

$$y_i(k) = [a_{i1}(1)q^{-1} + a_{i1}(2)q^{-2} + \cdots + a_{i1}(N-1)q^{-N+1} + a_{i1}(N)q^{-N} \frac{1}{1-q^{-1}}] \Delta U(k). \quad (3)$$

式中

$$a_{i1}(k) = [a_{i11}(k) \ a_{i12}(k) \ \cdots \ a_{i1m}(k)], \quad k = 1, 2, \dots, N.$$

由(3)式可得第 i 路预测状态空间表达式

$$\begin{cases} X_i(k+1) = AX_i(k) + B_i \Delta U(k), \\ y_i(k) = X_{i1}(k) = CX_i(k). \end{cases} \quad (4)$$

其中, $X_i(k) = [X_{i1}(k), X_{i2}(k), \dots, X_{iN}(k)]^T$ 为第 i 个输出的预测状态向量

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad C = [1 \ 0 \ \cdots \ 0]_{1 \times N},$$

$$B_i = [a_{i1}^T(1), a_{i1}^T(2), \dots, a_{i1}^T(N)]^T = [B_{i1}, B_{i2}, \dots, B_{im}]_{N \times m},$$

$$B_{ij} = [a_{ij}(1), a_{ij}(2), \dots, a_{ij}(N)]^T, \quad j = 1, 2, \dots, m.$$

(4)式可改写为

$$\begin{cases} X_i(k+1) = AX_i(k) + B_{i1} \Delta u_1(k) + B_{i2} \Delta u_2(k) + \cdots + B_{im} \Delta u_m(k), \\ y_i(k) = CX_i(k), \quad i = 1, 2, \dots, m. \end{cases} \quad (5)$$

由(5)式可得系统第 i 个输出的未来预测输出 $y_i(k+j)$, $j = 1, 2, \dots, p$, 表达式如下:

$$Y_i(k+1) = H_p X_i(k) + G_u \Delta U_M(k) + \sum_{j=1, j \neq i}^m G_{ij} \Delta U_{M_j}(k). \quad (6)$$

式中, $H_p = [\bar{0} \ I_p \ 0]_{p \times N}$, I_p 为 p 阶单位阵, $\bar{0}$ 为 p 维零向量, 0 为 $p \times (N-p-1)$ 的零阵.

$$G_{ij} = \begin{bmatrix} a_{ij}(1) & 0 & \cdots & 0 \\ a_{ij}(2) & a_{ij}(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{ij}(M) & a_{ij}(M-1) & \cdots & a_{ij}(1) \\ \vdots & \vdots & \vdots & \vdots \\ a_{ij}(p) & a_{ij}(p-1) & \cdots & a_{ij}(p-M+1) \end{bmatrix}_{i \times M}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, m.$$

$$\Delta U_M(k) = [\Delta u_1(k/k), \Delta u_1((k+1)/k), \dots,$$

$$\Delta u_1((k+M-1)/k)]^T, \quad i = 1, 2, \dots, m$$

为 k 时刻第 i 路待优化的未来控制增量序列, M 为控制优化步数.

$Y_i(k+1) = [y_i(k+1) \ \cdots \ y_i(k+p)]^T, i = 1, 2, \dots, m.$ 为第 i 路未来 p 步预测输出向量.

(6)式中 $X_i(k)$ 可由如下预测状态观测器获得其估计值,即

$$\begin{aligned} \hat{X}_i(k+1) &= A\hat{X}_i(k) + B_i \Delta U(k) + \\ &K_i [y_i(k) - C\hat{X}_i(k)], \quad i = 1, 2, \dots, m. \end{aligned} \quad (7)$$

$y_i(k)$ 为第 i 个输出在 k 时刻的实测值; $K_i = [k_{i1} \ \cdots \ k_{iN}]^T$ 为观测器反馈增益阵.

将(6)式中 $X_i(k)$ 换成其估计值 $\hat{X}_i(k)$, 便为系统的第 i 个输出的预测方程

$$\begin{aligned} \hat{Y}_i(k+1) &= H_p \hat{X}_i(k) + G_{ii} \Delta U_{M_i}(k) + \\ &\sum_{j=1, j \neq i}^m G_{ij} \Delta U_{M_j}(k), \quad i = 1, 2, \dots, m. \end{aligned} \quad (8)$$

2.2 分散优化的 DMC 算法 (DMC algorithm with decentralized optimization)

分散优化就是基于各路输出预测方程(8)式, 分别按如下分散二次型指标优化求解各路控制增量序列 $\Delta U_{M_i}(k)$:

$$J_i(\Delta U_{M_i}(k)) = \frac{1}{2} \{ \| Y_{di}(k+1) - \hat{Y}_i(k+1) \|_{Q_i}^2 + \| \Delta U_{M_i}(k) \|_{R_i}^2 \}. \quad (9)$$

$Y_{di}(k+1) = [y_{di}(k+1) \ \cdots \ y_{di}(k+p)]^T$ 为第 i 路参考输入序列.

但是, 用预测方程(8)式按分散二次型指标优化, 求解第 i 路控制增量序列 $\Delta U_{M_i}(k)$, 必须事先解出其余各路控制增量序列 $\Delta U_{M_j}(k), j \neq i$, 然而求解 $\Delta U_{M_j}(k), j \neq i$ 又必须先解出 $\Delta U_{M_i}(k)$. 因此多变量 DMC 通常是不能按上述分散优化方法分别求解 $\Delta U_{M_i}(k), i = 1, 2, \dots, m$. 只能按集中优化方法一次同时求出所有序列 $\Delta U_M(k) = [\Delta U_{M_1}^T(k) \ \cdots \ \Delta U_{M_m}^T(k)]^T$. 文献[2]中给出一种分散优化方法, 其基本思想是先不考虑耦合影响, 即令(8)式中的 $\Delta U_{M_j}(k) = 0, j \neq i$, 分别按指标(9)优化求得各路序列 $\Delta U_{M_i}^{(1)}(k), i = 1, 2, \dots, m$, 用解得的 $\Delta U_{M_i}^{(1)}(k)$ 代替(8)式相应的 $\Delta U_{M_j}(k)$, 再按(9)式优化求得各路序列 $\Delta U_{M_i}^{(2)}(k)$, 这样反复按(9)式迭代优化 L 次, 最后求得各路序列 $\Delta U_{M_i}^{(L)}(k)$ 作为分散优化解. 这种算法虽可行, 但计算量过大. 这里我们根据 DMC 滚动优化特点, 利用前一步 ($k-1$ 时刻) 已优化求出的各路未来控制增量序列

式时,可以将(10)式中的耦合通道的动态矩阵 G_y , 乘一个压缩因子 $K_y = k_y I_p, k_y < 1, j \neq i, I_p$ 为 p 阶单位阵. 即将分散优化控制律(10)式改为

$$\Delta U_{Mj}(k) = [G_u^T Q_i G_u + R_i]^{-1} G_u^T Q_i [Y_d(k+1) - H_p \hat{X}_i(k) - \sum_{j=1, j \neq i}^m K_{y_j} G_{y_j} S \Delta U_{Mj}(k-1)] \quad (16)$$

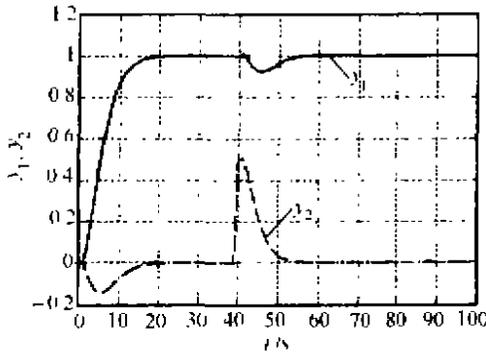
这样,矩阵 $K_y G_y S_0$ 的特征值也必相应减小,通过减小比例因子 K_y ,总可以使(13)式得以满足. 这样做虽能满足(13)式,但将使系统控制动态性能有所下降,闭环耦合作用随 K_y 减小而增强,所以 K_y 不可取得过小.

4 仿真试验(Simulation tests)

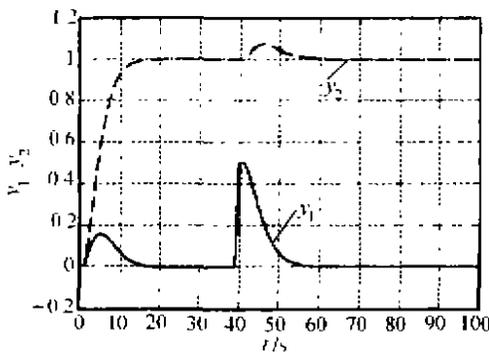
为检验上述分散优化算法的有效性,特取如下两个 2×2 系统. 系统1和2的传递函数矩阵分别为:

$$G_1(s) = \begin{bmatrix} \frac{1}{100s+1} & \frac{0.9}{50s+1} \\ \frac{-1}{50s+1} & \frac{1.75}{400s+1} \end{bmatrix}$$

$$G_2(s) = \begin{bmatrix} \frac{1}{100s+1} & \frac{5}{50s+1} \\ \frac{-6.25}{50s+1} & \frac{3.75}{100s+1} \end{bmatrix}$$



(a) 设定值 $Y_{r1}=1, Y_{r2}=0$

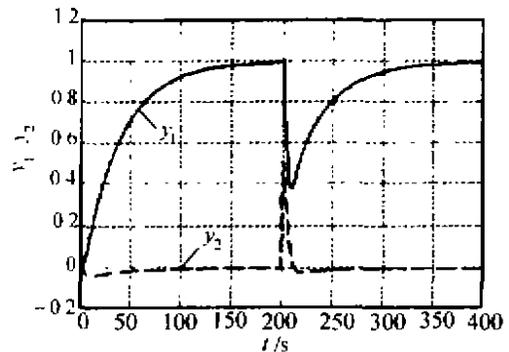


(b) 设定值 $Y_{r1}=0, Y_{r2}=1$

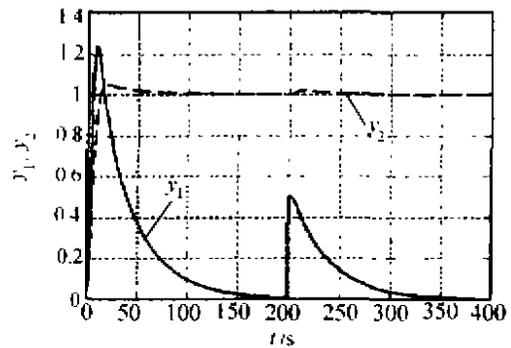
图2 系统1分散优化DMC仿真结果
Fig. 2 System 1 response under decentralized optimization DMC

对系统1,取 $r_1 = r_2 = 10, M = 2, p = 10$,从第40个采样时刻起,在设定值为零的一路输出加入幅值为0.5的阶跃扰动. 用文中不加比例因子的分散优化DMC控制算法,其结果如图2所示(y_1 为实线, y_2 为虚线). 可见该算法有良好的控制效果.

对于强耦合系统2,如果采用不加比例因子的算法,通过调节 M, p, r 值难以使系统稳定;即使稳定,系统的动态特性也很差. 如图3所示,取 $M = 2, p = 10, r_1 = r_2 = 54$,从第200个采样时刻起,加入幅值为0.5的阶跃扰动. 从图中可见,系统响应很慢,动态耦合严重.

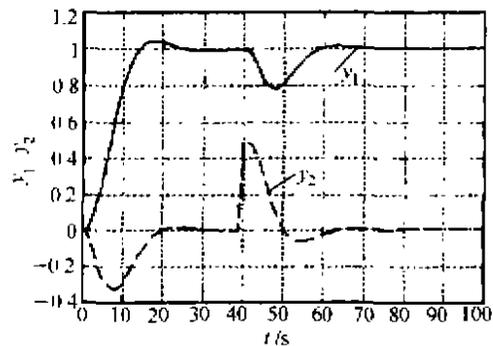


(a) 设定值 $Y_{r1}=1, Y_{r2}=0$



(b) 设定值 $Y_{r1}=0, Y_{r2}=1$

图3 系统2分散优化DMC仿真结果
Fig. 3 System 2 response under decentralized optimization DMC without contract factor



(a) 设定值 $Y_{r1}=1, Y_{r2}=0$

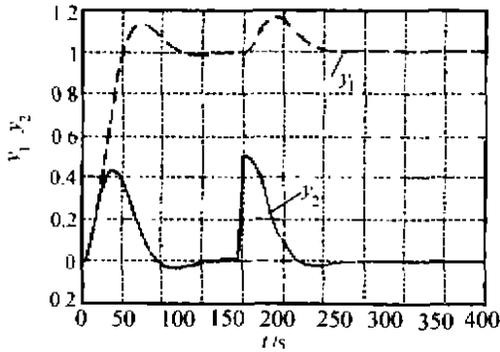
(b) 设定值 $y_{r1} = 0, y_{r2} = 1$

图 4 系统2加压缩因子分散优化DMC仿真结果

Fig. 4 System 2 response under decentralized optimization DMC with contract factor

改用加比例因子的算法,比例因子设为 $k_{12} = 0.4, k_{21} = 0.4$;其它参数同上,控制效果如图4所示.从图中可以看出,加比例因子后系统易于通过调节 k, M, p, r 值达到稳定,并且可有效改善系统的动态性能.

5 结论(Conclusion)

本文中提出的多变量系统分散优化 DMC 算法,根据预测控制的滚动优化策略的特点,利用前一步已优化的控制增量序列信息预测未来各控制序列对系统各输出的耦合影响,将集中优化 DMC 算法分解为若干单变量系统优化的 DMC 算法,使多变

量系统 DMC 算法的参数设计和算法求解计算大为简化.算法分析和仿真实验表明该算法是有效的,特别是对强耦合多变量系统,通过选择压缩因子,也可使系统稳定,且可实现稳态解耦,使各路输出均无静差.

参考文献(References)

- [1] 李嗣福.模型预测控制算法改进[J].信息与控制,1988,17(1): 15-20
- [2] 席裕庚.预测控制[M].北京:国防工业出版社,1993
- [3] 舒迪前.预测控制及其应用[M].北京:机械工业出版社,1996

本文作者简介

李嗣福 1939年生,1964年毕业于哈尔滨工业大学无线电系自控专业,1985—1986年在加拿大阿尔伯达大学化工系进修.现为中国科技大学自动化系教授.主要研究兴趣是多变量控制、预测及自适应控制和计算机控制工程等.

陈忠保 1970年生,1992年毕业于国防科技大学自控系,1997年在中国科技大学博士学位,并留校任教.研究方向为预测控制、自适应控制和计算机工业应用

刘勇 1972年生,1990年考入中国科技大学,1995年本科毕业,1999年获硕士学位.现在美国纽约工学院读博士学位.

赵守忠 1940年生,1965年毕业于中国科技大学自动化系.现是中国科技大学自动化系副教授.主要研究兴趣是多变量控制,传感器与自动测量技术以及计算机工业应用等.