Difference between Numerical Derivative and Physical Derivative Realization and Its Affection to Stability Analysis

Haibo Liu [1,a], Jinhuang Wu[1], Dingxiong Zhang[1], Hong Wang[2] and Junwei Lei[2]

[1]the 91872th unit of PLA, Beijing, China

[2]Department of control engineering, Naval aeronautical and Astronautical University

Yantai, 264001

[a]liuhaibo1024@126.com, [b]wanghong1024@163.com

Abstract. A kind derivative algorithm is discussed and the difference of numerical derivative and physical derivative is analyzed. What is more important is that the first step error problem is proposed and its bad affection to the system stability is shown by using a first order system as an example. Also the reason for unstable phenomenon is analyzed by using differential method. At last, detailed numerical simulation is given to shown the rightness of the proposed main conclusion.

Introduction

The design problem of differentiator is always a difficult and hot issue in the field of control[1-4]. For linear differential, Kahlil (1994) designed a linear high gain tracking differentiator, provides a signal to derivative, but the differentiator in each layer may contain the disturbance, so the anti interference performance is poor [5]. Wang Xinhua analyzed the high order linear tracking [6] with multiple integral chain structure in 2010 and the nature and limitations of Ibrir [7] in 2004 for linear derivative tracking. This paper is based on the above reasons, the physical and digital differential difference is studied, especially the first step error of the derivative, and the error influence on the stability of the system is analyzed. Although the first-order system simple is taken as an example, but the results are of general significance and can be extended to general system.

Model Description

To make the main result more easy to understand, the below first order system is taken as an example to show the whole analysis process.

$$\dot{x} = 3x + u \tag{1}$$

And the control objective is to design a control law such that the system state x can trace the desired value x^d. But the main purpose of this paper is to show the realization of derivative algorithm and all possible problems that will cause the system unstable.

Analysis of Difference between Numerical Derivative and Physical Derivative with Different First Step Value

Define a new variable as $e = x - x^d$, then it holds:

$$\dot{e} = 3e + u + 3x^d \tag{2}$$

Design a PD control law as below:

$$u = -6e - 3\dot{e} \tag{3}$$

Then it can be substitute into the original system as

$$\dot{e} = \frac{-3}{4}e + \frac{3}{4}x^d \tag{4}$$

And according to above design, the system should be stable, but in fact the system is unstable in real simulation, which can see below figures.
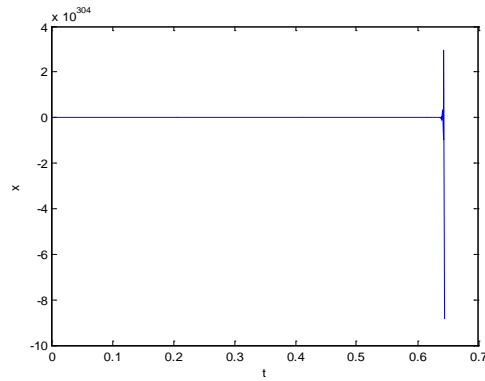
Fig.1. The unstable situation of PD controller

And the main reason is that a kind of approximate derivative algorithm is adopted in the PID control law design as bellows:
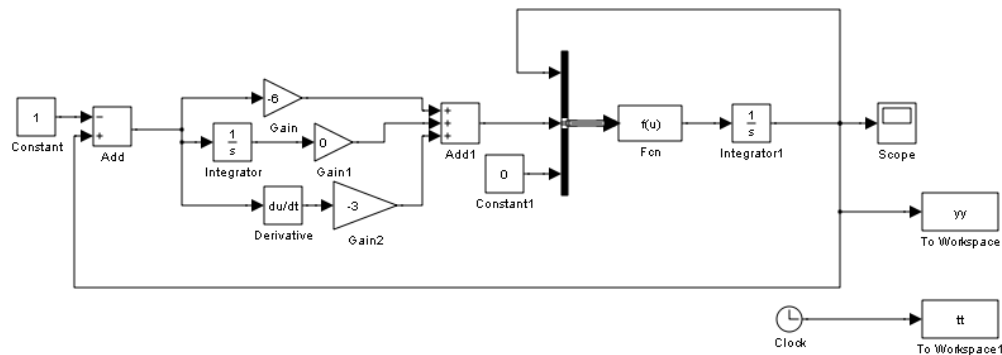

Fig.2. The approximate derivative structure with simulink

If a kind of physical derivative is adopted as below simulink diagram, then the system is stable.
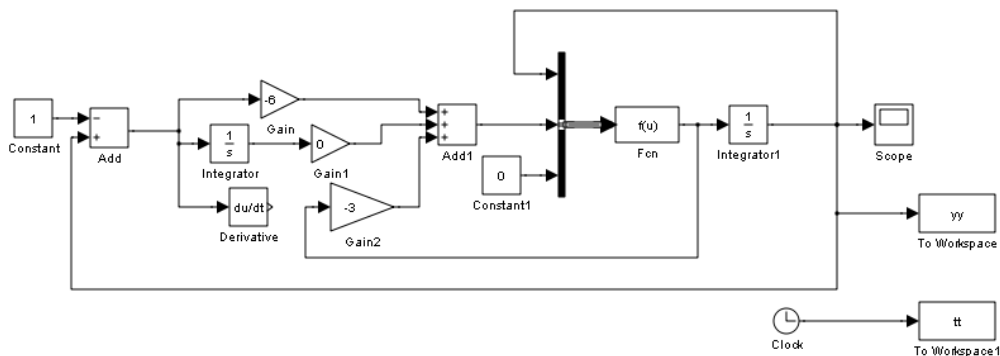

Fig.3. The physical derivative structure with simulink

Simulation result can see below figure 4. So we can make a conclusion that the main difference between numerical derivative algorithm and physical derivative algorithm is the first step, which will cause a big disturbance to the system and also will make the system unstable. But if a filter is adopt, then the control effect will be almost the same.
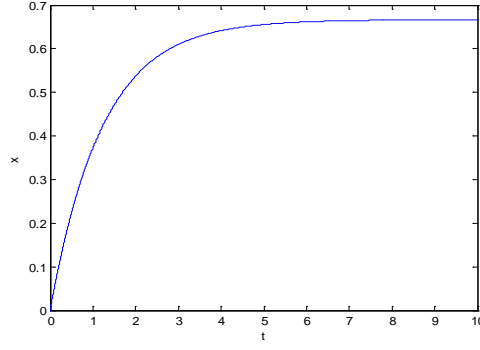
Fig.4. The stable situation of physical derivative algorithm

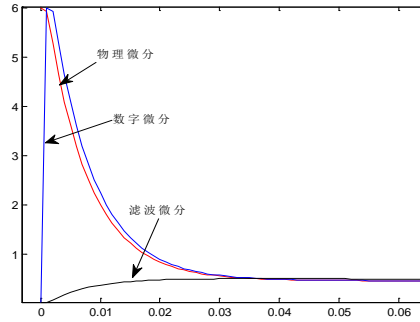And the simulation result can be shown as below figures.

Fig.5. The simulation method for three situation

Stability Analysis and Differential Coefficient Choosing

If the coefficient of derivative item is small than -1, then the system will has oscillations which will make system unstable sometimes. And the reason will be analyzed as following:

If the system is controlled with derivative algorithm then the differential equation can be written as bellows:

$$de/dt = 3e + u + 3x^d \tag{5}$$

where

$$u = -6e - 3\dot{e} \tag{6}$$

Then it holds:

$$de/dt = -3e - 3\dot{e} + 3x^d \tag{7}$$

And the above algorithm is realized in simulation program as

$$e(n) = e(n-1) + de \tag{8}$$

where

$$de = -3edt - 3\dot{e}dt + 3x^d dt \tag{9}$$

And for numerical derivative method, there holds:

$$\dot{e} = [e(n-1) - e(n-2)]/dt \tag{10}$$

Then it holds:

$$e(n) = e(n-1) - 3e(n-1)dt + ke(n-1) - ke(n-2) + 3x^d dt \tag{11}$$

so

$$e(n) = e(n-1)[1 + k - 3dt] - ke(n-2) + 3x^d dt \tag{12}$$

The above difference equation is unstable when $k < -1$.

Conclusion

The control parameter choosing for PID controller is $k_i < 0$, $k_w < 0$, $k_d < 1$. And the initial step of the numerical derivative algorithm will be greatly affect the stability of the whole system. So a filter

is very useful for this kind of approximate method to solve derivatives. But how to analyze the stability more accurately is still a big problem for the theory analysis and engineering application.

References

[1]Buschek H and Calise A J. Hypersonic flight control system design using fixed order robust controllers[A].In:AIAA Guidance,Navigation and Control Conference[C]. Chattanooga, TN, AIAA 95-6062

[2]Buschek H, Calise A J. Uncertainty modeling and fixed-order controller design for a hypersonicvehicle model. Journal of Guidance, Control and Dynamics, 1997, 20(1):42-48.

[3]Heller M and Sachs G. Flight dynamics and robust control of a hypersonic test vehicle with ramjet propulsion[A]. In:AIAA International Space Planes and Hypersonic Systems and Technologies Conference[C].Norfolk,VA,AIAA 98-1521

[4] M. Ohno, Y. Yamaguchi, T. Hata, M. Takahama, Robust flight control law design for an automatic landing flight experiment [J], Control Engineering Practice, 1999, 7(9):1143-1152

[5] H.Khalil. Robust servomechanism output feedback controller for feedback linearizable systems[J]. Automatica, 1994,30(5):1587-1599

[6] Zong Q, Wang J, Tian BL, et al. Quasi-continuous high-order sliding mode controller and observer design for flexible hypersonic vehicle. Aerosp Sci Technol 2013; 27: 127–137.

[7] S. Ibrir. Linear time-derivative trackers, Automatica. 2004,40(5):397-405