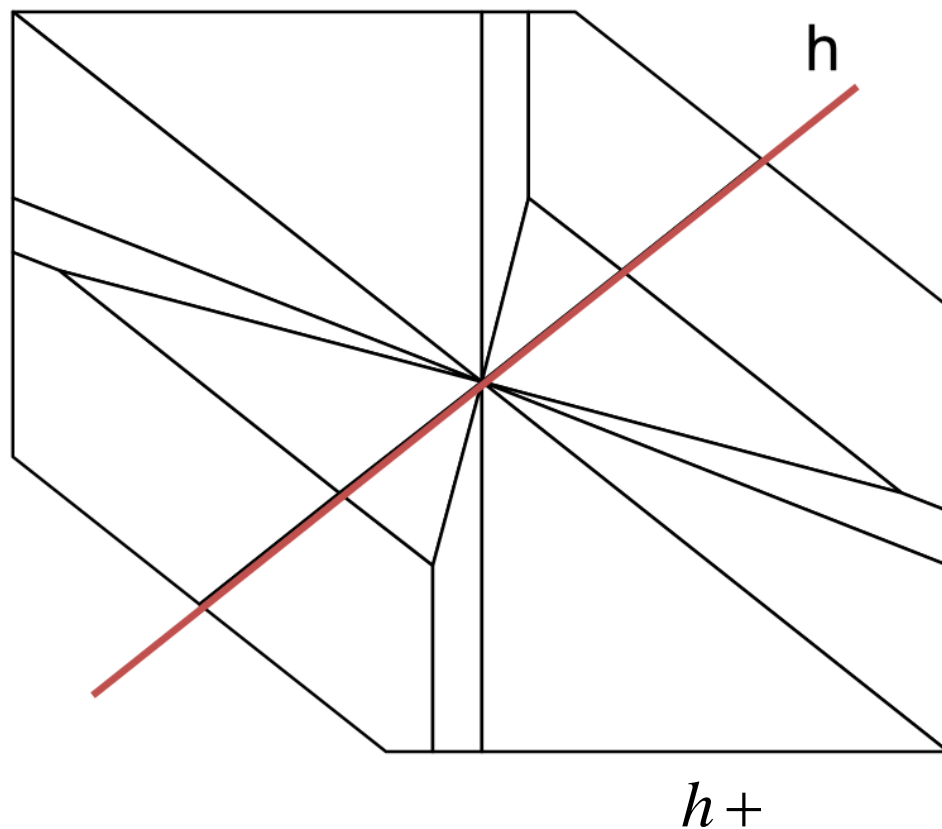
6.3 约束线性系统的显式预测控制

二叉树法

基本原理：引入一组超平面 h 将待查找的分区划分为两部分，并尽可能使左右两侧包含相同数量分区，不断重复直至找到分区或完成搜索。

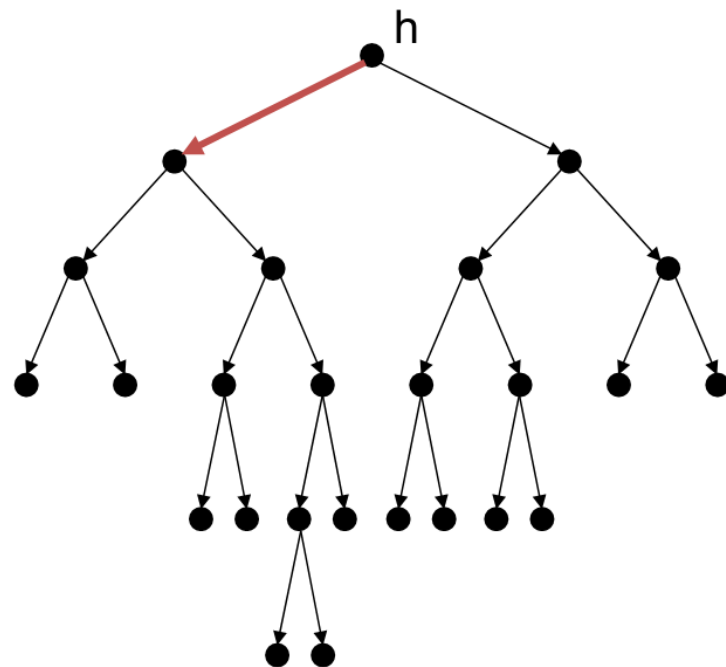
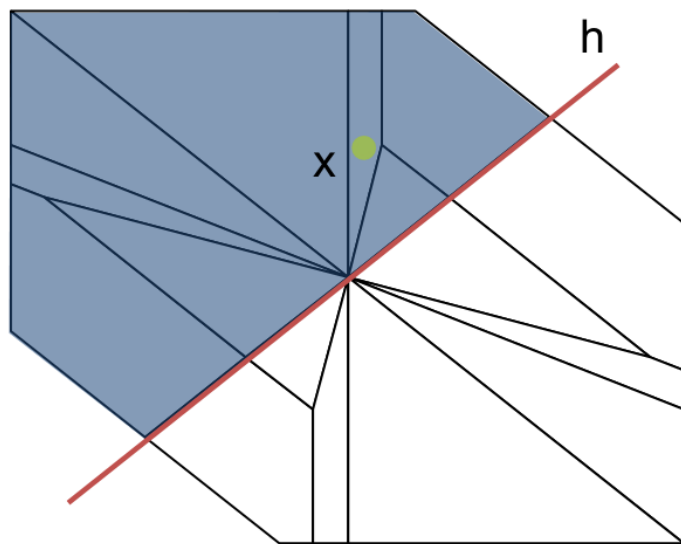
$$h = \{x \mid Hx = K\}$$

$h -$



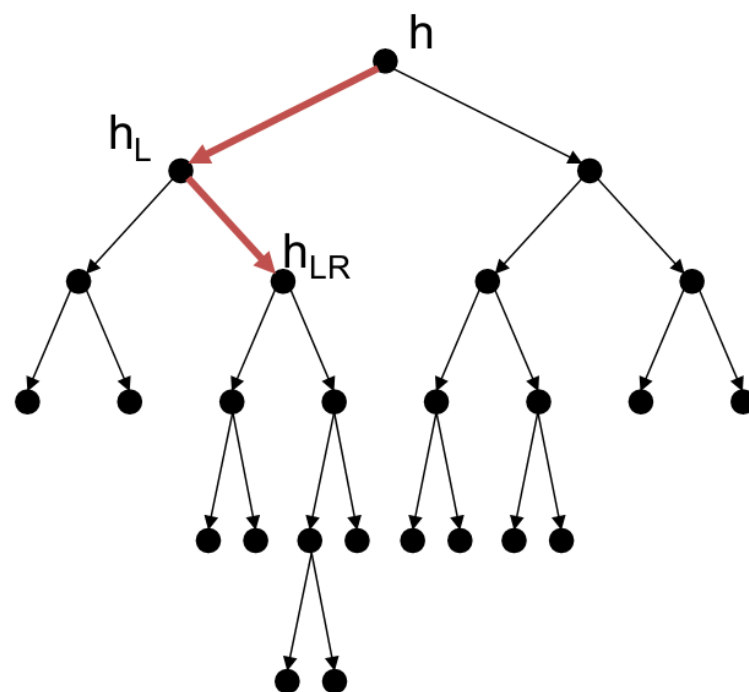
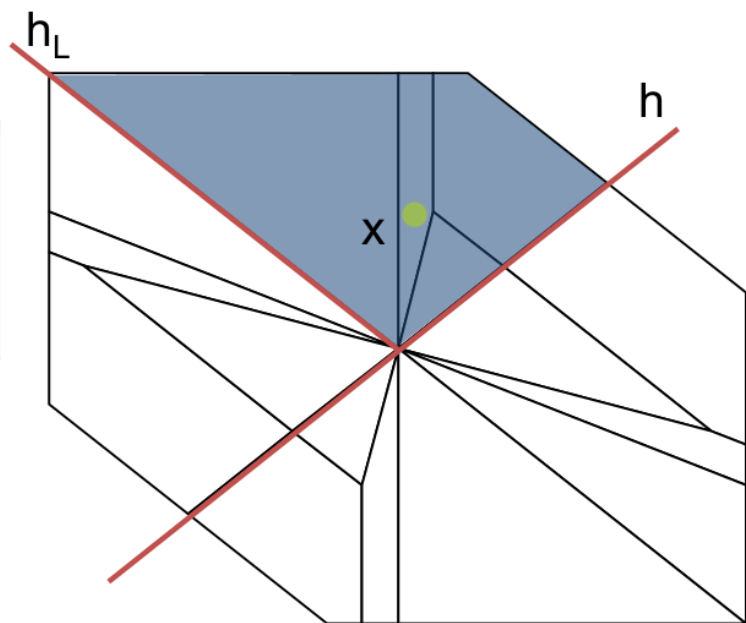
6.3 约束线性系统的显式预测控制

二叉树法



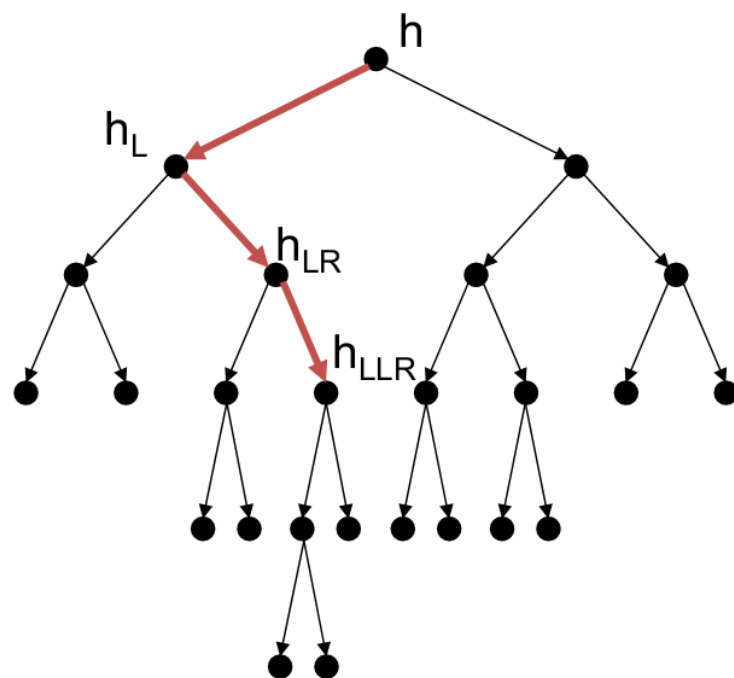
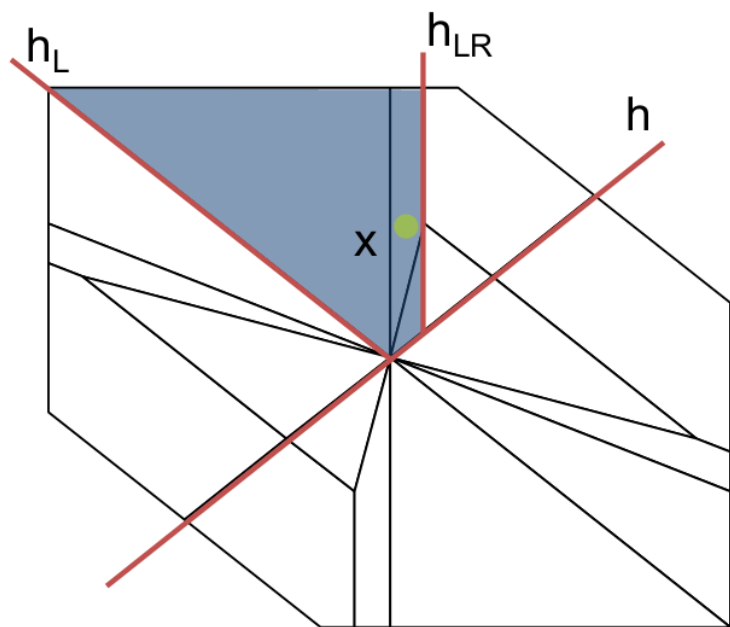
6.3 约束线性系统的显式预测控制

二叉树法



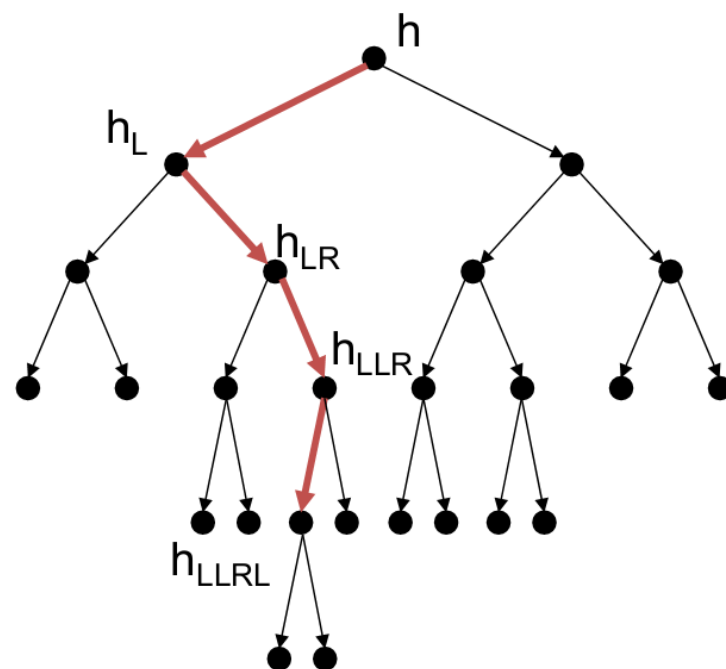
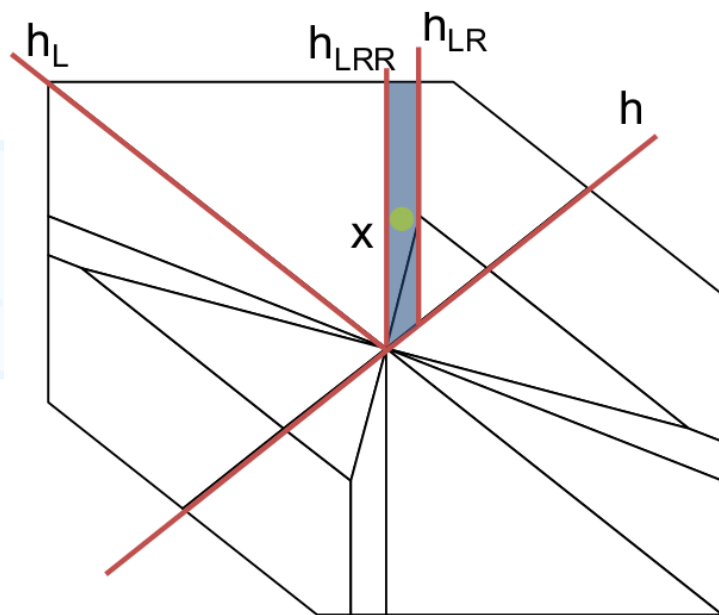
6.3 约束线性系统的显式预测控制

二叉树法



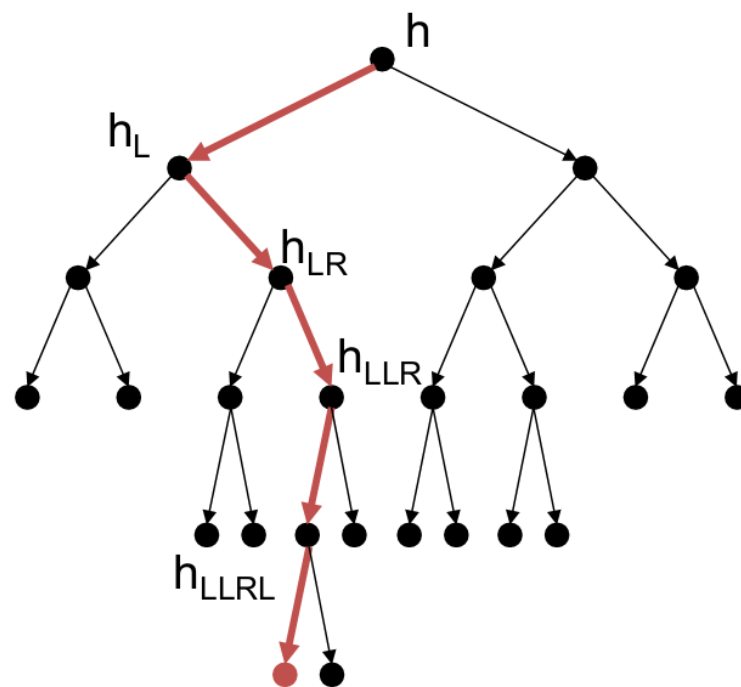
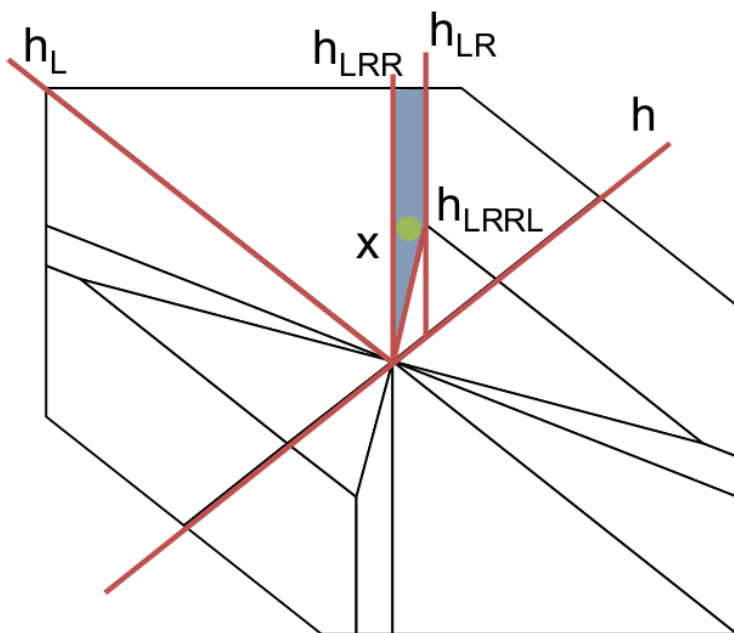
6.3 约束线性系统的显式预测控制

二叉树法



6.3 约束线性系统的显式预测控制

二叉树法



6.3 约束线性系统的显式预测控制

不同点定位算法的性能对比

例子

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

其中

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix}, C = 0$$

满足约束 $|u| \leq 0.25, |x| \leq 5$

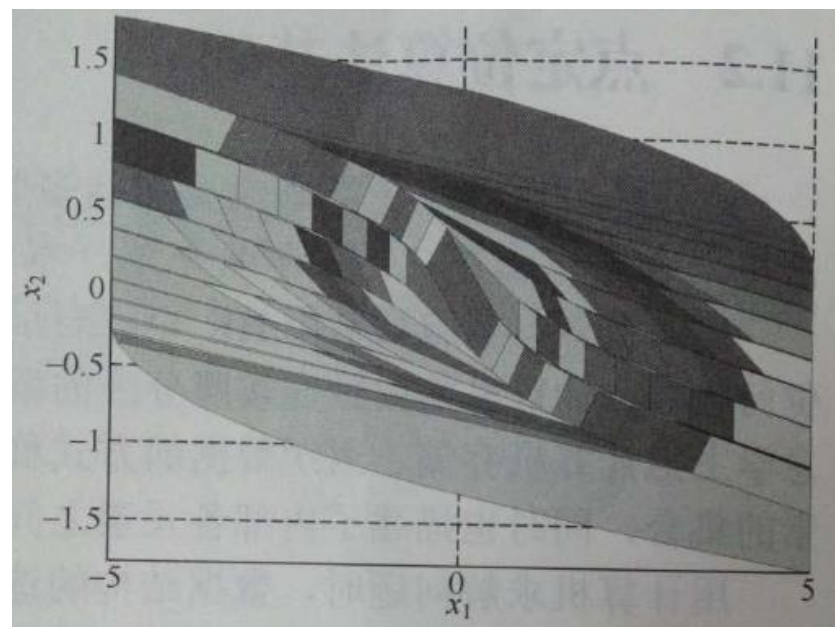
6.3 约束线性系统的显式预测控制

不同点定位算法的性能对比

系统经Matlab工具箱mpQP

求解后，将系统状态空间划

分为201个区



6.3 约束线性系统的显式预测控制

不同点定位算法的性能对比

对于同一系统的控制器分区：

算法	存储空间需求 (字节)	算法预处理时 间 (s)	平均在线计算 时间
直接查找法	27320	0	0.4624
可达分区法	34416	39	0.0404
哈希表法	31704	5	0.1525
二叉树法	3960	212	0.0045

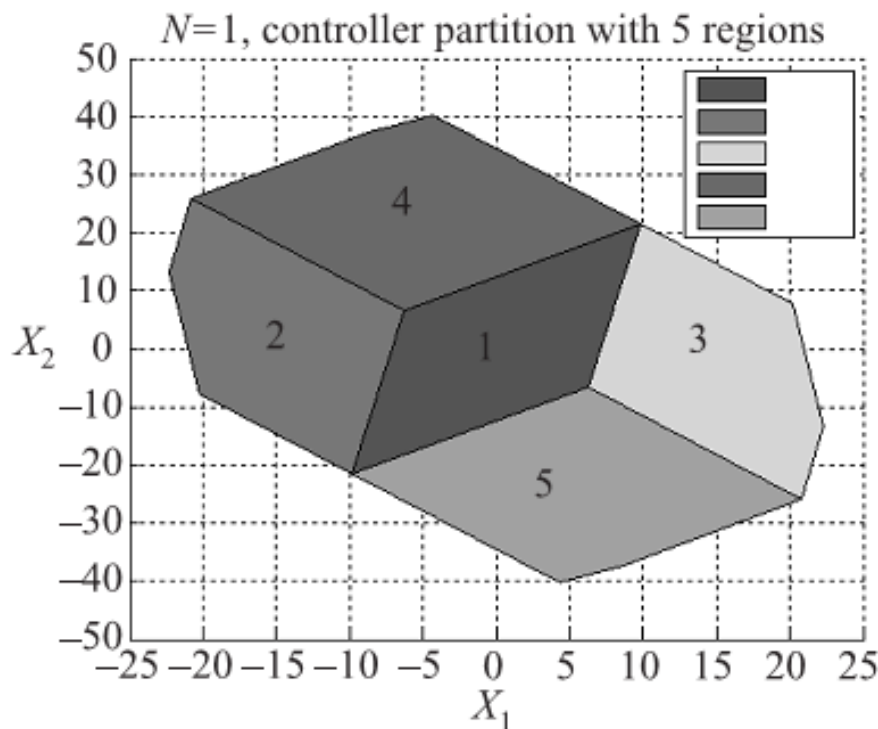
其中哈希表法的划分参数 N 为6。

6.3 约束线性系统的显式预测控制

显式预测控制的特点

利用多参数规划理论求解线性时不变对象的约束二次优化控制问题，可以获得：

1) 系统的状态分区



6.3 约束线性系统的显式预测控制

显式预测控制的特点

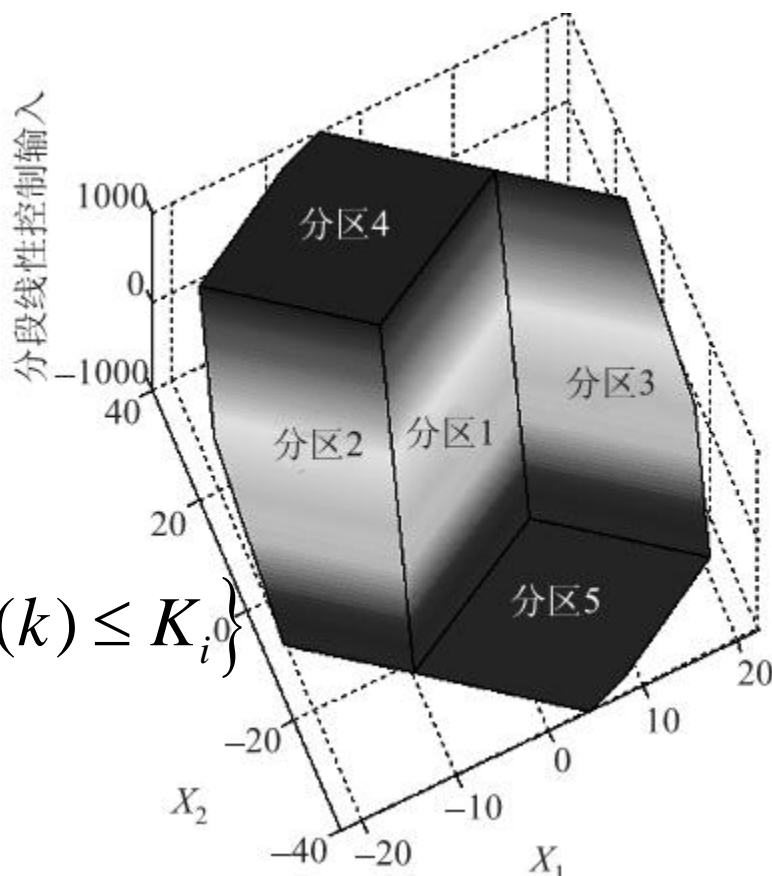
利用多参数规划方法求解线性时不变对象的约束二次优化控制问题，可以获得：

2) 每个状态分区对应一个
线性状态反馈控制律

$$u(k) = F_i x(k) + G_i$$

$$x(k) \in \Omega_i, \quad \Omega_i = \{x \in \mathbb{R}^n \mid H_i x(k) \leq K_i\}$$

F_i, G_i, H_i, K_i 为常数阵



6.3 约束线性系统的显式预测控制

显式预测控制的特点

线性时不变对象的闭环模型预测控制系统可以转化为显式自治的分段仿射（PWA）系统：

$$x(k+1) = A'x(k) + B'$$

$$z(k) = C'x(k)$$

$$x(k) \in \Omega_i, \quad \Omega_i = \{x \in \mathbb{R}^n \mid H_i x(k) \leq K_i\}$$

H_i, K_i, A', B', C' 为常数阵

6.3 约束线性系统的显式预测控制

显式预测控制的特点

显式预测控制算法特点：

- 1) 降低了在线计算时间，有利于应用于采样速度高的系统并且控制律的实现简单，有望在片上系统实现。
- 2) 获得控制律的解析表达式有助于进行控制器性能分析,如稳定性分析、可行性分析、鲁棒性分析等。

因此，扩大预测控制的应用范围到新的领域,如汽车、航空电子、机电一体化等，以及其它需要在线快速计算的领域。

6.3 约束线性系统的显式预测控制

显式预测控制的Matlab计算与仿真

多参数规划工具箱MPT(Multi-Parametric Toolbox)

1) 下载并安装MPT工具箱

✓ 下载网址 <https://www.mpt3.org/>

✓ 下载网页有详细的安装步骤

2) 使用MPT工具箱求解

✓ 下载网页有详细的使用说明

本章小结

本章主要内容：

- 1) 将约束预测控制在线优化问题转化为多参数规划问题
- 2) 求解多参数规划问题获得控制器分区及对应的控制律
- 3) 显式预测控制在线计算的点定位方法

后面章节重点讲解：非线性预测控制方法