第6卷 增刊1　　控制理论与应用　　Vol.6 Suppl.I. 1
1989年1月　　CONTROL THEORY AND APPLICATIONS　　Jan., 1986

用分割映射法解容差设计问题

叶庆凯　　毛剑琴　　吴智铭
（北京大学力学系）（北京航空学院自动控制系）（上海交通大学电机与计算机系）

摘　要

本文介绍控制系统的容差设计方法．它使设计参数在一定范围内变化时系统仍能满足指定的性能要求．文中给出了容差设计的可实现程序方案．讨论了具体的实现步骤，并给出了两个算例．

一、引　言

当用不等式方法设计控制系统时，对系统的性能要求归结为在n维欧氏空间R^n中寻找一个点x满足不等式

$$f^{(j)}(x)\leqslant 0,\ j=1,\ 2,\ \cdots,m. \tag{1}$$

称此x为标准设计值．但实际实现x时不可避免地会有误差．因而为保证实现的系统能满足要求应要求系统具有鲁棒性，即当实现的x对标准值有小偏离时系统仍应满足要求．但是，以往的鲁棒性讨论只是一种定性的讨论，即使对鲁棒的系统，到底容许实现的参数偏离标准值多少仍是不知道的．容差设计可以认为是一种定量的鲁棒性研究，即寻找点x满足不等式

$$f^{(j)}(x+t)\leqslant 0,\ j=1,\ 2,\ \cdots,\ n;\ t\in T, \tag{2}$$

其中T是R^n中的一个区域，可事先规范化为

$$T\triangleq\{t\in R^n|\,|t^i|\leqslant 1\}, \tag{3}$$

这里t^i是t的第i个分量．这样的容差设计也就是只要实现的参数偏离标准值不越出区域T的范围，就能保证系统仍满足性能要求．

问题（2）（称为P_T）是无限维的．它还可表达为：寻找一个点x满足不等式

$$\theta_T(x)\leqslant 0, \tag{4}$$

其中θ_T：$R^n\to R$定义为

$$\theta_T(x)\triangleq\max_t\{\phi(x+t)|t\in T\}, \tag{5}$$

而ϕ：$R^n\to R$定义为

$$\phi(x)\triangleq\max_j\{f^{(j)}(x)|j=1,\ 2,\ \cdots,\ m\}. \tag{6}$$

本文于1986年5月17日收到．1986年10月7日收到修改稿．

二、算 法 概 述

以F记问题（1）的解集，以G记问题（2）即问题P_T的解集，即

$$F \triangleq \{x \in R^n \mid \phi(x) \leqslant 0\}, \tag{7}$$

$$G \triangleq \{x \in R^n \mid \theta_T(x) \leqslant 0\}$$
$$= \{x \in R^n \mid \phi(x+t) \leqslant 0,\ t \in T\}. \tag{8}$$

集F与G的关系如图1所示．显然，G是F的子集．

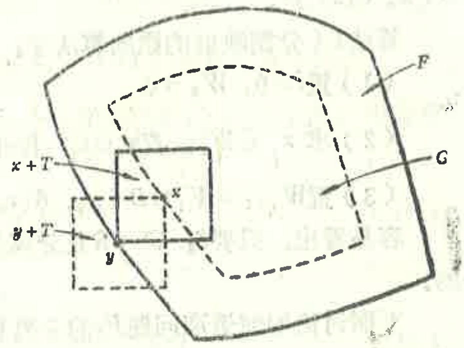

图 1　集F与集G之间的关系

可以用外逼近法[1]将无限维问题P_T化为一系列有限维问题P_{T_i}：

$$\phi(x+t) \leqslant 0, \qquad t \in T_i, \tag{9}$$

而T_i是T的子集，且只含有限个点．但在一般性算法中，确定G的一个外逼近是相当复杂的问题．而凭藉分割映射方法可以用相当简单的办法来构成G的一个外逼近．具体说来，在第i次迭代时，可以用

$$W_i = U\{B(x_j,\ \varepsilon_j) \mid j < i\} \tag{10}$$

或它的一个子集来近似G^c（集G的补集），其中$B(x_j,\ \varepsilon_j)$是中心在x_j，半径为$\varepsilon_j > 0$且与G不相交的开球．显然，$G \cap W_i = \phi$，因而$G \subset W_i^c$．这样，W_i^c乃是G的一个外逼近．

整个算法在于在W_{i-1}^c中找一点x_i，然后确定ε_i使$G \cap B(x_i,\ \varepsilon_i) = \phi$，再令$W_i = W_{i-1} \cup B(x_i,\ \varepsilon_i)$．反复迭代下去直至$\varepsilon_i = 0$，这表明相应的$x_i$已在$G$中．本文以后将讨论确定$\varepsilon_i$以及构造$W_i$的具体方法．

三、分离算子与分割映射

定义　设G是闭集，函数δ：$G^c \to R$是分离算子，如果

（1）对一切$x \in G^c$，有$\delta(x) > 0$；

（2）若$i \to \infty$时有$x_i \in G^c$，$x_i \to x^*$且$\delta(x_i) \to 0$，则$\delta(x^*) = 0$，即$x^* \in G$．

容易证明，若δ是分离算子，且无穷序列$\{x_i\}$满足

$$x_i \in W_i^c, \qquad i = 1,\ 2,\ \cdots \tag{11}$$

$$W_i \triangleq \cup\{B(x_j,\ \delta(x_j)) \mid j = 1,\ 2,\ \cdots,\ i-1\},$$

则$\{x_i\}$的任一聚点均属于G．

定义　若对一切$x \in G^c$有

$$0 < \delta(x) \leqslant d(x,\ G) \triangleq \min_y\{\|y-x\| \mid y \in G\},$$

$$B(x,\ \delta(x)) \triangleq \{\|y-x\| < \delta\},$$

则从点到集的映射$x \to B(x,\ \delta(x))^c$称为分割映射，显然有$x \in B(x,\ \delta(x))$，$G \subset$

$B(x,\ \delta(x))^c$。

算法1（分割映射的原则算法）：

（1）置$i=0$，$W_0=\phi$

（2）求$x_i\in W_i^c$。若$x_i\in G$，停止；否则进行（3）。

（3）置$W_{i+1}=W_i\cup B(x_i,\ \delta(x_i))$，$i=i+1$，返回（2）

容易看出，只要δ：$G^c\to R$是分离算子，由算法1得到的序列$\{x_i\}$的聚点x^*是属于G的。

下面讨论如何选取问题P_T的分离算子．首先，将分割映射定义中的范数取为无穷范数：

$$B_\infty(x,\ p)\triangleq\{y\in R^n \mid \|y-x\|_\infty<p\},\tag{12}$$

因而

$$x+T\triangleq\{y\in R^n \mid \|y-x\|_\infty\leqslant 1\}=\overline{B}_\infty(x,\ 1),\tag{13}$$

这里上面的一横表示集的闭包．这样，问题P_T成为寻找点x使$x+T\subset F$（看图1）．

再定义函数η：$G^c\to R$如下

$$\eta(x)\triangleq\min_y\{\|y-x\|_\infty \mid y\in x+T,\ y\in U\},\tag{14}$$

其中$U=\overline{F^c}\triangleq\{x\in R^n \mid \phi(x)\geqslant 0\}$．图2说明了$\eta(x)$的几何意义．以后为方便起见，记使式（14）取最小的y值为$w(x)$．因而$\eta(x)=\|w(x)-x\|_\infty$．

定义

$$\delta(x)=1-\eta(x).\tag{15}$$

可以证明，若问题P_T中的函数$f^{(i)}$均是连续的，则由式（15）定义的函数δ：$G^c\to R$是一个分离算子．

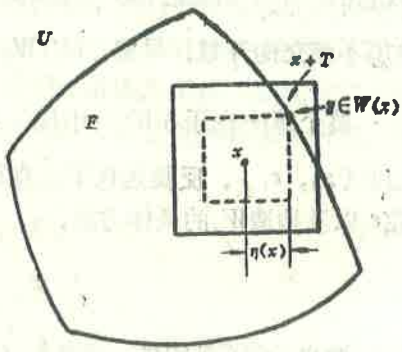

图2　$\eta(x)$的几何意义

现在，算法1可具体化为如下算法．

算法2：

（1）置$i=0$，$W_0=\phi$。

（2）求$x_i\in W_i^c$。若$x_i\in G$，停止；否则进行（3）。

（3）按式（14）计算$\eta(x_i)$，令$\delta(x_i)=1-\eta(x_i)$．

（4）置$W_{i+1}=W_i\cup B_\infty(x_i,\ \delta(x_i))$，$i=i+1$，返回（2）。

四、容差问题的分割映射算法

算法2仍是一个原则算法．按式（14）计算$\eta(x_i)$以及第（2）步中判断x_i是否属于G均需无穷次计算．而且，每迭代一次，W_i中将增加一个约束，使问题越来越复杂．为构造一个可实现的算法，可采取如下措施．

（1）用范数$\|\cdot\|_2$代替无穷范数$\|\cdot\|_\infty$算法 2 中第（4）步改为$W_{i+1}=W_i\cup B_2(x_i,$ $\delta(x_i))$，以使第（2）步中要处理的约束$x\notin B_2(\tilde{x},\ \delta(\tilde{x}))$具有可微形式。

（2）在算法2的第（3）步中，用近似的$\overline{\delta}(\cdot,\cdot)$来代替$\delta(\cdot)$。为此，用含有$x+T$中的$\tau(i)$个点的集合$V_{\tau(i)}$来代替$x+T$本身。式（14）成为

$$\overline{\eta}(x,\ i)\triangleq\min_{y}\{\|y-x\|_\infty|y\in V_{\tau(i)},\ \phi(y)\geqslant 0\}. \tag{16}$$

将使式（16）取最小的y值记为$w(x,\ i)$。相应地，

$$\overline{\delta}(x,\ i)\triangleq 1-\overline{\eta}(x,\ i)=1-\|w(x,\ i)-x\|_\infty\leqslant\delta(x) \tag{17}$$

是$\delta(x)$的一个近似。

为使算法收敛，$\tau(i)$应满足条件：（i）当$i\to\infty$时$\tau(i)\to\infty$；（ii）对一切$x\in G^c$，存在整数$I(x)$，对一切$i\geqslant I(x)$均有$w(x,i)\in U$且在$x+T$的内部；（iii）对G^c的任一紧的子集X，当$i\to\infty$时，$|\eta(x)-\|w(x,\ i)-x\|_\infty|$对$x\in X$应一致地趋于零。

（3）为不使算法2第（2）步中的W_i^c的结构过于复杂，可采取抛弃无效约束技术。为此将第（4）步中决定W_i的计算公式改为

$$W_i=U\{B_2(x_j,\ \overline{\delta}(x_j,\ j))|j\in J(i-1)\}, \tag{18}$$

其中$J(i-1)$是$\{1,\ 2,\ \cdots,\ i-1\}$的子集，定义为

$$J(i)\triangleq\{j<i\mid\overline{\delta}(x_j,\ j)>\varepsilon_{ij}\}, \tag{19}$$

而ε_{ij}具有性质：（i）对一切i，$j<i$有$\varepsilon_{ij}>0$，（ii）当$i\to\infty$时ε_{ij}对j一致地单调递升地趋向于某一$\overline{\varepsilon}_j$；（iii）当$j\to\infty$时$\overline{\varepsilon}_j$单调递减地趋于零。

式（18）中i为当前迭代次数，j为建立相应约束$x\notin B_2(x_j,\ \overline{\delta}(x_j,\ j))$时的迭代次数。

算法3：

（1）置$i=0$，$W_0=\phi$，$J(0)=\phi$，$\tau(0)=100$。

（2）求$x_i\in W_i^c$即求x_i满足

$$\overline{\delta}(x_j,\ j)^2-\|x_i-x_j\|_2^2\leqslant 0,\ j\in J(i).$$

（3）计算$w(x_i,\ i)$。

（4）计算$\overline{\delta}(x_i,\ i)=1-\|w(x_i,\ i)-x_i\|_\infty$。

（5）若全部$\tau(i)$个试验点均满足条件$\phi(z_i)\geqslant 0$，置$x_{i+1}=x_i$，$W_{i+1}=W_i$，$i=i+1$，计算$\tau(i+1)$后返回（3）；否则进行（6）。

（6）计算$J(i+1)$，$\tau(i+1)$。

（7）置$W_{i+1}=U\{B_2(x_j,\ \overline{\delta}(x_j,\ j))|j\in J(i+1)\}$。

（8）置$i=i+1$，返回（2）。

可以证明，由算法3求得的无穷序列$\{x_i\}$的任一聚点x^*均位于G中。

五、实现算法 3 时的一些具体考虑

（1）原则上讲，算法 3 并不需要给定初始迭代点。但若能给出一个合适的初始点，显然能加速算法的收敛。我们取$x_0 \in F$。

（2）每次计算$w(x_i, i)$可能要计算$\tau(i)$次ϕ值，我们限制$\tau(i)$的最大值$\tau_{max}=500\sim1000$。

（3）在$\tau(i)=\tau_{max}$且$\overline{\delta}(x_i, i)<0.05$时计算结束，此时$x_i$即所求的解。否则在$i=I_{max}$时计算结束，并认为解不存在（不妨另取初始点再试试）。

（4）为实现算法3中的第（2）步，即求$x_{i+1}\in w_{i+1}^c$，采用了带外罚函数的组合变尺度方法。这需要一个初始点。由于相应的约束函数

$$\overline{\delta}(x_j, j)-\|x-x_j\|_2^2$$

在$x=x_j$处的梯度为零，取点x_i为初始点是不合适的。我们的程序中，在求得新的$w(x_i, i)$后，在$w(x_i, i)$至x_i的方向上对$\phi(x)$进行一次极小值搜索，将所得结果x_i'作为下一次解问题$x\in W_{i+1}^c$的初始值。若上述搜索过程所得的x_i'十分靠近x_i，则取

$$x_i''=x_i+\lambda(x_i-w(x_i, i))$$

为解问题$x\in W_{i+1}^c$的初始值，其中

$$\lambda=a\overline{\delta}(x_i, i)/\|x_i-w(x_i, i)\|_2,\ a>1.$$

（5）算法执行过程中，若发现迭代点越出了F(即$\overline{\delta}(x_i, i)\geqslant1$)，就将它拉回到$F$中来。

最后，我们提出如下用分割映射解容差问题的计算方案。

算法4：

（1）置$i=0$，$W_0=\phi$，$J(0)=\phi$，$x_0'=x(0)$。

（2）以x_0'为初始点解问题$\min\phi(x)$得$x_0\in F$。

（3）计算$\tau(i)$，$y_i=w(x_i, i)$。

（4）计算$\overline{\delta}(x_i, i)=1-\|y_i-x_i\|_2$。

（5）若$\overline{\delta}(x_i, i)>0.99$，置$x_0'=x_i$，进行（2）；否则进行（6）。

（6）若$\tau(i)=\tau_{max}$且$\overline{\delta}(x_i, i)\leqslant0.05$，停止；否则进行（7）。

（7）若全部$\tau(i)$个试验点均满足条件$\phi(z_i)\geqslant0$，置$x_{i+1}=x_i$，$W_{+1}=W_i$，$i=i+1$，进行（8）；否则进行（9）。

（8）若$i\geqslant I_{max}$，停止；否则进行（3）。

（9）计算$J(i+1)$，$W_{i+1}=U\{B_2(x_j,\ \bar{\delta}(x_j,\ j))\,|\,j\in J(i+1)\}$.

（10）求点$x_{i+1}\in W^c_{i+1}$.

（11）置$i=i+1$. 若$i\geqslant I_{max}$，停止；否则进行（3）.

六、例　子

（1）集F定义为

$$f^{(1)}(x)=-x_1^2+x_2-1.5\leqslant 0,$$

$$f^{(2)}(x)=x_1-0.5(x_2-1)^2-1.5\leqslant 0,$$

$$f^{(3)}(x)=-0.2x_1^2-x_2-1\leqslant 0,$$

$$f^{(4)}(x)=-x_1-(2x_2-1)^2-1\leqslant 0,$$

$$f^{(5)}(x)=x_1^2+x_2^2-13\leqslant 0.$$

它的定义范围如图3所示.

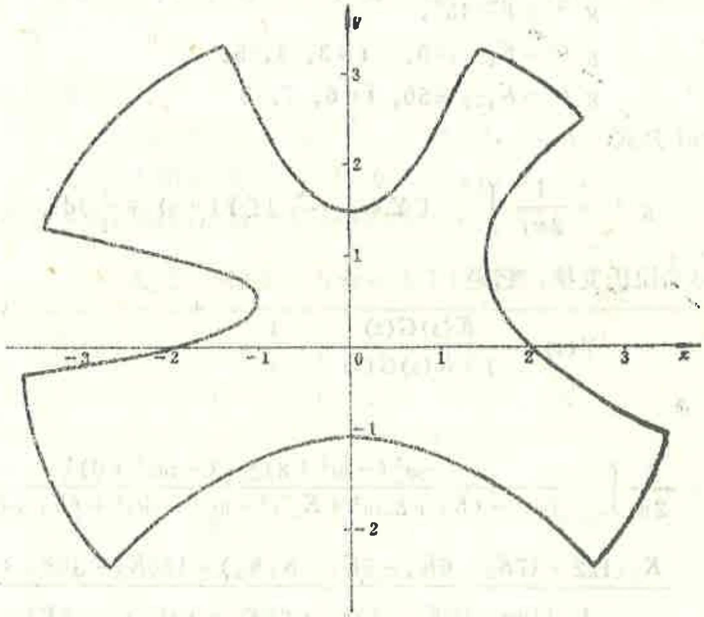

图 3　例1的可行集F

用我们的程序在各种不同的初始点下求得的结果示于表1.

（2）设系统的传递函数为

$$G(s)=\frac{1}{(s+3)(s^2+2s+2)},$$

要求设计一个PID控制器

表 1　例1在不同初始点下的结果

初始点		结果	
x_1	x_2	x_1	x_2
-1.0	-2.0	0.25557	0.00000
4.0	8.0	0.00000	0.00000
1.0	3.0	0.00000	0.24117
-1.0	2.0	0.01720	0.36344
3.0	-1.0	0.56020	0.23229

$$K(s)=K_1+K_2/s+K_3s$$

使$0\leqslant K_i\leqslant 50$ ($i=1,2,3$) 有±1的误差情况下系统在单位阶跃输入时的瞬态反应对于阶跃函数的误差的平方积分总小于0.2，且相位裕度总不小于45°，即要求

$$g^{(1)}=\int_0^{\infty}(y(t)-1)^2\mathrm{d}t\leqslant 0.2,$$

$$g^{(2)}=\theta\geqslant 45^\circ,$$

$$g^{(i)}=K_{i-2}\geqslant 0,\quad i=3,4,5,$$

$$g^{(i)}=K_{i-5}\leqslant 50,\quad i=6,7,8$$

由 Parseval 公式

$$g^{(1)}=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left[Y(s)-\frac{1}{s}\right]\left[Y(-s)+\frac{1}{s}\right]\mathrm{d}s,$$

其中$Y(s)$是$y(t)$的拉氏变换，它是

$$Y(s)=\frac{K(s)G(s)}{1+K(s)G(s)}\cdot\frac{1}{s}.$$

经计算后可得[2]

$$g^{(1)}=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{-\omega^2(-\omega^2+8)^2-(-5\omega^2+6)^2}{[\omega^4-(K_3+8)\omega^2+K_2]^2+\omega^2[-5\omega^2+(K_1+6)^2]}\mathrm{d}\omega$$

$$=\frac{K_2(122+17K_1+6K_3-5K_2+K_1K_3)+180K_3-36K_1+1224}{K_2(408+56K_1-50K_2+60K_3+10K_1K_3-2K_1^2)}.$$

因而不等式约束$g^{(1)}\leqslant 0.2$即约束

$$f^{(1)}=K_2(40.4+5.8K_1+5K_2-6K_3-K_1K_3+0.4K_1^2)+180K_3-36K_1+1224\leqslant 0.$$

现在令

$$G(j\omega)K(j\omega)=R(\omega)+jI(\omega),$$

经计算可得

$$R(\omega)=\frac{(K_2-K_3\omega^2)(\omega^2-8)+K_1(-5\omega^2+6)}{\omega^2(\omega^2-8)^2+(-5\omega^2+6)^2}$$

$$I(\omega)=\frac{K_1\omega(\omega^2-8)-(K_2-K_3\omega^2)(-5\omega_2+6)/\omega}{\omega^2(\omega^2-8)^2+(-5\omega^2+6)^2}.$$

要求相位裕度不小于45°，即要求

$$I(\omega_0)/R(\omega_0)\leqslant-1$$

其中ω_0由方程

$$R^2(\omega_0)+I^2(\omega_0)=1$$

决定．因而不等式约束$g^{(2)}\geqslant 45°$即约束

$$f^{(2)}=(\omega_0^2-8)(K_1\omega_0+K_2-K_3\omega_2^0)$$
$$+(-5\omega_0^2+6)(K_1-K_2/\omega_0+K_3\omega_0)\leqslant 0,$$

其中ω_0是由方程

$$\omega_0^8+9\omega_0^8+(4-K_3^2)\omega_0^4+(36-K_1^2+2K_2K_3)\omega_0^2-K_2^2=0$$

决定的正实数解．

其它的不等式约束等价于

$$f^{(i)}=-10000K_{i-2}\leqslant 0,\qquad i=3,\ 4,\ 5,$$
$$f^{(i)}=-10000(50-K_{i-5})\leqslant 0,\qquad i=6,\ 7,\ 8.$$

用我们的程序在各种不同的初始点下求得的结果示于表2．

表 2　例2在不同初始点下的结果

初始点			结果		
K_1	K_2	K_3	K_1	K_2	K_3
25.00	25.00	25.00	17.20	24.36	38.92
10.00	20.00	30.00	42.00	20.00	30.00
20.00	20.00	20.00	20.00	20.50	36.00
40.00	10.00	20.00	29.99	18.25	27.54

参 考 文 献

〔1〕 Mayne, D. Q., Polak, E., Trahan, R., An Outer Approximations Algorithm for Computer Aided Design Problems, JOTA, 28(1979).

〔2〕 Gonzaga, C., et al, An Improved Algorithm for Optimization Problems with Functional Imequility Constraints, IEEE Trans. Auto. Çont., AC-25, (1980).

Solve Design Problems with Parameter Tolerances by Cut Map Method

Ye Qingkai
(Department of Mechanics, Peking University)
Mao Jianqin
(Department of Automatic Control, Beijing Institute of Aeronautics and Astronautics)
Wu Zhiming
(Department of Electrical Engineering and Computer, Shanghai Jiaotong University)

Abstract

This paper introduces a design method for cyntrol systems with parameter tolerances. This method can consider the problem with parameter tolerances, i. e., the system will always satisfy the performance demands when the design parameters are changed in a given region. In this paper, an implementable algorithm for the tolerance problem is discussed. Two numerical examples are presented.