第1卷 创刊号　　控制理论与应用　　Vol.1, Initial Issue
1984年1月　　CONTROL THEORY AND APPLICATIONS　　January, 1984

线性多变量系统最优定常输出反馈设计的一种算法

毛 剑 琴
（北京航空学院）

摘　要

本文在Levine－Athans方法基础上提出了一种输出反馈最优化设计的算法。这一算法采用了函数最优化技术，比Levine－Athans算法在计算上大为简化，收敛性也得到了保证。

求梯度矩阵是这一算法的关键。文中给出了一种求梯度矩阵的方法，较其它方法更为直观和简单。

将共轭斜量法及Armijo法则推广运用到矩阵函数最优化问题。采用符号函数法求解矩阵Lyapunov方程。从而实现了这一算法的各主要步骤。

在CDC—6500上编制了实现这个算法的程序。

作为例子，文中对M＝2.7超音速运输机的侧向增稳控制系统进行了设计。

一、引　言

线性多变量系统最优状态反馈问题从理论上和算法上都已比较完善。但在实际问题中，常常遇到这样的困难，即全部状态量不是总能测到的，有时有些状态量不能直接测到，有时只能测到状态量的组合。为解决这个问题，有两种途径：1)设计状态观测器；2)直接用测到的输出量进行反馈。由于状态观测器的阶数比较高时，造成实际使用中的困难，特别是随着大系统的出现，这一方法已不能满足设计的要求。因此，输出反馈的研究就显得更为重要。

从理论上讲，输出反馈的研究与状态反馈的研究有着明显不同的地方。解决问题的途径也不相同。因此，输出反馈已形成了现代控制理论中的一类问题。至今，输出反馈设计从理论上和算法上并未得到完满的解决。它仍是一个正在发展的、相当活跃的领域。本文将对其算法问题进行讨论。

七十年代初以来，人们在输出反馈设计方面作了大量的工作。其中Levine－Athasn线性二次型最优定常输出反馈设计方法已引起了充分的重视。但由于Levine－Athans算法中要解两个非线性矩阵代数方程和一个Lyapunov方程，计算很繁重，算法的收敛性

本文于1982年11月30日收到。1993年6月21日收到修改稿。

也没有最后证明，于是有些文献中探讨如何用函数最优化技术来求解。

本文基于这一思想，将共轭斜量法推广到矩阵函数的寻优，进一步提出了输出反馈最优化设计的一种算法。这一算法只要解两个线性的矩阵Lyapunov 方程及用共轭斜量法寻优。这些问题的解决从计算技术上是比较成熟的。因此，这一算法较 Levine-Athans 算法简化了计算，保证了收敛性。

求梯度矩阵是共轭斜量法的关键。本文给出了一种更为简单和直观的推导方法。

在CDC—6500上编制了实现这一算法的程序。

为说明这一算法，对 $M=2.7$ 超音速运输机的侧向增稳控制系统进行设计，并给出了计算结果。

二、Levine-Athans 算法及问题的提出

线性二次型最优定常输出反馈问题可描述如下[1],[2]：

对系统

$$\begin{cases} \dot{x}(t)=Ax(t)+Bu(t), \quad x(0)=x_0, \\ y(t)=Cx(t), \end{cases} \tag{2.1}$$

其中 $x\in R^n$, $u\in R^m$, $y\in R^r$, $A\in R^{n\times n}$, $B\in R^{n\times m}$, $C\in R^{r\times n}$。

标准的无限时间线性二次型代价函数为

$$J=\frac{1}{2}\int_0^\infty [x'(t)Qx(t)+u'(t)Ru(t)]dt, \tag{2.2}$$

其中 $Q\in R^{n\times n}$ 且对称半正定，$R\in R^{m\times m}$ 对称正定，要求输出反馈规律$F^*\in R^{m\times r}$

$$u(t)=F^*y(t),$$

使 $J(F^*)\leq J(F)$, (2.3)

其中 $F\in \mathbf{F}$, $\mathbf{F}=\{F\in R^{m\times r}: A+BFC$ 渐近稳定$\}$。

由于代价函数J不仅是F的函数，也是x_0的函数，因此满足（2.3）的F^*也将是初始条件x_0的函数。这在实际应用中显然是有困难的。

Levine-Athans 采用了对x_0进行统计平均的方法，得到改进后的代价函数为

$$\hat{J}(F)=\frac{1}{2}\mathrm{trace}\left\{\int_0^\infty e^{(A+BFC)'t}[Q+C'F'RFC]e^{(A+BFC)t}dt X_0\right\}, \tag{2.4}$$

其中 $X_0=E[x_0x_0']$。

这时相应于（2.3）有

$$\hat{J}(F^*)\leq \hat{J}(F)。 \tag{2.5}$$

定理　（Levine-Athans）

（1）设A^*稳定。任何实常数矩阵F^*若满足（2.4）、（2.5），则一定满足下列积分方程

$$F^*=-R^{-1}B'K^*L^*C'[CL^*C']^{-1}, \tag{2.6}$$

其中

$$K^*=\int_0^\infty e^{A^{*\prime}t}[Q+C'F^{*\prime}RF^*C]e^{A^*t}dt, \tag{2.7}$$

$$L^* = \int_0^\infty e^{A^*\sigma} X_0 e^{A^{*\prime}\sigma} d\sigma, \tag{2.8}$$

$$A^* = A + BF^*C。 \tag{2.9}$$

（2）若 X_0 正定，并且 F^*、K^* 和 L^* 是（2.6）—（2.9）的解，那么 K^* 是一个半正定矩阵，满足方程

$$K^*A^* + A^{*\prime}K + Q + C'F^{*\prime}RF^*C = 0。 \tag{2.10}$$

L^* 是一个正定矩阵，满足方程

$$L^*A^{*\prime} + A^*L^* + X_0 = 0。 \tag{2.11}$$

由定理得到Levine－Athans 算法如下：

1）选择初始矩阵K_0；$n=1$。

2）迭代非线性矩阵方程组

$O = L_{n-1}[A + BF_{n-1}C]' + [A + BF_{n-1}C]L_{n-1} + X_0,$

$F_{n-1} = -R^{-1}B'K_{n-1}L_{n-1}C'[CL_{n-1}C']^{-1},$

得到L_{n-1}，F_{n-1}。

3）解方程

$$O = K_n[A + BF_{n-1}C] + [A + BF_{n-1}C]'K_n + C'F'_{n-1}RF_{n-1}C + Q,$$

得到K_n。

4）$n=n+1$，回到2）和3）。

对$n=1, 2, 3, \cdots$，得到$\{K_n\}$，$\{L_n\}$，$\{F_n\}$。

不难看出2）和3）的迭代计算过程是很复杂的。而且其收敛性并没有得到最后的证明。随着函数最优化技术的发展，使我们有可能进一步解决这个问题。

回到问题的原始提法，要对形如（2.4）的代价函数$\hat{J}(F)$求满足（2.5）的F^*，（不失一般性，不妨设 $X_0=I$）这实际上是一个典型的函数最优化问题。其一阶必要条件为

$$\left.\frac{\partial \hat{J}(F)}{\partial F}\right|_{F=F^*} = 0, \quad F \in \mathbf{F}。$$

因为$\hat{J}(F)$是二次型函数，而共轭斜量法是对二次型函数寻优特别有效的方法，Armijo 法则是二次型函数一维寻优的有效方法，所以下面我们将采用共轭斜量法和Armijo 法则给出一种新的算法。

梯度矩阵 $\partial\hat{J}(F)/\partial F$ 的计算是函数最优化方法中的关键步骤之一。因此，首先推导梯度矩阵。

三、梯度矩阵公式推导的一种新方法

定义 若$\hat{J}(F)$为Fre′chet 可微，则

$$\partial\hat{J}(F)/\partial F \triangleq (\partial\hat{J}(F)/\partial f_{ij}),$$

$$(i=1,\cdots,m,\ j=1,\cdots,r)$$

称为梯度矩阵。其中 $F=(f_{ij})$。

梯度矩阵公式可以用不同的方法得到。〔1〕中使用小扰动的方法。〔6〕中使用无穷维、动态最优化方法。这里我们用Fre′chet 微分和矩阵代数来推导$\partial\hat{J}(F)/\partial F$。

令 $$K(F)\triangleq\int_0^\infty e^{(A+BFC)'t}[Q+C'F'RFC]e^{(A+BFC)t}dt, \tag{3.1}$$

$$\hat{J}(F)\triangleq\frac{1}{2}\operatorname{trace}[K(F)], \tag{3.2}$$

$K(F)$满足方程

$$K(A+BFC)+(A+BFC)'K+Q+C'F'RFC=0, \tag{3.3}$$

对小扰动δF有

$$\hat{J}(F+\delta F)=\frac{1}{2}\operatorname{trace}[K(F)+\delta K]=\frac{1}{2}\operatorname{trace}[K(F)]+\frac{1}{2}\operatorname{trace}[\delta K]。$$

于是有

$$d\hat{J}[K(F),\delta K]=\frac{1}{2}\operatorname{trace}[\delta K]。$$

设 $K(F)$ Fte′chet 可微，则有

$$K(F+\delta F)\cong K(F)+\sum_{ij}\Gamma_{ij}(F)\delta f_{ij},$$

其中 $\Gamma_{ij}(F)=\Gamma_{ij}'(F)\in R^{n\times n}$，并满足方程

$$(A+BFC)'\Gamma_{ij}(F)+\Gamma_{ij}(F)(A+BFC)=-T, \tag{3.4}$$

这里

$$T=C'e_je_i'[B'K(F)+RFC]-[B'K(F)+RFC]'e_ie_j'C\ (e_i,\ e_j\text{为基础向量})。$$

所以，$$d\hat{J}[F,\ \delta F]=\frac{1}{2}\operatorname{trace}[\sum_{ij}\Gamma_{ij}(F)\delta f_{ij}],$$

由 Fre′chet 微分定义有

$$\partial\hat{J}(F)/\partial f_{ij}=\frac{1}{2}\operatorname{trace}[\Gamma_{ij}(F)]。 \tag{3.5}$$

对方程

$$(A+BFC)L+L(A+BFC)'=-I, \tag{3.6}$$

两端右乘Γ_{ij}，并取迹有

$$\operatorname{trace}[L(A+BFC)'\Gamma_{ij}]=-\frac{1}{2}\operatorname{trace}[\Gamma_{ij}]。 \tag{3.7}$$

对方程（3.4）两端左乘L，并取迹有

$$\operatorname{trace}[L(A+BFC)'\Gamma_{ij}]=-\frac{1}{2}\operatorname{trace}[LT]。 \tag{3.8}$$

比较（3.5）、（3.7）、（3.8）有

$$\partial \hat{J}(F)/\partial f_{ij}=\frac{1}{2}\operatorname{trace}[LT]。$$

根据T的表达式有

$$\partial \hat{J}(F)/\partial f_{ij}=e_i'[B'K+RFC]LC'e_j,$$

于是便得

$$\partial \hat{J}(F)/\partial F=[B'K+RFC]LC'。\tag{3.9}$$

（3.9）就是要求的$\partial \hat{J}(F)/\partial F$ 梯度矩阵公式。

其中 K、L 分别满足（3.3）和（3.6）。并有$K'=K$，K半正定，$L'=L$，L正定。

四、矩阵函数的共轭斜量法及Armijo法则

代价函数$\hat{J}(F)$的极小化问题，同一般函数最优化问题一样，可以化作一系列一维寻优问题来解决。即从某F_i出发，沿按某种规则所确定的寻优方向Z_i，求$\hat{J}(F)$在这个方向上的极小点F_{i+1}，也就是求W_i^*使

$\hat{J}(F_i+W_i^*Z_i)=\min\limits_{W_i}\hat{J}(F_i+W_iZ)$，$W_i\in(0,\infty)$，经过$N$步后，使$F_N$在一定精度要求下达到

$$\hat{J}(F_N)\simeq\min_F\hat{J}(F),\qquad F\in\mathbb{F}。$$

由于$\hat{J}(F)$是二次型函数，在我们的算法中将采用适用于二次型函数的共轭斜量法来定Z_i，及Armijo法则来定W_i^*。

通常使用的共轭斜量法是对向量函数的。即对向量函数$f(x)$，$x\in R^n$，其寻优方向P_i为

$$\begin{cases}P_0=-\dfrac{\partial f(x_0)}{\partial x},\\[2mm] P_i=-\dfrac{\partial f(x_i)}{\partial x}+\beta_{i-1}P_{i-1},\\[2mm] \beta_{i-1}=\left[\dfrac{\partial f(x_i)}{\partial x}-\dfrac{\partial f(x_{i-1})}{\partial x}\right]'\cdot\dfrac{\partial f(x_i)}{\partial x}\Big/\left[\dfrac{\partial f(x_{i-1})}{\partial x}\right]'\cdot\dfrac{\partial f(x_{i-1})}{\partial x},\end{cases}$$

其中 $\dfrac{\partial f(x_i)}{\partial x}=\dfrac{\partial f(x)}{\partial x}\Big|_{x=x_i}$，$i\geq1$。

而在我们的问题中，$\hat{J}(F)$是一个矩阵函数，所以，需要将共轭斜量法推广到矩阵

函数的情况。

不难得出，在矩阵函数情况下，共轭斜量法的寻优方向 Z_i 为

$$\begin{cases} Z_0 = -\dfrac{\partial \hat{J}(F_0)}{\partial F}, \\ Z_i = -\dfrac{\partial \hat{J}(F_i)}{\partial F} + \beta_{i-1} Z_{i-1}, \\ \beta_{i-1} = \sum\limits_{l,k} \left[\dfrac{\partial \hat{J}(F_i)}{\partial F} - \dfrac{\partial \hat{J}(F_{i-1})}{\partial F} \right]_{l,k} \cdot \left[\dfrac{\partial \hat{J}(F_i)}{\partial F} \right]_{l,k} \Big/ \\ \qquad \sum\limits_{l,k} \left[\dfrac{\partial \hat{J}(F_{i-1})}{\partial F} \right]^2_{l,k}, \end{cases}$$

其中
$$\frac{\partial \hat{J}(F_i)}{\partial F} \triangleq \left. \frac{\partial \hat{J}(F)}{\partial F} \right|_{F=F_i}, \quad Z_i \in R^{m\times r}。$$

如上所述，给定寻优方向 Z_i 后，需要在这个方向进行一维寻优。将Armijo法则推广到矩阵函数情况有

1）$\beta=1.2$，$\gamma=0.6$，$W_i=1$，Z_i 给定。

2）作
$$\hat{J}_z(F_i, W_i) = \hat{J}(F_i) + 0.5W_i \cdot \sum_{l,k} \left[\frac{\partial \hat{J}(F_i)}{\partial F} \right]_{l,k} \cdot [Z_i]_{l,k}。 \tag{4.2}$$

3）计算 $\hat{J}_z(F_i, W_i)$ 及 $\hat{J}(F_i+W_i \cdot Z_i)$，并比较。

4）若 $\hat{J}(F_i+W_i \cdot Z_i) - \hat{J}_z(F_i, W_i) < 0$，则 $W_i=W_i \cdot \beta$，回到3），否则 $W_i=W_i \cdot \gamma$。

5）计算 $\hat{J}_z(F_i, W_i)$ 及 $\hat{J}(F_i+W_i \cdot Z_i)$，并比较。

6）若 $\hat{J}(F_i+W_i \cdot Z_i) - \hat{J}_z(F_i, W_i) > 0$，则 $W_i=W_i \cdot \gamma$，回到5），否则停。这时 $W_i \cong W_i^*$。

五、符号函数法求解矩阵 Lyapunov 方程

为求梯度矩阵 $\partial \hat{J}(F)/\partial F$，需要解（3.3）和（3.6）。当取定 F_i 时，它们是矩阵Lyapunov方程。这种方程的解法有多种。在我们的算法中，当系统阶数不高时，可采用Kronecker 乘积法来解矩阵Lyapunov 方程；当系统阶数较高时，可采用符号函数法来解矩阵Lyapunov方程。

若系统阶数不高，或使用的计算机具有较大的内存时，可采用Kronecker 乘积将矩阵代数方程重新排列成线性代数方程组来求解[8]。这一方法的优点是可以直接调用求解线性代数方程组的标准子程序来求解，不须证明其收敛性问题。这一方法的缺点是计算量较大。

对于阶数较高的系统，可采用符号函数法[7]。

定义　设矩阵 H 不具有实部为零的特征根并有非奇异阵 M，使得

$$H = M \wedge M^{-1},$$

其中　$\wedge$ 为由 A 的特征根构成的若当标准型。则 $\mathrm{sign}\,(H) = M\mathrm{sign}(\wedge)M^{-1}$ 称为矩阵H的符号函数。

不难看出，

$$\mathrm{sign}(\wedge) = \begin{pmatrix} \mathrm{sign}(J_1) & 0 & \cdots & 0 \\ 0 & \mathrm{sign}(J_2) & \cdots & 0 \\ \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \mathrm{sign}(J_K) \end{pmatrix},$$

其中　J_i 是对应于 A 的特征根 λ_i 的若当块。并有

$$\mathrm{sign}(J_i) = \begin{pmatrix} \mathrm{sign}(\lambda_i) & 0 & \cdots & 0 \\ 0 & \mathrm{sign}(\lambda_i) & \cdots & 0 \\ \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \mathrm{sign}(\lambda_i) \end{pmatrix},$$

$$\mathrm{sign}(\lambda_i) = \begin{cases} +1 & \text{若 } \lambda_i \text{ 实部} > 0, \\ -1 & \text{若 } \lambda_i \text{ 实部} < 0。 \end{cases}$$

已经证明[7]，对下列方程

$$\widetilde{B} + \widetilde{A}P + P\widetilde{A}' = 0, \tag{5.3}$$

利用符号函数可得其解为

$$P = \frac{1}{2}S_{12}(N),$$

其中

$$\begin{cases} S_{12}(\overline{K}+1) = \dfrac{1}{2}[S_{12}(\overline{K}) + S_{11}^{-1}(\overline{K})S_{12}(\overline{K})S_{11}^{-1\prime}(\overline{K})], \\ S_{11}(\overline{K}+1) = \dfrac{1}{2}[S_{11}(\overline{K}) + S_{11}^{-1}(\overline{K})], \\ S_{22}(\overline{K}+1) = -S_{11}'(\overline{K}+1)。 \end{cases} \tag{5.4}$$

$S_{11}(\overline{K})$、$S_{12}(\overline{K})$、$S_{22}(\overline{K}) \in R^{n\times n}$，并按下列形式构成 $S(\overline{K}) \in R^{2n\times 2n}$

$$S(\overline{K}) = \begin{pmatrix} S_{11}(\overline{K}) & S_{12}(\overline{K}) \\ 0 & S_{22}(\overline{K}) \end{pmatrix}。$$

$S(\overline{K})$ 具有性质

$$S(\overline{K}+1) = \frac{1}{2}[S(\overline{K}) + S^{-1}(\overline{K})],\ (\overline{K} = 1, 2, \cdots N)。$$

当$\overline{K}=1$时，$S(1)$由矩阵H得到

$$S(1)=\frac{1}{2}(H+H^{-1})。$$

对方程（5.3），H具有如下形式

$$H=\begin{pmatrix}\widetilde{A} & \widetilde{B} \\ 0 & -\widetilde{A}'\end{pmatrix}。$$

因此，

$$S(1)=\frac{1}{2}\begin{bmatrix}\widetilde{A}+\widetilde{A}^{-1} & \widetilde{B}+\widetilde{A}^{-1}\widetilde{B}\widetilde{A}^{-1\prime} \\ 0 & -\widetilde{A}'-\widetilde{A}^{-1\prime}\end{bmatrix}。$$

对给定$\varepsilon>0$，当$\|S_{12}(\overline{K})-S_{12}(\overline{K}+1)\|<\varepsilon$时，$N=\overline{K}$，便得到我们要求的$S_{12}(N)$。

对方程（3.3）、（3.6），只须将$A+BFC$，$Q+C'F'RFC$及X_0代替方程（5.3）中相应的$\widetilde{A}$，$\widetilde{B}$，便可解出K及L。

由于$A+BFC$稳定，H不具有实部为零的特征根，于是保证了每次迭代中$S_{11}^{-1}(\overline{K})$的存在，也就保证了迭代的收敛及解的存在。计算表明，一般的迭代次数<10，精度可达10^{-5}。

六、算　　法

综上所述，我们可归纳出一种用函数最优化进行输出反馈最优化设计的算法如下：

根据的方程

$$\begin{cases}[A+BFC]'K+K[A+BFC]=-[Q+C'F'RFC], & (3.3)\\ [A+BFC]L+L[A+BFC]'=-I, & (3.6)\\ \hat{J}(F)=\frac{1}{2}\mathrm{trace}[K(F)], & (3.2)\\ S=\sum_{i,j}\left(\frac{\partial\hat{J}}{\partial F}\right)_{ij}^2, & (6.1)\\ \partial\hat{J}(F)/\partial F=[B'K+RFC]LC'。 & (3.9)\end{cases}$$

算法步骤

1）给定$\varepsilon>0$，$n=0$。选定$F_0\in R^{m\times r}$，$F_0\in \mathbf{F}$。

2）将F_n代入（3.3）、（3.6），求出K_n、L_n。

3）从（3.2）、（3.9）得出$\hat{J}(F_n)$，$\partial\hat{J}(F)/\partial F|_{F=F_n}$。

4）若$S<\varepsilon$，则停，$F_n\cong F^*$。否则，取寻优方向Z_i，如（4.1）所示。

5）在Z_i方向一维寻优，用Armijo法则选定W_n^*。

6）$n=n+1$，$F_{n+1}=F_n+W_n^*\cdot Z_n$。

7）检验$A+BF_{n+1}C$是否稳定，若是，回到2）。否则，缩小W_n^*，直到$A+BF_{n+1}C$稳定为止，再回到2）。

几点说明：

1.由于共轭斜量法是二阶收敛的，在一定的精度要求下，选取适当的初条件和步长，上述算法是收敛的和稳定的。因此，就收敛性来说，这一算法优于Levine－Athans算法。

2.用共轭斜量法求极值是近似的方法，其精度是人为选定的。通常取$\varepsilon=10^{-3}$或10^{-5}。当问题非病态时，这将大于计算过程中舍入或截断造成的误差。

3.整个过程中，只要求解矩阵Lyapunov方程和函数寻优，显然，比Levine－Athans算法的计算量大为减少。

实现这一算法的程序是用FortranⅣ写成的，最初是在CDC—6500上工作的。

七、M＝2.7超音速运输机侧向增稳控制系统的设计

一个具体的例子是飞机侧向增稳控制系统的设计。线性化以后的运动方程为[3]

$$\begin{pmatrix}\dot\beta\\ \dot\phi\\ \dot\rho_s\\ \dot R_s\end{pmatrix}=A\begin{pmatrix}\beta\\ \phi\\ \rho_s\\ R_s\end{pmatrix}+B\begin{bmatrix}\delta R\\ \delta A\end{bmatrix},$$

其中

$$\left.\begin{array}{ll}\beta: & \text{侧滑角}\\ \phi: & \text{欧拉倾斜角}\\ \rho_s: & \text{滚动速率}\\ R: & \text{偏航速率}\end{array}\right\}\text{状态变量，}$$

$$\left.\begin{array}{ll}\delta R: & \text{方向舵面转角}\\ \delta A: & \text{副翼面转角}\end{array}\right\}\text{控制变量。}$$

具体的参数为

$$A=\begin{pmatrix}-0.037 & 0.0123 & 0.00055 & -1\\ 0 & 0 & 1.0 & 0\\ -6.37 & 0 & -0.23 & 0.0618\\ 1.25 & 0 & 0.016 & -0.0457\end{pmatrix},$$

$$B=\begin{pmatrix}0.00084 & 0.000236\\ 0 & 0\\ 0.08 & 0.804\\ -0.0862 & -0.0665\end{pmatrix}.$$

由于侧滑角β在飞行过程中是很难测到的，所以这个问题不能实现全状态量反馈。于是对这一输出反馈问题有

$$C=\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}。$$

取　$Q=I_{4\times4}$，$R=I_{2\times2}$。

根据（六）中的算法，得到结果为

$$K^*=\begin{pmatrix} 67.2771 & -4.1231 & -5.57678 & -26.8279 \\ -4.12318 & 2.1277 & 1.53297 & 3.73094 \\ -5.57678 & 1.53297 & 3.6374 & 11.0674 \\ -26.8279 & 3.73094 & 11.0674 & 86.0271 \end{pmatrix},$$

$$F^*=\begin{pmatrix} 0.39624 & 1.58858 & 7.84149 \\ -1.25170 & -3.4681 & -4.95186 \end{pmatrix},$$

$J^*=79.53$。

计算精度为　$S=0.00089$。

计算时间为　130 秒。

寻优迭代次数为　80。

比较〔4〕中的结果，说明这一算法及程序可以用于实际问题的设计。

结论　采用函数最优化技术的输出反馈最优化设计的算法，较之 Levine－Athans 算法简化了计算，保证了收敛性，这是一个值得探讨的方向，将使最优输出反馈设计理论更加切实可行地应用到工程实际中去。

致谢　本工作是作者在英访问进修时工作的一部分，曾与Dr.J.Allwright 进行过有益的讨论，回国后补充整理时得到林士谔先生和王振钧先生的支持，在此一并感谢。

参 考 文 献

〔1〕 Levine, W.S.and Athans, M., On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems, IEEE Trans., AC－**15**, Feb.(1970).

〔2〕 Levine, W. S., Johson, T. L. and Athans, M., Optimal Limited State Variable Feedback Controls for Linear Systems, IEEE Trans., AC－**16**, Dec. (1971).

〔3〕 Markland, C. A., The Design of Optimal and Suboptimal Stability Augmentation Systems, AIAA J., **8**(1970), 673.

〔4〕 Choi, S. S. and Sirisena, H. R., Computation of Optimal Output Feedback Gains for Linear Multivariable Systems, IEEE Trans., AC－**19**, June(1974).

〔5〕 Horisberger, Hans. P. and Belanger, Pierre. R., Solution of the Optimal Constant Output Feedback Problem by Conjugate Gradients, IEEE Trans., AC-19, Aug. (1974).

〔6〕 Knapp, C. H. and Basuthakur, S., On Optimal Output Feedback, IEEE Trans., AC-17, Dec. (1972).

〔7〕 Denman, E. D., Beavers, A. N., The Matrix Sign Function and Computation in Systems, Applied Mathematics and Computation, 2 (1976), 63—94.

〔8〕 Barnett, Stephen., Matrix Methodsfor Engineering and Scientists, Mc Graw-Hill (1978).

AN ALGORITHM OF OUTPUT FEEDBACK OF LINEAR SYSTEMS

Mao Jianqin

(Beijing Institute of Aeronautics and Astronautics)

Abstract

In this paper an algorithm is developed, which is based on Levine-Athans work, by using functional optimization, This approach requires considerably less computation than for the algorithm of Levine-Athans and ensures the convergence of solution.

Derivation of the gradient matrix is important in this algorithm. A more simple and explicit derivation is presented here.

The conjugate gradient technique is here generalized and applied to the problem of matrix function.

A program is produced to make this algorithm viable on CDC-6500.

As a numerical example, a stability augmentation system of M-2.7 flight condition of a supersonic transport aircraft is computed.