Using Integrals of the State Transition Matrix for Efficient Transient-Response Computations

Arthur J. Grunwald*
Technion—Israel Institute of Technology
Haifa, Israel

Introduction

ONBOARD digital flight control systems usually require a fast and memory-efficient computation of the real-time response of linear systems. Typical examples are discrete filters and state estimators, digital control laws, predictive computations, and so on. Frequently, the computation of the state transition matrix is required in obtaining the real-time system response. Since the computational power of onboard systems is limited, techniques requiring minimal computation time and Central Processing Unit (CPU) storage space are essential.

Reference 1 discusses 19 different methods for computing the state transition matrix. The closed-form solutions,[2-4] though exact, have the disadvantage that the eigenvalues and eigenvectors of the system matrix have to be computed first. Methods based on numerical integration of ordinary differential equations are inefficient because they do not take advantage of the linear, time-invariant nature of the problem. The polynomial methods[5] have the disadvantage that the characteristic polynomial must be known and are also relatively inefficient for low-order systems. Methods based on truncated Taylor series[6-8] or Pade approximations[9] have the distinct advantage that no a priori knowledge of the eigenvalues is required. The Taylor series method is simple to implement and therefore, for relatively low-order systems, very suitable for onboard computers.

Integrals of the state transition matrix can also be computed by a series expansion method, but the result may become highly inaccurate for large transition intervals. An effective technique is to subdivide the interval into sufficiently small segments and then apply an iterative scheme of repeated squaring.[10-12]

In contrast to the sophisticated and universal schemes mentioned above, the technique in this paper applies to specific systems of relatively low order which have one or more poles at the origin (e.g., the lateral dynamics of an aircraft), or to systems where the input is a polynomial function of the time t. These systems require integration of either the input or the output signals. Usually, these integrations are included in the system model, thus raising the system order. Since transition matrix computations by series methods are based on matrix multiplications and additions, the number of arithmetic operations increases approximately with the third power of the system order. In the technique applied in this paper, the integrations are not included in the system, instead, integrals of the state transition matrix are used in computing the required integrated signals. Since the system is of reduced order, considerable computational effort is saved.

Technique Description

Given the linear time-invariant system

$$\dot{x}_0(t) = A_0 x_0(t) + B_0 u(t) \tag{1}$$

Submitted Feb. 7, 1983; revision received June 8, 1983.

*Senior Lecturer, Dept. of Aeronautical Engineering.

where x_0 is the m-dimensional state vector, u the n-dimensional vector of control inputs, and A_0 and B_0 the system and input matrices of dimension $(m \times m)$ and $(m \times n)$, respectively. Three additional state vectors are defined, each of dimension m, the first, second, and third integrals of x_0. An augmented system of order $4m$ is defined as

$$\begin{bmatrix} \dot{x}_0(t) \\ \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} A_0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} B_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{2}$$

which may be written more compactly as

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3}$$

The system response at $t = t_0 + \tau$ is given by

$$x(t_0+\tau) = e^{A\tau}x(t_0) + \int_0^\tau e^{A\tau}Bu(t_0+\sigma)\,d\sigma \tag{4}$$

Assuming $u(t_0+\sigma) = u(t_0)$ over the interval $0<\sigma<\tau$, Eq. (4) can be written as

$$x(t_0+\tau) = \phi(\tau)x(t_0) + \Gamma(\tau)Bu(t_0) \tag{5}$$

where $\phi(\tau)$ is the state transition matrix and $\Gamma(\tau)$ the first integral of $\phi(\tau)$. The matrices $\phi(\tau)$ and $\Gamma(\tau)$ can be approximated by the following series expansions:

$$\phi(\tau) = I + A\tau + A^2\frac{\tau^2}{2!} + \cdots + A^n\frac{\tau^n}{n!} + \cdots \tag{6}$$

$$\Gamma(\tau) = I\tau + A\frac{\tau^2}{2!} + A^2\frac{\tau^3}{3!} + \cdots + A^{n-1}\frac{\tau^n}{n!} + \cdots \tag{7}$$

Making use of the system matrix A of Eq. (2) in Eqs. (6) and (7) yields:

$$\phi(\tau) = \begin{bmatrix} \phi_0(\tau) & 0 & 0 & 0 \\ \Gamma_0(\tau) & I & 0 & 0 \\ \Lambda_0(\tau) & I\tau & I & 0 \\ \Xi_0(\tau) & \dfrac{I\tau^2}{2!} & I\tau & I \end{bmatrix} \tag{8}$$

and

$$\Gamma(\tau) = \begin{bmatrix} \Gamma_0(\tau) & 0 & 0 & 0 \\ \Lambda_0(\tau) & I\tau & 0 & 0 \\ \Xi_0(\tau) & \dfrac{I\tau^2}{2!} & I\tau & 0 \\ \Omega_0(\tau) & \dfrac{I\tau^3}{3!} & \dfrac{I\tau^2}{2!} & I\tau \end{bmatrix} \tag{9}$$

where $\phi_0(\tau)$ is the state transition matrix of A_0 at τ, and $\Gamma_0(\tau)$, $\Lambda_0(\tau)$, $\Xi_0(\tau)$, and $\Omega_0(\tau)$ are the first, second, third, and fourth integrals of ϕ_0, respectively, which are given by the following series expansions:

$$\phi_0(\tau) \triangleq e^{A_0\tau} = I + A_0\tau + A_0^2\frac{\tau^2}{2!} + \cdots + A_0^n\frac{\tau^n}{n!} + \cdots \tag{10}$$

$$\Gamma_0(\tau) \triangleq \int_0^\tau \phi_0(s)\,\mathrm{d}s = I\tau + A_0\frac{\tau^2}{2!} + A_0^2\frac{\tau^3}{3!} + \cdots + A_0^{n-1}\frac{\tau^n}{n!} + \cdots \tag{11}$$

$$\Lambda_0(\tau) \triangleq \int_0^\tau \Gamma_0(r)\,\mathrm{d}r = I\frac{\tau^2}{2!} + A_0\frac{\tau^3}{3!} + A_0^2\frac{\tau^4}{4!} + 999 + A_0^{n-2}\frac{\tau^n}{n!} + \cdots \tag{12}$$

$$\Xi_0(\tau) \triangleq \int_0^\tau \Lambda_0(w)\,\mathrm{d}w = I\frac{\tau^3}{3!} + A_0\frac{\tau^4}{4!} + A_0^2\frac{\tau^5}{5!} + \cdots + A_0^{n-3}\frac{\tau^n}{n!} + \cdots \tag{13}$$

$$\Omega_0(\tau) \triangleq \int_0^\tau \Xi_0(v)\,\mathrm{d}v = I\frac{\tau^4}{4!} + A_0\frac{\tau^5}{5!} + A_0^2\frac{\tau^6}{6!} + \cdots + A_0^{n-4}\frac{\tau^n}{n!} + \cdots \tag{14}$$

Using Eqs. (8) and (9) in Eq. (5) yields the states x_1, x_2, and x_3 at $t = t_0 + \tau$:

$$x_1(t_0+\tau) = \Gamma_0(\tau)x_0(t_0) + \Lambda_0(\tau)Bu(t_0) + x_1(t_0) \tag{15}$$

$$x_2(t_0+\tau) = \Lambda_0(\tau)x_0(t_0) + \Xi_0(\tau)Bu(t_0) + \tau x_1(t_0) + x_2(t_0) \tag{16}$$

$$x_3(t_0+\tau) = \Xi_0(\tau)x_0(t_0) + \Omega_0(\tau)Bu(t_0) + \frac{\tau^2}{2!}x_1(t_0) + \tau x_2(t_0) + x_3(t_0) \tag{17}$$

Equations (15-17) show that the integrals of the state x_0 can be computed without raising the dimension of the system. In the following section an efficient scheme for the computation of the state transition matrix and the integral matrices is presented.

Computation of the State Transition and Integral Matrices

The scheme starts with the computation of the highest integral matrix which is required; that is, the matrix $\Xi_0(\tau)$, by means of the series expansion of Eq. (13). The expansion is continued until the norm of the nth term is sufficiently small. The state transition matrix and the integral matrices are computed by using the following expressions, which are derived from Eqs. (10-13):

$$\Lambda_0(\tau) = A_0\Xi_0(\tau) + I\frac{\tau^2}{2!} \tag{18}$$

$$\Gamma_0(\tau) = A_0\Lambda_0(\tau) + I\tau \tag{19}$$

$$\phi_0(\tau) = A_0\Gamma_0(\tau) + I \tag{20}$$

Computational accuracy and convergence properties depend on the value of τ relative to the shortest time constant of the system. Computational difficulties may arise when τ is considerably larger than the shortest time constant. This might result either in very slow convergence, or in large accumulative errors, due to the fact that the terms in the series may grow very large before they decay. In computing the state transition matrix, this problem is commonly solved by subdividing the interval τ into N sufficiently small subintervals $\Delta\tau$, according to $\Delta\tau = \tau/N$, and then using the relation

$$\phi_0(\tau) \triangleq \phi_0(N\Delta\tau) = \{\phi_0(\Delta\tau)\}^N \tag{21}$$

This relation, however, is not valid for the integrals of ϕ_0, and the correct expressions are derived hereafter.

Consider the system of Eq. (2) for which the augmented system matrix $\phi(\tau)$ is given by Eq. (8). The first column of partition of $\phi(\tau)$ contains the required state transition matrix and integral matrices. The relation given in Eq. (21) is applied to the augmented state transition matrix of Eq. (8); thus $\phi(\Delta\tau)$ is raised to the power N by $(N-1)$ repeated multiplications. The nth multiplication yields the transition matrix at $(n+1)\Delta\tau$, from which the first partition column contains the following expressions:

$$\phi_0^{n+1} = \phi_0^n\phi_0^1 \tag{22}$$

$$\Gamma_0^{n+1} = \Gamma_0^n\phi_0^1 + \Gamma_0^1 \tag{23}$$

$$\Lambda_0^{n+1} = \Lambda_0^n\phi_0^1 + n\tau\Gamma_0^1 + \Lambda_0^1 \tag{24}$$

$$\Xi_0^{n+1} = \Xi_0^n\phi_0^1 + \frac{n^2\tau^2}{2!}\Gamma_0^1 + n\tau\Lambda_0^1 + \Xi_0^1 \tag{25}$$

where the superscripts 1, n, and $n+1$ denote the values of the matrices at $\Delta\tau$, $n\Delta\tau$,and $(n+1)\Delta\tau$, respectively; for example, $\phi_0^n \triangleq \phi_0(n\Delta\tau)$, and so on.

The algorithm starts with computing Ξ_0^1, Λ_0^1, Γ_0^1, and ϕ_0^1 by means of Eqs. (13) and (18-20). Substitution of these matrices into Eqs. (22-25) yields the value of the matrices at $2\Delta\tau$. The substitution is repeated until $n = N-1$, which yields the values of the matrices at $\tau = N\Delta\tau$. The recursive form of Eqs. (22-25) is easily translated into simple computer software.

The abovementioned technique was successfully used in computing the predicted lateral response of a DC-8 aircraft.[13] In order to compute the lateral displacement of the vehicle, two integrations were required, in addition to the basic fourth-order system. The matrices ϕ_0, Γ_0, Λ_0, and Ξ_0 were computed for $\tau = 6$ s, by subdivision into six intervals of 1 s each. The expansion of Ξ_0 was continued until the 27th term, for which the norm was less than 10^{-30}. The computations were performed in floating point arithmetic at 32-bit precision and required 7,824 arithmetic operations (multiplication or addition). For comparison, the computations were performed for an augmented system of order six, in which the two integrations were included. In order to reach the same accuracy in the results, this scheme required 15,012 arithmetic operations, which is about twice as many as with the proposed method.

Computational Examples

Example 1: Consider the system of Eq. (1) with $x_0(t) \triangleq [x_1(t), x_2(t)]'$, $u(t) \triangleq u_0(t)$, and

$$A_0 = \begin{bmatrix} -0.5 & -1 \\ 1 & 0 \end{bmatrix} \tag{26a}$$

$$B_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{26b}$$

where the superscript t indicates the transpose. Two additional states are defined, the first and second integrals of x_2 according to $\dot{x}_3(t) \triangleq x_2(t)$ and $\dot{x}_4(t) \triangleq x_3(t)$. Given the initial conditions $x(0) = [0, 1, 0, 0]'$ with $u_0(t)$ being a unit step function [$u_0(t) = 0$ for $t \le 0$; $u_0(t) = 1$ for $t > 0$] the value of x_4 is required at $t = 2$ s.

The state transition matrix and the integral matrices computed according to Eqs. (13) and (18-20) at $t = 2$ s are:

$$\phi_0(2) = \begin{bmatrix} -0.363 & -0.585 \\ 0.585 & -0.071 \end{bmatrix} \tag{27a}$$

$$\Gamma_0(2) = \begin{bmatrix} 0.585 & -1.071 \\ 1.071 & 1.120 \end{bmatrix} \tag{27b}$$

$$\Lambda_0(2) = \begin{bmatrix} 1.071 & -0.880 \\ 0.880 & 1.510 \end{bmatrix} \tag{27c}$$

$$\Xi_0(2) = \begin{bmatrix} 0.880 & -0.489 \\ 0.489 & 1.124 \end{bmatrix} \tag{27d}$$

Then according to Eq. (16) the value of $x_4(2)$ is givey by

$$x_4(2) = [0.880\ 1.510]\,[0\ \ 1]' + [0.489\ 1.124]\,[2\ \ 2]' \cdot 1 = 4.738 \tag{28}$$

Example 2: For the system of the first example the value of state x_1 is required at $t = 2$ s, while the input $u(t) \triangleq u_1(t)$ is a polynomial function of t according to $u_1(t) = (c_1 + c_2 t + c_3 t^2/_2)\, u_0(t)$, where $u_0(t)$ is a unit step function. The initial conditions are set to zero, i.e., $x_0(0) = [0,0]'$. In order to include the first and second powers of t in the system, an augmented state equation is defined according to

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \hline \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \left[\begin{array}{c|cc} A_0 & c_2 B_0 & c_3 B_0 \\ \hline \mathbf{0} & \begin{matrix} 0 \\ 1 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \end{array}\right] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \hline x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} c_1 B_0 \\ \hline 1 \\ 0 \end{bmatrix} u_0(t) \tag{29}$$

Expanding the system matrix of Eq. (29) with Eqs. (6) and (7) and applying Eq. (5) yields the solution for $x(t)$, from which the first row is given by

$$x_1(t_0 + \tau) = \mathrm{row}_1\{\phi_0(\tau)x_0(t_0) + [\Gamma_0(\tau)c_1 + \Lambda_0(\tau)c_2 + \Xi_0(\tau)c_3]B_0 u(t_0)\} \tag{30}$$

With $t_0 = 0$, $\tau = 2$ s, $x_0 = [0\ 0]'$, and $c_1 = 1$, $c_2 = 2$, and $c_3 = 3$, $x_1(2)$ becomes

$$x_1(2) = \{[0.585\ -1.071]1 + [1.071\ -0.880]2 + [0.880\ -0.489]3\}[2\ 2]' \cdot 1 = 2.133 \tag{31}$$

Conclusions

The method proves efficient and sufficiently accurate for low-order systems with one or more poles at the origin. The simplicity of the method makes it very suitable for implementation in onboard computers. For higher-order systems (i.e., above order 10), the advantage of the method will be marginal, and efficiency and accuracy, in particular for systems with large differences in time constants, is not yet known.

Acknowledgments

The technique in this paper originated from a research project on predictor symbology in computer-generated pictorial flight displays, carried out in the framework of the Terminal Configured Vehicle Program at Langley Research Center, Hampton, Virginia, and sponsored by the National Aeronautics and Space Administration under Contract NASW-3302.

References

[1] Moler, C.B. and Van Loan, C.F., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review*, Vol. 20, No. 4, Oct. 1978, pp. 801-836.

[2] Kirchner, R.B., "An Explicit Formula for e^{At}," *American Mathematical Monthly*, Vol. 74, 1967, pp. 1200-1204.

[3] Apostol, T.M., "Some Explicit Formulas for the Exponential Matrix e^{At}," *American Mathematical Monthly*, Vol. 76, 1969, pp. 289-292.

[4] Vidylsagar, M., "A Novel Method of Evaluating e^{At} in Closed Form," *IEEE Transactions on Automatic Control*, Vol. AC-15, Oct. 1970, pp. 600-601.

[5] Bierman, G.J., "Finite Series Solutions for the Transient Matrix and Covariance of a Time-Invariant System," *IEEE Transactions on Automatic Control*, Vol. AC-16, 1971, pp. 173-175.

[6] Liou, M.L., "A Novel Method of Evaluating Transient Response," *Proceedings of IEEE*, Vol. 54, 1966, pp. 20-23.

[7] Everling, W., "On the Evaluation of e^{At} by Power Series," *Proceedings of IEEE*, Vol. 55, March 1967, p. 413.

[8] Bickart, T.A., "Matrix Exponential: Approximation by Truncated Power Series," *Proceedings of IEEE*, Vol. 56, May 1968, pp. 872-873.

[9] Ward, R.C., "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM Journal of Numerical Analysis*, Vol. 14, 1977, pp. 600-610.

[10] Johnson, J.C. and Phillips, C.L., "An Algorithm for the Computation of the Integral of the State Transition Matrix," *IEEE Transactions on Automatic Control*, Vol. AC-16, April 1971, pp. 204-205.

[11] Armstrong, E.S. and Caglayan, A.K., "An Algorithm for the Weighting Matrices in the Sampled-Data Optimal Linear Regulator Problem," NASA TN D-8376, 1976.

[12] Van Loan, C.F., "Computing Integrals Involving the Matrix Exponential," *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 3, June 1978, pp. 395-404.

[13] Grunwald, A.J., "Predictor Symbology in Computer-Generated Pictorial Displays," Technion Aeronautical Engineering, TAE No. 470, Haifa, Israel, Nov. 1981.