Journal of Intelligent Learning Systems and Applications, 2013, 5, 99-107
http://dx.doi.org/10.4236/jilsa.2013.52011 Published Online May 2013 (http://www.scirp.org/journal/jilsa)

Randomized Algorithm for Determining Stabilizing Parameter Regions for General Delay Control Systems*

Chao Yu, Binh-Nguyen Le, Xian Li, Qing-Guo Wang

Department of Electrical and Computer Engineering, National University of Singapore, Singapore City, Singapore.
Email: yuchao@nus.edu.sg, binhnguyen.le@nus.edu.sg, lixian@nus.edu.sg, elewqg@nus.edu.sg

Received January 29th, 2013; revised March 25th, 2013; accepted April 2nd, 2013

ABSTRACT

This paper proposes a method for determining the stabilizing parameter regions for general delay control systems based on randomized sampling. A delay control system is converted into a unified state-space form. The numerical stability condition is developed and checked for sample points in the parameter space. These points are separated into stable and unstable regions by the decision function obtained from some learning method. The proposed method is very general and applied to a much wider range of systems than the existing methods in the literature. The proposed method is illustrated with examples.

Keywords: Stabilizing Parameter Regions; Delay Control Systems; Randomized Sampling; LMI Stability Criterion; Support Vector Machines

1. Introduction

Finding stabilizing regions for control systems in parameter space becomes important in recent years. Stabilizing parameter regions will be instructive for controller tuning with greatest robustness or controller optimization with regard to other specific indexes. Most papers in the literature discuss about the stabilizing parameter regions for proportional-integral-derivative (PID) controllers. Wang *et al.* [1] designed a quasi-Linear Matrix Inequality method to compute the stabilizing parameter regions of multi-loop PID controllers, but it only dealt with systems with no time delays. Lee *et al.* [2-4] established some stability conditions by simple P or PI controllers for a class of unstable processes with time delays, but the application of their methods is confined to single-input single-output (SISO) systems whose transfer functions only have one zero. Nie *et al.* [5] gave a frequency method to calculate the loop gain margins of multivariable feedback system. Liu *et al.* [6] introduced a fast calculation approach for PI controller stable region based on D-partition method. Wang *et al.* [7] presented an effective graphical method to obtain exact P controller gain ranges for two input two output (TITO) systems with input time delay. However, this approach could not handle systems with state-delays. Some other methods can be found in [8-13]. All the methods seek the solutions for the stabilizing parameter regions for limited classes of plants or controllers.

In this paper, we design a general algorithm for determining stabilizing parameter regions for delay control systems based on randomized sampling. Each unknown parameter is assumed to follow the uniform distribution in a given range and a certain number of independent and identically distributed (i.i.d.) random sample points are generated in the parameter space based on the randomized algorithms [14]. Next, given a delay control system, we convert it into a unified state-space form. Efficient LMI stability criterion is developed for a control system with multiple delays in both input and state. Then each point in the parameter space is checked by the developed stability criterion. After that, these points are separated into stable and unstable regions by the decision function obtained from some learning method. The effectiveness of the proposed method is illustrated by simulation examples.

The rest of this paper is organized as follows. Section 2 presents the idea of proposed method. Section 3 develops the stability criterion. Determining stabilizing parameter regions is discussed in Section 4. Section 5 gives

*This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

simulation examples and Section 6 concludes the paper.

2. The Proposed Method

We consider a unity feedback control system as shown in **Figure 1**. The plant may have some unknown parameters that may affect the system stability and the parameters of the controller are also needed to be designed. Hence, knowing stabilizing parameter regions is instructive for robustness analysis and design. Some methods [2-4] can give analytical solutions for stabilizing parameter regions, but these methods usually have many constraints and could only be applied to limited plants or controllers. Some numerical methods [11,12] also have some restrictions on system structures and their algorithms might be difficult to be implemented. The objective of this paper is to provide stabilizing parameter regions with a new approach which is totally different from the existing methods in this specific area. We illustrate the idea of our method with a simple example.

We consider the model in [10] as follows,

$$G(s)=\frac{s^3+4s^2-s+1}{s^5+2s^4+32s^3+14s^2-4s+50},$$

with a PI controller

$$C(s)=K_p+\frac{K_i}{s},$$

where K_p and K_i are unknown parameters. With the method in [10], the stabilizing parameter region is shown in **Figure 2(a)**.

The randomized algorithms have been applied to design robust controllers [14]. In their context, a fixed single controller is obtained for an uncertain plant. The uncertainty lies in some plant parameters. These parameters are sampled randomly to get a set of plants which represent and replace the original uncertain plant. One single controller (fixed controller parameters) is found to meet a performance measure such as H_∞ for these sampled points. In our context, we want to find the entire regions of controller parameters which stabilize a plant. Furthermore, the plant may also have some uncertain parameters such as delay. In the later case, we want to find the regions of combined parameter vector p from the controller and the plant which stabilize the control system. We employ the idea of randomized sampling. Suppose that each unknown parameter follows the uniform distribution in a given range, that is $K_p\in[-10,15]$, $K_i\in[10,40]$ and they distribute uniformly in their respective range. Then a certain number of i.i.d. random points are sampled in the parameter space. Repeated samples are omitted. According to the randomized algorithms [14], the number of points, N, should satisfy

$$N\ge\frac{1}{2\varepsilon^2}\ln\frac{2}{\delta}, \tag{1}$$

where we set a priori $\varepsilon\in(0,1)$ as the accuracy parameter and $\delta\in(0,1)$ as the confidence level. Both ε and δ are usually taken small values, say less than 0.1. We choose $\varepsilon=0.02$ and $\delta=0.05$ for our example. It could be calculated from (1) that $N\ge4611$ and then we choose $N=5000$. Throughout this paper, $N=5000$ is used for all simulation cases.

Next, we check whether each of these points could stabilize the system by some stability criterion. The characteristic equation of the closed-loop system is

$$s^6+2s^5+(K_p+3)s^4+(4K_p+K_i+14)s^3$$
$$+(4K_i-K_p-4)s^2+(K_p-K_i+50)s+K_i=0.$$

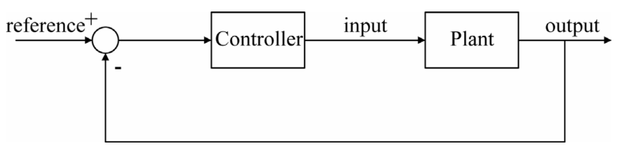
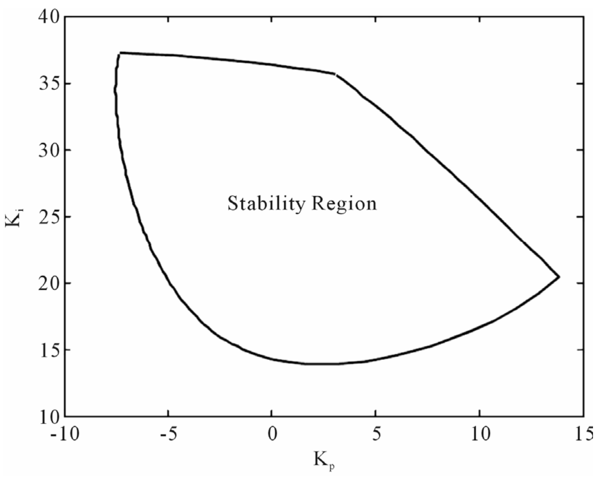
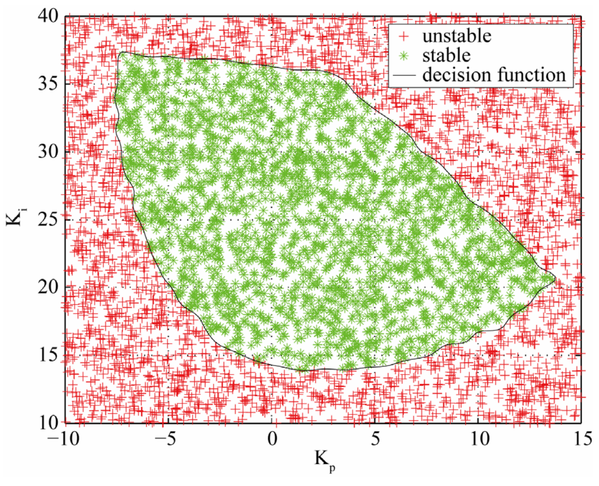


Figure 1. Unity feedback control system.



(a)



(b)

Figure 2. Stabilizing parameter region for the example in [10]. (a) Result in [10]; (b) Result with the proposed method.

We can simply calculate the closed-loop poles for stability testing. If a point of $\left[K_p, K_i\right]$ could stabilize the system, it is labeled as "stable". Otherwise, if a point could not stabilize the system, it is labeled as "unstable". However, calculating the closed-loop poles is not possible for systems with time delays. In this case, we present a Linear Matrix Inequality (LMI) stability criterion which will be discussed in next section.

Lastly, the points in the parameter space are divided into stable and unstable regions by the decision function obtained from some learning method, such as the Neural Networks and the Support Vector Machines (SVM) [15]. We choose SVM as the classification tool and employ the LibSVM [16] kit with its arguments "-t" = 2 (Radial Basis Function (RBF) as kernel) and "-c" = 1,000,000 (penalty parameter) to solve the problem. The resulting stabilizing parameter region is shown in **Figure 2(b)**. It is seen from **Figures 2(a)** and **(b)** that the stable region from the proposed method is almost same as that in [10]. Hence, our method is effective and straightforward.

3. Stability Criterion

As stated in previous section, it is impossible to calculate the closed-loop poles for systems with time delays. Therefore, in this section, we present an effective algorithm for stability testing which can be applied to a much wider range of systems. Given a delay system with PI or PID controller, we first convert it into a unified state-space form, which is a generalization of the method in [17] where a delay-free system is considered. Next, we present a conversion of delay systems with general dynamic controllers. Lastly, we present an LMI stability criterion for the unified state-space form.

3.1. PI Control for Input-Delay Plant

Consider a plant:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t-d), \\ y(t) = Cx(t), \end{cases} \tag{2}$$

with a PI controller:

$$u(t) = F_1 y(t) + F_2 \int_0^t y(\tau)\mathrm{d}\tau.$$

Let

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \int_0^t y(t)\mathrm{d}\tau \end{bmatrix},$$

so that

$$z(t-d) = \begin{bmatrix} z_1(t-d) \\ z_2(t-d) \end{bmatrix} = \begin{bmatrix} x(t-d) \\ \int_0^{t-d} y(\tau)\mathrm{d}\tau \end{bmatrix}.$$

The vector $z(t)$ can be viewed as a new state variable of the system, whose dynamics is governed by

$$\dot{z}(t) = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} z(t) + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t-d), \tag{3}$$

where

$$\begin{aligned} u(t) &= F_1 C z_1(t) + F_2 z_2(t) \\ &= F_1 \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + F_2 \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}. \end{aligned} \tag{4}$$

Let $\bar{C}_1 = \begin{bmatrix} C & 0 \end{bmatrix}$ and $\bar{C}_2 = \begin{bmatrix} 0 & I \end{bmatrix}$. Equation (4) can be rewritten as

$$u(t) = \left(F_1\bar{C}_1 + F_2\bar{C}_2\right) z(t),$$

or

$$u(t-d) = \left(F_1\bar{C}_1 + F_2\bar{C}_2\right) z(t-d). \tag{5}$$

Substituting (5) into (3) yields

$$\dot{z}(t) = \tilde{A}z(t) + \tilde{A}_1 z(t-d), \tag{6}$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix},$$

and

$$\tilde{A}_1 = \begin{bmatrix} B \\ 0 \end{bmatrix} \left(F_1\bar{C}_1 + F_2\bar{C}_2\right).$$

When (2) is with a PID controller

$$u(t) = F_1 y(t) + F_2 \int_0^t y(\tau)\mathrm{d}\tau + F_3 \frac{\mathrm{d}y(t)}{\mathrm{d}t},$$

the conversion could not be proceeded. This is because $u(t)$ depends on $u(t-d)$ since $\frac{\mathrm{d}y(t)}{\mathrm{d}t} = CAx(t) + CBu(t-d)$. Then the control signal cannot be expressed only by state vectors as (4) or (5). In such a case, we could use a practical D controller:

$$\frac{s}{1 + \frac{s}{N_d}},$$

where N_d is chosen by users to limit derivative gain on higher frequencies. Then, the practical PID controller falls in a format of general dynamic controller, which is handled in Section 3.3 below.

3.2. PID Control for State-Delay Plant

Consider a plant:

$$\begin{cases} \dot{x}(t) = Ax(t) + A_1 x(t-d) + Bu(t), \\ y(t) = Cx(t), \end{cases} \tag{7}$$

with a PID controller:

$$u(t) = F_1 y(t) + F_2 \int_0^t y(\tau)\,\mathrm{d}\tau + F_3 \frac{\mathrm{d}y(t)}{\mathrm{d}t}.$$

Let $z_1(t) = x(t)$ and $z_2(t) = \int_0^t y(\tau)\,\mathrm{d}\tau$. We have

$$\dot{z}_1(t) = \dot{x}(t) = Az_1(t) + A_1 z_1(t-d) + Bu(t),$$

and

$$\dot{z}_2(t) = y(t) = Cz_1(t).$$

Denoting $z(t) = \left[z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t)\right]^{\mathrm{T}}$, we have

$$\dot{z}(t) = \bar{A}z(t) + \bar{A}_1 z(t-d) + \bar{B}u(t),$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}, \bar{A}_1 = \begin{bmatrix} A_1 & 0 \\ 0 & 0 \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

Combining (7) and the definition of z yields

$$y(t) = Cz_1(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix},$$

$$\int_0^t y(\tau)\,\mathrm{d}\tau = z_2(t) = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix},$$

and

$$\begin{aligned} \frac{\mathrm{d}y(t)}{\mathrm{d}t} &= C\dot{x}(t) = CAx(t) + CA_1 x(t-d) + CBu(t) \\ &= [CA, 0]z(t) + [CA_1, 0]z(t-d) + CBu(t). \end{aligned}$$

Denoting $\bar{C}_1 = \begin{bmatrix} C & 0 \end{bmatrix}$, $\bar{C}_2 = \begin{bmatrix} 0 & I \end{bmatrix}$, $\bar{C}_3 = \begin{bmatrix} CA & 0 \end{bmatrix}$, $C_d = \begin{bmatrix} CA_1 & 0 \end{bmatrix}$, $\bar{y}_1(t) = \bar{C}_1 z(t)$, $\bar{y}_2(t) = \bar{C}_2 z(t)$ and $\bar{y}_3(t) = \bar{C}_3 z(t) + C_d z(t-d)$, we have

$$u(t) = F_1 \bar{y}_1(t) + F_2 \bar{y}_2(t) + F_3 \bar{y}_3(t) + F_3 CBu(t).$$

Suppose that $(I - F_3 CB)$ is invertible. Let $\bar{y}(t) = \left[\bar{y}_1^{\mathrm{T}}(t), \bar{y}_2^{\mathrm{T}}(t), \bar{y}_3^{\mathrm{T}}(t)\right]^{\mathrm{T}}$, $\bar{C} = \left[\bar{C}_1^{\mathrm{T}}, \bar{C}_2^{\mathrm{T}}, \bar{C}_3^{\mathrm{T}}\right]^{\mathrm{T}}$, $\bar{C}_d = \left[0, 0, C_d^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\bar{F} = \left[\bar{F}_1, \bar{F}_2, \bar{F}_3\right]$, where

$$\bar{F}_1 = (I - F_3 CB)^{-1} F_1,$$

$$\bar{F}_2 = (I - F_3 CB)^{-1} F_2,$$

$$\bar{F}_3 = (I - F_3 CB)^{-1} F_3.$$

Then (7) is equivalent to

$$\begin{cases} \dot{z}(t) = \bar{A}z(t) + \bar{A}_1 z(t-d) + \bar{B}u(t), \\ \bar{y}(t) = \bar{C}z(t) + \bar{C}_d z(t-d), \end{cases}$$

with

$$u(t) = \bar{F}\bar{y}(t),$$

i.e.,

$$\begin{aligned} \dot{z}(t) &= \bar{A}z(t) + \bar{A}_1 z(t-d) + \bar{B}\bar{F}\bar{C}z(t) + \bar{B}\bar{F}\bar{C}_d z(t-d) \\ &= \left(\bar{A} + \bar{B}\bar{F}\bar{C}\right)z(t) + \left(\bar{A}_1 + \bar{B}\bar{F}\bar{C}_d\right)z(t-d), \end{aligned} \tag{8}$$

which is also in the form of (6) with $\tilde{A} = \left(\bar{A} + \bar{B}\bar{F}\bar{C}\right)$ and $\tilde{A}_1 = \left(\bar{A}_1 + \bar{B}\bar{F}\bar{C}_d\right)$.

Remark 1. The systems (2) and (7) only contain one time delay. However, it would not be difficult to make conversion for systems with multiple time delays, which is omitted here for brevity.

The previous two cases only tackle delay systems with PI or PID controller whose parameters appear in a linear form. In practical control systems, the controllers may be of higher orders and the parameters of controllers may also appear in a nonlinear form, such as the lead-lag compensators [18]. Thus, we consider the conversion for delay systems with general dynamic controller as follows.

3.3. General Dynamic Controller for a Plant with Multiple Delays in Input and State

Consider a plant (9)

$$\begin{cases} \dot{x}(t) = Ax(t) + A_1 x(t-d_1) + A_2 x(t-d_2) \\ \qquad + \cdots + A_h x(t-d_h) + Bu(t) + B_1 u(t-d_{h+1}) \\ \qquad + B_2 u(t-d_{h+2}) + \cdots + B_l u(t-d_{h+l}), \\ y(t) = Cx(t). \end{cases} \tag{9}$$

under the following dynamic controller:

$$C(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1}s + b_m}{s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n},$$

whose minimal state-space realization can be expressed by

$$\begin{cases} \dot{v}(t) = A_c v(t) + B_c y(t), \\ u(t) = C_c v(t) + D_c y(t). \end{cases}$$

Let $z_1(t) = x(t)$ and $z_2(t) = v(t)$. Denoting $z(t) = \left[z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t)\right]^{\mathrm{T}}$, we have

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix},$$

and

$$z(t-d_i) = \begin{bmatrix} z_1(t-d_i) \\ z_2(t-d_i) \end{bmatrix} = \begin{bmatrix} x(t-d_i) \\ v(t-d_i) \end{bmatrix}.$$

Combining the above expressions gives (10)

$$\begin{aligned}\dot{z}_1(t) = {} & Az_1(t) + A_1 z_1(t-d_1) + \cdots + A_h z_1(t-d_h) \\ & + BD_cCz_1(t) + BC_c z_2(t) + B_1 D_c C z_1(t-d_{h+1}) \\ & + B_1 C_c z_2(t-d_{h+1}) + \cdots + B_l D_c C z_1(t-d_{h+l}) \\ & + B_l C_c z_2(t-d_{h+l}).\end{aligned} \tag{10}$$

and

$$\dot{z}_2(t) = B_c C z_1(t) + A_c z_2(t),$$

i.e.,

$$\dot{z}(t) = \tilde{A}z(t) + \sum_{i=1}^{k} \tilde{A}_i z(t-d_i), \tag{11}$$

where

$$\tilde{A} = \begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{bmatrix},$$

$$\tilde{A}_i = \begin{cases} \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, & \text{for } 0 < i \le h, \\ \begin{bmatrix} B_{i-h}D_cC & B_{i-h}C_c \\ 0 & 0 \end{bmatrix}, & \text{for } h < i \le k, \end{cases}$$

and $k = h + l$.

Remark 2. The system (6) is a special case of (11).

3.4. The LMI Stability Criterion for a System with Multiple Delays in Input and State

Theorem 1. *The system* (11) *is asymptotically stable if there exist symmetric positive definite matrices* $P, Q_1, \cdots, Q_k$, *and* $W_1, \cdots, W_k$, *such that*

$$\begin{bmatrix} \Omega & \Psi \\ * & \Lambda \end{bmatrix} < 0, \tag{12}$$

where (13) holds,

$$\Omega = \begin{bmatrix} \tilde{A}^{\mathrm{T}}P + P\tilde{A} + \sum_{i=1}^{k} Q_i - \sum_{i=1}^{k} W_i & P\tilde{A}_1 + W_1 & P\tilde{A}_2 + W_2 & \cdots & P\tilde{A}_k + W_k \\ * & -Q_1 - W_1 & 0 & \cdots & 0 \\ * & * & -Q_2 - W_2 & \ddots & \vdots \\ * & * & * & \ddots & 0 \\ * & * & * & * & -Q_k - W_k \end{bmatrix}. \tag{13}$$

$$\Psi = \begin{bmatrix} d_1\tilde{A}^{\mathrm{T}}W_1 & \cdots & d_k\tilde{A}^{\mathrm{T}}W_k \\ d_1\tilde{A}_1^{\mathrm{T}}W_1 & \cdots & d_k\tilde{A}_1^{\mathrm{T}}W_k \\ \vdots & \ddots & \vdots \\ d_1\tilde{A}_k^{\mathrm{T}}W_1 & \cdots & d_k\tilde{A}_k^{\mathrm{T}}W_k \end{bmatrix}, \text{ and } \Lambda = \begin{bmatrix} -W_1 & 0 & \cdots & 0 \\ * & -W_2 & \ddots & \vdots \\ * & * & \ddots & 0 \\ * & * & * & -W_k \end{bmatrix}.$$

Here and in the sequel, a block induced by symmetry is denoted by an ellipsis *.

Proof. Define the Lyapunov functional as

$$V(z(t)) = z^{\mathrm{T}}(t)Pz(t) + \sum_{i=1}^{k}\left(\int_{t-d_i}^{t} z^{\mathrm{T}}(s)Q_i z(s)\mathrm{d}s\right) + \sum_{i=1}^{k}\left(d_i \int_{-d_i}^{0}\int_{t+\alpha}^{t} z^{\mathrm{T}}(s)W_i z(s)\mathrm{d}s\mathrm{d}\alpha\right).$$

The derivative of $V(z(t))$ is

$$\begin{aligned}\dot{V}(z(t)) = {} & z^{\mathrm{T}}(t)P\dot{z}(t) + \dot{z}^{\mathrm{T}}(t)Pz(t) + \sum_{i=1}^{k}\left(z^{\mathrm{T}}(t)Q_i z(t)\right) - \sum_{i=1}^{k}\left(z^{\mathrm{T}}(t-d_i)Q_i z(t-d_i)\right) \\ & + \sum_{i=1}^{k}\left(d_i^2 \dot{z}^{\mathrm{T}}(t)W_i\dot{z}(t)\right) - \sum_{i=1}^{k}\left(d_i \int_{t-d_i}^{t} \dot{z}^{\mathrm{T}} s W_i \dot{z}(s)\mathrm{d}s\right).\end{aligned}$$

It follows from Jensen's inequality [19] that

$$-d_i \int_{t-d_i}^{t} \dot{z}^{\mathrm{T}}(s)W_i\dot{z}(s)\mathrm{d}s \le -\left[z(t) - z(t-d_i)\right]^{\mathrm{T}} W_i \left[z(t) - z(t-d_i)\right].$$

Then we have (14).

$$\begin{aligned}\dot{V}(z(t)) \le\ & z^{\mathrm{T}}(t)P\left[\tilde{A}z(t)+\sum_{i=1}^{k}\left(\tilde{A}_i z(t-d_i)\right)\right]+\left[\tilde{A}z(t)+\sum_{i=1}^{k}\left(\tilde{A}_i z(t-d_i)\right)\right]^{\mathrm{T}} Pz(t)\\ &+\sum_{i=1}^{k}\left(z^{\mathrm{T}}(t)Q_i z(t)\right)-\sum_{i=1}^{k}\left(z^{\mathrm{T}}(t-d_i)Q_i z(t-d_i)\right)\\ &+\sum_{i=1}^{k}\left\{d_i^2\left[\tilde{A}z(t)+\sum_{i=1}^{k}\left(\tilde{A}_i z(t-d_i)\right)\right]^{\mathrm{T}} W_i\left[\tilde{A}z(t)+\sum_{i=1}^{k}\left(\tilde{A}_i z(t-d_i)\right)\right]\right\}\\ &-\sum_{i=1}^{k}\left\{\left[z(t)-z(t-d_i)\right]^{\mathrm{T}} W_i\left[z(t)-z(t-d_i)\right]\right\}.\end{aligned} \tag{14}$$

Let

$$w(t)=\left[z^{\mathrm{T}}(t)\quad z^{\mathrm{T}}(t-d_1)\quad \cdots \quad z^{\mathrm{T}}(t-d_k)\right]^{\mathrm{T}},$$

and

$$\Gamma=\left[\tilde{A}\quad \tilde{A}_1\quad \cdots \quad \tilde{A}_k\right].$$

One sees

$$\dot{V}(z(t)) \le w^{\mathrm{T}}(t)\left[\Omega+\sum_{i=1}^{k}\left(d_i^2\Gamma^{\mathrm{T}}W_i\Gamma\right)\right]w(t).$$

By Schur complement, (12) guarantees

$$\left[\Omega+\sum_{i=1}^{k}\left(d_i^2\Gamma^{\mathrm{T}}W_i\Gamma\right)\right]<0.$$

Therefore, the system (11) is asymptotically stable.

4. Stabilizing Parameter Regions

Each point in the parameter space corresponds to a sample of the parameter vector *p*, which is denoted by p_i, $i=1,\cdots,N$. We check whether each of these points could stabilize the system by the developed LMI stability criterion. If a point p_i could stabilize the system, it is labeled as "stable". Otherwise, if p_i could not stabilize the system, it is labeled as "unstable".

The points in the parameter space can be separated into stable and unstable regions by the decision function obtained from some learning method. In this paper, we choose SVM as the learning method due to its superior performance in a wide range of applications. Support Vector Machines (SVM), which was first introduced by Vapnik [20], has shown many attractive features in the fields of small sample, non-linear and high dimensional pattern recognition [21]. It can be promoted to classification and regression problems. It employs the Structural Risk Minimization principle [21]. The goal of SVM is to find a decision function that minimizes the structural risk, which could be converted into a quadratic programming problem. In addition, the solution of an SVM problem is a globally optimal solution [22].

In this paper, SVM is employed to solve a binary classification problem. Given the data set $\boldsymbol{S}=\{S_1,S_2,\cdots,S_N\}$ with $S_i=(p_i,y_i), i=1,2,\cdots,N,$ where p_i is a point in the parameter space and $y_i=1$ (stable) or −1 (unstable) is the label of the point, SVM is to solve the following problem:

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(p_i)^{\mathrm{T}}\phi(p_j),$$

$$\text{subject to}\sum_{i=1}^{N}\alpha_i y_i=0, 0\le\alpha_i\le C,$$

where α is the Lagrange multiplier, $C>0$ is the penalty parameter which can be set by users and $\phi(\cdot)$ is a mapping from p_i to a higher dimensional space.

There have already been many SVM tool kits that can be used to solve the classification problems. LIBSVM [16] is a simple and effective one developed by Chih-Jen Lin's research group. Throughout this paper, the LibSVM kit is employed to do simulation with proper arguments.

5. Simulation Examples

In this section, four examples are presented to illustrate the effectiveness of the proposed method.

Example 1. The analytical method in [2] cannot deal with a process containing multiple zeros, while our method does not have this constraint. Consider the plant:

$$G(s)=\frac{(0.4s+1)(0.2s+1)}{(s-1)(0.5s+1)(0.1s+1)}\mathrm{e}^{-ds},$$

with a *P* controller $C(s)=kI_2$. This control system is converted to the form in (11) with

$$\tilde{A}=\begin{bmatrix}-11 & -8 & 20\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix},$$

$$\tilde{A}_1=\begin{bmatrix}-1.6k & -12k & -20k\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}.$$

Let $p=[d,k]$. Performing our method with the LibSVM arguments "-t" = 2 and "-c" = 100, the stabilizing parameter region is obtained and shown in **Figure 3**.

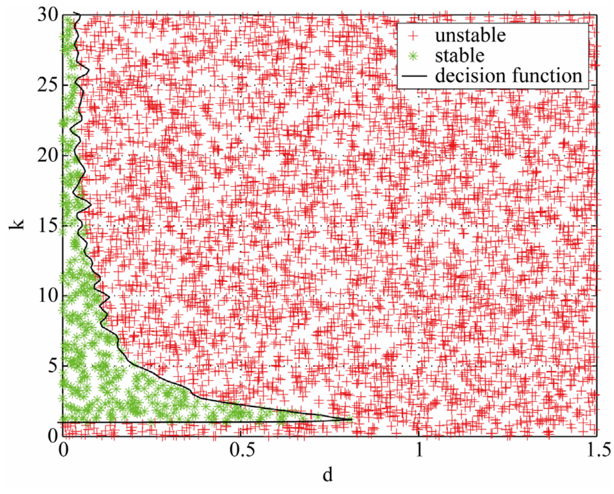


Figure 3. Stabilizing parameter region for Example 1.

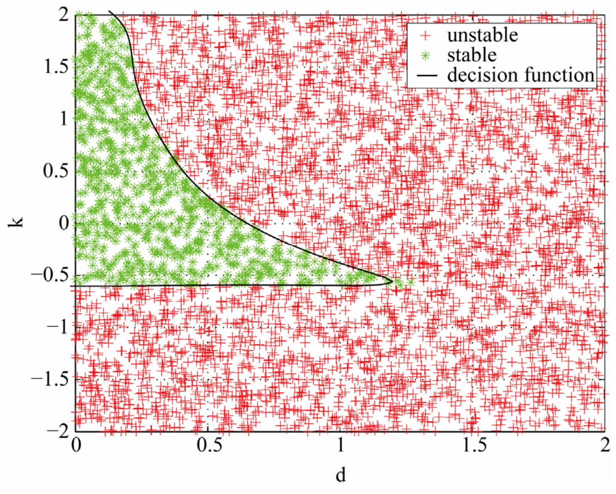


Figure 4. Stabilizing parameter region for Example 2.

Example 2. The graphical method in [7] cannot deal with a process containing state-delays. However, our method does not have this restriction. Consider the plant:

$$\begin{cases} \dot{x}(t)=\begin{bmatrix}-12.5 & -25\\ 1 & 0\end{bmatrix}x(t) \\ \qquad +\begin{bmatrix}0 & 10\\ 1.5 & 0\end{bmatrix}x(t-d)+\begin{bmatrix}1\\0\end{bmatrix}u(t), \\ y(t)=\begin{bmatrix}0 & 25\end{bmatrix}x(t), \end{cases} \qquad (15)$$

with a P controller $u=-ky$. This control system is converted to the form in (11) with

$$\tilde{A}=\begin{bmatrix}-12.5 & -25-25k\\ 1 & 0\end{bmatrix},$$

$$\tilde{A}_1=\begin{bmatrix}0 & 10\\ 1.5 & 0\end{bmatrix}.$$

Let $p=[d,k]$. Performing our method with "-t" = 2 and "-c" = 1000, the stabilizing parameter region is obtained and shown in **Figure 4**.

Example 3. Consider the plant (15) with $d=0.5$ under the controller

$$C(s)=\frac{a}{s+b}. \qquad (16)$$

Note that b appears in a nonlinear fashion, which is different from parameters of PID controllers. We can rewrite (16) as

$$\begin{cases} \dot{v}(t)=-bv(t)+y(t), \\ u(t)=av(t). \end{cases}$$

This control system is converted to the form in (11) with

$$\tilde{A}=\begin{bmatrix}-12.5 & -25 & a\\ 1 & 0 & 0\\ 0 & 25 & -b\end{bmatrix}, \tilde{A}_1=\begin{bmatrix}0 & 10 & 0\\ 1.5 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}.$$

Let $p=[b,a]$. Performing our method with "-t" = 2 and "-c" = 1000, the stabilizing parameter region is obtained and shown in **Figure 5**.

Example 4. The proposed method also works well with a high-dimensional parameter space. Consider the plant:

$$\begin{cases} \dot{x}(t)=\begin{bmatrix}-12.5 & -25\\ 1 & 0\end{bmatrix}x(t) \\ \qquad +\begin{bmatrix}0 & 10\\ 1.5 & 0\end{bmatrix}x(t-d_1)+\begin{bmatrix}1\\0\end{bmatrix}u(t-d_2), \\ y(t)=\begin{bmatrix}0 & 25\end{bmatrix}x(t), \end{cases}$$

with a controller:

$$\begin{cases} \dot{v}(t)=-bv(t)+y(t), \\ u(t)=v(t). \end{cases}$$

This control system is converted to the form in (11) with

$$\tilde{A}=\begin{bmatrix}-12.5 & -25 & 0\\ 1 & 0 & 0\\ 0 & 25 & -b\end{bmatrix}, \tilde{A}_1=\begin{bmatrix}0 & 10 & 0\\ 1.5 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

$$\text{and } \tilde{A}_2=\begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}.$$

Let $p=[d_1,d_2,b]$. Performing our method with "-t" = 2 and "-c" = 1000, the stabilizing parameter region is obtained and shown in **Figure 6**.

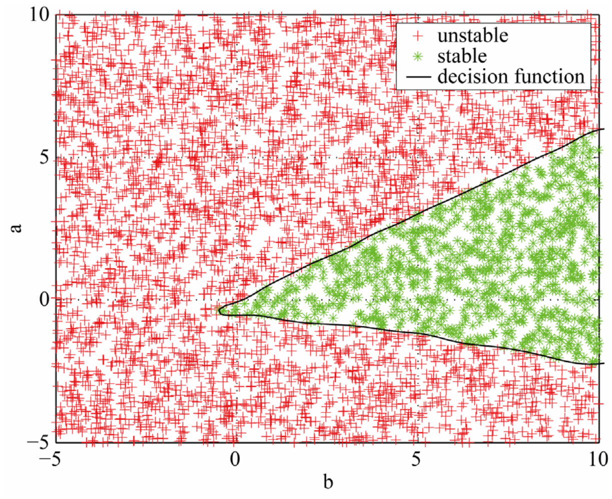


Figure 5. Stabilizing parameter region for Example 3.

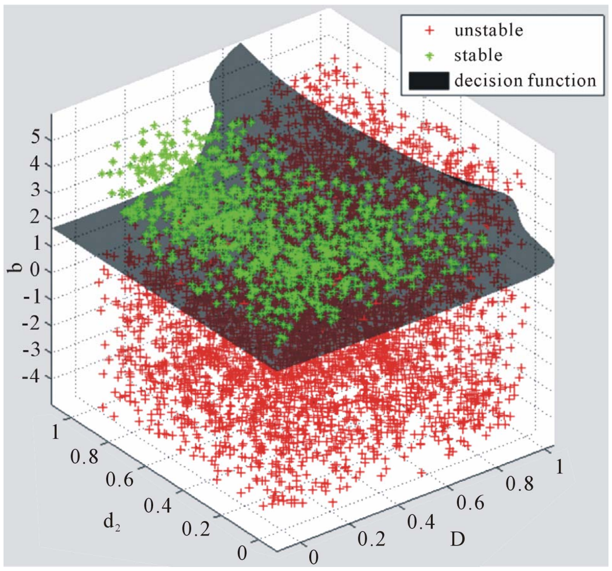


Figure 6. Stabilizing parameter region for Example 4.

The above examples have well illustrated the effectiveness of the proposed method which can be applied to a much wider range of systems than the existing methods in the literature.

6. Conclusions

This paper proposes a new and general method for determining the stabilizing parameter regions for delay control systems. We first take a certain number of random sample points in the parameter space. Next, we represent a delay control system in a unified state-space form. Then the numerical stability condition is developed and checked for sample points in the parameter space. These points are divided into two classes according to whether they can stabilize the system. The stabilizing parameter regions could be well defined by the decision function obtained from some learning method. The effectiveness of the proposed method is well illustrated with examples. The proposed method does not have essential constraints and has a wide range of applications. Note that our method could be applied to a higher-dimensional parameter space, though the stabilizing parameter regions are difficult to be shown by graphics.

It should be pointed out that the presented LMI stability criterion is only sufficient since it is based on Lyapunov theory. A sufficient and necessary stability criterion and the additional potential values of the proposed method are to be investigated in future works.

REFERENCES

[1] Q. Wang, C. Lin, Z. Ye, G. Wen, Y. He and C. Hang, "A Quasilmi Approach to Computing Stabilizing Parameter Ranges of Multi-Loop Pid Controllers," *Journal of Process Control*, Vol. 17, No. 1, 2007, pp. 59-72. doi:10.1016/j.jprocont.2006.08.006

[2] S. Lee and Q. Wang, "Stabilization Conditions for a Class of Unstable Delay Processes of Higher Order," *Journal of the Taiwan Institute of Chemical Engineers*, Vol. 41, No. 4, 2010, pp. 440-445. doi:10.1016/j.jtice.2010.03.001

[3] S. Lee, Q. Wang and C. Xiang, "Stabilization of All-Pole Unstable Delay Processes by Simple Controllers," *Journal of Process Control*, Vol. 20, No. 2, 2010, pp. 235-239. doi:10.1016/j.jprocont.2009.05.005

[4] S. Lee, Q. Wang and L. Nguyen, "Stabilizing Control for a Class of Delay Unstable Processes," *ISA transactions*, Vol. 49, No. 3, 2010, pp. 318-325. doi:10.1016/j.isatra.2010.03.006

[5] Z. Nie, Q. Wang, M. Wu and Y. He, "Exact Computation of Loop Gain Margins of Multivariable Feedback Systems," *Journal of Process Control*, Vol. 20, No. 6, 2010, pp. 762-768. doi:10.1016/j.jprocont.2010.04.006

[6] L. Jinggong, X. Yali and L. Donghai, "Calculation of Pi Controller Stable Region Based on d-Partition Method," 2010 *International Conference on Control Automation and Systems* (*ICCAS*), Gyeonggi-do, 27-30 October 2010, pp. 2185-2189.

[7] Q. Wang, B. Le and T. Lee, "Graphical Methods for Computation of Stabilizing Gain Ranges for Tito Systems," 2011 9*th IEEE International Conference on Control and Automation* (*ICCA*), Santiago, 19-21 December 2011, pp. 82-87.

[8] Q. Wang, Y. He, Z. Ye, C. Lin and C. Hang, "On Loop Phase Margins of Multivariable Control Systems," *Journal of Process Control*, Vol. 18, No. 2, 2008, pp. 202-211. doi:10.1016/j.jprocont.2007.06.004

[9] M. S¨oylemez, N. Munro and H. Baki, "Fast Calculation of Stabilizing Pid Controllers," *Automatica*, Vol. 39, No. 1, 2003, pp. 121-126. doi:10.1016/S0005-1098(02)00180-2

[10] N. Tan, I. Kaya, C. Yeroglu and D. Atherton, "Computation of Stabilizing Pi and Pid Controllers Using the Stability Boundary Locus," *Energy Conversion and Man-*

agement, Vol. 47, No. 18, 2006, pp. 3045-3058. doi:10.1016/j.enconman.2006.03.022

[11] B. Fang, "Computation of Stabilizing Pid Gain Regions Based on the Inverse Nyquist Plot," *Journal of Process Control*, Vol. 20, No. 10, 2010, pp. 1183-1187. doi:10.1016/j.jprocont.2010.07.004

[12] E. Gryazina and B. Polyak, "Stability Regions in the Parameter Space: D-Decomposition Revisited," *Automatica*, Vol. 42, No. 1, 2006, pp. 13-26. doi:10.1016/j.automatica.2005.08.010

[13] K. Saadaoui, S. Testouri and M. Benrejeb, "Robust Stabilizing First-Order Controllers for a Class of Time Delay Systems," *ISA transactions*, Vol. 49, No. 3, 2010, pp. 277-282. doi:10.1016/j.isatra.2010.02.001

[14] G. Calafiore, F. Dabbene and R. Tempo, "Research on Probabilistic Methods for Control System Design," *Automatica*, Vol. 47, No. 7, 2011, pp. 1279-1293. doi:10.1016/j.automatica.2011.02.029

[15] T. Hastie, R. Tibshirani and J. H. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer, New York, 2009. doi:10.1007/978-0-387-84858-7

[16] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology* (*TIST*), Vol. 2, No. 3, 2011, p. 27. doi:10.1145/1961189.1961199

[17] F. Zheng, Q. Wang and T. Lee, "On the Design of Multivariable PID Controllers via LMI Approach," *Automatica*, Vol. 38, No. 3, 2002, pp. 517-526. doi:10.1016/S0005-1098(01)00237-0

[18] G. Franklin, J. Powell, A. Emami-Naeini and J. Powell, "Feedback Control of Dynamic Systems," , Vol. 3, Addison-Wesley, Reading, 1994.

[19] K. Gu, V. Kharitonov and J. Chen, "Stability of Time-Delay Systems," Birkhauser, Boston, 2003. doi:10.1007/978-1-4612-0039-0

[20] V. Vapnik, "The Nature of Statistical Learning Theory," 1995.

[21] S. R. Gunn, "Support Vector Machines for Classification and Regression," *ISIS Technical Report*, Vol. 14, 1998.

[22] P. H. Chen, C. J. Lin and B. Schölkopf, "A Tutorial on ν-Support Vector Machines," *Applied Stochastic Models in Business and Industry*, Vol. 21, No. 2, 2005, pp. 111-136.