Yugoslav Journal of Operations Research
4 (1994), Number 1, 27-34

DISCRETIZATION AND CONTINUALIZATION OF MIMO SYSTEMS

Stanoje BINGULAC

Kuwait University, Department of Electrical and Computer Engineering, POB 5969, 13060 - Safat, Kuwait

Hugh F. VANLANDINGHAM

Virginia Polytechnic Institute and State University, Electrical Engineering, Blacksburg, VA 24061-0111, USA

Abstract:* New numerically robust algorithms are presented for converting linear continuous-time constant-parameter state models into equivalent discrete-time state models (discretization) as well as the reverse problem of determining continuous-time models to represent given discrete-time models (continualization). Two methods of discretizing linear uniformly sampled systems have been considered for their utility in computer-aided design. These methods are the standard zero-order hold method which assumes that inputs are held constant at their previous sample value for the duration of the sample interval, and a method which assumes that the inputs are linearly interpolated between samples.

Keywords: Modeling, recursive algorithms, discrete time systems, state-space methods, computer programming, computer-aided design.

1. INTRODUCTION

With the widespread use of computers in control loops it is inevitable that control engineers will face problems associated with sampled-data systems. Such systems by their very definition contain a mixture of continuous-time (C-T) and discrete-time (D-T) signals. A common problem that arises with sampled-data control systems is to find the equivalent effect of C-T operations as seen by the computer in the loop. Typically, the modeling of the signal converters assumes an ideal uniform sampler for

---

Communicated by S.Dajović

* The research presented in this paper has been supported in part by the Research Administration, Kuwait University, under the Grant EDE 105.

the analog-to-digital converter and a simple (zero-order) hold device synchronized with the samples for a digital-to-analog converter. With these assumptions one may find in many references the standard *zero-order hold* model, also known as the *step invariant* (SI) model which will be discussed subsequently.

In addition to simple plant modeling with SI equivalents there are occasions, such as in digital redesign, that demand more accuracy between a given C-T system and its D-T equivalent model. In these instances higher-order discrete models are required. Such a model is one which assumes a linearly interpolated input. This method is referred to as a *ramp invariant* (RI) model in contrast to the standard ZOH model's being a *step invariant* (SI) model. There are many other useful models, but this paper will focus on only the SI and RI methods of discretization as being the most useful in practice.

The reverse problem, called *continualization*, is that of reconstructing a C-T model from a given D-T model. This problem could arise, for instance, when measured discrete data are used to identify a C-T system [2]. The particular method of continualization selected would depend on how the discrete data were derived. The method of continualization is presented for the two discretization techniques, thereby offering the designer flexibility in going between the continuous and the discrete domains.

2. PROBLEM FORMULATION

We assume a basic *state space realization* for a linear system consisting of a 4-tuple of matrices; namely,

$$R_c = \{ A_c, B_c, C_c, D_c \} \tag{1}$$

which defines the state model

$$\begin{aligned} \dot{x}(t) &= A_c\, x(t) + B_c\, u(t) \\ y(t) &= C_c\, x(t) + D_c\, u(t) \end{aligned} \tag{2}$$

where $x(t)$, $u(t)$ and $y(t)$ are the state, input and output vectors with dimensions n, m and p, respectively, while the matrices A_c, B_c, C_c and D_c are constant matrices with compatible dimensions.

2.1. DISCRETIZATION PROCEDURES

In this paper some computational issues of the discretization and continualization procedures will be discussed with emphasis on explaining different algorithms which are easily implementable. The problem of discretization will be discussed first.

The familiar SI (ZOH) equivalent D-T model assumes that the input vector $u(t)$ in Equation (2) is constant between (uniform) samples. The equivalent D-T model can be represented as

$$R_d = \{ A_d, B_d, C_d, D_d \} \tag{3}$$

which implies the D–T state model

$$x(k+1) = A_d x(k) + B_d u(k)$$
$$y(k) = C_d x(k) + D_d u(k) \tag{4}$$

The matrices A_d and B_d in Equation (3) are related to A_c and B_c in Equation (2) by the well known relations [4]

$$A_d = e^{A_c T} = \sum_{i=0}^{\infty} \frac{(A_c T)^i}{i!}$$
$$B_d = \int_0^T e^{A_c t} B_c \, dt = \sum_{i=0}^{\infty} \frac{(A_c T)^i}{(i+1)!} B_c T \tag{5}$$

$$C_d = C_c \text{ and } D_d = D_c \tag{6}$$

Step Invariant (SI) Equivalent Model. This algorithm is a numerically robust procedure for calculating A_d and B_d described above. The standard general method for calculating A_d is to compute a truncated version of Equation (5). The problem with this approach is that for matrices A_c and sampling intervals T satisfying that

$$\|A_c T\| > 1 \tag{7}$$

a truncated version of Equation (5) may either require large N, leading to considerable round-off errors, or may not converge at all [6]. The concept of *norm* is used here to have a scalar measure of the relative size of the entries of a matrix, usually for the comparison of convergence errors after different numbers of steps of a particular algorithm. For this purpose the *Frobenius (F) norm*, defined as the square root of the sum of squares of all matrix elements, is used. Any other standard matrix norm could be used to measure the same relative effects.

It has been shown in [7] that the SI model can be calculated using an intermediate matrix E as follows:

$$A_d = I + EA_c T, \text{ and } B_d = EB_c T \tag{8}$$

where $E = \sum_{i=0}^{\infty} \frac{(A_c T)^i}{(i+1)!}$

It is well known that to resolve the problem associated with Equation (7), it is possible to utilize the property of the exponential function that

$$\exp(x) = e^x = (e^{(x/r)})^r \tag{9}$$

The present method extends this techniques to permit calculation of both A_d and B_d under the condition of Equation (7) as well as the condition that A_c may be singular.

It is shown in [3] that the truncated version of E in Equation (8) can be calculated by the following recursive process:

$$T_{k+1} = 2T_k$$
$$E_{k+1} = E_k (I + E_k A_c T_k / 2) \tag{10}$$

for $k = 1, 2, 3, \ldots, j$ where

$$T_1 = \frac{T}{r} \quad \text{and} \quad E_1 = \sum_{i=0}^{N} \frac{(A_c T/r)^i}{(i+1)!} \tag{11}$$

$$\text{for } r = 2^j \text{ and } j = \left[\frac{\ln\|A_c T\|}{\ln(2)} \right]_{\substack{\text{integer}\\ \text{part}}} + 1 \tag{12}$$

The desired $E = E_{j+1}$. The series will converge satisfactorily with the value of j given in Equation (12) since $\|A_c T/r\| < 1$. Once E has been calculated, A_d can be obtained using Equation (8).

Ramp invariant (RI) Equivalent Model. This algorithm provides a robust method for the conversion from a C–T model, Equation (2), to a five matrix D–T state model [1], represented by

$$R_{dr} = \{ A_d, B_{d0}, B_{d1}, C_d, D_d \} \tag{13}$$

which, in turn, can be written as

$$\begin{aligned} x(k+1) &= A_d x(k) + B_{d0} u(k) + B_{d1} u(k+1) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned} \tag{14}$$

The matrices A_d, E, C_d and D_d have been described previously, see Equations (5), (6) and (8). To specify the remaining matrices, we define

$$F = \sum_{i=0}^{\infty} \frac{(A_c T)^i}{(i+2)!} \tag{15}$$

from which we obtain

$$B_{d0} = (E - F)\, B_c T, \quad \text{and} \quad B_{d1} = P B_{d0} \tag{16}$$

where $P = F\,(E - F)^{-1}$

It is desirable to create an algorithm which allows the condition of Equation (7) and singular A_c matrices. The development for this algorithm is given in [3] and is summarized by the following recursive process:

$$\begin{aligned} T_{k+1} &= 2T_k \\ F_{k+1} &= 0.5F_k + 0.25\,(I + F_k A_c T_k)^2 \end{aligned} \tag{17}$$

for $k = 1, 2, 3, ..., j$ where (with j as in Equation (12) and $r = 2^j$ as before)

$$T_1 = \frac{T}{r}, \quad F_1 = \sum_{i=0}^{N} \frac{(A_c T/r)^i}{(i+2)!} \tag{18}$$

and the desired $F = F_{j+1}$. Once F has been calculated, it follows that

$$E = (I + F A_c T), \quad A_d = I + E A_c T \tag{19}$$

Equation (17) may be used when either the SI or RI equivalent model is required, as well as when only the transition matrix $A_d = \exp(A_c T)$ is sought.

Equivalent Standard State Model. Since the algorithm of Equation (17) results in a non-standard five matrix model, it is useful to have a method of converting to a standard model as given in Equation (4). Specifically, we describe the transformation from Equation (14) to the following *equivalent* model:

$$\begin{aligned} x(k+1) &= A_{de}\, x(k) + B_{de}\, u(k) \\ y(k) &= C_{de}\, x(k) + D_{de}\, u(k) \end{aligned} \tag{20}$$

The simplest computational procedure for converting to a standard state model is derived using the identity of transfer function matrices, i.e.

$$C_d\,(zI - A_d)^{-1}\,(B_{d0} + zB_{d1}) + D_d = C_{de}\,(zI - A_{de})^{-1}\,B_{de} + D_{de} \tag{21}$$

The development is presented in [3].

2.2. Continualization Procedures

The reverse process of converting from a D–T model to an *equivalent* C–T model will now be considered, i.e. converting between the model in Equation (4) and the model in Equation (2), $R_d \rightarrow R_c$ in the SI case, or between Equation (14) and Equation (2), $R_{dr} \rightarrow R_c$ in the RI sense.

SI to Continuous–Time Model. The algorithms for continualization require *logarithmic* operations instead of matrix exponentiation. When $(A_d - I)$ or A_c is non-singular, it is easily concluded that the matrices of R_c in Equation (2) may be obtained from:

$$A_c = \frac{1}{T}\ln(A_d),\;\; B_c = (A_d - I)^{-1}\,A_c B_d \tag{22}$$

with the understanding that $C_c = C_d$ and $D_c = D_d$ as before. An outline of the method is given in the following, details can be found in [3]. In a manner similar to the series definition of the exponential function in Equation (5), the Taylor series expansion for the function $\ln(x)$ in the neighborhood of $x = 1$ leads to

$$A_c = \frac{1}{T}\sum_{i=1}^{\infty}\frac{(A_d - I)^i}{i}(-1)^{(i+1)} \tag{23}$$

The problem of using a truncated version of Equation (23) is that for matrices A_d with

$$|\,\lambda_{\max}\,| > 0.5 \tag{24}$$

where $\lambda_{\max}$ is the maximum magnitude eigenvalue of $(A_d - I)$, the series may require large N, leading to considerable round-off errors if it converges at all. This algorithm resolves this problem [5], by using the following basic property of the logarithm function.

$$\ln(x) = r\ln\left[(x)^{1/r}\right] = -r\sum_{i=1}^{\infty}\frac{(1 - x^{1/r})^i}{i} \tag{25}$$

With this approach the truncated series for calculation becomes

$$A_c = -\frac{r}{T}\sum_{i=1}^{N}\frac{(I - A_d{}^{1/r})^i}{i} \tag{26}$$

where the integer j satisfies that

$$|\lambda(A_d^{1/r} - I)|_{\max} < 0.5, \quad \text{with } r = 2^j \tag{27}$$

It has been experimentally verified that the accuracy of using Equation (26) is satisfactory even for matrices A_d where some eigenvalues of $L = A_d - I$ have magnitude greater than one.

Having determined A_c, the remaining matrices in the C–T equivalent state space model of Equation (2) could be calculated using the matrix E, appearing in Equation (8) using the procedure given in Equations (10)–(12). It follows that $C_c = C_d$, $D_c = D_d$ and

$$B_c = \frac{1}{T} E^{-1} B_d \tag{28}$$

RI to Continuous-Time Model. It is easily determined that the C–T model in Equation (2) can be obtained from the five matrix D–T model in Equation (14), or Equations (17)–(19), by using the logarithm algorithm to calculate A_c and from the availability of F in Equation (15), i.e. Equations (17)–(19), solving Equation (16) for B_c,

$$B_c = \frac{1}{T} F^{-1} B_{d1} = \frac{1}{T} (E - F)^{-1} B_{d0} \tag{29}$$

with $B_{d1} = PB_{d0}$, where $P = F(E - F)^{-1}$

The required five matrix D–T model in Equation (14), containing B_{d0} and B_{d1}, could be obtained from a standard four matrix D–T model as in Equation (20) by applying the conversion from a four-to a five-matrix model [3].

3. NUMERICAL EXAMPLES

Two examples are presented in this section. They were selected to illustrate the computational accuracy that can be achieved using the exponential and the logarithmic matrix calculations discussed previously. The first example demonstrates convergence rates when calculating A_d given a 5×5 singular, non-diagonalizable matrix A_c, followed by a similar development in the second example in calculating A_c given A_d.

3.1. Example 1: Discretization

For this example a matrix A_c with eigenvalues was used

$$\lambda(A_c) = \{\, 0, -1, -1, -1 + j1, -1 - j1 \,\} \tag{30}$$

Note that A_c is singular and has multiple eigenvalues. In addition, the Jordan form, A_j, corresponding to A_c was not diagonal. The desired sampling interval for the discretization is T=2 sec.; and the (Frobenius) norm of A_cT was calculated to be 15.65. The matrix A_d was determined from Equation (19) using the matrix F calculated from Equations (17)–(19). Equations (5) and (8) combined provide the following truncated summation for calculating the exponential matrix.

$$A_d = \left[\sum_{i=0}^{N} \frac{(A_c T / r)^i}{i!} \right]^r \tag{31}$$

As before, $r = 2^j$ where j is given in Equation (12). Both the truncation number N and the scaling parameter j are of key interest to this development. Several combinations of N and j were used to calculated $A_d(N,j)$. Each $A_d(N,j)$ is compared to a numerically exact matrix A_d calculated by first reducing A_c to its Jordan form to find $\exp(A_c T)$.

The log10 of the norm of the error matrix $E_d = A_d - A_d(N,j)$ was calculated for each combination of N and j. It was seen that N=16 terms are sufficient for A_d in Equation (31) even for matrices $A_c T$ with relatively high norms. And N may be chosen as low as N=6 with judicious choice of the parameter j, e.g. better than 7-place accuracy was achieved using N=6 and j=5, and for the same j, better than 14-place accuracy was obtained with N=10.

3.2. Example 2: Continualization

In this example the matrix A_d was taken to be the exact A_d given in Example 1. The calculation used to determine A_c is the truncated series in Equation (26). The eigenvalues $\lambda(L)$ were given by

$$\{ 0, -0.86, -0.86, -1.06 + j0.12, -1.06 - j0.12 \}$$

Evaluation of A_c was done for several combinations of the parameters N and j. The matrix $A_c(N,j)$ was found to be accurate even when the maximum eigenvalue of L is greater than unity. It is also noted that the truncation may be as low as N=10.

4. CONCLUSIONS

A newly developed set of numerically robust algorithms has been presented. These algorithms deal with the often encountered problems of *discretization* of C–T models as well as the inverse problem of *continualization*, recreating a C–T model from a given D–T model. The algorithms described in the paper comprise, in addition to the standard *Step Invariant (SI or ZOH)* procedures, a method which is referred to as the *Ramp Invariant (RI)* method, representing a piecewise linear approximation to the input functions. With these algorithms the design engineer can operate easily between the continuous and discrete time domains.

REFERENCES

[1] Bingulac,S., and Cooper,D., "Derivation of discrete- and continuous-time ramp invariant representations", *Electronics Letters* 26/10 (1990) 664–666.

[2] Bingulac,S., and Sinha,N.K., "On the identification of continuous-time multivariable systems", *Math. Comput. Modeling* 14 (1990) 203–208.

[3] Bingulac,S., and VanLandingham,H.F., *Algorithms for Computer-Aided Design of Multivariable Control Systems*, Marcel Dekker, New York, 1993.

[4] Brogan,W.L., *Modern Control Theory*, Prentice-Hall Pub. Co., Englewood Cliffs, NJ, 1984.

[5] Cooper,D., and Bingulac,S., "Computational improvement in the calculation of the natural log of a square matrix", *Electronics Letters* 26/13 (1990) 861-862.

[6] Golub,G., and Van Loan,C.F., *Matrix Computations*, John-Hopkins University Press, Baltimore, MD, 1991.

[7] VanLandingham,H.F., *Introduction to Digital Control Systems*, Macmillan Pub. Co., New York, 1985.