系统工程与电子技术
1990年第5期

广义系统的时域解析

西安工业学院　窦惠芳　陈阳泉

摘要　本文提出了适合于计算机求解的广义系统时域解析方法。主要解决了 $L^{-1}\left\{(E_s-A)^{-1}\right\}$ 当 E、A 都奇异时一种不需人机交互的计算方法。

主题词　+广义系统，时域，计算方法，拉普拉斯变换。

一、引　言

广义系统

$$E\dot{X}=AX+BU \tag{1}$$

$$Y=CX$$

是在电网络、电力及社会经济等系统中经常遇到的一类描述。其中 E、$A\in R^{n\times n}$，$X\in R^n$，$U\in R^m$，$B\in R^{m\times n}$，$C\in R^{l\times n}$ 且 E 为奇异。式(1)的零输人响应为

$$Z_0(t)=L^{-1}\left\{(E_s-A)^{-1}\right\}X(t_0)$$

其中 $X(t_0)$为状态初值。若记

$$Z(t)=L^{-1}\left\{(E_s-A)^{-1}\right\} \tag{2}$$

则$X(t)=Z(t)\cdot X(t_0)+\int_{t_0}^{t} Z(t-\tau)\cdot BU(\tau)d\tau$

可见，广义系统（1）的时域响应求取的关键在于 $Z(t)$的求取。本文主要提出求取 $Z(t)$ 解析表达式的计算方法。为便于讨论，记

$$(E_s-A)^{-1}=(T(s))^{-1}=\sum_{k=0}^{n-1}P_k s^k \Big/ \sum_{k=0}^{n}a_k s^k \tag{3}$$

其中 $P_k\in R^{n\times n}$。

二、a_k $(k=0,1,\cdots,n)$的求取

1. 解线性方程组法

由 $\det(E_s-A)=\sum_{k=0}^{n}a_k s^k$

取 $n+1$ 个互不相同的数 s_j $(j=0,1,\cdots,n)$，显然有关于 a_k 的线性方程组

$$\sum_{k=0}^{n}a_k s_j^k=\det(Esj-A)\quad (j=0,1,2,\cdots,n) \tag{4}$$

记　$d_j=\det(Es_j-A)$

显然取 $s_0=0$，则 $a_0=d_0=\det(-A)$，式(4)写作

收稿日期：1989年12月20日

$$\begin{bmatrix} 1 & s_1 & s_1^2 & \cdots & s_1^{n-1} \\ 1 & s_2 & s_2^2 & \cdots & s_2^{n-1} \\ \vdots & \vdots & \vdots & & \\ 1 & s_n & s_n^2 & \cdots & s_n^{n-1} \end{bmatrix}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (d_1-d_0)/s_1 \\ (d_2-d_0)/s_2 \\ \vdots \\ (d_n-d_0)/s_n \end{bmatrix}$$

根据 Vondermonde 矩阵的性质，知 a_k 可唯一解出。

2. DFT 法

适当选取 s_j 可以使 Vondermode 阵求逆更为方便。现取

$$s_j = \exp\left(\frac{2\pi j}{n+1}\right)i$$

$$(j=0,1,2,\cdots,n),\ i=\sqrt{-1}$$

相当于在单位圆上均匀取 $n+1$ 个点。记

$$\omega = \exp\left(\frac{2\pi i}{n+1}\right)$$

则式(4)可写作

$$W(\omega)[a_0,a_1,\cdots,a_n]^T = [d_0,d_1,\cdots,d_n]^T$$

d_k 的定义同前，且

$$W(\omega) = \{\omega^{(i-1)(j-1)}\}_{i,j}$$

$$(i,j=1,2,\cdots,n+1)$$

由离散付氏变换中的算法易知

$$W^{-1}(\omega) = \frac{1}{n+1}W(\omega^{-1}) \tag{5}$$

所以，

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \frac{1}{n+1}\ W(\omega^{-1}) \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{6}$$

可见，如果计算机程序设计语言具有复数型数据，这种方法就更为实用简便。

3. 广义 Faddeeva 递推算法

我们知道，当 E 为单位阵时，a_k，P_k 可由 Faddeeva 递推算法得到[1]:

$$\begin{cases} P_{n-1} = I_n,\ a_n = 1 \\ a_{n-k} = -\dfrac{1}{k}\mathrm{tr}(AP_{n-k}),\ k=1,2,\cdots,n \\ P_{n-k} = AP_{n-k+1} + a_{n-k+1}I_n,\ k=2,3,\cdots,n \end{cases} \tag{7}$$

当 E 奇异，由定理 1 可得到广义 Faddeeva 递推算法。

引理 1　设 $n\times n$ 矩阵 $A(t)$的元 $a_{ij}(t)$的代数余子式为 $A_{ij}(t)$，则有

$$\frac{d}{dt}\begin{vmatrix} a_{11}(t) & \cdots & a_{1n}(t) \\ \vdots & & \vdots \\ a_{n1}(t) & \cdots & a_{nn}(t) \end{vmatrix}$$

$$= \sum_{j=1}^{n}\begin{vmatrix} a_{11}(t) & \cdots & \frac{d}{dt}a_{1j}(t) & \cdots & a_{1n}(t) \\ \vdots & & \vdots & & \vdots \\ a_{n1}(t) & \cdots & \frac{d}{dt}a_{nj}(t) & \cdots & a_{nn}(t) \end{vmatrix}$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{n}\left[\frac{d}{dt}a_{ij}(t)\right]A_{ij}(t) \tag{8}$$

引理 1 的证明是显然的。

定理 1　E、A 为任意 $n\times n$ 常数矩阵，s 为变量，则

$$\frac{d}{ds}[\det(E_s - A)] = \mathrm{tr}[E.\mathrm{adj}(E_s - A)] \tag{9}$$

证明　记 $E=[e_{ij}]_{ij}$，$A=[a_{ij}]_{ij}$，$\mathrm{adj}(E_s-A)=[P_{ij}(s)]_{ij}^T$，由引理 1

$$\frac{d}{ds}[\det(E_s - A)] = \sum_{i=1}^{n}\sum_{j=1}^{n}e_{ij}P_{ij}(s)$$

而 $\operatorname{tr}\left[E\cdot\operatorname{adj}(E_s-A)\right]$

$$=\operatorname{tr}\left\{\begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & & \\ e_{n1} & \cdots & e_{nn}\end{bmatrix}\cdot\begin{bmatrix} P_{11}(s) & \cdots & P_{n1}(s) \\ \vdots & & \vdots \\ P_{1n}(s) & \cdots & P_{nn}(s)\end{bmatrix}\right\}$$

$$=\operatorname{tr}\begin{bmatrix} \sum_{j=1}^{n} e_{1j}P_{ij}(s) & \cdots & \sum_{j=1}^{n} e_{1j}P_{nj}(s) \\ \vdots & & \vdots \\ \sum_{j=1}^{n} e_{nj}P_{1j}(s) & \cdots & \sum_{j=1}^{n} e_{nj}P_{nj}(s)\end{bmatrix}$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n} e_{ij}P_{ij}(s)$$

定理证得。

由于 $\det(E_s-A)=\sum_{k=0}^{n} a_k s^k$

$$\operatorname{adj}(E_s-A)=\sum_{k=0}^{n-1} p_k s^k$$

所以由定理 1 有

$$\sum_{k=1}^{n} k a_k s^{k-1}=\operatorname{tr}\left[E\sum_{k=0}^{n-1} P_k s^k\right]$$

且 $$\sum_{k=0}^{n} a_k s^k I_n=(E_s-A)\sum_{k=0}^{n-1} P_k s^k$$

设 $\det(A)\neq 0$，广义 Faddeeva 递推公式为

$$\begin{cases} a_k=\dfrac{1}{k}\operatorname{tr}(EP_{k-1}), \\ \qquad k=1,2,\cdots,n \\ P_k=A^{-1}(EP_{k-1}-a_kI_n), \\ \qquad k=1,2,\cdots,n-1 \\ a_0=\det(-A),\ P_0=\operatorname{adj}(-A) \\ \quad=-a_0A^{-1}\end{cases}\tag{10}$$

当 A 奇异但 E 非奇异时，作 $v=s^{-1}$，则

$$(E_s-A)^{-1}=-v(A_v-E)^{-1}$$

利用广义 Faddeeva 递推公式经变换可得

$$\begin{cases} a_n=\det(E),\ P_{n-1}=E^{-1}a_n=\operatorname{adj}(E) \\ a_{n-k}=-\dfrac{1}{k}\operatorname{tr}(AP_{n-k}) \\ \qquad k=1,2,\cdots,n \\ P_{n-k}=E^{-1}(AP_{n-k+1}+a_{n-k+1}I_n), \\ \qquad k=2,3,\cdots,n\end{cases}\tag{11}$$

当 $E=I_n$ 时，式(11)与式(7)是相同的，可见 Faddeeva 递推算法或称 Leveierr−Faddeeva 算法或 Sourian− Frame 算法是广义 Faddeeva 递推算法的特例。

三、$P_k(k=0,1,\cdots,n-1)$的求法

利用广义 Faddeeva 递推算法可以同时求取 a_k 及 P_k 阵。若由二、1、二、2 的方法得到 a_k，则 P_k 亦可由式(10)或式(11)得到，但要求 A、E 不同时奇异。

当 A、E 同时奇异，如果 $(E_s-A)^{-1}$ 存在则总能找到常数 λ_0 使(λ_0E+A)非奇异。令 $v=s+\lambda_0$，则

$$(E_s-A)^{-1}=[E_v-(A+\lambda_0E)]^{-1}$$

$$=\sum_{k=0}^{n-1} Q_k v^k \Big/ \sum_{k=0}^{n} b_k v^k$$

由式(10),

$$\begin{cases} b_k=\dfrac{1}{k}\operatorname{tr}(EQ_{k-1}), \qquad k=1,2,\cdots,n \\ Q_k=(A+\lambda_0E)^{-1}(EQ_{k-1}-b_kI_n), \\ \qquad k=1,2,\cdots,n-1 \\ b_0=\det(-A-\lambda_0E),\ Q_0 \\ \quad=-b_0(A+\lambda_0E)^{-1}\end{cases}\tag{12}$$

则 $(E_s-A)^{-1}=\sum_{k=0}^{n-1} Q_k(s+\lambda_0)^k \Big/ \sum_{k=0}^{n} b_k(s$

$+\lambda_0)^k=\sum_{k=0}^{n-1}P_ks^k/\sum_{k=0}^{n}a_ks^k$ 将 Q_k、b_k 转化为 P_k、a_k 可由下式来完成:

$$\begin{cases}a_k=\sum_{i=k}^{n}b_i\lambda_0^{i-k}\gamma_{i,k+1}\\ \quad=b_k+\sum_{i=k+1}^{n}b_i\lambda_0^{i-k}\gamma_{i,k+1},\ k=0,1,\cdots,n\\ P_k=Q_k+\sum_{i=k+1}^{n}Q_i\lambda_0^{i-k}\cdot\gamma_{i,k+1},\\ \quad k=0,1,\cdots,n-1\end{cases}\tag{13}$$

其中系数 $\gamma_{i,j}$ 定义为

$$\begin{cases}\gamma_{i,j}=\gamma_{i-1,j}+\gamma_{i-1,j-1},\\ \quad(i=1,2,\cdots,n,\ j=2,3,\cdots,i)\\ \gamma_{0,1}=1,\\ \gamma_{i,1}=1,\gamma_{i,i+1}=1,\ \gamma_{0,0}=0\end{cases}\tag{14}$$

这里 λ_0 的取值要由人机交互来决定。为便于应用，下面介绍一种方法可以不必考虑 E、A 是否奇异或同为奇异，避开计算过程中的人机交互。

四、一种与 E、A 奇异性无关的新算法

记 $T(s)=E_s-A$，那么

$$\begin{aligned}(E_s-A)^{-1}&=T^{-1}(s)\\ &=-[vI_n-T(s)]^{-1}\Big|_{v=0}\\ &=-\frac{\sum_{k=0}^{n-1}P_k(s)v^k}{\sum_{k=0}^{n}a_k(s)\cdot v^k}\Bigg|_{v=0}\end{aligned}\tag{15}$$

其中 $P_k(s)$为 $n\times n$ 多项式矩阵，$a_k(s)$为 s 的多项式。

应用式(7)有

$$\begin{cases}P_{n-1}(s)=I_n,\ a_n(s)=1\\ a_{n-k}(s)=-\dfrac{1}{k}\mathrm{tr}[T(s)P_{n-k}(s)],\\ \quad k=1,2,\cdots,n\\ P_{n-K}(s)=T(s)P_{n-k+1}(s)\\ \quad+a_{n-k+1}(s)I_n,\ k=2,3,\cdots,n\end{cases}\tag{16}$$

显然，$P_{n-k}(s)$为 s 的 $k-1$ 次矩阵多项式，$a_{n-k}(s)$为 s 的 k 次多项式。记

$$P_{n-k}(s)=\sum_{i=0}^{k-1}Q_{n-k,i}s^i,$$

$$a_{n-k}(s)=\sum_{i=0}^{k}b_{n-k,i}s^i$$

$$T(s)P_{n-k}(s)=\sum_{i=0}^{k}R_{n-k-1,i}s^i$$

代入式(16)得到递推式

$$\begin{cases}Q_{n-1,0}=I_n\\ R_{n-k-1,i}=-AQ_{n-k,i}+EQ_{n-k,i-1},\\ \quad(k=1,2,\cdots,n_j\ i=0,1,\cdots,k)\\ b_{n-k,i}=-\dfrac{1}{k}\mathrm{tr}(R_{n-k-1,i}),\\ \quad(k=1,2,\cdots,n_j\ i=0,1,\cdots,k)\\ Q_{n-k,i}=R_{n-k,i}+b_{n-k+1,i}I_n,\\ \quad(k=2,3,\cdots,n_j\ i=0,1,\cdots,k-1)\end{cases}\tag{17}$$

注意到 $i<0$ 或 $i>k-1$ 时 $Q_{n-k,i}=0$，最后得到

$$(E_s-A)^{-1}=-\frac{\sum_{i=0}^{n-1}Q_{0,i}s^i}{\sum_{i=0}^{n}b_{0,i}s^i}\tag{18}$$

可见这种算法与 A、E 的奇异性无关。但需指出的是，这种算法需要 n^2-n 次矩阵乘法，而 Faddeeva 算法只要 n 次矩阵乘法，其计算量相对较大。

五、求时域解析 $L^{-1}[(E_s-A)^{-1}]$ 的新算法

文献[2]、[3]、[4]中都介绍了使用数值拉氏反变换求传函的时域解析式的方法。这里已知 $(E_s-A)^{-1}$ 分子矩阵多项式的系数阵 P_k 以及分母多项式系数 a_k 使用更适合计算机的新方法。

$$L^{-1}\left\{[E_s-A]^{-1}\right\}$$

$$=L^{-1}\left\{\sum_{k=0}^{n-1}P_k s^k \Big/ \sum_{k=0}^{n}a_k s^k\right\}$$

$$=\sum_{k=0}^{n-1}P_k\frac{d^k}{dt^k}L^{-1}\left(\frac{1}{\sum\limits_{j=0}^{n}a_j s^j}\right) \qquad (19)$$

记 $D(s)=\left(\sum\limits_{i=0}^{n}a_i s^i\right)^{-1}=\prod\limits_{j=1}^{m}(s-r_j)^{-n_j}$

其中 γ_j 为 $D(s)$ 的 n_j 重极点，且 $\sum\limits_{j=1}^{m}n_j=n$

又记 $D_k(s)=(s-r_k)^{n_k}$，$D(s)=\prod\limits_{\substack{j=1\\ i\neq k}}^{m}$

$(s-r_j)^{-n_j}$，$k=1,2,\cdots,m$

令 $F_k(s)=\ln D_k(s)=-\sum\limits_{\substack{j=1\\ j\neq k}}^{m}n_j\ln(s-r_j)$ (20)

所以

$$F_k^{(l)}(s)=(-1)^l(l-1)!\sum_{\substack{j=1\\ j\neq k}}^{m}n_j/(s-r_j)^l \qquad (21)$$

由式(20)又有 $D_k^{(i)}(s)$

$$=\sum_{j=0}^{j-1}C_{i-1}^{j}D_k^{(j)}(s)F_k^{(i-j)}(s) \qquad (22)$$

将 $D(s)$ 部分分式展开可写作

$$D(s)=\sum_{k=1}^{m}\sum_{j=1}^{n_k}\frac{C_{kj}}{(s-r_k)^{(n_k-j+1)}} \qquad (23)$$

其中 $C_{kj}=\dfrac{1}{(j-1)!}\dfrac{d^{j-1}}{ds^{j-1}}[D_k(s)]\Big|_{s=r_k}$ (24)

C_{kj} 可由式(22)、(21)递推得到。所以式(19)中

$$L^{-1}\left(\frac{1}{\sum\limits_{j=0}^{n}a_j s^j}\right)$$

$$=\sum_{k=1}^{m}\sum_{j=1}^{n}\frac{C_{n_{k-j+1}}}{(j-1)!}t^{j-1}e^{r_k t} \qquad (25)$$

代入式(19)得

$$L^{-1}\left\{(E_s-A)^{-1}\right\}$$

$$=\sum_{l=0}^{n-1}P_l\sum_{k=1}^{m}\sum_{j=1}^{n_k}\frac{C_{n_k-j+1}}{(j-1)!}\frac{d^l}{dt^l}(t^{j-1}e^{r_k t}) \qquad (26)$$

记 $\delta(i,j)=\dfrac{1}{2}[1+\mathrm{sign}(j-i+1)]$，其中

$$\mathrm{sign}(i)=\begin{cases}1, & i>0\\ 0, & i=0\\ -1, & i<0\end{cases}$$

则 $\dfrac{d^l}{dt^l}(t^{j-1}e^{r_k t})=\sum\limits_{i=0}^{l}C_l^i(t^{j-1})^{(i)}(e^{r_k t}))^{(l-i)}$

$$=\sum_{i=0}^{l}C_e^i\delta(j-1,i)\frac{(j-1)!}{(j-i-1)!}$$

$$\cdot t^{j-1-i}r_k^{e-i}e^{r_k t} \qquad (27)$$

其中 $C_l^i=l!/[i!(l-i)!]$

最后得到

$$L^{-1}\left\{(E_s-A)^{-1}\right\}$$

$$=\sum_{l=0}^{n-1}P_l\sum_{k=1}^{m}\sum_{j=1}^{n_k}C_{n_{k-j+1}}\sum_{i=0}^{l}\frac{l!\delta(j-1,i)r_k^{l-1}}{i!(l-i)!(j-i-1)!}$$

$$\cdot t^{j-1-i}e^{r_k t}) \qquad (28)$$

这就是本文所得到的 $Z(t)$ 的时域解析表达式。

六、结论

本文解决了广义系统的时域解析解求取

的计算方法。它们很容易由计算机语言实现，限于篇幅这里没有给出算例。使用本文提供的一些算法还可以解其它相关的问题，比如线性系统方框图描述的直接传函化简等。

参考文献

[1] [日]须田信英等著，曹长修译，《自动控制中的矩阵理论》，科学出版社，北京，1979年。

[2] 吴智铭编，《控制系统计算机辅助设计》，电子工业出版社，北京，1986年。

[3] 孙增圻等编，《控制系统的计算机辅助设计程序汇编（BASIC语言）》，计算机辅助设计丛书，清华大学出版社，北京，1988 年。

[4] 付明义，《状态转移矩阵的一种解析算法》，信息与控制，1987年第4期第60页。

(上接 75 页)

第 5 次耶路撒冷信息技术会议，10 月 22～25 日，耶路撒冷，以色列

重复数据管理学术讨论会，10 月 24～26 日，休斯敦，美国

欧洲计算机安全研讨会，10 月 24～26 日，图卢兹，法国

国际信息技术会议，10 月 29～31 日，伯恩默思，英国

第 14 次国际计算机软件和应用会议，10月 31 日～11 月 2 日，芝加哥，美国

第 14 次计算机在医疗保健中应用会议，11 月 4～7 日，华盛顿，美国

智能机器人系统设计与应用会议，11 月 6～7 日，费城，美国

第 2 次国际人工智能工具会议，11 月 6～9 日，华盛顿，美国

IEEE 国际计算机辅助设计会议，11 月 12～15 日，圣克拉拉，美国

1990 年超级计算会议，11 月 12～16 日，纽约，美国

1990 年软件维护会议，11 月 26～29 日，圣地亚哥，美国

第 23 次微程序设计与微体系结构学术讨论会，11 月 27～29 日，奥兰多，美国

第 4 次软件开发环境学术讨论会，12 月 3～5 日，欧文，美国

第 3 次国际计算机视觉会议，12 月 4～7 日，大阪，日本

第 11 次实时系统专题讨论会，12 月 5～7 日，奥兰多，美国

第 4 次国际计算机辅助软件工程讨论会，12 月 5～8 日，欧文，美国

IEEE 第 2 次“并行与分布处理”学术讨论会，12 月 10～12 日，达拉斯，美国

ANALYTICAL TIME DOMAIN SOLUTION OF GENERALIZED SYSTEMS

Dou Huifang and **Chen Yangquan**

Xian Institute of Technology

In this paper a computer oriented analytical method for solving a generalized system in time domain is proposed. The authors main contribution is to put forward a computation method without man–machine interaction for determining $L^{-1}\ \{(E_s-A)^{-1}\}$ when both E and A are singular.

Keywords: Generalized system, Numerical inverse Laplace transformation.

DETECTION OF THE WEAK COHERENT–PULSE SIGNAL IN NOISE

Liang Min and **Sun Zhongkang**

National University of Defence Technology

In the paper, a model and a criterion are proposed for the generalized locally optimum detector (LOD). On the basis of such criterion, the generalized LOD structure is derived for detecting the weak coherent–pulse sequence signal with known initial phase in the narrow–band non–Gaussian noise, and the asymptotic relative efficiency (ARE) propounded by Pitman is used to evaluate the LOD performance. Finally, numerical calculations are carried out for ARE of the LOD and some interesting results are obtained.

Keywords: Locally optimum detector, Signal detection, White noise.

NONLINEAR EDGE DETECTION METHOD FOR THE LOW SNR IMAGE

Lei Xiangkang and **Zhou Manli**

Huazhong University of Science and Technology