Discrete-continuous model conversion

L. S. Shieh and H. Wang
Department of Electrical Engineering, University of Houston, Houston, Texas 77004, USA

R. E. Yates
Guidance and Control Directorate, US Army Missile Command, Redstone Arsenal, Alabama 35809, USA
(Received March 1980)

This paper presents methods for model conversions of continuous-time state-space equations and discrete-time state-space equations. An improved geometric-series method is presented for converting continuous-time models to equivalent discrete-time models. Also, a direct truncation method, a matrix continued fraction method and a geometric-series method are presented for converting discrete models to equivalent continuous models. As a result, many well-developed theorems and methods in either continuous or discrete domains can be effectively applied to a suitable model in either domain.

Introduction

Many industrial processes and social systems are formulated via discrete models. Chemical processes and economic systems are typical examples. The recursive features of these discrete models and the recent availability of high performance low cost microprocessesors have enabled a new consideration of control and analysis of these systems. On the other hand, many control systems are formulated via continuous models for which many theories and practical methods have been developed. Many large practical control systems consist of both continuous-time and discrete-time subsystems. For effective analysis and synthesis of these composite systems it is often necessary to convert a discrete subsystem to an equivalent continuous model and vice versa.

Harris[1] has recently studied the problem of converting a discrete state equation to an equivalent continuous model. In his method a logarithmic matrix function is converted to a logarithmic scalar function via a model transformation such that both approximate and exact continuous models can be formulated from the discrete models. On the other hand, Shieh *et al.*[2] have converted continuous-time state equations to equivalent discrete-time models via a block-pulse function method[3,4] and a geometric-series method.[2] In this paper, the geometric-series method[2] will be improved and extended for accurately converting continuous state equations to equivalent discrete models. Also, a direct truncation method, a matrix continued fraction method, and a geometric-series method are proposed for converting discrete state equations to equivalent continuous equations.

Continuous-to-discrete conversion

Consider a linear, time-invariant continuous state equation:

$$\dot{X}_0(t) = A_c X_0(t) + B_c U_0(t) \tag{1a}$$

$$X_0(0) = \alpha \tag{1b}$$

where $X_0 \in R^n, U_0 \in R^m, A_c \in R^{n \times n}$, and $B_c \in R^{n \times m}$.

The exact solution of equation (1) is:

$$X_0(t) = \Phi_c(t)\, X_0(0) + \int_0^t \Phi_c(t-\lambda)\, B_c U_0(\lambda)\, \mathrm{d}\lambda \tag{2a}$$

where:

$$\Phi_c(t) = \mathrm{e}^{A_c t} = [\mathrm{e}^{A_c T}]^k = [\Phi_d(T)]^k = \Phi_d(kT) \tag{2b}$$

$$\Phi_d(T) \triangleq \mathrm{e}^{A_c T} \tag{2c}$$

$$t = kT \quad \text{for} \quad k = 0, 1, 2, \ldots \tag{2d}$$

$\Phi_c(t)$ is a continuous state transition matrix and $\Phi_d(kT)$ is a discrete state transition matrix, and T is the sampling period. $U_0(t)$ may be a known analytical function or a set of discrete data whose function is unknown. To simplify the convolution integral in equation (2a), $U_0(t)$ can be approximated by using either of the following piecewise-constant functions:

$$U_d(t) \triangleq U_r(kT) = U_0(kT) \cong U_0(t) \quad \text{for} \quad kT \leqslant t < (k+1)\,T; \quad k = 0, 1, 2, \ldots \tag{3a}$$

0307-904X/80/060449-07/$02.00

or

$$U_d(t) \triangleq U_t(kT) = \tfrac{1}{2}[U_0(kT) + U_0(kT+T)] \cong U_0(t) \quad \text{for } kT \leqslant t < (k+1)T; \quad k = 0, 1, 2, \ldots \tag{3b}$$

where $U_r(kT)$ is a rectangular wave which can be considered as obtained by applying a sampler and zero-order-hold device to $U_0(t)$, while $U_t(kT)$ is another rectangular wave formed by sampling and polygonal holding the $U_0(t)$. The approximate state due to $U_d(t)$ is $X_d(t)$. The new state equation becomes:

$$\dot{X}_d(t) = A_c X_d(t) + B_c U_d(t) \tag{4a}$$

$$X_d(0) = X_0(0) \tag{4b}$$

The solution of equation (4) evaluated at $t = kT$ is:

$$X_d(kT) = \Phi_d^k(T)\, X_d(0) + \sum_{j=0}^{k-1} \Phi_d(kT - jT - T) H_d U_d(jT) \quad \text{for } k = 1, 2, \ldots \tag{5a}$$

where:

$$H_d = \int_0^T e^{A_c\alpha} B_c \, d\alpha = [\Phi_d(T) - I_n]\, A_c^{-1} B_c \tag{5b}$$

I_n is an $n \times n$ identity matrix. Dropping the T in $X_d(kT)$ and $U_d(jT)$ in equation (5), the corresponding discrete model becomes:

$$X_d(k+1) = G_d X_d(k) + H_d U_d(k) \tag{6a}$$

$$X_d(0) = X_0(0) \tag{6b}$$

where:

$$G_d \triangleq \Phi_d(T) = e^{A_cT} \tag{6c}$$

$$H_d = [G_d - I_n]\, A_c^{-1} B_c \tag{6d}$$

For a large scale system, on-line evaluation of the exact transition matrix e^{A_cT} is a difficult task. An approximate transition matrix is often used for on-line operations of a control system. Using the geometric-series method, Shieh *et al.*[2] have shown that G_d can be expressed using approximinants of e^{A_cT} as:

$$G_d = e^{A_cT} = [e^{-\frac{1}{2}A_cT}]^{-1}[e^{\frac{1}{2}A_cT}] \cong Q_j^{-1}P_j \triangleq G_{dj} \quad \text{for } j = 1, 2, \ldots \tag{7a}$$

where:

$$Q_j = \left[I_n - \frac{1}{2j} A_cT\right]\left[I_n + \sum_{i=1}^{j-1} \frac{(-1)^i (j-i)}{(2^i)(j)(i!)} (A_cT)^i\right] \tag{7b}$$

$$P_j = \left[I_n + \frac{1}{2j} A_cT\right]\left[I_n + \sum_{i=1}^{j-1} \frac{(j-i)}{(2^i)(j)(i!)} (A_cT)^i\right] \tag{7c}$$

$$T < 2j/\|A_c\|, \quad j = 1, 2, \ldots \tag{7d}$$

where $\|A_c\|$ is the matrix norm of the matrix A_c.

Some approximate discrete-time system matrices G_{dj} for $j = 1, 2, 3, 4$ are:

$$G_{d1} = [I_n - \tfrac{1}{2}A_cT]^{-1}[I_n + \tfrac{1}{2}A_cT] = Q_1^{-1}P_1 \tag{8a}$$

$$G_{d2} = [I_n - \tfrac{1}{2}A_cT + \tfrac{1}{16}(A_cT)^2]^{-1} \times [I_n + \tfrac{1}{2}A_cT + \tfrac{1}{16}(A_cT)^2] = Q_2^{-1}P_2 \tag{8b}$$

$$G_{d3} = [I_n - \tfrac{1}{2}A_cT + \tfrac{7}{72}(A_cT)^2 - \tfrac{1}{144}(A_cT)^3]^{-1} \times [I_n + \tfrac{1}{2}A_cT + \tfrac{7}{72}(A_cT)^2 + \tfrac{1}{144}(A_cT)^3] = Q_3^{-1}P_3 \tag{8c}$$

$$G_{d4} = [I_n - \tfrac{1}{2}A_cT + \tfrac{7}{64}(A_cT)^2 - \tfrac{5}{384}(A_cT)^3 + \tfrac{1}{1536}(A_cT)^4]^{-1} \times [I_n + \tfrac{1}{2}A_cT + \tfrac{7}{64}(A_cT)^2 + \tfrac{5}{384}(A_cT)^3 + \tfrac{1}{1536}(A_cT)^4] = Q_4^{-1}P_4 \tag{8d}$$

The corresponding input matrices H_{dj} for $j = 1, 2, 3, 4$ are:

$$H_{d1} = T[I_n - \tfrac{1}{2}A_cT]^{-1}B_c \tag{9a}$$

$$H_{d2} = T[I_n - \tfrac{1}{2}A_cT + \tfrac{1}{16}(A_cT)^2]^{-1}B_c \tag{9b}$$

$$H_{d3} = T[I_n - \tfrac{1}{2}A_cT + \tfrac{7}{72}(A_cT)^2 - \tfrac{1}{144}(A_cT)^3]^{-1} \times [I_n + \tfrac{1}{72}(A_cT)^2]B_c \tag{9c}$$

$$H_{d4} = T[I_n - \tfrac{1}{2}A_cT + \tfrac{7}{64}(A_cT)^2 - \tfrac{5}{384}(A_cT)^3 + \tfrac{1}{1536}(A_cT)^4]^{-1}[I_n + \tfrac{5}{192}(A_cT)^2]B_c \tag{9d}$$

G_{dj} in equation (8) are approximate models of the exponential matrix function e^{A_cT} obtained by taking a finite number of dominant terms and approximating the infinite summation of the other terms of the Taylor-series matrix expansion. For example, G_{d1} can be expressed as:

$$G_{d1} = [I_n - \tfrac{1}{2}A_cT]^{-1}[I_n + \tfrac{1}{2}A_cT] \quad \text{for } T < 2/\|A_c\| \tag{10a}$$

$$= I_n + A_cT + \frac{1}{2!}(A_cT)^2 + \sum_{j=3}^{\infty} \frac{1}{2^{j-1}} (A_cT)^j \tag{10b}$$

while the exact expression is:

$$G_d = e^{A_cT} = I_n + A_cT + \frac{1}{2!}(A_cT)^2 + \frac{1}{3!}(A_cT)^3 + \frac{1}{4!}(A_cT)^4 + \ldots \tag{11a}$$

$$= I_n + A_cT + \frac{1}{2!}(A_cT)^2 + \sum_{j=3}^{\infty} \frac{1}{j!}(A_cT)^j \tag{11b}$$

The first three dominant terms in both equations (10b) and (11b) are identical and the other terms differ in their weighting factors $1/2^{j-1}$ in equation (10b) and $1/j!$ in equation (11b). It has been shown that if G_{d1} and H_{d1} of equations (8a) and (9a) and $U_d(t)(=U_t(kT))$ in equation (3b) are used in equation (6) to solve the discrete-time system of equation (1), the solution so obtained is identical to that obtained by using the Walsh function approach[5] and the block-pulse function approach.[4] Also, it has been shown that equation (6) may be used to obtain the numerical integration of a stiff state equation.[4]

From equation (7) we observe that $[I_n - (1/2j)A_cT]^{-1}$ is a geometric series that converges if $T < 2j/\|A_c\|$ and a better approximation can be obtained if $T \ll 2j/\|A_c\|$. Thus, we can improve the approximants of equation (7a) as follows.

Rewriting equation (7a) and modifying the sampling period T with $T_m \triangleq T/m$ where m is a positive integer yields:

$$G_d = [e^{-\frac{1}{2}A_cT}]^{-1}[e^{\frac{1}{2}A_cT}] \tag{12a}$$

$$= \{[e^{-\frac{1}{2}A_cT_m}]^{-1}[e^{\frac{1}{2}A_cT_m}]\}^m \tag{12b}$$

$$= \{[e^{-\frac{1}{2}\hat{A}_cT}]^{-1}[e^{\frac{1}{2}\hat{A}_cT}]\}^m \tag{12c}$$

$$\triangleq G_g \tag{12d}$$

$$\cong \{\hat{Q}_j^{-1}\hat{P}_j\}^m \triangleq G_{gj} \tag{12e}$$

where:

$$T_m = T/m \tag{12f}$$

$$\hat{A}_c = A_c/m \tag{12g}$$

$$\hat{Q}_j = \left[I_n - \frac{1}{2(j)(m)} A_c T\right] \times \left[I_n + \sum_{i=1}^{j-1} \frac{(-1)^i (j-i)}{(2^i)(j)(i!)(m)^i} (A_c T)^i\right] \tag{12h}$$

$$\hat{P}_j = \left[I_n + \frac{1}{2(j)(m)} A_c T\right] \times \left[I_n + \sum_{i=1}^{j-1} \frac{(j-i)}{(2^i)(j)(i!)(m)^i} (A_c T)^i\right] \tag{12i}$$

$$T < (2)(j)(m)/\|A_c\| \tag{12j}$$

Our improved discrete state equation approximation to the continuous system becomes:

$$X_g(k+1) = G_g X_g(k) + H_g U_d(k) \tag{13a}$$

$$X_g(0) = X_d(0) = X_0(0) \tag{13b}$$

where $X_g(k) \cong X_d(k) \cong X(t)$ at $t = kT$. Also:

$$H_g = [G_g - I_n] A_c^{-1} B_c \tag{13c}$$

For comparing equations (8) and (12) we list as follows:

$$G_g \cong \left\{\left[I_n - \frac{1}{2(m)}(A_c T)\right]^{-1}\left[I_n + \frac{1}{2(m)}(A_c T)\right]\right\}^m$$

$$\triangleq G_{g1} \quad \text{for } j = 1 \tag{14a}$$

$$\cong \left\{\left[I_n - \frac{1}{2(m)}(A_c T) + \frac{1}{16(m)^2}(A_c T)^2\right]^{-1} \times \left[I_n + \frac{1}{2(m)}(A_c T) + \frac{1}{16(m)^2}(A_c T)^2\right]\right\}^m \tag{14b}$$

$$\triangleq G_{g2} \quad \text{for } j = 2 \tag{14c}$$

$$\cong \ldots$$

When $m = 1$ in equation (14a), we observe that G_{g1} in equation (14a) is identical to G_{d1} in equation (8a), or:

$$G_{g1} = G_{d1} = [I_n - \tfrac{1}{2}A_c T]^{-1}[I_n + \tfrac{1}{2}A_c T] \qquad T < 2/\|A_c\|$$

$$= I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(1)}(A_c T)^3 + \frac{1}{2^3(1)}(A_c T)^4 + \frac{1}{2^4(1)}(A_c T)^5 + \ldots \tag{15a}$$

When $m = 2$, equation (14a) becomes:

$$G_{g1} = \left\{\left[I_n - \frac{1}{2(2)} A_c T\right]^{-1}\left[I_n + \frac{1}{2(2)} A_c T\right]\right\}^2 \qquad T < 4/\|A_c\|$$

$$= I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(\frac{4}{3})}(A_c T)^3 + \frac{1}{2^3(2)}(A_c T)^4 + \frac{1}{2^4(\frac{16}{5})}(A_c T)^5 + \ldots \tag{15b}$$

If $m = 4$, we have:

$$G_{g1} = \left\{\left[I_n - \frac{1}{2(4)} A_c T\right]^{-1}\left[I_n + \frac{1}{2(4)} A_c T\right]\right\}^4 \qquad T < 8/\|A_c\|$$

$$= I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(\frac{16}{11})}(A_c T)^3 + \frac{1}{2^3(\frac{8}{3})}(A_c T)^4 + \frac{1}{2^4(\frac{256}{45})}(A_c T)^5 + \ldots \tag{15c}$$

An exact expansion for $e^{A_c T}$ is:

$$G_d = e^{A_c T} = I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(1.5)}(A_c T)^3 + \frac{1}{2^3(3)}(A_c T)^4 + \frac{1}{2^4(7.5)}(A_c T)^5 + \ldots \tag{16}$$

Comparing equations (15a), (15b), (15c), and (16) we observe that the first three terms in all four equations are identical and that each of the weighting factors for the other terms $(A_c T)^j$ in equation (15) approaches the corresponding terms in equation (16) as the value of m increases. Therefore, the discrete-time model of equation (13) is better than that of equation (6).

It has been reported[4] that the simplest G_{d1} in equation (10a) or G_{g1} in equation (15a) has been successfully applied to solve a stiff state-space equation for which the fourth-order Runge-Kutta method fails. This is due to the fact that the Runge-Kutta method[6] approximates the infinite series expansion of $e^{A_c T}$ by taking only the first five terms, i.e.,

$$e^{A_c T} = I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(1.5)}(A_c T)^3 + \frac{1}{2^3(3)}(A_c T)^4 + \frac{1}{2^4(7.5)}(A_c T)^5 + \ldots \tag{17a}$$

$$\cong I_n + A_c T + \tfrac{1}{2}(A_c T)^2 + \frac{1}{2^2(1.5)}(A_c T)^3 + \frac{1}{2^3(3)}(A_c T)^4 \tag{17b}$$

Comparing equations (15a) and (17b) we observe that the first five dominant terms of G_{g1} in equation (15a) are nearly equal to those of equation (17b), while G_{g1} contains an additional infinite series which apparently improves the accuracy of the numerical integration of a stiff differential equation. Therefore, the other more sophisticated models will be more effective in solving these stiff differential equations.

A matrix inversion has also been introduced in the proposed method. However, a matrix inversion is a common exercise for control system applications such as identification, estimation, and optimal control problems.

From equation (12j) we also observe that a more sophisticated model, which uses more terms, enables the use of a larger sampling period. This can be verified as follows. Letting T_j designate the sampling periods if the model G_{gj} for $j = 1$ and 2 in equation (13) we have:

$$G_{g1} = \left\{\left[I_n - \frac{1}{2m}(A_cT_1)\right]^{-1}\left[I_n + \frac{1}{2m}(A_cT_1)\right]\right\}^m \quad \text{for } j = 1 \quad (18a)$$

and

$$G_{g2} = \left\{\left[I_n - \frac{1}{2m}(A_cT_2) + \frac{1}{16m^2}(A_cT_2)^2\right]^{-1} \times \left[I_n + \frac{1}{2m}(A_cT_2) + \frac{1}{16m^2}(A_cT_2)^2\right]\right\}^m \quad \text{for } j = 2 \quad (18b)$$

Consider that A_c is a diagonal matrix with all eigenvalues λ_j located on the diagonal. The absolute value of the largest eigenvalue, λ_m, is often used to evaluate the minimal sampling period. Now we can compare the sampling periods T_1 and T_2 in equations (18a) and (18b) by solving the following equation:

$$\left\{\left(1 - \frac{1}{2m}\lambda_m T_1\right)^{-1}\left(1 + \frac{1}{2m}\lambda_m T_1\right)\right\}^m = \left\{\left(1 - \frac{1}{2m}\lambda_m T_2 + \frac{1}{16m^2}\lambda_m^2 T_2^2\right)^{-1} \times \left(1 + \frac{1}{2m}\lambda_m T_2 + \frac{1}{16m^2}\lambda_m^2 T_2^2\right)\right\}^m \quad (19a)$$

or

$$T_1 = \frac{1}{1 + \frac{1}{16m^2}(\lambda_m T_2)^2} T_2 < T_2 \quad (19b)$$

from equation (19b) we conclude that a larger sampling period T_2 can be used if G_{g2} instead of G_{g1} is used to formulate the equivalent discrete-time model. As a result, we have greatly increased our flexibility in selecting the minimal common sampling period among the various subsystems of a large composite system.

Given a unit-feedback continuous control system with an input function $r(t)$ and error function $U_0(t)$ it is often necessary to determine the equivalent discrete model with a sample and hold acting on the $U_0(t)$ for digital control.[7] A solution to this problem is to first convert the open-loop continuous system to a discrete-time model via our new method. Then, the piecewise constant function $U_d(t)$ will be a good approximation of the $U_0(t)$. As a result, the output response of a closed-loop digital system to an input $r(t)$ will closely follow the original continuous closed-loop system. If the continuous closed-loop system is directly converted to an equivalent closed-loop discrete model with a sample and hold acting on $r(t)$, the discrete model gives good output response only if the input frequency of $r(t)$ is sufficiently low in comparison with the sampling frequency.

Discrete-to-continuous conversion

Consider a discrete state equation:

$$X_d(k+1) = G_d X_d(k) + H_d U_d(k) \quad (20a)$$

$$X_d(0) = X_0(0) \quad (20b)$$

It is necessary to convert the discrete-time model to an equivalent continuous-time model so that well developed theorems and methods in the continuous-time and frequency domains[8] can be effectively applied. The corresponding continuous state equation is:

$$\dot{X}_c(t) = A_c X_c(t) + B_c U_c(t) \quad (21a)$$

$$X_c(0) = X_d(0) \quad (21b)$$

where $X_d(kT) \cong X_c(t)$ at $t = kT$ and $U_d(kT) = U_c(t)$ at $t = kT$ and:

$$U_c(t) \cong U_d(kT) + \frac{U_d(kT+T) - U_d(kT)}{T}(t - kT)$$

$$k = 0, 1, 2, \ldots, kT \leqslant t \leqslant (k+1)T \quad (21c)$$

From equation (6) we can solve for the equivalent system matrix A_c and input vector B_c as:

$$A_c = \frac{1}{T}\ln G_d \quad (21d)$$

and

$$B_c = A_c[G_d - I_n]^{-1} H_d \quad (21e)$$

The logarithmic matrix function in equation (21d) can be expressed by:

$$\ln G_d = 2\left[R + \tfrac{1}{3}R^3 + \tfrac{1}{5}R^5 + \ldots + \frac{1}{n}R^n + \frac{1}{n+2}R^{n+2} + \frac{1}{n+4}R^{n+4} + \ldots\right] \quad (22a)$$

where:

$$R = [G_d - I_n][G_d + I_n]^{-1} \quad (22b)$$

The matrix R is expressed by a rational matrix function in a bilinear expansion form.[7]

Three methods are proposed for approximating $\ln G_d$ in determining A_c as follows.

Direct truncation method

The approximate A_c can be determined by taking the first several dominant terms of the expansion in equation (22a) and truncating others. Some approximate models are:

$$A_c = \frac{1}{T}\ln G_d \quad (23a)$$

$$= \frac{2}{T}[R + \tfrac{1}{3}R^3 + \tfrac{1}{5}R^5 + \tfrac{1}{7}R^7 + \ldots] \quad (23b)$$

$$\cong \frac{2}{T}R \quad (23c)$$

$$\cong \frac{2}{T}[R + \tfrac{1}{3}R^3] \quad (23d)$$

$$\cong \frac{2}{T}[R + \tfrac{1}{3}R^3 + \tfrac{1}{5}R^5] \quad (23e)$$

$$\cong \ldots$$

The equivalent B_c can be obtained by substituting equation (23) into equation (21e). If the model in equation (23c) is used, we have:

$$A_c \cong \frac{2}{T}(G_d - I_n)(G_d + I_n)^{-1} \quad (24a)$$

and

$$B_c \cong \frac{1}{T}(I_n - \tfrac{1}{2}A_cT)H_d \tag{24b}$$

The approximate models in equation (24) are identical to those developed from the discrete-time model G_{d1} in equation (8a) and H_{d1} in equation (9a). Since R is a bilinear form, the numerical errors due to matrix multiplications in equation (23) are minimized.

Matrix continued fraction method

A method using a matrix continued fraction approximation[9] has been shown to be effective in approximating high-order multivariable systems by a low-order one. The same approach can be applied to approximate the logarithmic matrix function in equation (22a). Rewriting equations (21d) and (22a) and expanding into a matrix continued fraction results in:

$$A_c = \frac{1}{T}\ln G_d \tag{25a}$$

$$= \frac{2}{T}R[I_n + \tfrac{1}{3}N + \tfrac{1}{5}N^2 + \tfrac{1}{7}N^3 + \ldots] \tag{25b}$$

$$= \frac{2}{T}R[K_1 + N[K_2 + N[K_3 + N[K_4 + N[\ldots]^{-1}]^{-1}]^{-1}]^{-1}]^{-1} \tag{25c}$$

where $N = R^2$:

$$R = [G_d - I_n][G_d + I_n]^{-1}$$

The matrices K_i $(=k_iI_n)$ are diagonal matrices and are the matrix quotients in the matrix continued fraction expansion. The scalars k_i can be determined from the following scalar Routh algorithm[10]:

$$a_{1,1} = 1, a_{1,j} = 0;\quad j = 2, 3, \ldots \tag{26a}$$

$$a_{2,j} = \frac{1}{2j-1};\quad j = 1, 2, \ldots \tag{26b}$$

$$a_{i,j} = a_{i-2,j+1} - k_{i-2}a_{i-1,j+1};\quad j = 1, 2, \ldots, i = 3, 4, \ldots \tag{26c}$$

$$k_i = a_{i,1}/a_{i+1,1};\quad i = 1, 2, \ldots \tag{26d}$$

The $k_i, i = 1, 2, \ldots$ so obtained are $k_1 = 1, k_2 = -3$, $k_3 = 5/4$, $k_4 = -28/9$, etc. The approximate models can be obtained by keeping the first several dominant matrix quotients K_i and discarding others, that is:

$$A_c \cong \frac{2}{T}R[K_1]^{-1} = \frac{2}{T}R \tag{27a}$$

$$\cong \frac{2}{T}R[K_1 + N[K_2]^{-1}]^{-1} = \frac{2}{T}R[K_2][N + K_1K_2]^{-1}$$

$$= \frac{-6}{T}R(R^2 - 3I_n)^{-1} \tag{27b}$$

$$\cong \frac{2}{T}R[K_1 + N[K_2 + N[K_3]^{-1}]^{-1}]^{-1}$$

$$= \frac{2}{T}R[N + K_2K_3][(K_1 + K_3)N + K_1K_2K_3]^{-1}$$

$$= \frac{2}{T}R\ (R^2 - \tfrac{15}{4}I_n)(\tfrac{9}{4}R^2 - \tfrac{15}{4}I_n)^{-1} \tag{27c}$$

$$\cong \frac{2}{T}R[K_1 + N[K_2 + N[K_3 + N[K_4]^{-1}]^{-1}]^{-1}]^{-1}$$

$$= \frac{2}{T}R\ [(K_2 + K_4)N + (K_2K_3K_4)]\,[N^2 + (K_1K_2 + K_1K_4 + K_3K_4)N + (K_1K_2K_3K_4)]^{-1}$$

$$= \frac{2}{T}R(-\tfrac{55}{9}R^2 + \tfrac{35}{3}I_n)(R^4 - 10R^2 + \tfrac{35}{3}I_n)^{-1} \tag{27d}$$

Note that A_c in equation (27a) is identical to that in equation (23c). The B_c can be obtained by substituting A_c of equation (27) into equation (21e).

Geometric-series method

Rewriting equations (21) and (22) yields:

$$A_c = \frac{1}{T}\ln G_d$$

$$= \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n + \frac{1}{n+2}R^{n+2} + \frac{1}{n+4}R^{n+4} + \frac{1}{n+6}R^{n+6} + \ldots\right] \tag{28a}$$

$$= \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n + \frac{1}{n+2}R^{n+2} + \sum_{i=2}^{\infty}\frac{1}{n\left(1+\frac{2i}{n}\right)}R^{n+2i}\right] \tag{28b}$$

The weighting factor of the term R^{n+2i} in the infinite series in equation (28b) can be approximated by:

$$\frac{1}{n\left(1+\frac{2i}{n}\right)} \cong \frac{1}{n\left(1+\frac{2}{n}\right)^i} \tag{29}$$

Thus the logarithmic matrix function in equation (28b) can be approximated by using a geometric series, that is:

$$A_c = \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n + \frac{1}{n+2}R^{n+2} + \sum_{i=2}^{\infty}\frac{1}{n\left(1+\frac{2i}{n}\right)}R^{n+2i}\right] \tag{30a}$$

$$\cong \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n + \frac{1}{n+2}R^{n+2} + \sum_{i=2}^{\infty}\frac{1}{n\left(1+\frac{2}{n}\right)^i}R^{n+2i}\right] \tag{30b}$$

$$= \frac{2}{T}\left\{R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n\right.$$

$$\times\left[I_n + \frac{1}{\left(1+\frac{2}{n}\right)}R^2 + \frac{1}{\left(1+\frac{2}{n}\right)^2}R^4 + \frac{1}{\left(1+\frac{2}{n}\right)^3}R^6 + \ldots\right]\Bigg\} \quad (30c)$$

$$= \frac{2}{T}\left\{R + \tfrac{1}{3}R^3 + \ldots + \frac{1}{n}R^n\left[I_n - \frac{1}{\left(1+\frac{2}{n}\right)}R^2\right]^{-1}\right\}$$

$$\text{for } \|R^2\| < \left(1+\frac{2}{n}\right) \quad (30d)$$

Observe that $[I_n - (1/(1+2/n))R^2]^{-1}$ is a geometric series. Comparing equations (30a), (30b), and (23) we note that all equations have the same first several dominant terms, but other terms differ by weighting factors; $1/n(1+2i/n)$ in equation (30a), $1/n(1+2/n)^i$ in equation (30b) and zero in equation (23). The A_c for $n = 1, 3$, and 5 are:

$$A_c \cong \frac{2}{T}\left[R + \frac{1}{(1+\frac{2}{1})}R^3 + \frac{1}{(1+\frac{2}{1})^2}R^5 + \ldots\right]$$

$$= \frac{2}{T}R[I_n - \tfrac{1}{3}R^2]^{-1} \quad \text{as } n = 1 \quad (31a)$$

$$A_c \cong \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \frac{1}{3(1+\frac{2}{3})}R^5 + \frac{1}{3(1+\frac{2}{3})^2}R^7 + \ldots\right]$$

$$= \frac{2}{T}R[I_n - \tfrac{4}{15}R^2][I_n - \tfrac{3}{5}R^2]^{-1} \quad \text{as } n = 3 \quad (31b)$$

$$A_c \cong \frac{2}{T}\left[R + \tfrac{1}{3}R^3 + \tfrac{1}{5}R^5 + \frac{1}{5(1+\frac{2}{5})}R^7 + \frac{1}{5(1+\frac{2}{5})^2}R^9 + \ldots\right]$$

$$= \frac{2}{T}R[I_n - \tfrac{8}{21}R^2 - \tfrac{4}{105}R^4][I_n - \tfrac{5}{7}R^2]^{-1} \quad \text{as } n = 5 \quad (31c)$$

The corresponding B_c can be obtained by substituting the A_c from equation (31) into the following equation:

$$B_c \cong A_c[G_d - I_n]^{-1}H_d \quad (32)$$

Thus, we obtain the equivalent continuous-time models in equation (21).

Conclusions

A geometric-series method has been presented for accurate representation of a continuous-time state equation by an equivalent discrete-time state equation. This allows the use of well-developed theorems and algorithms in the discrete-time domain to solve the continuous-time problems indirectly. It has also been shown that if a more accurate model is used, a larger sampling period can be used. Thus, the difficulty of selecting a common sampling period among subsystems of a composite system can be greatly reduced. Three methods have been presented for developing equivalent continuous-time state equations from discrete-time state equations. Therefore, well-developed design methods in the continuous-time and frequency domains can be effectively used for discrete systems. The rapid conversion of the mixed models of continuous and discrete subsystems will greatly simplify design procedures.

Acknowledgements

This work was supported in part by US Army Missile Research and Development Command, DAAH01-80-C-0323, and US Army Research Office, DAAG29-80-K-0077.

References

1 Harris, E. L. *Automatica* 1979, **15**, 97
2 Shieh, L. S. *et al. Int. J. Syst. Sci.* 1979, **10**, 1415
3 Shieh, L. S. *et al. Int. J. Control* 1978, **28**, 383
4 Shieh, L. S. *et al. Comput. Elec. Eng.* 1979, **6**, 3
5 Chen, C. F. and Hsiao, C. H. *Int. J. Syst. Sci.* 1975, **6**, 833
6 Vidyasagar, M. 'Nonlinear systems analysis', Prentice-Hall, Inc., Englewood Cliffs, USA, 1978, p. 113.
7 Jury, E. I., 'Theory and application of the Z-transform method', John Wiley, New York, 1964
8 Thaler, G. J. 'Design of feedback systems', Dowden, Hutchinson and Ross, Inc., Stroudsburg, USA, 1973
9 Shieh, L. S. and Gaudiano, F. F. *IEEE Trans. Circuits Syst.* 1975, **CAS-22**, (9), 721
10 Jury, E. I. 'Inners and stability of dynamic systems', John Wiley, New York, 1974

Appendix

An illustrative example

Consider a continuous-time state equation:

$$\dot{X}_0(t) = A_cX_0(t) + B_cU_0(t) \quad (33a)$$

$$X_0(0) = 0 \quad (33b)$$

where:

$$A_c = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}, \quad B_c = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad X(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \quad (33c)$$

When $T = 0.25$ s (where $T = \frac{1}{4} < 2/\|A_c\| = 2/7$ as $j = 1$ and $m = 1$ in equation (12j)) and $U_0(t)$ is a unit-step function, we have the exact discrete-time model from equation (6):

$$X_d(k+1) = G_dX_d(k) + H_dU_d(k) \quad (34a)$$

$$X_d(0) = X_0(0) = 0 \quad (34b)$$

where $X_0(t) = X_d(kT)$ as $t = kT$

$$G_d = \begin{bmatrix} 1.45412 & 0.38920 \\ 0.58381 & 0.48111 \end{bmatrix} \quad \text{and} \quad H_d = \begin{bmatrix} 0.64872 & 0.05190 \\ 0.32437 & 0.16865 \end{bmatrix} \quad (34c)$$

For on-line real-time use, it is desirable to use an approximate model. If the approximate models G_{g1} in equation (14a) for $m = 1, 2$, and 3 are chosen, then we have:

$$G_d \cong G_{g1} \cong \begin{bmatrix} 1.4615 & 0.4102 \\ 0.6154 & 0.4359 \end{bmatrix} \quad \text{as } m = 1$$

$$\cong \begin{bmatrix} 1.4561 & 0.3939 \\ 0.5909 & 0.4713 \end{bmatrix} \quad \text{as } m = 2$$

$$\cong \begin{bmatrix} 1.4550 & 0.3913 \\ 0.5869 & 0.4769 \end{bmatrix} \quad \text{as } m = 3 \quad (35a)$$

and

$$H_d \cong H_{g1} \cong \begin{bmatrix} 0.6666 & 0.0513 \\ 0.3334 & 0.1795 \end{bmatrix} \quad \text{as } m = 1$$

$$\cong \begin{bmatrix} 0.6530 & 0.0518 \\ 0.3265 & 0.1711 \end{bmatrix} \quad \text{as } m = 2$$

$$\cong \begin{bmatrix} 0.6507 & 0.0519 \\ 0.3254 & 0.1697 \end{bmatrix} \quad \text{as } m = 3 \qquad (35b)$$

If the approximate models G_{g2} in equation (14b) are used, we have:

$$G_d \cong G_{g2} \cong \begin{bmatrix} 1.4561 & 0.3939 \\ 0.5909 & 0.4713 \end{bmatrix} \quad \text{as } m = 1$$

$$\cong \begin{bmatrix} 1.4546 & 0.3903 \\ 0.5856 & 0.4788 \end{bmatrix} \quad \text{as } m = 2$$

$$\cong \begin{bmatrix} 1.4538 & 0.3894 \\ 0.5844 & 0.4800 \end{bmatrix} \quad \text{as } m = 3 \qquad (36a)$$

and

$$H_d \cong H_{g2} \cong \begin{bmatrix} 0.6531 & 0.0518 \\ 0.3266 & 0.1711 \end{bmatrix} \quad \text{as } m = 1$$

$$\cong \begin{bmatrix} 0.6498 & 0.0519 \\ 0.3250 & 0.1692 \end{bmatrix} \quad \text{as } m = 2$$

$$\cong \begin{bmatrix} 0.6485 & 0.0518 \\ 0.3244 & 0.1689 \end{bmatrix} \quad \text{as } m = 3 \qquad (36b)$$

Comparing equations (35), (36), and (34c), the approximation is satisfactory. Observe that the results in equation (36) are better than those in equation (35).

On the other hand, if G_d and H_d in equation (34) are given and the equivalent A_c and B_c of equation (33) are required, the approximate models in equations (23c), (23d), and (23e) via the direct truncation method are:

$$A_c \cong \begin{bmatrix} 1.04560 & 1.82744 \\ 2.74120 & -3.52304 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.95932 & 0.02630 \\ 0.97968 & 0.90054 \end{bmatrix}$$

$$A_c \cong \begin{bmatrix} 1.01419 & 1.96861 \\ 2.95293 & -3.90733 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.99850 & 0.00598 \\ 0.99927 & 0.98132 \end{bmatrix}$$

and

$$A_c \cong \begin{bmatrix} 1.00341 & 1.99299 \\ 2.98951 & -3.97907 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.99991 & 0.00138 \\ 0.99998 & 0.99581 \end{bmatrix} \qquad (37a)$$

The approximate models in equations (27b), (27c), and (27d) via the matrix continued fraction method are:

$$A_c \cong \begin{bmatrix} 1.007460 & 1.983676 \\ 2.975556 & -3.951780 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.999298 & 0.003124 \\ 0.999666 & 0.990289 \end{bmatrix}$$

$$A_c \cong \begin{bmatrix} 1.000747 & 1.998432 \\ 2.997691 & -3.995378 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.999963 & 0.000306 \\ 1.000018 & 0.999076 \end{bmatrix}$$

and

$$A_c \cong \begin{bmatrix} 1.000061 & 1.999824 \\ 2.999780 & -3.999546 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.999973 & 0.000029 \\ 1.000007 & 0.999911 \end{bmatrix} \qquad (37b)$$

The approximate models in equations (31a), (31b), and (31c) via the geometric series method are:

$$A_c \cong \begin{bmatrix} 1.00746 & 1.98368 \\ 2.97556 & -3.95178 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.99930 & 0.00313 \\ 0.99967 & 0.99029 \end{bmatrix}$$

$$A_c \cong \begin{bmatrix} 1.00075 & 1.99844 \\ 2.99772 & -3.99538 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 2.00672 & 0.00166 \\ 1.00003 & 0.99908 \end{bmatrix}$$

and

$$A_c \cong \begin{bmatrix} 1.00009 & 1.99977 \\ 2.99965 & -3.99937 \end{bmatrix} \quad B_c \cong \begin{bmatrix} 1.999975 & 0.00004 \\ 0.99997 & 0.99987 \end{bmatrix} \qquad (37c)$$

Comparing the results in equation (37) with the exact results in equation (33), illustrates that the approximation is very satisfactory.