第 3 章

基于状态空间模型的极点配置设计法

第 2 章介绍了设计计算机控制系统的常规的设计方法，它们都是基于传递函数模型。本章介绍利用状态空间模型的设计方法。基于状态空间模型设计控制系统的方法主要有两类：一类是最优设计方法，它包括最优控制和最优估计两个方面。它也就是通常所称的 LQG(Linear Quadratic Gaussian)设计问题。另一类是按极点配置的设计方法，它也包括按极点配置设计控制规律和按极点配置设计观测器两个方面。本章首先讨论按极点配置的设计方法。

3.1 连续控制对象模型的离散化

若忽略量化效应，计算机控制系统即为采样控制系统。如果将连续的控制对象连同它前面的零阶保持器一起进行离散化，那么采样控制系统即可简化为纯粹的离散系统来进行分析和设计，从而使问题进一步得到简化。本章即是按这个思想来讨论计算机控制系统按极点配置的设计方法。为此首先讨论连续控制对象模型离散化的问题。

3.1.1 不带延时的连续控制对象模型的离散化

设连续控制对象的模型可用如下的状态方程来描述：

$$\left.\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}\end{aligned}\right\} \tag{3-1}$$

其中设 $\boldsymbol{x}$ 为 n 维状态向量，$\boldsymbol{u}$ 为 m 维控制向量，$\boldsymbol{y}$ 为 r 维输出向量。同时设在连续的控制对象前面有一零阶保持器，即

$$\boldsymbol{u}(t) = \boldsymbol{u}(k) \quad kT \leqslant t < (k+1)T \tag{3-2}$$

其中 T 为采样周期。现在要求将连续控制对象模型连同零阶保持器一起进行离散化，从而使整个系统变为纯粹的离散系统。

式(3-1)的解的一般形式为

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\mathrm{d}\tau \tag{3-3}$$

若令 $t_0 = kT$，$t = (k+1)T$，同时考虑到零阶保持器的作用，则上式变为

$$\boldsymbol{x}(k+1) = e^{\boldsymbol{A}T}\boldsymbol{x}(k) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}(kT+T-\tau)}\mathrm{d}\tau\boldsymbol{B}\boldsymbol{u}(k) \tag{3-4}$$

若令 $t = kt + T - \tau$，则上式可进一步化为

$$\boldsymbol{x}(k+1) = \boldsymbol{F}\boldsymbol{x}(k) + \boldsymbol{G}\boldsymbol{u}(k) \tag{3-5}$$

其中

$$\boldsymbol{F} = e^{\boldsymbol{A}T}, \quad \boldsymbol{G} = \int_0^T e^{\boldsymbol{A}T}\mathrm{d}t\boldsymbol{B} \tag{3-6}$$

式(3-1)中的输出方程可以很容易离散化为

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) \tag{3-7}$$

式(3-5)和式(3-7)便是式(3-1)所示连续模型的等效离散状态方程。由于在连续的控制对象前面包含有零阶保持器,因而上面求得的离散等效关系是完全准确的。可见,离散化的关键在于式(3-6)所示的关于矩阵指数及其积分的计算。关于它们的计算在下一节再专门讨论。

3.1.2 包含延时的连续控制对象模型的离散化

在控制系统中,尤其是工业过程控制中,由于管道传输、热传导等均需要一定时间,因此控制对象中常常包含有延时。例如,若通过控制阀门来改变液体的流量,并进一步改变容器中混合液的浓度,由于从阀门到容器之间的管道传输需要一定时间,因此从给出控制(改变阀门开度)到控制发生作用(改变浓度)便产生了一定的延时。

设连续的控制对象模型为

$$\begin{cases} \boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t-\lambda) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \tag{3-8}$$

其中 $\boldsymbol{x}$ 是 n 维状态向量,$\boldsymbol{u}$ 是 m 维控制向量,$\boldsymbol{y}$ 是 r 维输出向量,λ 是控制作用的延迟时间,并设

$$\lambda = lT - \overline{m} \tag{3-9}$$

其中 l 是大于零的整数,$0 \leqslant \overline{m} < T$,$T$ 是采样周期。也即假设延迟时间不一定是采样周期 T 的整数倍。对于 $\overline{m}=0$(延迟时间是 T 的整数倍)的情况,其离散化计算将较为简单。

由于现在讨论的是计算机控制系统,因而在控制对象前面有一零阶保持器,如式(3-2)所示,即

$$\boldsymbol{u}(t) = \boldsymbol{u}(k), \quad kT \leqslant t < (k+1)T$$

式(3-8)的解可写为

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau-\lambda)\mathrm{d}\tau \tag{3-10}$$

上式中,令 $t_0 = kT$,$t = (k+1)T$,可得

$$\boldsymbol{x}(k+1) = e^{\boldsymbol{A}T}\boldsymbol{x}(k) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}(kT+T-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau-\lambda)\mathrm{d}\tau \tag{3-11}$$

作变量置换 $\eta = (k+1)T - \tau$,并代入式(3-9),上式变为

$$\begin{aligned} \boldsymbol{x}(k+1) &= e^{\boldsymbol{A}T}\boldsymbol{x}(k) + \int_0^T e^{\boldsymbol{A}\eta}\boldsymbol{B}\boldsymbol{u}(kT+T-\eta-\lambda)\mathrm{d}\eta \\ &= e^{\boldsymbol{A}T}\boldsymbol{x}(k) + \int_0^T e^{\boldsymbol{A}\eta}B\boldsymbol{u}(kT+T-lT+\overline{m}-\eta)\mathrm{d}\eta \end{aligned} \tag{3-12}$$

下面分别讨论 $\overline{m}=0$ 和 $\overline{m}\neq 0$ 的两种情况。

1. $\overline{m}=0$

当 $\overline{m}=0$ 时，同时考虑到控制对象前面有零阶保持器式(3-2)、式(3-12)可以化为

$$\begin{aligned}\boldsymbol{x}(k+1) &= e^{\boldsymbol{A}T}\boldsymbol{x}(k)+\int_0^T e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B}\boldsymbol{u}(k-l)\\ &= \boldsymbol{F}\boldsymbol{x}(k)+\boldsymbol{G}\boldsymbol{u}(k-l)\end{aligned} \tag{3-13}$$

其中 F 和 G 仍如式(3-6)所示。

式(3-13)便是式(3-8)的离散状态方程。但是为了便于进一步的分析和设计，式(3-1)需化为标准离散状态方程的形式，即等式左边为 $(k+1)$ 时刻的量，等式右边均为 k 时刻的量。为此可令

$$\left.\begin{aligned}x_{n+1}(k) &= u(k-l)\\ x_{n+2}(k) &= u(k-l+1)\\ &\vdots\\ x_{n+l}(k) &= u(k-1)\end{aligned}\right\} \tag{3-14}$$

并令增广状态为

$$\bar{\boldsymbol{x}}(k)=\begin{Bmatrix}\boldsymbol{x}(k)\\ x_{n+1}(k)\\ \vdots\\ x_{n+l}(k)\end{Bmatrix} \tag{3-15}$$

则式(3-13)变为

$$\bar{\boldsymbol{x}}(k+1)=\bar{\boldsymbol{F}}\bar{\boldsymbol{x}}(k)+\bar{\boldsymbol{G}}\boldsymbol{u}(k) \tag{3-16}$$

其中

$$\bar{\boldsymbol{F}}=\begin{bmatrix}F & G & 0 & & \cdots & 0\\ 0 & 0 & I & 0 & \cdots & 0\\ \vdots & \vdots & & \ddots & & \\ 0 & 0 & \cdots & 0 & & I\\ 0 & 0 & \cdots & & & 0\end{bmatrix},\quad \bar{\boldsymbol{G}}=\begin{bmatrix}0\\ \vdots\\ \vdots\\ 0\\ I\end{bmatrix} \tag{3-17}$$

式(3-8)中的输出方程很容易离散化为

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)=\bar{\boldsymbol{C}}\bar{\boldsymbol{x}}(k) \tag{3-18}$$

其中

$$\bar{\boldsymbol{C}}=[C\quad 0] \tag{3-19}$$

可见离散化计算的关键也在于计算如式(3-6)所示的 $\boldsymbol{F}$ 和 $\boldsymbol{G}$。

2. $\overline{m}\neq 0$

考虑到控制对象前面有零阶保持器，因而 $u(t)$ 呈阶梯形，如图 3.1 所示。式(3-12)的积分可以分两段进行，即

$$\begin{aligned}\boldsymbol{x}(k+1) &= e^{\boldsymbol{A}T}\boldsymbol{x}(k)+\int_0^{\overline{m}} e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B}\boldsymbol{u}(k-l+1)+\int_{\overline{m}}^{T} e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B}\boldsymbol{u}(k-l)\\ &= \boldsymbol{F}\boldsymbol{x}(k)+\boldsymbol{G}_a\boldsymbol{u}(k-l)+\boldsymbol{G}_b\boldsymbol{u}(k-l+1)\end{aligned} \tag{3-20}$$

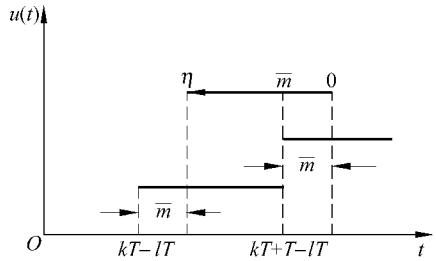

图 3.1 包含延时的 $u(t)$简图

其中

$$\boldsymbol{F}=e^{\boldsymbol{A}T},\quad \boldsymbol{G}_a=\int_{\overline{m}}^{T}e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B},\quad \boldsymbol{G}_b=\int_{0}^{\overline{m}}e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B} \tag{3-21}$$

若令

$$\boldsymbol{F}(t)=e^{\boldsymbol{A}T},\quad \boldsymbol{G}(t)=\int_{0}^{t}e^{\boldsymbol{A}\tau}\mathrm{d}\tau\boldsymbol{B} \tag{3-22}$$

则式(3-21)中各量可表示为

$$\left.\begin{aligned}\boldsymbol{F}&=e^{\boldsymbol{A}T}=\boldsymbol{F}(T)\\ \boldsymbol{G}_a&=\int_{\overline{m}}^{T}e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B}=\int_{0}^{T-\overline{m}}e^{\boldsymbol{A}(\overline{m}+\sigma)}\mathrm{d}\sigma\boldsymbol{B}\\ &=e^{\boldsymbol{A}\overline{m}}\int_{0}^{T-\overline{m}}e^{\boldsymbol{A}\sigma}\mathrm{d}\sigma\boldsymbol{B}=\boldsymbol{F}(\overline{m})\boldsymbol{G}(T-\overline{m})\\ \boldsymbol{G}_b&=\int_{0}^{\overline{m}}e^{\boldsymbol{A}\eta}\mathrm{d}\eta\boldsymbol{B}=\boldsymbol{G}(\overline{m})\end{aligned}\right\} \tag{3-23}$$

可见,式(3-20)中系数矩阵 $\boldsymbol{F}$,$\boldsymbol{G}_a$ 和 $\boldsymbol{G}_b$ 的计算最后也归结为计算矩阵指数及其积分。

(1) $l=1$

式(3-20)变为

$$\boldsymbol{x}(k+1)=\boldsymbol{F}\boldsymbol{x}(k)+\boldsymbol{G}_a\boldsymbol{u}(k-1)+\boldsymbol{G}_b\boldsymbol{u}(k) \tag{3-24}$$

令新的状态

$$\boldsymbol{x}_{n+1}(k)=\boldsymbol{u}(k-1) \tag{3-25}$$

合并以上两式得

$$\begin{bmatrix}\boldsymbol{x}(k+1)\\ \boldsymbol{x}_{n+1}(k+1)\end{bmatrix}=\begin{bmatrix}\boldsymbol{F}&\boldsymbol{G}_a\\ 0&0\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{x}_{n+1}(k)\end{bmatrix}+\begin{bmatrix}\boldsymbol{G}_b\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{u}(k) \tag{3-26}$$

上式中令

$$\bar{\boldsymbol{x}}(k)=\begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{x}_{n+1}(k)\end{bmatrix},\quad \bar{\boldsymbol{F}}=\begin{bmatrix}\boldsymbol{F}&\boldsymbol{G}_a\\ 0&0\end{bmatrix},\quad \bar{\boldsymbol{G}}=\begin{bmatrix}\boldsymbol{G}_b\\ \boldsymbol{I}\end{bmatrix} \tag{3-27}$$

同时令

$$\bar{\boldsymbol{C}}=[\boldsymbol{C}\quad 0] \tag{3-28}$$

则可得到标准的离散状态方程为

$$\left.\begin{aligned}\bar{\boldsymbol{x}}(k+1)&=\bar{\boldsymbol{F}}\,\bar{\boldsymbol{x}}(k)+\bar{\boldsymbol{G}}\,\boldsymbol{u}(k)\\ \boldsymbol{y}(k)&=\bar{\boldsymbol{C}}\,\bar{\boldsymbol{x}}(k)\end{aligned}\right\} \tag{3-29}$$

(2) $l>1$

这时令

$$\left.\begin{aligned} \boldsymbol{x}_{n+1}(k) &= \boldsymbol{u}(k-l) \\ \boldsymbol{x}_{n+2}(k) &= \boldsymbol{u}(k-l+1) \\ &\vdots \\ \boldsymbol{x}_{n+l}(k) &= \boldsymbol{u}(k-1) \end{aligned}\right\} \tag{3-30}$$

合并上式与式(3-24)可得增广的标准离散状态方程如式(3-29)所示。其中

$$\bar{\boldsymbol{x}}(k)=\begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{x}_{k+1}(k) \\ \vdots \\ \boldsymbol{x}_{k+1}(k) \end{bmatrix},\quad \bar{\boldsymbol{F}}=\begin{bmatrix} \boldsymbol{F} & \boldsymbol{G}_a & \boldsymbol{G}_b & 0 & \cdots & 0 \\ 0 & 0 & \boldsymbol{I} & 0 & \cdots & 0 \\ \vdots & \vdots & & & \ddots & \\ 0 & 0 & \cdots & & 0 & \boldsymbol{I} \\ 0 & 0 & \cdots & & & 0 \end{bmatrix} \tag{3-31}$$

$$\bar{\boldsymbol{G}}=\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{I} \end{bmatrix},\quad \bar{\boldsymbol{C}}=[\boldsymbol{C}\quad 0] \tag{3-32}$$

在以上讨论的问题中，当 $m=1$ 时表示只有一个控制量及一个延时，当 $m>1$ 时，表示有 m 个控制量及 m 个延时。以上假定 m 个延时都相同(见式(3-8)，延时均为 λ)，在实际系统中，常常对于不同的控制量其延时也是不同的，或者有的控制量有延时，有的没有延时。因此式(3-8)写成一般形式应为

$$\left.\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}_1 u_1(t-\lambda_1)+\cdots+\boldsymbol{B}_q u_q(t-\lambda_q) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) \end{aligned}\right\} \tag{3-33}$$

其中 $\boldsymbol{x}$ 是 n 维向量，u_i 是 m_i 维向量 $i=1,\cdots,q$，y 是 r 维输出向量。$\lambda_i(i=1,\cdots,q)$是对于控制作用 u_i 的延迟时间，并设 $\lambda_i=l_iT-\bar{m}_i$，$i=1,\cdots,q$，其中 l_i 是大于零的整数，$0\leqslant\bar{m}_i<T$。

仿照与前面类似的推导，不难求得式(3-33)的离散状态方程为

$$\left.\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{F}\boldsymbol{x}(k)+\sum_{i=1}^{q}\boldsymbol{G}_{ai}u_i(k-l_i)+\sum_{i=1}^{q}\boldsymbol{G}_{bi}u_i(k-l_i+1) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) \end{aligned}\right\} \tag{3-34}$$

其中

$$\boldsymbol{F}=\boldsymbol{F}(T),\quad \boldsymbol{G}_{ai}=\boldsymbol{F}(\bar{m}_i)\boldsymbol{G}(T-\bar{m}_i),\quad \boldsymbol{G}_{bi}=\boldsymbol{G}(\bar{m}_i),\quad i=1,\cdots,q \tag{3-35}$$

这里 $\boldsymbol{F}(t)$，$\boldsymbol{G}(t)$的含义仍同式(3-22)。仿照前面的方法，利用增广矩阵可进一步将式(3-34)按照具体情况化为标准的离散状态方程。

从以上离散化过程看出，对于控制对象有延时的情况，最终均化成了标准的离散状态方程的形式，它与没有延时的离散状态的模型在形式上完全一样。因此离散系统的各种分析和设计方法可以用来统一处理包括带延时的情况。与连续系统相比，这是离散系统分析的一个优点。

以上各种情况的离散化，最后均归结为求式(3-6)所示的关于矩阵指数及其积分的计算，下面将专门讨论它们的计算问题。

3.2 矩阵指数及其积分的计算

本节专门讨论如式(3-6)所示的矩阵指数及其积分的计算。为便于参考,这里重写式(3-6)如下

$$\boldsymbol{F} = e^{\boldsymbol{A}T} \tag{3-36}$$

$$\boldsymbol{G} = \int_0^T e^{\boldsymbol{A}t}\,\mathrm{d}t\boldsymbol{B} \tag{3-37}$$

计算矩阵指数及其积分的方法很多,下面介绍几种典型的方法,并着重介绍其中最为实用的直接级数求和法。

3.2.1 拉普拉斯反变换法

利用拉普拉斯反变换可以求得

$$e^{\boldsymbol{A}t} = \mathscr{L}^{-1}(s\boldsymbol{I} - \boldsymbol{A})^{-1} \tag{3-38}$$

若求得 $e^{\boldsymbol{A}t}$ 的表达式,则简单地令 $t = T$ 即可求得式(3-36)中的 $\boldsymbol{F}$,同时按照式(3-37)的积分式可进一步求得 $\boldsymbol{G}$。

这里关键的计算有两个:一是计算 $(s\boldsymbol{I}-\boldsymbol{A})^{-1}$;二是求 $(s\boldsymbol{I}-\boldsymbol{A})^{-1}$ 的反变换。下面首先介绍 $(s\boldsymbol{I}-\boldsymbol{A})^{-1}$ 的算法。

定理 3.1

$$(s\boldsymbol{I} - \boldsymbol{A})^{-1} = \frac{\mathrm{adj}(s\boldsymbol{I} - \boldsymbol{A})}{|\,s\boldsymbol{I} - \boldsymbol{A}\,|} = \frac{\boldsymbol{B}(s)}{\phi(s)} \tag{3-39}$$

其中

$$\phi(s) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n \tag{3-40}$$

$$\boldsymbol{B}(s) = \boldsymbol{B}_1 s^{n-1} + \cdots + \boldsymbol{B}_{n-1}s + \boldsymbol{B}_n \tag{3-41}$$

α_k 和 $\boldsymbol{B}_k$ 可由下列迭代公式算出

$$\left.\begin{aligned} &\boldsymbol{B}_1 = \boldsymbol{I} \\ &a_k = -\frac{1}{k}\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}_k) \quad k = 1,2,\cdots,n \\ &\boldsymbol{B}_k = \boldsymbol{A}\boldsymbol{B}_{k-1} + \alpha_{k-1}\boldsymbol{I} \quad k = 2,3,\cdots,n \end{aligned}\right\} \tag{3-42}$$

证明 式(3-39)两边同乘$(s\boldsymbol{I}-\boldsymbol{A})$得

$$(s\boldsymbol{I} - \boldsymbol{A})\boldsymbol{B}(s) = \phi(s)\boldsymbol{I} \tag{3-43}$$

将式(3-40)和式(3-41)代入式(3-43)并比较方程两边的系数便可得式(3-42)中计算 $\boldsymbol{B}_k$ 的递推公式。

下面推导 α_k 的递推公式。设 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 是 $\phi(s)$ 的 n 个根(即矩阵 $\boldsymbol{A}$ 的 n 个特征值),现令

$$\lambda_1^k + \lambda_2^k + \cdots + \lambda_n^k \triangleq S_k \tag{3-44}$$

根据根与系数关系的牛顿公式,有

$$-k\alpha_k = S_k + \alpha_1 S_{k-1} + \cdots + \alpha_{k-1}S_1 \tag{3-45}$$

同时,设 $\boldsymbol{A}$ 的若当标准形为 $\boldsymbol{J}$,变换矩阵为 $\boldsymbol{P}$,即 $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{-1}$,则 $\boldsymbol{A}^k = \boldsymbol{P}\boldsymbol{J}^k\boldsymbol{P}^{-1}$。同时根据 $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A})$ 的性质,则有

$$\begin{aligned}\mathrm{tr}\boldsymbol{A}^k &= \mathrm{tr}(\boldsymbol{P}\boldsymbol{J}^k\boldsymbol{P}^{-1}) = \mathrm{tr}(\boldsymbol{J}^k\boldsymbol{P}^{-1}\boldsymbol{P}) \\ &= \mathrm{tr}\boldsymbol{J}^k = S_k, \quad k=1,2,\cdots,n\end{aligned} \tag{3-46}$$

因此，由式(3-44)、式(3-45)及式(3-46)得

$$\begin{aligned}\alpha_k &= -\frac{1}{k}(\mathrm{tr}\boldsymbol{A}^k + \alpha_1\mathrm{tr}\boldsymbol{A}^{k-1} + \cdots + \alpha_{k-1}\mathrm{tr}\boldsymbol{A}) \\ &= -\frac{1}{k}\mathrm{tr}(\boldsymbol{A}^k + \alpha_1\boldsymbol{A}^{k-1} + \cdots + \alpha_{k-1}\boldsymbol{A}), \quad k=1,2,\cdots,n\end{aligned} \tag{3-47}$$

根据式(3-42)有

$$\begin{aligned}\boldsymbol{B}_{k+1} &= \boldsymbol{A}\boldsymbol{B}_k + \alpha_k\boldsymbol{I} = \boldsymbol{A}(\boldsymbol{A}\boldsymbol{B}_{k-1} + \alpha_{k-1}\boldsymbol{I}) = \cdots \\ &= \boldsymbol{A}^k + \alpha_1\boldsymbol{A}^{k-1} + \cdots + \alpha_{k-1}\boldsymbol{A} + \alpha_k\boldsymbol{I}\end{aligned} \tag{3-48}$$

即

$$\boldsymbol{A}\boldsymbol{B}_k = \boldsymbol{A}^k + \alpha_1\boldsymbol{A}^{k-1} + \cdots + \alpha_{k-1}\boldsymbol{A} \tag{3-49}$$

将式(3-49)代入式(3-47)即得式(3-42)中计算 α_k 的递推公式，从而定理得证。

上述求 $\boldsymbol{B}_k$ 和 α_k 的方法通常称为 Leverrier-Faddeeva 算法。

计算 $e^{\boldsymbol{A}t}$ 的第二个关键问题是计算 $(s\boldsymbol{I}-\boldsymbol{A})^{-1}$ 的拉普拉斯反变换。根据式(3-39)～式(3-41)，$e^{\boldsymbol{A}t}$ 可以表示为：

$$e^{\boldsymbol{A}t} = \sum_{k=1}^{n}\mathscr{L}^{-1}\left[\frac{s^{n-k}}{\phi(s)}\right]\boldsymbol{B}_k \tag{3-50}$$

其中 $\mathscr{L}^{-1}[s^{n-k}/\phi(s)]$ 可以通过先对方括号内的有理分式进行部分分式展开，然后再求出各部分分式的反变换。

这个方法中所包含的两个关键问题的计算均较复杂，且根据式(3-37)计算 G 还需要作进一步的积分计算。然而在有些情况下需要计算定常系统转移矩阵的解析解时，它是一种比较好的方法。同时这种方法也比较适合于手工计算。

3.2.2　凯莱-哈密尔顿法

根据凯莱-哈密尔顿(Caley-Hamilton)定理，设 $\phi(s)$ 为如式(3-40)所示的 $\boldsymbol{A}$ 的特征多项式，则 $\boldsymbol{A}$ 阵满足 $\phi(\boldsymbol{A})=0$，即：

$$\boldsymbol{A}^n + \alpha_1\boldsymbol{A}^{n-1} + \cdots + \alpha_{n-1}\boldsymbol{A} + \alpha_n\boldsymbol{I} = 0 \tag{3-51}$$

式(3-51)表示，$\boldsymbol{A}^n$ 可由 $A^i(i<n)$ 的线性组合来表示，反复使用式(3-51)，则 $\boldsymbol{A}$ 的更高次幂 $\boldsymbol{A}^{n+m-1}$ 表达式的系数 $\boldsymbol{A}^{n+m}(m\geqslant 0)$ 也可用 $\boldsymbol{A}^i(i<n)$ 的线性组合来表示，即可写成

$$\boldsymbol{A}^{n+m} = \alpha_{1m}\boldsymbol{A}^{n-1} + \cdots + \alpha_{n-1,m}\boldsymbol{A} + \alpha_{nm}\boldsymbol{I} \tag{3-52}$$

假定 $\boldsymbol{A}^{n+m-1}$ 表达式的系数 $\alpha_{1,m-1},\cdots,\alpha_{n-1,m-1},\alpha_{n,m-1}$ 为已知，则根据 $\boldsymbol{A}^{n+m}=\boldsymbol{A}^{n+n-1}\boldsymbol{A}$，并在式中代入式(3-51)和(3-52)，通过比较等式两边 $\boldsymbol{A}$ 的同次幂的系数，则可得到如下的递推计算公式

$$\left.\begin{aligned}\alpha_{km} &= \alpha_{k+1,m-1} - \alpha_k\alpha_{1,m-1}, \quad k=1,2,\cdots,n-1 \\ \alpha_{nm} &= -\alpha_n\alpha_{1,m-1}\end{aligned}\right\} \tag{3-53}$$

根据式(3-51)和式(3-52)可以求得当 $m=0$ 时的系数初值为：

$$a_{k0} = -a_k, \quad k=1,2,\cdots,n \tag{3-54}$$

根据式(3-52)和式(3-54)即可以算得 m 依次递增时式(3-52)中的各个系数。进一步利用

e^{AT}的幂级数展开式,得

$$\begin{aligned} e^{AT} &= \sum_{k=0}^{\infty}\frac{1}{k!}(AT)^k = \sum_{k=0}^{n-1}\frac{T^k}{k!}A^k + \sum_{m=0}^{\infty}\frac{T^{n+m}}{(n+m)!}A^{n+m} \\ &= \sum_{k=0}^{n-1}\frac{T^k}{k!}A^k + \sum_{m=0}^{\infty}\frac{T^{n+m}}{(n+m)!}\sum_{k=0}^{n-1}a_{n-k,m}A^k \\ &= \sum_{k=0}^{n-1}A^k\left[\frac{T^k}{k!} + \sum_{m=0}^{\infty}\frac{a_{n-k,m}T^{n+m}}{(n+m)!}\right] \end{aligned} \tag{3-55}$$

实际计算时并不需要计算无穷多项。若根据精度要求,在e^{AT}的幂级数表达式取到第$(N+1)$项,相当在式(3-55)中的方括号内的第二项取到$m=N-n$。使用式(3-55)计算e^{AT}的步骤如下:

(1) 计算$A^2, A^3, \cdots, A^n$;

(2) 按式(3-47)计算A的特征多项式$\phi(s)$的系数$\alpha_k(k=1,2,\cdots,n)$;

(3) 根据给定的精度要求确定级数求和的项数$(N+1)$(后面将专门讨论确定的方法),并计算$m=N-n$;

(4) 根据式(3-53)~式(3-55)计算出e^{AT}。

以上讨论了计算$F=e^{AT}$的方法,同时还必须求出如式(3-37)所示的G。G也可以表示成如下所示的极数求和的形式:

$$\begin{aligned} G &= \int_0^T e^{At}\mathrm{d}t = \sum_{k=0}^{\infty}\int_0^T \frac{1}{k!}A^k t^k \mathrm{d}t \\ &= \sum_{k=0}^{\infty}\frac{1}{(k+1)!}A^k T^{k+1} \end{aligned} \tag{3-56}$$

仿照与前面类似的推导,不难求得

$$G = \left\{\sum_{k=0}^{n-1}A^k\left[\frac{T^{k+1}}{(k+1)!} + \sum_{m=0}^{\infty}\frac{a_{n-k,m}T^{n+m+1}}{(n+m+1)!}\right]\right\}B \tag{3-57}$$

以上介绍的算法只需要一些最基本的运算如矩阵的相乘等,这是该方法的优点。然而当用式(3-47)及式(3-53)计算各系数时,这些递推公式对数据误差比较灵敏,同时该算法需存贮$A, A^2, \cdots, A^{n-1}$各矩阵,因而占用较多的内存,这是该算法的缺点。

若已经求得A的特征值为$\lambda_1, \lambda_2, \cdots, \lambda_n$,且它们是两两相异时,则$e^{AT}$可由如下的公式算出

$$e^{At} = \beta_0(t)I + \beta_1(t)A + \cdots + \beta_{n-1}(t)A^{n-1} \tag{3-58}$$

其中

$$\begin{bmatrix}\beta_0(t)\\ \beta_1(t)\\ \vdots\\ \beta_{n-1}(t)\end{bmatrix} = \begin{bmatrix}1 & \lambda_1 & \lambda_1^2 & \cdots\lambda_1^{n-1}\\ 1 & \lambda_2 & \lambda_2^2 & \cdots\lambda_2^{n-1}\\ \cdots & & & \\ 1 & \lambda_n & \lambda_n^2 & \cdots\lambda_n^{n-1}\end{bmatrix}\begin{bmatrix}e^{\lambda_1 t}\\ e^{\lambda_2 t}\\ \vdots\\ e^{\lambda_n t}\end{bmatrix} \tag{3-59}$$

当A具有重特征值时,公式将较复杂。利用该法来计算e^{AT}需要计算A的特征值,而且特征值一般为复数,因而包含有复数的运算。另外,当A有重特征值时,计算将更为复杂。因此,只有当矩阵的特征值由于别的问题已经求出时,采用该算法才是有利的。

3.2.3　特征值和特征向量法

设矩阵 $\boldsymbol{A}$ 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，相应的特征向量为 $\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_n$。如果 $\boldsymbol{A}$ 的特征值两两相异。则 $\boldsymbol{A}$ 可经相似变换变为对角标准形，即

$$\boldsymbol{A}=\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1} \tag{3-60}$$

其中

$$\left.\begin{aligned}\boldsymbol{P}&=[\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_n]\\ \boldsymbol{\Lambda}&=\mathrm{diag}[\lambda_1\lambda_2\cdots\lambda_n]\end{aligned}\right\} \tag{3-61}$$

式中 $\boldsymbol{P}_i$ 是相应于特征值 $\lambda_i(i=1,2,\cdots,n)$ 的特征向量。不难推得

$$\begin{aligned}e^{\boldsymbol{A}t}&=e^{\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}t}=\boldsymbol{P}e^{\boldsymbol{\Lambda}t}\boldsymbol{P}^{-1}\\&=\boldsymbol{P}\begin{bmatrix}e^{\lambda_1 t}&&&\\&e^{\lambda_2 t}&&\boldsymbol{0}\\&&\ddots&\\\boldsymbol{0}&&&e^{\lambda_n t}\end{bmatrix}\boldsymbol{P}^{-1}\end{aligned} \tag{3-62}$$

从而矩阵指数的计算化为了标量指数的计算。当矩阵 $\boldsymbol{A}$ 不能化为对角标准形时，其结果要复杂些。当由于别的问题需要计算矩阵的特征值和特征向量时，利用该方法进行计算才是比较合算的。

3.2.4　直接级数求和法

根据矩阵指数的定义，有

$$\boldsymbol{F}=e^{\boldsymbol{A}T}=\boldsymbol{I}+\boldsymbol{A}T+\frac{\boldsymbol{A}^2T^2}{2!}+\cdots=\sum_{k=0}^{\infty}\frac{\boldsymbol{A}^kT^k}{k!} \tag{3-63}$$

同时不难求得

$$\boldsymbol{G}=\int_0^T e^{\boldsymbol{A}t}\mathrm{d}t\boldsymbol{B}=\sum_{k=0}^{\infty}\frac{\boldsymbol{A}^kT^{k+1}}{(k+1)!}\boldsymbol{B}=\sum_{k=1}^{\infty}\frac{\boldsymbol{A}^{k-1}T^k}{k!}\boldsymbol{B} \tag{3-64}$$

利用式(3-63)及式(3-64)计算矩阵指数及其积分，其计算方法及程序实现均较简单，这是目前应用最广泛的一种实用算法。下面将较详细地对它进行讨论。

1. $e^{\boldsymbol{A}T}$ 的收敛性

容易证明，利用式(3-63)和式(3-64)的无穷级数求和来计算 $\boldsymbol{F}$ 和 $\boldsymbol{G}$ 在任何情况下都是收敛的。取式(3-63)的范数得：

$$\|\boldsymbol{F}\|=\|e^{\boldsymbol{A}T}\|=\left\|\sum_{k=0}^{\infty}\frac{(\boldsymbol{A}T)^k}{k!}\right\|\leqslant\sum_{k=0}^{\infty}\frac{\|\boldsymbol{A}T\|^k}{k!}e^{\|\boldsymbol{A}T\|} \tag{3-65}$$

由于 $e^{\|\boldsymbol{A}T\|}$ 是标量指数函数，只要 $\|\boldsymbol{A}T\|$ 有界，$e^{\|\boldsymbol{A}T\|}$ 也一定有界，从而根据式(3-65)，式(3-63)的无穷级数一定收敛。同样可以证明，式(3-64)的无穷级数也一定收敛。从而说明，对于式(3-63)和式(3-64)的无穷级数，从理论上讲，只要计算足够多的项数，就一定能够求得 $\boldsymbol{F}$ 和 $\boldsymbol{G}$ 的准确解。

2. F 和 G 的计算

根据式(3-63)和式(3-64)计算 $\boldsymbol{F}$ 和 $\boldsymbol{G}$,需要计算两个无穷级数求和。为了减少计算工作量,经过适当变换,只需计算其中一个无穷级数。例如,若 $\boldsymbol{A}$ 可逆,式(3-64)的积分可求得为

$$\boldsymbol{G}=\boldsymbol{A}^{-1}(e^{\boldsymbol{A}T}-\boldsymbol{I})\boldsymbol{B}=\boldsymbol{A}^{-1}(\boldsymbol{F}-\boldsymbol{I})\boldsymbol{B} \tag{3-66}$$

从而只需按式(3-63)计算一个级数即可。然而,对于 $\boldsymbol{A}$ 是奇异阵的情况(如对象中包含有积分环节,则 $\boldsymbol{A}$ 中包含零特征值,从而 $\boldsymbol{A}$ 为奇异阵),则不能用式(3-66)来进行计算。因此,先计算 $\boldsymbol{F}$ 再由式(3-66)来计算 $\boldsymbol{G}$ 不是一个好的方法。如果令

$$\boldsymbol{G}_1=\int_0^T e^{\boldsymbol{A}t}\mathrm{d}t=\sum_{k=1}^{\infty}\frac{\boldsymbol{A}^{k-1}T^k}{k!} \tag{3-67}$$

则

$$\boldsymbol{G}=\boldsymbol{G}_1\boldsymbol{B} \tag{3-68}$$

根据式(3-63)有

$$\boldsymbol{F}=\sum_{k=0}^{\infty}\frac{\boldsymbol{A}^kT^k}{k!}=I+\sum_{k=1}^{\infty}\frac{\boldsymbol{A}^{k-1}T^k}{k!}\boldsymbol{A}=\boldsymbol{I}+\boldsymbol{G}_1\boldsymbol{A} \tag{3-69}$$

可见,只要按式(3-67)计算一个无穷级数 $\boldsymbol{G}_1$ 即可求得 $\boldsymbol{F}$ 和 $\boldsymbol{G}$。为便于计算机计算,通常将式(3-67)的级数求和写成递推求和的公式。归纳以上的讨论,最后得到计算 $\boldsymbol{F}$ 和 $\boldsymbol{G}$ 的实用算法如下

$$\left.\begin{aligned}&\boldsymbol{G}_1=\sum_{k=1}^{\infty}\boldsymbol{G}_1(k)\\&\boldsymbol{G}_1(k)=\frac{\boldsymbol{A}T}{k}\boldsymbol{G}_1(k-1)\\&\boldsymbol{G}_1(1)=T\boldsymbol{I}\\&\boldsymbol{G}=\boldsymbol{G}_1\boldsymbol{B}\\&\boldsymbol{F}=\boldsymbol{I}+\boldsymbol{G}_1\boldsymbol{A}\end{aligned}\right\} \tag{3-70}$$

3. 确定级数求和项数

下面以式(3-63)计算 $e^{\boldsymbol{A}T}$ 的展开式为例来说明截断误差上限的估计及根据精度要求来确定级数的求和项数。设计算 $e^{\boldsymbol{A}T}$ 到 $N+1$ 项,即

$$e^{\boldsymbol{A}T}=\sum_{k=0}^{N}\frac{\boldsymbol{A}^kT^k}{k!}+\sum_{k=N+1}^{\infty}\frac{\boldsymbol{A}^kT^k}{k!}=\hat{e}^{\boldsymbol{A}T}+\boldsymbol{R} \tag{3-71}$$

其中

$$\boldsymbol{R}=\sum_{k=N+1}^{\infty}\frac{\boldsymbol{A}^kT^k}{k!} \tag{3-72}$$

便是计算的截断误差。式(3-72)两边取范数得

$$\|\boldsymbol{R}\|=\left\|\sum_{k=N+1}^{\infty}\frac{\|\boldsymbol{A}T\|^k}{k!}\right\|\leqslant\sum_{k=N+1}^{\infty}\frac{\|\boldsymbol{A}T\|^k}{k!} \tag{3-73}$$

若令