Time-Discretization of Delayed Multi-Input Nonlinear System Using A new algorithm

Time-Discretization of Delayed Multi-Input Nonlinear System Using A new algorithm

장강, 장정, 김성중, 정길도
Zhang Qiang, Zheng Zhang, Sung Jung Kim , Kil To Chong

Abstract - In this paper, a new approach for a sampled-data representation of nonlinear system that has time-delayed multi-input is proposed. That is largely devoid of illconditioning and is suitable for any nonlinear problem. The new scheme is applied to nonlinear systems with two or three inputs; and then the delayed multi-input general equation is derived. The method is based on thematrix exponential theory. Itdoes not require excessive computational resources and lends itself to a short and robust piece of software that can be easily inserted into large simulation packages. A performance of the proposed method is evaluated using a nonlinear system with time-delay: maneuvering an automobile.

Key Words : Algorithms Time-Discretization; Time-delay; Multi-input; Taylor-series Matrix exponential; Nonlinear system

1. Introduction

First I will introduce some basic knowledge for Taylor-Lie series algorithm and our algorithm

Then I will compare the Taylor-Lie series way and our way with using time and accuracy.

Initially, delay-free (D = 0) nonlinear control systems are considered with a state-space representation of the form:

$$\frac{dx(t)}{dx} = f(x(t)) + g(x(t))u(t) \tag{1}$$

Under the ZOH assumption and within the sampling interval, the solution of (1) is expanded in a uniformly convergent Taylor-series as

$$\begin{aligned} x_i(k+1) &= \Phi^{N_i T}(x(k), u(k)) \\ &= x_i(k) + \sum_{l=1}^{\infty}(L_f + uL_g)^l x_i|_{(x(k),u(k))}\frac{T^l}{l!} \end{aligned} \tag{2}$$

The resulting coefficients can be easily computed by taking successive partial derivatives of the right-hand side of equation (1):

$$\begin{aligned} x_i(k+1) &\equiv x(k) + \sum_{l=1}^{\infty}\frac{T^l}{l!}\frac{d^l x}{dt^l}|_{t_k} \\ &= x(k) + \sum_{l=1}^{\infty}A^{[l]}(x(k), u(k))\frac{T^l}{l!} \end{aligned} \tag{3}$$

Where $x(k)$ is the value of the state vector x at time $t = t_k = kT$ and $A^{[l]}(x, u)$ are determined recursively by

$$\begin{aligned} A^{[l]}(x,u) &= f(x) + ug(x) \\ A^{[l+1]}(x,u) &= \frac{\partial A^{[l]}(x,u)}{\partial x}(f(x) + ug(x)) \end{aligned} \tag{4}$$

where l = 1, 2, 3⋯

The Taylor series expansion of equation (2) can offer either an exact sampled-data representation (ESDR) of (1) by retaining the full infinite series representation of the state vector

$$\begin{aligned} x(k+1) &= \Phi_T(x(k), u(k)) \\ &= x(k) + \sum_{l=1}^{\infty}A^{[l]}(x(k), u(k))\frac{T^l}{l!} \end{aligned} \tag{5}$$

or an approximate sampled-data representation (ASDR) of equation (1) resulting from a truncation of the Taylor series of order N:

$$\begin{aligned} x(k+1) &= \Phi_T^N(x(k), u(k)) \\ &= x(k) + \sum_{l=1}^{N}A^{[l]}(x(k), u(k))\frac{T^l}{l!} \end{aligned} \tag{6}$$

where the subscript of the map Φ_T^N denotes the dependence on the sampling period T of the sampled-data representation obtained under the above scheme of discretization, and the superscript N denotes the finite series truncation order associated with the ASDR of the equation given above.

Similarly the single input case can be expanded to multi-input case. The discretization method of general nonlinear system with multi-input delay is developed using Taylor series expansion. A system with only two time-delayed inputs will be considered for simplicity in this section. A time-delayed two-input nonlinear continuous-time control system can be expressed with the following state-space form.

$$\begin{aligned} \frac{dx(t)}{dt} &= f(x(t)) + u_1(t - D_1)g_1(x(t)) \\ &\quad + u_2(t - D_2)g_2(x(t)) \end{aligned} \tag{7}$$

$$u_1(t - D_1) \to (D_1 = q_1 T + \gamma_1)$$
$$u_2(t - D_2) \to (D_2 = q_2 T + \gamma_2)$$

So the inputs are as follows;

저자 소개
* 장강 : 全北大學 電子情報工學科 碩士課程
** 장정 : 全北大學 電子情報學科 博士
*** 김성중 : 全北大學 電子情報工學科 敎授・工博
**** 정길도 : 全北大學 電子情報工學科 敎授・工博

$$u_1(t-D_1)=\begin{cases} u(KT-q_1T-T)\equiv u(k-q_1-1) & \text{if } KT\le t < KT+\gamma_1 \\ u(KT-q_1T)\equiv u(k-q_1) & \text{if } KT+\gamma_1 \le t < KT+T \end{cases} \quad (8)$$

$$u_2(t-D_2)=\begin{cases} u(KT-q_2T-T)\equiv u(k-q_2-1) & \text{if } KT\le t < KT+\gamma_2 \\ u(KT-q_2T)\equiv u(k-q_2) & \text{if } KT+\gamma_2 \le t < KT+T \end{cases}$$

Below is another way to solve this kind of problems with matrix computation algorithm.

Anautonomous nonlinear ordinary (vector) differential equation for the time function $\underline{X}(t)$ in explicit form can be denoted as

$$\dot{\underline{x}}=\underline{f}(\underline{x}) \quad (9)$$

with the initial condition $\underline{X}(t_0) = \underline{X}_0$, where the underlined symbols denote column vectors of length n.

Any nonautonomous equation $\dot{\underline{X}}=\underline{f}(\underline{X},t)$ can always be put in autonomous form by simply adding an auxiliary variable.

Let x(t) be the (continuous and piecewise smooth) approximation to the solution, defined below. For k1, 2, 3 . . . , let t_k, t_{k+1} be the initial and final times of step k + 1 of the procedure $h_{k+1}=t_{k+1}-t_k$, be the time step size, and let $x_0=X_0$ and $\underline{x}_k=\underline{x}(t_k)$ for k > 0.

We integrate a linear form of our differential equation along step k + 1 as follows. We denote the n n Jacobian matrix of $\underline{f}(\underline{x})$ by $J(\underline{x})$. For the sake of simplicity, $f_k=f(x_k)$ and $J_k=J(x_k)$. Consider the linearization $f_k+J_k\cdot(\underline{x}-\underline{x}_k)$ of $\underline{f}(\underline{x})$ around $\underline{x}_k$. The approximate solution $\underline{x}_t$ along step k + 1, i.e., from $t=t_k$ to $t=t_{k+1}$, is the solution of the differential equation

$$\frac{d\underline{x}(t)}{dt}=f_k+J_k\bullet(\underline{x}(t)-\underline{x_k}) \quad (10)$$

with the initial condition $\underline{x}(t_k) = \underline{x}_k$. The analytical solution of the differential equation within step k + 1 is

$$\underline{x}=\tau\bullet\phi(\tau\bullet J_k)\bullet\underline{f_k}+\underline{x_k} \quad (11)$$

where = t - t_k *and (G) G^{-1} (e^G - 1). The initial point of the following step will therefore be* $\underline{x}_{k+1} = \underline{x}(t_{k+1})$, *which is the final point of the current step.*

We denote

$$\zeta_k(\tau)=\underline{x}(t)-\underline{x_k} \quad (12)$$

and combine this equation with the last equation,

$$\underline{x}(t)=\tau\bullet\phi(\tau\bullet J_k)\bullet\underline{f_k}+\underline{x_k} \quad (13)$$

We can then form a new equation:

$$\zeta(\tau)=J^{-1}\bullet(e^{\tau\bullet J}-I)\bullet\underline{f} \quad (14)$$

where I is the n n identity matrix.

To avoid analytical or numerical annoyances when J is singular or nearly singular, we consider an augmented problem that avoids the matrix inversion altogether and is therefore simpler and more robust. We define the augmented (n + 1)-dimensional vectors

$$\eta\equiv(\frac{\zeta}{1}) \qquad \eta_0\equiv(\frac{0}{1}) \quad (15)$$

Then we can get an augmented (n + 1) (n + 1) matrix

$$A=\begin{pmatrix} J & \underline{f} \\ \underline{0}^T & 0 \end{pmatrix} \quad (16)$$

where the n-vectors $\underline{\xi}$, $\underline{f}$ and θ are considered column vectors, and the superscript T indicates transposition. Apart from an added zero eigenvalue, A has the same eigenvalue spectrum as J. The above nonhomogeneous differential equation becomes a homogeneous equation:

$$\frac{d\underline{\eta}}{d\tau}=A\bullet\underline{\eta} \quad (17)$$

with initial condition $\underline{\eta}(0)=\underline{\eta_0}$. The exact solution of the augmented problem is

$$\underline{\eta}(\tau)=e^{\tau A}\underline{\eta_0} \quad (18)$$

which only requires the computation of a matrix exponential and is also valid if J is singular.

Due to the definitions of and 0, the vector will be given by the last column of e - A, while the vector ξ will be given by the first n elements of . Therefore, at the end of step k + 1, the value of the solution x(t) provided by our method is

$$\underline{x_{k+1}}=\underline{x_k}+\underline{\zeta_k}(h_{k+1}) \quad (19)$$

where the n-vector $\zeta(h_{k+1})$consists of the first n elements of

$$\underline{\eta}(h_{k+1})=e^{h_{k+1}A}\eta_0 \quad (20)$$

Now we only need to focus on the details of the computation of $e^{h\cdot A}$ .We will describe a simple method to compute the exponential of a matrix. Let Z be a square matrix and I the corresponding identity matrix. The exact formula is

$$e^Z=\lim_{N\to\infty}(I+\frac{Z}{N})^N \quad (21)$$

A truncated approximation with a suitable b is

$$e^Z\cong(I+\frac{Z}{2^b})^{2^b} \quad (22)$$

This idea, which according to Knuth can be traced back to Pingala in 200 B.C., is much more economical in terms of the number of matrix multiplications.

For the nonlinear system with input control

A time-free single-input nonlinear continuous-time control system can be expressed with the following state-space form.

$$\frac{dx(t)}{dt}=f(x(t))+g(x(t))u(t-D) \quad (23)$$

Where $x(0)=x_k$, $t\in[t_k,t_{k+1}]$

Consider time interval $t\in[t_k,t_{k+1}]$ and suppose $u(t)=u_k$, $t\in[t_k,t_{k+1}]$

Denote $\xi(t)=x(t)-x_k$, $t\in[t_k,t_{k+1}]$, $x_k=x(t_k)$

Similarly the single input case can be expanded to multi-input case. The discretization method of general nonlinear system with multi-input delay is developed using matrix exponential algorithm. A system with only two time-delayed inputs will be considered for simplicity in this section. A time-delayed two-input nonlinear continuous-time control system can be expressed with the following state-space form.

$$\frac{dx(t)}{dt}=f(x(t))+u_1(t-D_1)g_1(x(t))+u_2(t-D_2)g_2(x(t)) \quad (24)$$

2. Simulation and Results

Simplified model of maneuvering an automobile

One example is considered in the computer simulations. The example is a simplified model of maneuvering an automobile(Henk Nijmeijer and Arfan van der Schaft, 1990). Exact solutionsfor the systems are required in order to validate the proposed discretization method of nonlinear systems with the delayed multi-input. In this paper the continuous Matlab ODE solver is used as an exact solution. In the simulation the discrete values obtained using the Taylor series expansion method are compared with the values obtained through the continuous Matlab ODE solver at the corresponding sampled period.

The front axle of a simplified automobile maneuvering system is shown in Fig 3. The middle of the axles linking the front wheels has position $(x_1,x_2)\in R^2$, while the rotation of this axis is given by the angle x_3. The states x_1,x_2 related with rolling are directly controlled by input u_1 and the state x_3 related with rotation is directly controlled by u_2, thus the governing nonlinear differential equation can be obtained as followings;

$$\frac{d}{dt}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\begin{pmatrix}\sin x_3\\\cos x_3\\0\end{pmatrix}u_1(t-D_1)+\begin{pmatrix}0\\0\\1\end{pmatrix}u_2(t-D_2) \qquad (25)$$

At first we choose a small sampling period and small time delay to verify the discretization method proposed in this paper. The inputs of u_1 and u_2 are assumed to be step functions respectively whose magnitudes are $u_1=1$ and $u_2=2.5$. The simulation result is shown in Table 1.

Here N of Taylor part is 1,3,5 and b of matrix part is 1,3,5.

Time step	Matlab (x1)	Taylor (x1)	Matrix(x1)	Matlab (x2)	Taylor(x2)	Matrix(x2)
200	0.1377	0.1377	0.1377	0.1414	0.1414	0.1414
400	0.3268	0.3268	0.3268	0.1997	0.1997	0.1997
600	0.5208	0.5208	0.5208	0.1603	0.1602	0.1603
800	0.6721	0.6721	0.6721	0.0326	0.0326	0.0327
1000	0.7436	0.7436	0.7436	-0.1518	-0.1518	-0.1518
1200	0.7180	0.7180	0.7180	-0.3481	-0.3481	-0.3481
1400	0.6014	0.6014	0.6014	-0.5080	-0.5080	-0.5080
1600	0.4224	0.4224	0.4224	-0.5924	-0.5924	-0.5924
1800	0.2248	0.2248	0.2248	-0.5807	-0.5807	-0.5807
2000	0.0570	0.0570	0.0570	-0.4757	-0.4757	-0.4757

Table 1 simulation results

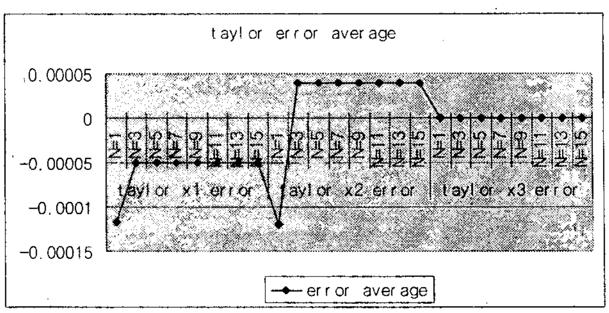

Figure 1 taylor error average

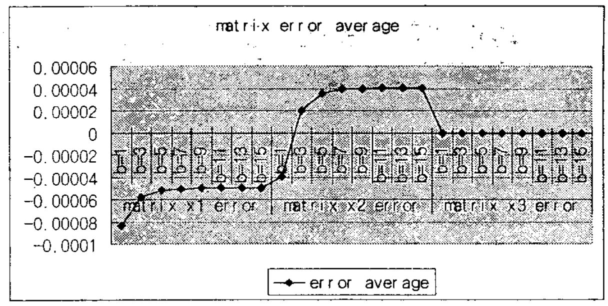

Figure 2 matrix error average

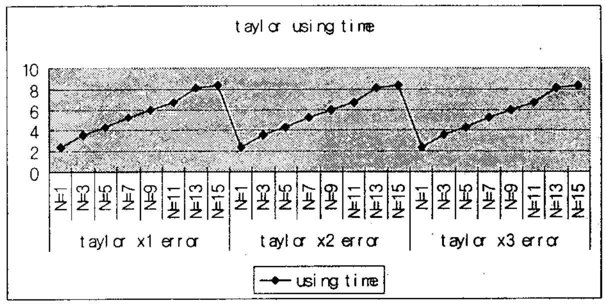

Figure 3 using time with taylor way

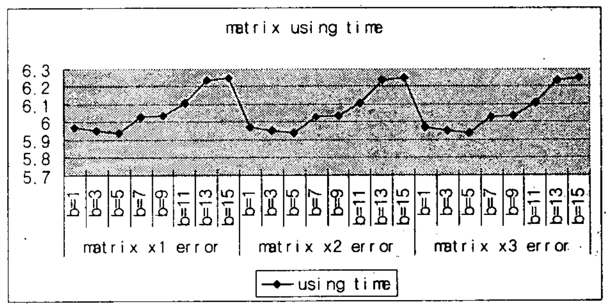

Figure 4 using time with matrix way

We got this algorithmwith many simulations and computation and proved that it was really better than the taylor way when we need more exactly results without much computing time.

참 고 문 헌

[1] G.R. Gavalas, Nonlinear Differential Equations of Chemically Reacting Systems (Springer - Verlag, Berlin, 1968).

[2] E.H. Kerner, J. Math. Phys. 22, pp. 1366 (1981).

[3] N. Kazantzis and C. Kravaris, "System-theoretic properties of sampled-data representations of nonlinear systems obtained via Taylor - Lie series," Int. J. Control, 67, pp. 997 - 1020 (1997).

[4] G.H. Golub and C.F. van Loan, Matrix Computations (Johns Hopkins University Press, Baltimore, 1996).

[5] G.F. Franklin, J.D. Powell, and M.L. Workman, Digital Control of Dynamic Systems (Addison - Wesley, New York, 1998).

[6] K.R. Harris and A. Palazoglu, "Studies on the analysis of nonlinear processes via functional expansions. I. Solution of nonlinear ODEs," Chem. Eng. Sci., 52, pp. 3183 (1997).

[7] N. Kazantzis and C. Kravaris, "Time-discretization of nonlinear control systems via Taylor methods."Comp. Chem. Engn., 23,