Sensors & Transducers, Vol. 21, Special Issue, May 2013, pp. 218-222

Sensors & Transducers

Multifunctional Simulation Instrument for Control Systems Based on MATLAB GUI

[1] **XING XUENING**
[1] School of Electrical and Electronic Engineering, Shandong University of Technology, 255049, China
[1] E-mail: xingxuening@yeah.net

Received: 12 April 2013 /Accepted: 14 May 2013 /Published: 30 May 2013

Abstract: In the automatic control principle course teaching, the experiment is a kind of important teaching methods used to help students understand and grasp what they learned in class. MATLAB GUI is used to design a kind of multifunctional control system simulation. When input the zero pole parameters of transfer function, and the open loop transfer function of the corresponding control system can be got the bode diagram and Nyquist diagram, the root locus diagram and the unit step response curve. The simulator can make up for the deficiency of the traditional laboratory, moreover it can increase the students' interest in learning about the course. Students are the biggest beneficiaries. *Copyright © 2013 IFSA.*

Keywords: MATLAB GUI, Simulation instrument, Automatic control, Transfer function, Root locus.

1. Introduction [1-3]

Principle of automatic control technology is the study of automatic control theory, which is one of the main courses of professional. In the automatic control principle of the course, we must learn the control system's basic theory, the basic concept and the analysis and the design method. In the automatic control principle course teaching, the experiment is a kind of important teaching methods used to help students understand and grasp what they learned in class. In the traditional hardware experimental platform, you must change or adjust the corresponding components in the experimental stage if you want to change the parameters of the system and structure more often and sometimes it is difficult to achieve due to various reasons. In recent years, with the wide application of MATLAB in automatic control theory and the use of MATLAB to assist teaching, which has played a good role in the process of teaching. But simply use MATLAB to the simulation needs certain programming basis. With the emergence of virtual instrument technology and the development of computer technology, using the virtual experiment system developed by MATLAB GUI can do virtual experiments.

GUIDE from the Graphical User Interface Development Environment, it is short for MATLAB Graphical User Interface Development Environment, it can provide the GUI with a series of tools, which greatly simplifies the design and the process of creating a GUI. Using GUI editor to edit needs to editing respectively two files: A FIG file (FIG), containing the attribute set of GUI objects and layout information; the other one is M file. (M), containing the control GUI objects implement the callback function.

In this paper, a control system analysis and design of simulation software based on MATLAB GUI platform was designed, combined with control system basic theory and MATLAB control system toolbox. And the bode diagram, the step response

Article number P_SI_369

curve, Nyquist diagram and the root locus diagram according to the given transfer function can be plotted.

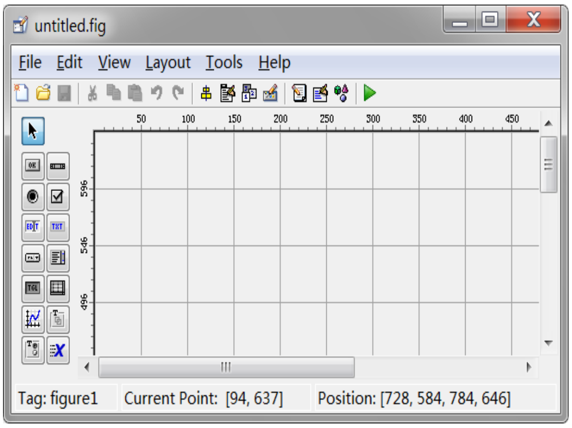

Fig. 1. GUI editor interface.

2. Classical Control Theory and Simulation [4-6]

System's transfer function is the basis of the classical control theory, mainly studies the analysis and design problems of single input and single output of the linear time-invariant systems with frequency domain method .In constant value system and servo system, the theory has more extensive application, lying a solid foundation for the further study of the intelligent control theory and the modern control theory [4-6].

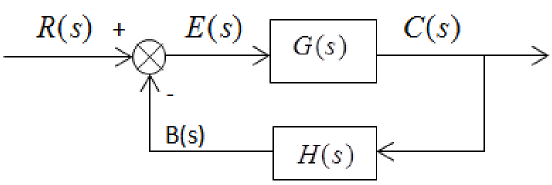

Fig. 2. Typical structure of control system.

2.1. Transfer Function and Simulation

Transfer function is a mathematical model for describing the dynamic characteristics of linear systems, but is only applicable to linear time-invariant system and initial conditions are zero.

A linear time-invariant system consists of the following n order linear ordinary differential equations.

$$a_0\frac{d^n}{dt^n}y(t)+a_1\frac{d^{n-1}}{dt^{n-1}}y(t)+\cdots+a_{n-1}\frac{d}{dt}y(t)+a_ny(t)$$
$$=b_0\frac{d^m}{dt^m}u(t)+b_1\frac{d^{m-1}}{dt^{m-1}}u(t)+\cdots+b_{m-1}\frac{d}{dt}u(t)+b_mu(t) \quad (1)$$

In this type, $y(t)$ it is the output of the system; $u(t)$ is the system input; $a_i(i=0,1,2,\cdots,n)$ and $b_j(j=0,1,2,\cdots,m)$ is associated with the system structure and parameters of constant coefficient.

Set $u(t)$, $y(t)$ and each derivative are all zero when $t=0$, i.e. zero initial conditions, in the type of the Lap lace transform, respectively, set $Y(s)=\lambda[y(t)]$ and $U(s)=\lambda[u(t)]$, s available for algebraic equation.

Then, the system transfer is defined as

$$G(s)=\frac{Y(s)}{U(s)}=\frac{b_0s^m+b_1s^{m-1}+\cdots+b_{m-1}s+b_m}{a_0s^n+a_1s^{n-1}+\cdots+a_{n-1}s+a_n} \quad (2)$$

$$b_j(j=0,1,2,\cdots,m)$$
$$a_i(i=0,1,2,\cdots,n)$$

Therefore, molecules can be used in MATLAB, the denominator coefficient vectors num and den to represent the transfer function $G(s)$, realizing the function of tf(), it calls the format is as follows:

num = [b_0 , b_1 ,…, b_{m-1} , b_m]
den = [a_0 , a_1 ,…, a_{n-1} , a_n]
sys = tf (num，den)

2.2. The Unit Step Response and Simulation

For the unit step input r (t) = 1 (t), r (s) = $\frac{1}{s}$, and so

$$C(s)=\frac{1}{s(Ts+1)}=\frac{1}{s}-\frac{1}{s+\frac{1}{T}} \quad (3)$$

Therefore,

$$c(t)=c_{ss}(t)-c_t(t)=1-e^{-\frac{t}{T}} \quad (4)$$

$c_{ss}(t)$ for steady state component, $c_{ss}(t)=1$;

$c_t(t)$ As the transient component, $c_t(t)=e^{-\frac{t}{T}}$.

step(sys)
step(sys,tfinal)
step(sys,t)
step(sys1,sys2,…,t)
[n, d]=step(sys)
[n,d,x]=step(sys)

The t final as a response to a termination time variables; T for a given time variables, such as: t = 0:0.01:10; Return variable, y for the response vector, vector t is time, as the state vector x.

2.3. Bode Diagram and Simulation

The performance of the control system with time domain measurement is intuitive, but a high order control system characteristics in time domain, it is difficult to determine with the analytical method, we usually use frequency domain method to solve. Frequency domain method is a graphical method, mainly includes three kinds of methods: Bode (amplitude / phase frequency characteristic curve), Nyquist curve and Nichole diagram. Bode diagram is the logarithmic frequency characteristic curve. A known system transfer

$$H(s) = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \cdots + a_{n+1}} \qquad (5)$$

The frequency response of the system can be directly calculated

$$H(jw) = \frac{b_1(j\omega)^m + b_2(j\omega)^{m-1} + \cdots + b_{m+1}}{a_1(j\omega)^n + a_2(j\omega)^{n-1} + \cdots + a_{n+1}} \qquad (6)$$

Bode diagrams of the system is H ($j\omega$) of rendering, so it is also called the amplitude frequency and phase frequency characteristic curve. Abscissa is angular frequency w, is commonly used in logarithmic scale, namely unit is rad/s. The ordinate is the logarithmic phase frequency characteristics $\varphi(\omega)$ and the unit for Angle; Ordinate is logarithmic amplitude-frequency characteristics $L(\omega) = 20\log A(\omega)$ and the unit is decibels (dB).

Bode diagram drawing function of MATLAB for Bode(), which calls the format is as follows:

bode(sys)
bode(sys, w)
[mag, phase, w]=bode(sys)

Among them: artificial give the frequency of the input vector for the vector; mag for the returned amplitude vector; phase to return to the phase Angle of the vector. In addition, MATLAB provides a very convenient computing system function of the amplitude and phase Angle margin, which calls the format is as follows:

margin(sys)
[gm, pm, wg, wp]=margin(sys)
[gm, pm, wg, wp]=margin(mag, phase, w)

Among them: gm for the returned vector amplitude margin; pm as the phase Angle margin; wg for phase through frequency; wp for the cutoff frequency.

2.4. Nyquist Curve and the Simulation

Nyquist curve is based on open loop frequency characteristic in the complex plane on the amplitude and phase trajectory of the paint. According to the Nyquist curve of open loop, it can judge the stability of the closed-loop system.

Feedback control is necessary and sufficient conditions for system stability, Nyquist curve counterclockwise around the critical point (-1, j0) circle number P is equal to the open-loop transfer function poles book in right half of s plane, otherwise the closed-loop system instability. When the open-loop transfer function contains the poles on the imaginary axis of closed curves, should be from the right side of the semicircle around the pole. This is the Nyquist criterion.

MATLAB Nyquist frequency curve drawing function Nyquist(), which calls the format is as follows:

nyquist(sys)
nyquist(sys,w)
[re,im,w]=nyquist(sys)

2.5. Root Locus and Simulation

The basic performance of the closed loop system transient response is composed of closed-loop poles distribution in root plane more determined. The closed-loop poles are the roots of the characteristic equation. The root locus is when a variable parameter system by, the closed-loop poles are portrayed in the S plane trajectory.

Set up control system of the open loop transfer function can be expressed as:

$$G(s)H(s) = K^* \frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} = K^* \frac{\prod_{j=1}^{m}(s-z_j)}{\prod_{i=1}^{n}(s-p_i)} \qquad (7)$$

In the formula, K* as the root locus gain open-loop system; Z_j for the system open loop zero (j=1,2,... , m); P_i for the system open loop poles (i=1,2,... , n). The closed-loop characteristic equation of system is:

$$1 + G(s)H(s) = 0 \qquad (8)$$

The formula is called the root locus equation. It can be drawn when K changed from zero to infinity, continuous root locus system.

Using root locus function provided by the MATLAB can be conveniently and accurately plot the root locus of control system, and can use the root locus diagram of control system are analyzed.

Used to map the root locus of system, its call format is as follows:

rlocus(sys)
rlocus(sys,k)
r=rlocus(sys,k)
[r,k]=rlocus(sys)

3. Simulation Instrument Design [7-10]

Main parts include start, exit button, input parameters and graphic display. Which use the GUI static text in the selection area, editable text, touch button and axes. The designed simulator is only applicable to the system transfer function is zero pole models:

$$G(s) = \frac{K(s - z_1)}{(s - p_1)(s - p_2)(s - p_3)} \tag{9}$$

In the simulation instrument panel plot the Bode graph, the system step response curve, Nyquist diagram and root locus diagram at the same time, this provides great convenience for system research.

Design and Simulation of instrument is divided into design and software programming of the control panel. According to the idea of publishing on the control panel in the GUI interface, modify each text box or button, and then write the program of the multi-function control system simulation instrument in the callback function, repeated debugging and modification, until it reaches the expected goal.

3.1. Layout of GUI Objects

According to their overall conception of multifunctional control system simulation, the simulation instrument panel consists of three parts.

The first part is the parameter input, on the left side of the object and selects the Static Text zone (static text); it is often used to display other objects, such as numerical state. Drag the mouse to determine the size of the text box in the GUI layout area of the properties dialog box, double-click the text box pops up Static Text, modify the text box static text in String. If you want to enter the numbers in the text box in the corresponding parameters, you need to select the object selection in Edit Text (editable text); it allows the user to modify the text content, used for data input and display. Drag the mouse to determine the size of the text box, and then click the text box to modify its properties. Delete the contents of the String correspond to change the color to white and its interface is shown in Fig. 3.

The second part is the beginning, exit Button, choose the Push Button object selection area on the left (touch Button), drag the mouse in the GUI layout area to determine the size of the Button, double-click the Button pop-up Push Button in the properties dialog box, and amend the content in the String to start or exit.

The third part is the graphical display, select the axes (axis) on the left side of the object, drag the mouse to determine the coordinate axis size, double-click the coordinate axis modify its properties. You can use the GUI menu bar tool will these axes alignment.

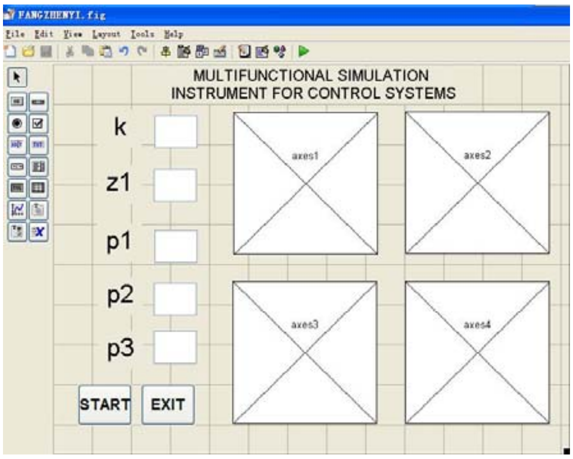

Fig. 3. Simulator interface editor.

3.2. Callback Function

Right click on the interface of Fig. 3 in the start button, select View Callbacks callback; in the popup editor page in the function pushbutton1_Callback :(hObject, eventdate, handles) enter the following procedures:

```
K =str2double(get(handles.edit1,'String'));
z1=str2double(get(handles.edit2,'String'));
p1=str2double(get(handles.edit3,'String'));
p2=str2double(get(handles.edit4,'String'));
p3=str2double(get(handles.edit5,'String'));
k=k;
z=[z1];
p=[p1,p2,p3];
y=zpk(z,p,k);
axes(handles.axes1);
bode(y)
grid on
axes(handles.axes2);
y1=feedback(y,1)
step(y1)
grid on
axes(handles.axes3);
nyquist(y)
grid on
axes(handles.axes4);
rlocus(y)
grid on
```

Right click the exit button interface as shown in Fig. 3, select the callback in the View Callbacks , in the popup editor page in the function pushbutton2_Callback (hObject, eventdate, handles) enter the close, point to save. Click the toolbar in the Run Figure, run the program, get the control panel of multifunctional control system simulation instrument.

4 The Use of the Simulator

The system model has been built, you can start the simulation. In the simulation, you should set the simulation parameters. The simulation system is designed in this paper is based on pole-zero system model design, so the zero and pole of the system should be calculated before the simulation.

Sets the transfer function:

$$G(s)=\frac{100s+300}{s^3+11s^2+10s}.$$

(1) Start MATLAB, at the MATLAB command window, type the following sequence Program:

num=[100,300];

den=[1,11,10,0]; [z,p,k]=tf2zp(num,den)

sys=zpk(z,p,k)

Press the Enter key, getting the pole zero negative feedback system

```
Zero/pole/gain:
  100 (s+3)
--------------
s (s+10) (s+1)
```

(2) Operating system, as shown in Fig. 4.

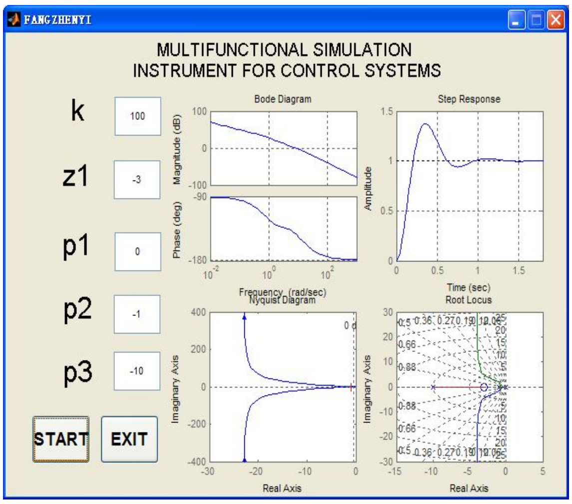

Fig.4. Test Result.

Input the pole-zero transfer function parameters in the parameter input control panel on the left side of the simulation system, we can get the corresponding control system open-loop transfer function of the Bode diagram, Nyquist diagram, and the root locus diagram and unit step response curve. Fig. 4 shows an open-loop gain of 100, zero - 3, pole respectively, 1, 0-10 to the simulation of the system. From this interface can obtain some parameters of system dynamic performance and steady-state performance, and through parameter controls the system, make the system stable, accurate and fast running. After the analysis of system press the exit button to end.

5 Conclusions

The simulator put bode diagram, unit step response curve, Nyquist diagram and the root locus diagram which are commonly used in control system in the same control panel, can more convenient to study the stability, accuracy and rapidity of the control system. Curve has the characteristics of intuitive; it is very convenient to study the steady-state performance and dynamic performance of the control system used by curve. The instrument also has some shortcomings, such as the control system to study the model is the best model of zero-pole, if not zero pole model and using MATLAB language into pole-zero model and then use the simulation instrument simulation; transfer function of control system simulation number order is relatively low, and form a fixed. Also can be used for further modification and perfection of the simulation system, it can simulate order transfer function better, but at this time of the MATLAB programming requirements will be higher.

Reference

[1]. Dub, Michal, Brotan, Lukas, MATLAB GUI for DC drive parameter identification, *Mechatronika*, 2012, 2012.

[2]. Mitchell, R. J. Using MATLAB GUIs to improve the learning of frequency response methods, *Control*, 2012, 2012.

[3]. Elmagroud, Bachir, Educational software for wave propagation phenomena using MATLAB GUIs and FDTD method, *EDUCON 2012*, 2012.

[4]. Li Suling, Automatic Control Theory (3rd Edition) *Machinery Industry Press*, 2012.

[5]. Zhang Defeng, MATLAB and automatic control system design, *Machinery Industry Press*, 2010.

[6]. Zhang Jing, Application of MATLAB in control systems, *Publishing House of Electronics Industry*, 2007.

[7]. Chen Chao, MATLAB application examples succinctly -- image processing and GUI design, *Publishing House of Electronics Industry*, 2011.

[8]. Huang Wenmei, Yang Yong, System simulation analysis and Design - MATLAB application, *National University of Defense Technology Press*, 2001.

[9]. Wang Zhenglin, MATLAB/Simulink and Control System Simulation (2nd Edition), *Electronic Industry Press*, 2008.

[10]. Sun Liang, MATLAB simulation and control system, *Beijing University of Technology Press*, 2006.