

## 基于粒子群优化的有约束模型预测控制器

董 娜, 陈增强, 孙青林, 袁著祉

(南开大学 自动化系, 天津 300071)

**摘要:** 研究了模型预测控制(MPC)中解决带约束的优化问题时所用到的优化算法, 针对传统的二次规划(QP)方法的不足, 引入了一种带有混沌初始化的粒子群优化算法(CPSO), 将其应用到模型预测控制中, 用于解决同时带有输入约束和状态约束的控制问题. 最后, 引入了一个实际的带有约束的线性离散系统的优化控制问题, 分别用二次规划和粒子群优化两种算法去解决, 通过仿真结果比较, 说明了基于粒子群优化(PSO)的模型预测控制算法的优越性.

**关键词:** 模型预测控制; 粒子群优化算法; 带约束的优化; 线性离散系统  
**中图分类号:** TP273      **文献标识码:** A

## Particle-swarm optimization algorithm for model predictive control with constraints

DONG Na, CHEN Zeng-qiang, SUN Qing-lin, YUAN Zhu-zhi

(Department of Automation, Nankai University, Tianjin 300071, China)

**Abstract:** We investigate the optimization algorithms for solving the constrained optimization problems in model predictive control(MPC). To deal with the disadvantage of the quadratic programming(QP) algorithm, we introduce and apply the chaotic particle-swarm optimization(CPSO) algorithm to solve the control problem with simultaneous constraints on inputs and states. A practical constrained optimization problem of the discrete-time linear system is solved by QP and PSO, respectively. By comparing the simulation results, we show the advantages of the PSO-based MPC algorithm.

**Key words:** model predictive control; particle swarm optimization; optimization with constraints; discrete-time linear systems

### 1 引言(Introduction)

模型预测控制(MPC)是一种优化控制算法, 利用它可以来优化系统的未来输出预测值和通过系统模型得到的未来的系统响应值. 近年来, MPC 得到了学者们的广泛关注, D. Q. Mayne等<sup>[1]</sup>对有约束MPC稳定性及最优做了较为详尽的阐述, 席裕庚等<sup>[2]</sup>较为系统的综述和评价了预测控制性能方面的理论研究. MPC可以控制许多拥有大量控制参数的系统, 最重要的是它可以解决同时带有输入约束和状态约束的控制问题. 然而, 当把这种带约束的优化问题转化为二次规划问题求解时, 可能会出现寻找不到全局最优解的情况, 从而得不到正确的结果. 为了解决这一问题, 引入了粒子群优化算

法(PSO)<sup>[3]</sup>, PSO是一种随机的、并行的优化算法, 在解决带有约束的优化问题<sup>[4]</sup>时, 可以得到较为理想的结果. 近年来, 很多学者在PSO算法改进方面进行了研究. Lovbjerg和Krink<sup>[5]</sup>将自组织临界控制(self-organized criticality, SOC)引入PSO来增加种群的多样性. Krink等<sup>[6]</sup>提出了一种粒子空间扩展的方法来解决粒子间的冲突和聚集问题, 并增强了粒子逃脱局优的能力. 还有结合混沌优化<sup>[7]</sup>算法所提出的基于Tent映射的混沌粒子群优化算法(TCPSO)<sup>[8]</sup>, 等等. 本文将PSO应用到模型预测控制中, 并引入了一个实际的带有约束的线性离散系统的优化控制问题, 分别用二次规划和粒子群优化两种算法去解决, 最后通过仿真结果比较, 充分说明了基于PSO的模

型预测控制算法的有效性.

## 2 模型预测控制策略(Model predictive control)

考虑如下的线性离散系统:

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

$x(k)$ 和 $u(k)$ 分别表示系统在第 $k$ 个采样时刻的状态向量和输入向量. 为了方便, 将预测序列用向量 $X(k)$ 和 $U(k)$ 表示:

$$\begin{aligned} U(k) &= [u(k|k) \ u(k+1|k) \ \cdots \ u(k+N-1|k)]^T, \quad (2) \\ X(k) &= [x(k+1|k) \ x(k+2|k) \ \cdots \ x(k+N|k)]^T. \quad (3) \end{aligned}$$

在这里:  $N$ 为预测时域,  $u(k+i|k)$ 和 $x(k+i|k)$ 分别表示第 $k$ 个采样时刻的输入向量和状态向量在 $k+i$ 时刻的预测值. 是通过如下的预测模型得到的:

$$\begin{aligned} X(k+i+1|k) &= Ax(k+i|k) + Bu(k+i|k), \\ i &= 0, 1, \cdots. \end{aligned} \quad (4)$$

其中定义

$$X(k|k) = x(k). \quad (5)$$

预测控制律是通过优化计算, 从而使得目标函数最小而得到的. 目标函数通常用如下形式表示:

$$\begin{aligned} J(k) &= \sum_{i=0}^N [x^T(k+i|k)Qx(k+i|k) + \\ & \quad u^T(k+i|k)Ru(k+i|k)]. \end{aligned} \quad (6)$$

其中 $Q$ 和 $R$ 为半正定矩阵. 目标函数 $J(k)$ 为 $u(k)$ 的函数, 所以系统的理想输入序列 $U^*(k)$ 可通过下式求解:

$$U^*(k) = \arg \min_u J(k). \quad (7)$$

### 2.1 优化策略(Optimization)

由于预测值与实际的闭环系统响应值之间存在差异, 所以, 选择基于有限时域的目标函数(6)就不能够保证控制闭环系统时可以得到理想的预测效果. 为了解决这一问题, 引入了基于无限时域的目标函数:

$$\begin{aligned} J(k) &= \sum_{i=0}^{\infty} [x^T(k+i|k)Qx(k+i|k) + \\ & \quad u^T(k+i|k)Ru(k+i|k)]. \end{aligned} \quad (8)$$

为了使得两者等价, 将式(6)改写为如下形式:

$$\begin{aligned} J(k) &= \sum_{i=0}^{N-1} [x^T(k+i|k)Qx(k+i|k) + \\ & \quad u^T(k+i|k)Ru(k+i|k)] + \\ & \quad x^T(k+N|k)\bar{Q}x(k+N|k). \end{aligned} \quad (9)$$

$\bar{Q}$ 的计算要通过解如下的Lyapunov等式得到:

$$\bar{Q} - (A + BK)^T \bar{Q} (A + BK) = Q + K^T R K. \quad (10)$$

其中 $K$ 为利用线性规划(LQ)方法解决目标函数为(8)的优化问题时所得到的反馈增益向量.

结合线性状态空间模型(1)和输入预测序列 $U(k)$ , 可以将状态预测值向量表示为如下简化形式:

$$x(k+i|k) = A^i x(k) + c_i U(k), \quad i = 1, \cdots, N. \quad (11)$$

其中

$$c = \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{pmatrix}. \quad (12)$$

$c^i$ 为矩阵 $c$ 的第 $i$ 行.

将目标函数(9)中的 $x(k+i|k)$ 通过式(11)写为用 $U(k)$ 表示的形式:

$$\begin{aligned} J(k) &= U^T(k) H U(k) + 2x^T(k) F^T U(k) + \\ & \quad x^T(k) G x(k). \end{aligned} \quad (13)$$

其中:

$$\begin{cases} H = \sum_{i=0}^{N-1} c_i^T Q c_i + c_N^T \bar{Q} c_N + \mathfrak{R}, \\ F = \sum_{i=0}^{N-1} c_i^T Q A^i + c_N^T \bar{Q} A^N, \\ G = \sum_{i=0}^{N-1} A^{N-i} Q A^i + A^N \bar{Q} A^N. \end{cases} \quad (14)$$

矩阵 $H$ ,  $F$ 和 $G$ 可以离线计算得到, 且有

$$\mathfrak{R} = \begin{pmatrix} R & & \\ & \ddots & \\ & & R \end{pmatrix}. \quad (15)$$

### 2.2 带有约束的优化(Optimization with constraints)

MPC的真正优点在于它可以通过一些在线的计算, 来解决带有约束的优化问题. 下面主要来解决约束条件的转化问题. 系统的输入以及状态的约束条件为

$$\begin{cases} \underline{u} \leq u(k) \leq \bar{u}, \\ \underline{x} \leq x(k) \leq \bar{x}. \end{cases} \quad (16)$$

将输入预测值 $u(k+i|k)$ ,  $i = 0, \cdots, N-1$ 写为向量 $U(k)$ 的形式, 则约束条件就改写为

$$\begin{bmatrix} I \\ -I \end{bmatrix} U(k) \leq \begin{bmatrix} \mathbf{1}\bar{u} \\ -\mathbf{1}\underline{u} \end{bmatrix}. \quad (17)$$

其中: 对于单输入的情况,  $\mathbf{1}$ 为元素全部为1的向量, 当输入维数为 $n_u$ 时,  $\mathbf{1} = [I_{n_u} \cdots I_{n_u}]$ . 同理, 应用式(11), 状态预测值 $x(k+i|k), i = 0, \dots, N$ 的约束可等价地表示为

$$\begin{bmatrix} c_i \\ -c_i \end{bmatrix} U(k) \leq \begin{bmatrix} \bar{x} \\ -\underline{x} \end{bmatrix} + \begin{bmatrix} -A^i \\ A^i \end{bmatrix} x(k),$$

$$i = 1, \dots, N. \quad (18)$$

结合式(17)与式(18), 可以将系统的约束条件写为如下形式:

$$A_c U(k) \leq b_0 + B_x x(k). \quad (19)$$

其中 $A_c, b_0, B_x$ 为可以离线计算的常数矩阵.

### 2.3 二次规划(QP)方法解决带有约束的优化问题(Solve the constrained optimization problems with QP)

结合目标函数表达式(13)以及2.2节中所推导出的约束条件, 这种带有约束的优化问题就可以转化为如下的二次规划问题:

$$\begin{aligned} \min_u & U^T(k) H U(k) + 2x^T(k) F^T U(k) \\ \text{s.t.} & A_c U(k) \leq b_0 + B_x x(k). \end{aligned} \quad (20)$$

这种优化问题可以通过调用MATLAB中的quadprog函数来实现:

$$U^* = \text{quadprog}(H, F * x, A_c, b_0 + B_x * x).$$

然而, 在用这种方法进行优化计算时, 系统中加入约束条件后, 有时会出现寻找不到系统全局最优值的情况, 从而不能得到正确的优化结果.

### 3 粒子群优化算法(Particle swarm optimization)

针对QP算法在解决带有约束的优化问题时所存在的缺陷, 这里引入了一种新的智能优化算法—粒子群优化算法(PSO), 来进行优化计算.

#### 3.1 算法介绍(Algorithm introduction)

在PSO中, 每个优化问题的潜在解都可以想象成D维搜索空间上的一个点, 称之为“粒子”(particle). 一个由 $m$ 个粒子组成的群体(swarm)在D维搜索空间中以一定的速度飞行. 第 $i$ 个粒子的位置表示为:  $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ; 第 $i$ 个粒子的速度表示为:  $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ;  $1 \leq i \leq m, 1 \leq d \leq D$ ; 第 $i$ 个粒子的经历过的历史最好点(所谓最好, 是指计算得到的适应值为最小, 即pbest)表示为:  $\vec{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ; 群体内(或邻域内)所有粒子所经过的最好的点(即gbest)表示为:  $\vec{p}_g =$

$(p_{g1}, p_{g2}, \dots, p_{gD})$ , 其中 $g \in \{1, 2, \dots, m\}$ . 粒子速度和位置的更新方程为

$$\begin{aligned} v_{id}^{k+1} &= \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k), \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^{k+1}. \end{aligned} \quad (21)$$

式中:  $c_1$ 和 $c_2$ 称为学习因子, 为正常数, 通常都等于2;  $r_1, r_2$ 是均匀分布于[0,1]之间的两个随机数;  $\omega$ 称为惯性权重, 其大小决定了对粒子当前速度继承的多少.

#### 3.2 对于约束条件的处理(Constraints management)

将约束条件(18)写为不等式组的形式, 即

$$\begin{cases} c_i U(k) \leq \bar{x} - A^i x(k), \\ c_i U(k) \geq \underline{x} - A^i x(k), \end{cases} \quad i = 1, \dots, N. \quad (23)$$

以2阶系统为例, 来进行推导.

将控制信号 $U(k)$ 中各个预测值的最大值和最小值分别写入两个向量中, 记作 $U_{\max}$ 和 $U_{\min}$ :

$$\begin{cases} U_{\max} = [m_1, m_2, \dots, m_N]^T, \\ U_{\min} = [n_1, n_2, \dots, n_N]^T, \end{cases} \quad (24)$$

并且记

$$\begin{cases} \bar{x} - A^i x(k) = [\bar{a}_{i1} \quad \bar{a}_{i2}]^T, \\ \underline{x} - A^i x(k) = [\underline{a}_{i1} \quad \underline{a}_{i2}]^T. \end{cases} \quad (25)$$

1) 当 $B = [b_{11} \quad b_{12}]^T$ 中各个元素均不为0时: 通过求解不等式组(23), 便可以得到此时约束条件的一般表达式:  $i = 1$ 时:

$$\begin{cases} m_1 = \min\left\{\frac{\bar{a}_{11}}{b_{11}}, \frac{\bar{a}_{12}}{b_{12}}\right\}, \\ n_1 = \max\left\{\frac{\underline{a}_{11}}{b_{11}}, \frac{\underline{a}_{12}}{b_{12}}\right\}. \end{cases} \quad (26)$$

$i = j(j = 2, 3, \dots, N)$ 时:

$$\begin{cases} m_j = \min\left\{\frac{\bar{a}_{j1} - \sum_{q=2}^j b_{q1} n_{j-q+1}}{b_{11}}, \frac{\bar{a}_{j2} - \sum_{q=2}^j b_{q2} n_{j-q+1}}{b_{12}}\right\}, \\ n_j = \max\left\{\frac{\underline{a}_{j1} - \sum_{q=2}^j b_{q1} m_{j-q+1}}{b_{11}}, \frac{\underline{a}_{j2} - \sum_{q=2}^j b_{q2} m_{j-q+1}}{b_{12}}\right\}. \end{cases} \quad (27)$$

2) 当 $B = [b_{11} \quad b_{12}]^T$ 有一个元素为0时:

将1)中所有推导结果中含有这一元素的项忽略即可.

#### 3.3 算法流程(Algorithm flow)

本文中应用的是带混沌初始化的粒子群优化算

法,其算法流程如下:

1) 初始化粒子群. 设定变量的取值范围 $[x_{\min,j}, x_{\max,j}]$ , 群体规模 $m$ , 令 $k = 0$ . 利用Tent映射( $\mu = 1$ )产生 $D$ 个混沌变量:

$$z_j^{i+1} = \mu(1 - 2|z_j^{(i)} - 0.5|), j = 1, 2, \dots, D. \quad (28)$$

式中:  $j$ 表示混沌变量的序号,  $j = 1, 2, \dots, D$ ;  $i$ 表示粒子群序号,  $i = 0, 1, \dots, m$ ;  $z_j$ 表示混沌变量,  $0 \leq z_j \leq 1$ . 根据混沌搜索, 取 $i = 0$ , 给式(28)赋 $D$ 个微小差异的初值, 得到 $D$ 个混沌变量 $z_j (j = 1, 2, \dots, D)$ . 依次取 $i = 0, 1, \dots, m$ , 可得到 $m$ 个初始粒子群.

2) 将混沌变量 $z_j^i (i = 1, 2, \dots, m)$ 分别引入到优化变量中, 使混沌变量的变化范围分别变换到相应的优化变量的取值范围:

$$x_{ij} = x_{\min,j} + z_j^i(x_{\max,j} - x_{\min,j}), j = 1, 2, \dots, D. \quad (29)$$

定义

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}), i = 1, 2, \dots, m.$$

3) 计算适应度值 $f_i$ , 并更新个体历史最优位置 $P_i$ 和全局最优位置 $P_g$ .

4) 根据式(21)和(22)分别对粒子的速度 $V_i$ 和位置 $X_i$ 进行更新.

5) 如果满足停止条件, 则停止, 输出最优解; 否则转步骤6).

6) 令 $k := k + 1$ , 转步骤3).

#### 4 基于粒子群优化算法的MPC(PSO based model predictive control)

将输入预测值向量 $U(k)$ 作为优化变量, 则粒子群中各个粒子的维数 $D$ 就等于预测的步数 $N$ , 并选取函数(13)作为计算各个粒子适应度的目标函数. 算法的具体步骤如下:

**Step 1** 在 $k$ 时刻, 采样获取过程对象已有输入量和状态量;

**Step 2** 根据已有的输入向量和状态向量, 以及离线计算出的矩阵 $H, F$ 和 $G$ 确定目标函数 $J(U(k))$ ;

**Step 3** 利用PSO优化计算得到新的控制序列, 其中评价目标函数为(13);

**Step 4** 重复Step 2和3, 直到满足精度要求的最优控制序列 $U^*(k)$ 找到;

**Step 5** 将最优控制序列 $U^*(k)$ 的第一个分量 $u(k)$ 反馈到控制系统中;

**Step 6** 令 $k := k + 1$ , 转Step 1.

#### 5 仿真研究(Simulations)

为了更好地说明基于粒子群优化算法的MPC的良好性能, 这里引入了一个带有约束的线性离散系统的优化控制问题, 系统的状态方程如下:

$$\begin{cases} x(k+1) = \begin{bmatrix} 1.1 & 2.0 \\ 0.0 & 0.95 \end{bmatrix} x(k) + \begin{bmatrix} 0.0 \\ 0.0787 \end{bmatrix} u(k), \\ y(k) = [1 \ 1]x(k). \end{cases} \quad (30)$$

约束条件为:  $-1 \leq u(k) \leq 1$ , 目标函数定义为式(13)的形式, 其中, 预测步数 $N = 2$ ,  $R = 0.01$ ,  $Q = C^T \cdot C$ ,  $\bar{Q}$ 通过求解李雅普诺夫等式得到, 采用QP算法进行优化控制, 状态初始值取为: $x(0) = (0.8, -0.8)$ , 运行100步, 优化控制仿真结果如图1所示.

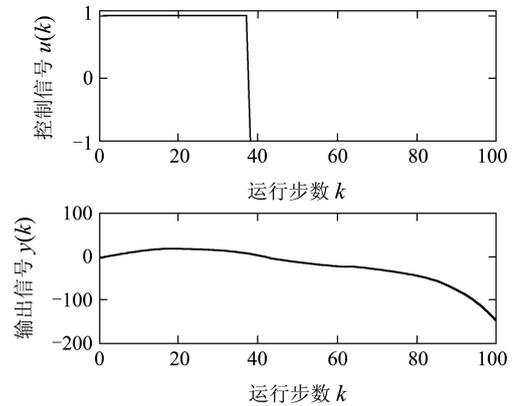


图1 QP算法优化控制结果

Fig. 1 Control result by QP algorithm

接下来使用基于PSO的MPC算法来对系统进行优化控制, PSO的参数选择为: 粒子群规模 $m = 20$ ,  $c_1 = c_2 = 2$ , 惯性权重 $\omega = 0.5$ , 最大迭代次数 $k_{\max} = 1000$ , 最大速度 $v_{\max} = 1$ , 其余的系统参数选择同前面一致, 其优化控制仿真结果如图2所示.

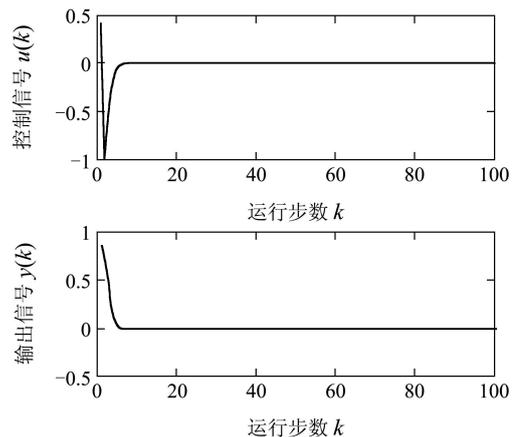


图2 基于PSO的MPC算法优化控制结果

Fig. 2 Control result by PSO based MPC

从仿真结果可以看出,采用二次规划的方法对系统(30)进行优化控制时,系统中加入约束条件后,得到了不稳定的系统响应结果.而使用基于PSO的MPC算法时,就不会出现这种情况,而且可以很快地使系统的响应值收敛到最优.

## 6 结论(Conclusions)

本文将PSO应用到模型预测控制中,用于解决带有约束的线性离散系统的优化控制问题,并在一个实际的线性离散系统上做了仿真实验,分别用二次规划和粒子群优化两种算法去解决这个系统的带有约束的优化控制问题,最后通过仿真结果比较,充分说明了基于PSO的模型预测控制算法的有效性.

## 参考文献(References):

- [1] MAYNEA D Q, RAWLINGSB J B, RAOB C V, et al. Constrained model predictive control: Stability and optimality[J]. *Automatica*, 2000, 36(6): 789 – 814.
- [2] 席裕庚, 庚晓军, 陈虹. 预测控制性能研究新进展[J]. *控制理论与应用*, 2000, 17(4): 469 – 485.  
(XI Yugeng, GENG Xiaojun, CHEN Hong. Recent advances in research on predictive control performance[J]. *Control Theory & Applications*, 2000, 17(4): 469 – 485.)
- [3] KENNEDY J, EBERHART R C. Particle swarm optimization[C] // *Proceeding of IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE Press, 1995: 1942 – 1948.
- [4] RAY T, LIEW K M. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problems[C] // *Proceeding of IEEE International Conference on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 2001, 1: 75 – 80.
- [5] LOVBJERG M, KRINK T. Extending particle swarms with self-organized criticality[C] // *Proceeding of the 4th Congress on Evolutionary Computation*. Washington, DC: IEEE Computer Society, 2002, 2: 1588 – 1593.
- [6] KRINK T, VESTERSTROM J S, RIGET J. Particle swarm optimization with spatial particle extension[C] // *Proceeding of the 4th Congress on Evolutionary Computation*. Washington, DC: IEEE Computer Society, 2002, 2: 1474 – 1479.
- [7] 宋莹, 陈增强, 袁著祉. 基于混沌优化的非线性预测控制器[J]. *控制理论与应用*, 2007, 24(4): 561 – 564.  
(SONG Ying, CHEN Zengqiang, YUAN Zhuzhi. A nonlinear predictive controller based on chaos optimization[J]. *Control Theory & Applications*, 2007, 24(4): 561 – 564.)
- [8] SONG Y, CHEN Z Q, YUAN Z Z. New chaotic PSO-based neural network predictive control for nonlinear process[J]. *IEEE Transactions on Neural Networks*, 2007, 18(2): 595 – 600.

## 作者简介:

**董娜** (1983–), 女, 博士研究生, 主要从事智能预测控制方面的研究, E-mail: dn1983@mail.nankai.edu.cn;

**陈增强** (1964–), 男, 教授, 博士生导师. 目前研究方向为智能预测控制、混沌系统与复杂动态网络、多智能体系统控制, E-mail: chenzq@nankai.edu.cn;

**孙青林** (1963–), 男, 教授, 目前研究方向为自适应控制、嵌入式控制系统, E-mail: sunql@nankai.edu.cn;

**袁著祉** (1937–), 男, 教授, 博士生导师, 目前研究方向为自适应控制及应用、智能控制、计算机控制与管理, E-mail: yuanzhzh@nankai.edu.cn.