Faculty of Electrical Engineering
Universiti Teknologi Malaysia

VOL. 9, NO. 2, 2007, 41-48

ELEKTRIKA

http://fke.utm.my/elektrika

Development of a Modern Control System Analysis Package Using Visual Basic Programming

Mohd Fua'ad Rahmat[1*] and **Lee Shu Khan**[2]

[1]Department of Control and Instrumentation Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
[2]Jaalaa Malaysia Sdn Bhd, Kuala Lumpur, Malaysia.

[*]Corresponding author: fuaad@fke.utm.my (Mohd Fua'ad Rahmat), Tel: 607-5535900, Fax: 607-5566272

Abstract: This paper introduces a Computer Aided Control System Design (CACSD) program that furnishes a background necessary for studying modern control theory. The program focuses on state-space analysis which performs conversion of a system from state-space representation to transfer function and via versa. Besides, system transformation on different state coordinates, time domain solution, controller design, observer design and steady state error evaluation with interactive graph response are also being emphasized in this program.

Keywords: CACSD, Modern control system, State-space analysis, Time domain.

1. INTRODUCTION

Over the past thirty years, CACSD has emerged as an indispensable tool for analyzing control system [1]. As the trend moving from lecturer to student-oriented type of teaching, exploratory learning in virtual control systems environment with interactive desktop visualization has become more common nowadays.

The "Modern Control System Analysis Package (MCSAP)" program uses the state-space approach and matrix operation as the major development tools. Modern control theory which based on state-space approach is computationally more powerful than the frequency domain technique for a high-order system [2].

This program is developed to build an interactive learning tool to help in understanding the modern control theory and visualize system behaviour in effective way using graph response.

1.1 State-space Concepts

A system can be depicted by a black box with a number of accessible terminals, as shown in Figure 1. The input terminals represent a set of input variables, u_i; the output terminals describe a set of output variables, or response, y_j [3]. The state variables, x_k, are embedded inside the box and are thus inaccessible.

If the system is describable by linear differential equation, the state equation and output equation of the system can be represented in matrix notation as:

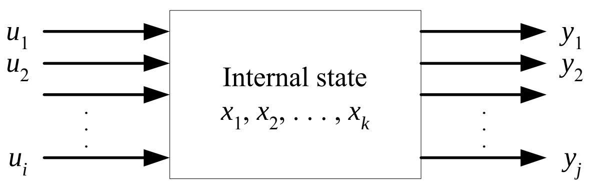

Figure 1. State variable representation of a system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ . \\ . \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & a_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & a_{nn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ x_n(t) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & . & . & b_{1r} \\ b_{21} & b_{22} & . & . & b_{2r} \\ . & . & . & . & . \\ . & . & . & . & . \\ b_{n1} & b_{n2} & . & . & b_{nr} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ . \\ . \\ u_r(t) \end{bmatrix} \tag{1}$$

$$\begin{aligned} \begin{bmatrix} y_1(t) \\ y_2(t) \\ . \\ . \\ y_m(t) \end{bmatrix} &= \begin{bmatrix} c_{11} & c_{12} & . & . & c_{1n} \\ c_{21} & c_{22} & . & . & c_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ c_{m1} & c_{m2} & . & . & c_{mn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ x_n(t) \end{bmatrix} \\ &+ \begin{bmatrix} d_{11} & d_{12} & . & . & d_{1r} \\ d_{21} & d_{22} & . & . & d_{2r} \\ . & . & . & . & . \\ . & . & . & . & . \\ d_{m1} & d_{m2} & . & . & d_{mr} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ . \\ . \\ u_r(t) \end{bmatrix} \end{aligned} \tag{2}$$

which can be simplified to the general form of state-space representation as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{Ax}(t) + \mathbf{Bu}(t) \\ \mathbf{y}(t) &= \mathbf{Cx}(t) + \mathbf{Du}(t) \end{aligned} \tag{3}$$

where $\mathbf{A}(t)$ is the system matrix, $\mathbf{B}(t)$ is the input matrix, $\mathbf{C}(t)$ is the output matrix and $\mathbf{D}(t)$ is the feed-forward matrix of the system.

2. MODERN CONTROL SYSTEM THEORY

A state-space representation consists of Equation (1), the simultaneous, first-order differential equations from which the state variables can be solved and Equation (2), the algebra output equation from which all other system variables can be found [4]. State-space representation is not unique since a different choice of state variables leads to different representation of the same system. The transformation of the similar systems is done by manipulating the transfer function, drawing signal-flow graph and then writing the state equation from signal-flow graph [5].

Similarity transformation can be done without using the transfer function and signal-flow graph but only transformation matrix $\mathbf{P}$ [6]. Transformation matrix can convert a state vector $\mathbf{x}$ in x_1x_2 plane to state vector $\mathbf{z}$ in z_1z_2 plane. To obtain the transformation matrix, eigenvalues and eigenvectors of the system have to be studied. The solution of a state-space representation can be obtained in the time domain by solving the corresponding matrix differential equation directly using transition matrix $\mathbf{\Phi}(t)$.

Normally, $\mathbf{\Phi}(t)$ can be found using Sylvester's interpolation formula. State equation can be obtained from the equation

$$\mathbf{x}(t) = \mathbf{\Phi}(t)\mathbf{x}(0) + \int_0^t \mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \qquad (4)$$

The time domain method, expressed in terms of state variables, can be utilized to design a suitable compensation scheme for a control system. Typically, the system with a control signal, $\mathbf{u}(t)$, which is a function of several measurable state variables is controlled. The controller design concept is based on achieving a desired closed-loop state equation or a desired control ratio [7].

To control the location of all closed-loop poles, state feedback method is introduced. However, state feedback design often cannot be realize directly since it requires that all elements of the state vector to be available for measurement. Therefore, it is important to find an approximation of the state variables, $\hat{\mathbf{x}}(t) \cong \mathbf{x}(t)$. This can be done by constructing another scheme, called the observer or estimator, connected to the system under consideration, whose role is to produce good estimates of the state-space variable [7]. The main purpose of designing controller and observer is to find their gains.

Controller and observer design only manage to obtain the desired characteristic of transient response but not zero steady state error [8]. Steady state error is also a significant study in modern control system. Steady state analysis via input substitution can be used for multiple-input multiple-output systems.

3. SOFTWARE IMPLEMENTATION

The "Modern Control System Analysis Package" program is developed to provide a system which minimizes engineering and programming resources for the analysis of control system. Once the program has been opened, a menu page will be shown as in Figure 2. The user may choose to perform any of the function in the list. This program contains modules for state-space analysis, transfer function analysis, transformation analysis, time domain solution, controller design, observer design and steady state error analysis.

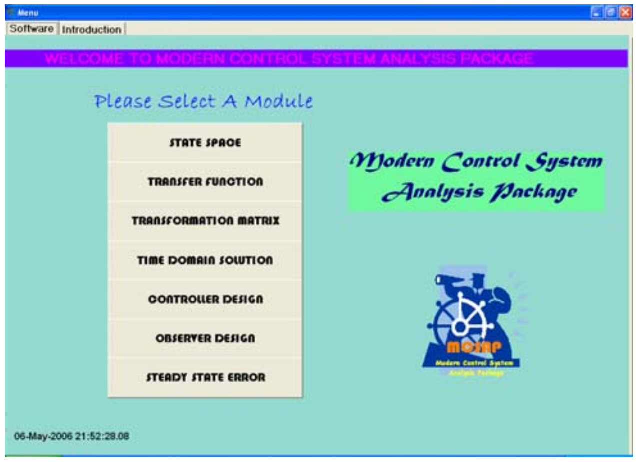

Figure 2. Program menu

State-space module includes the functions of converting a system in state-space representation to its transfer function and representing the state-space representation in phase variable form, controller canonical form, observer canonical form, diagonal form or cascade form. Transfer function module provides the function of converting a system transfer function to its state-space representation in phase variable form, controller canonical form, observer canonical form, diagonal form or cascade form. Transformation matrix module includes the functions of finding the transformed system in other state coordinate with the information on a state-space representation and transformation matrix or inverse transformation matrix; and finding transformation matrix and inverse transformation matrix with the information on two state-space representations in different state coordinate [9]. Time Domain Solution module provides the analysis of transition matrix, state equation and output equation, with interactive graph response of these three outputs. Controller design involves the functions of finding the state variable feedback gains, K_i, tests the controllability of a system, displaying the system and desired characteristic equation, and also the graph response of the controlled system. Observer design involves the functions of finding the state variable feedback gains, L_i, tests the observability of a system, displaying the system and desired characteristic equation, and graph response of designed system. Finally, steady state error includes the function of finding the steady state error with a unit step input or a unit ramp input and obtaining their corresponding graph response. All these modules also include the function of finding eigenvalues and eigenvectors of the system [10].

4. RESULTS

Figure 3 shows the State-space module on state-space to transfer function analysis. User enters the value of the system represented in state-space and gets the transfer function of the system as numerator and denominator. Figure 4 shows the Transfer Function module on transfer function to state-space analysis. User enters the value of the system transfer function and gets the state-space model system matrix, $\mathbf{A}$, input matrix, $\mathbf{B}$, output matrix, $\mathbf{C}$ and feed-forward matrix, $\mathbf{D}$.

Figures 5 and 6 show the Transformation Matrix module on analyzing the transformation matrix. User enters the value of the system with state vector **x** and inverse transformation matrix $\mathbf{P}^{-1}$ to gets the transformed system with state vector **z**, which shown in Figure 5. Eser can also enter the value of the system with state vector **x** and system with state vector **z** to gets the transformation matrix **P** and inverse transformation matrix $\mathbf{P}^{-1}$, which shown in Figure 6. Other than that, user can obtain diagonal system from the module [11]. Solution of state-space equation can be obtained in Time Domain Solution module. User enters the value of the system in state-space representation and the initial **x** value to get the transition matrix, state equation and output equation as shown in Figure 7. User can choose to analyse step input or ramp input in this module. Graph display of the three outputs also can be obtained as shown in Figure 8.

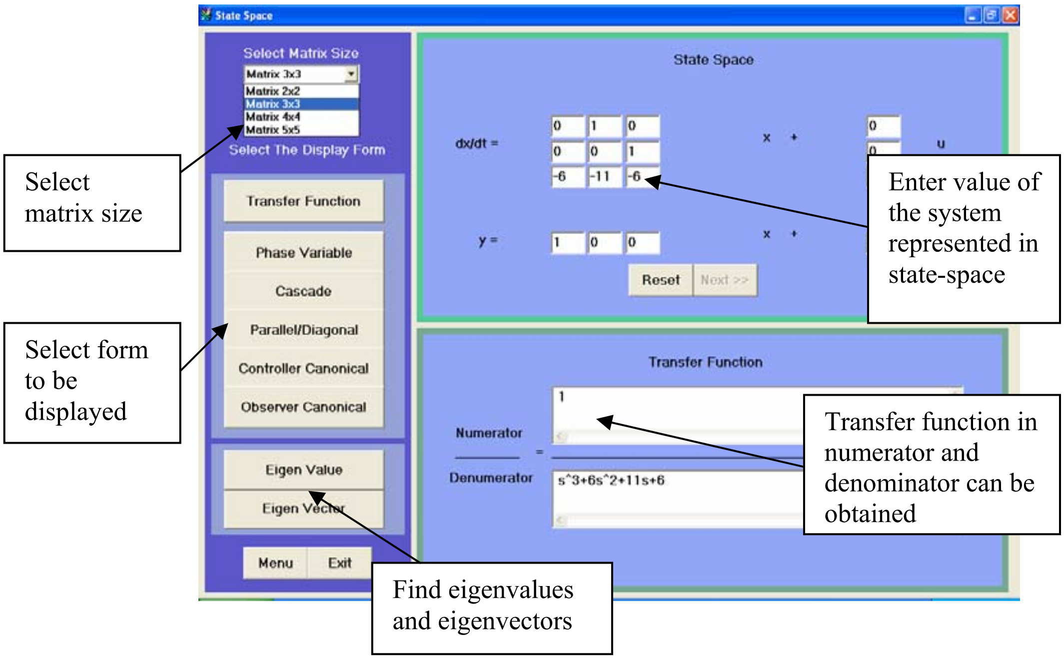

Figure 3 . State-space Module to obtain transfer function representation.

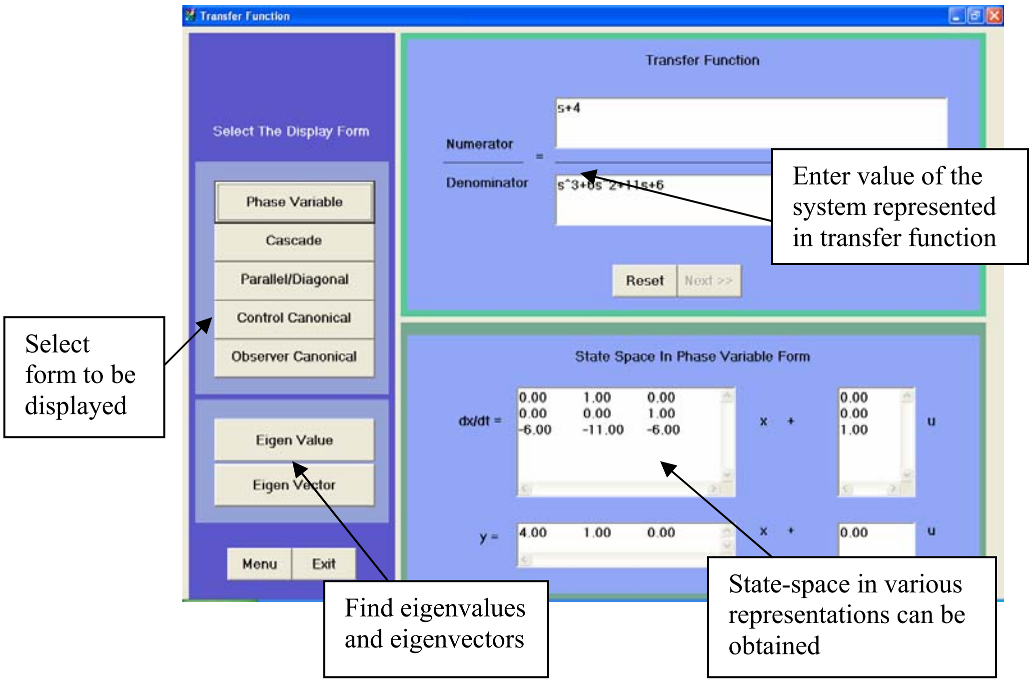

Figure 4 . Transfer Function module to obtain state.

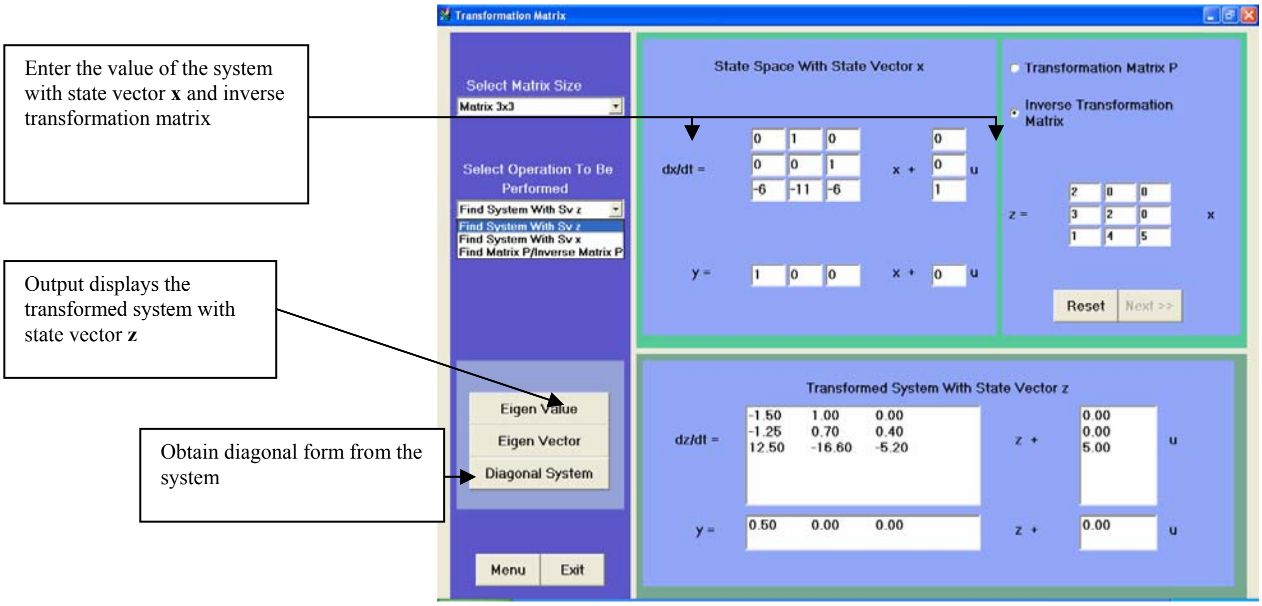

Figure 5 . Transformation Matrix module to obtain transformed system.

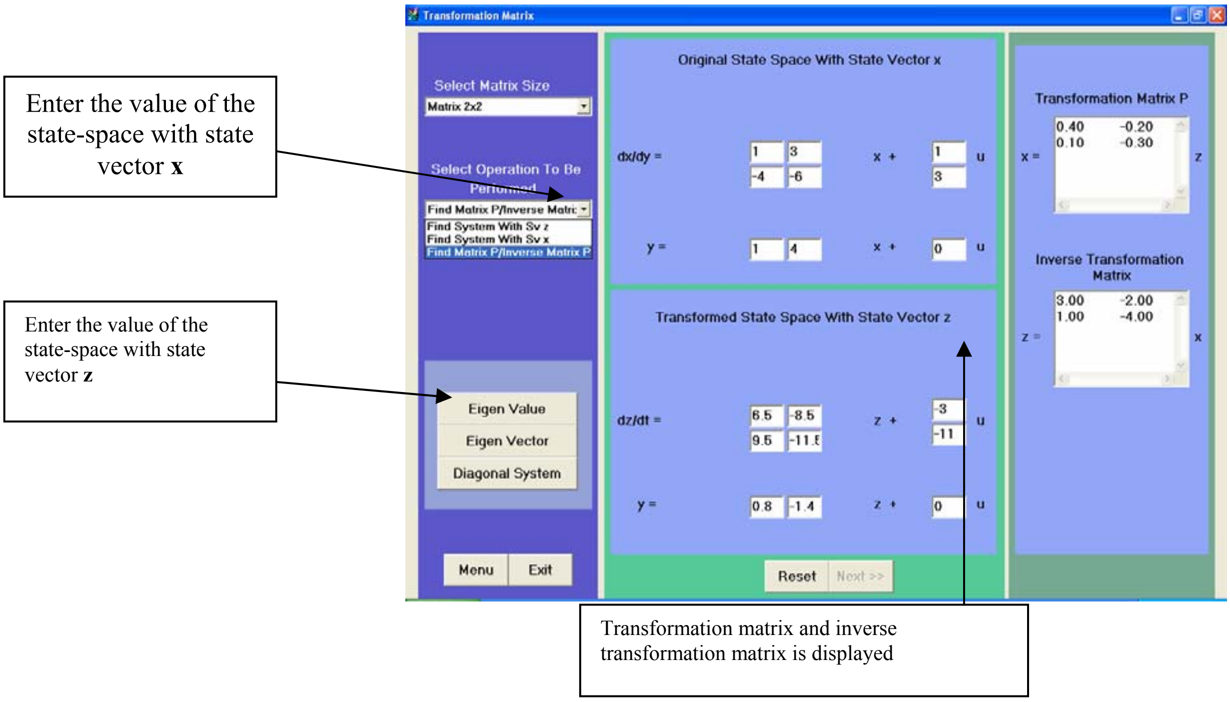

Figure 6. Transformation Matrix module to obtain transformation matrix and inverse transformation matrix.

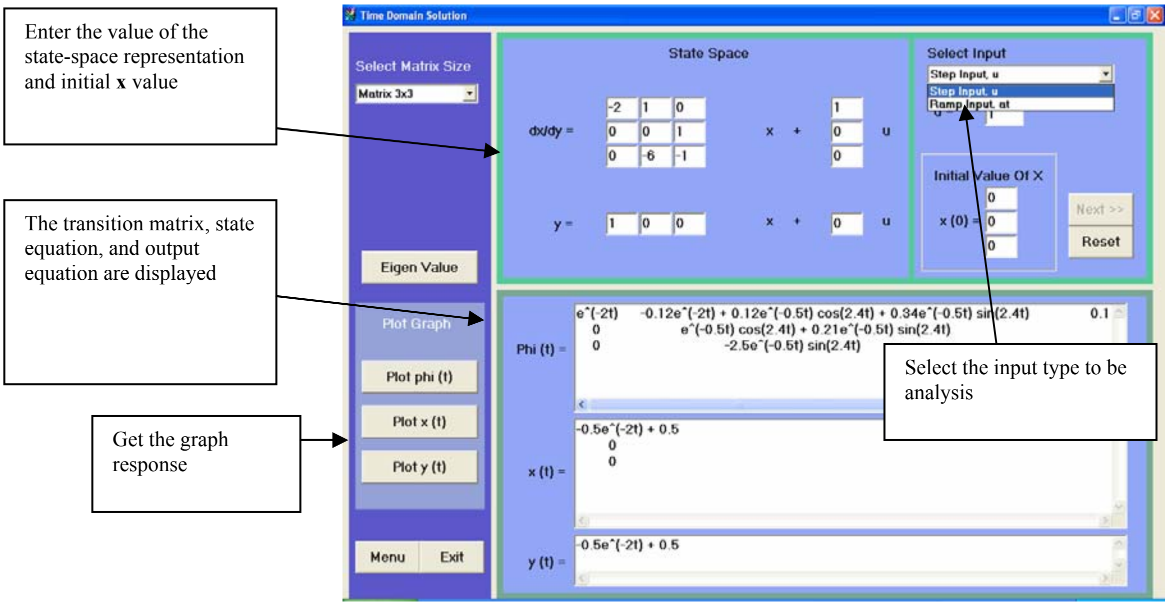

Figure 7. Time Domain Solution module.

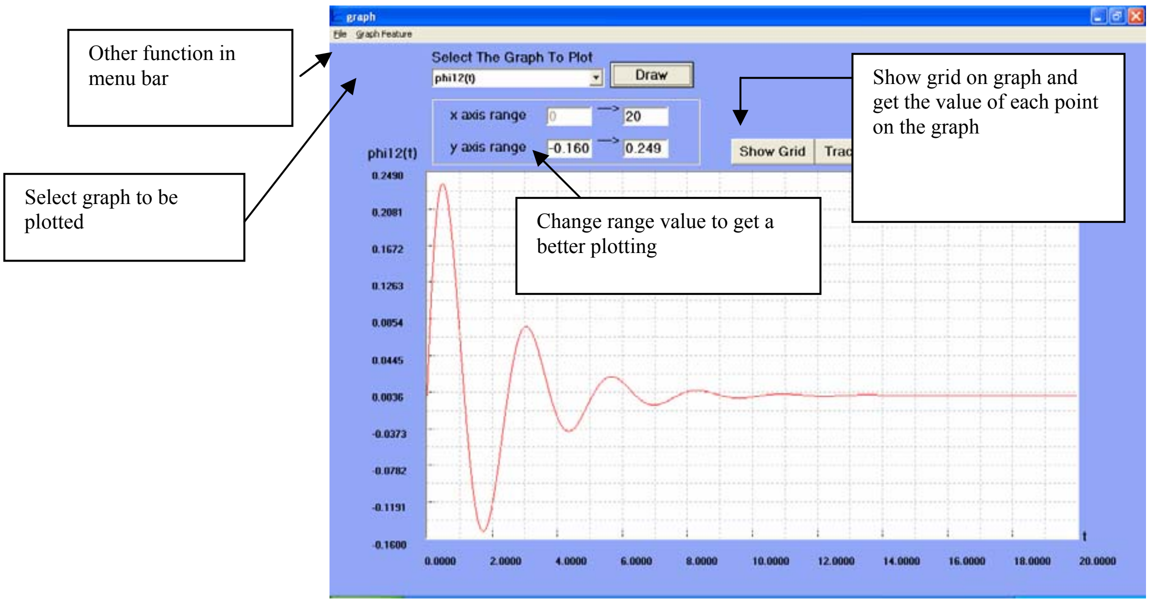

Figure 8. Graph response of Time Domain Solution module.

Figure 9 shows the Controller Design module. User enters the value of the system represented in state-space and the desired response characteristics. The system will be tested for the controllability. If the system is controllable, the state variable feedback gains, K_i can be obtained [12]. Figure 10 displays the graph response of the controlled system.

Figure 11 shows the Observer Design module. User enters the value of the system represented in state-space and the desired response characteristics included how fast the observer's transient response compares to controlled loop. The system is tested for the observability. If the system is observable, observer gains, L_i can be obtained. Graph response of $\mathbf{y}$ and $\hat{\mathbf{y}}$ of the designed system is shown in Figure 12.

Figure 13 shows the Steady State Error module with step unit response analysis. User enters the value of the system represented in state-space and gets the steady state error evaluation with a unit step input or a unit ramp input. Figure 14 shows the graph response of the system for Steady State Error module with step response analysis.

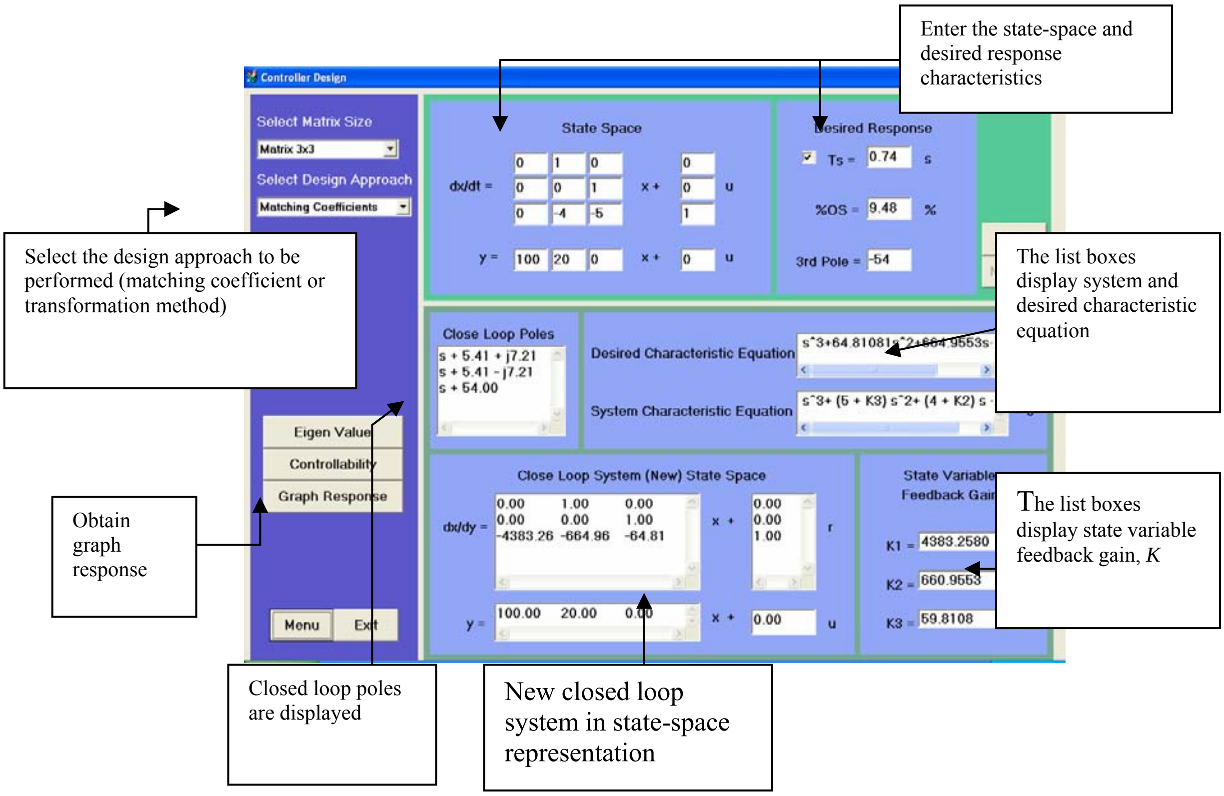

Figure 9. Controller Design module to get controller gains.

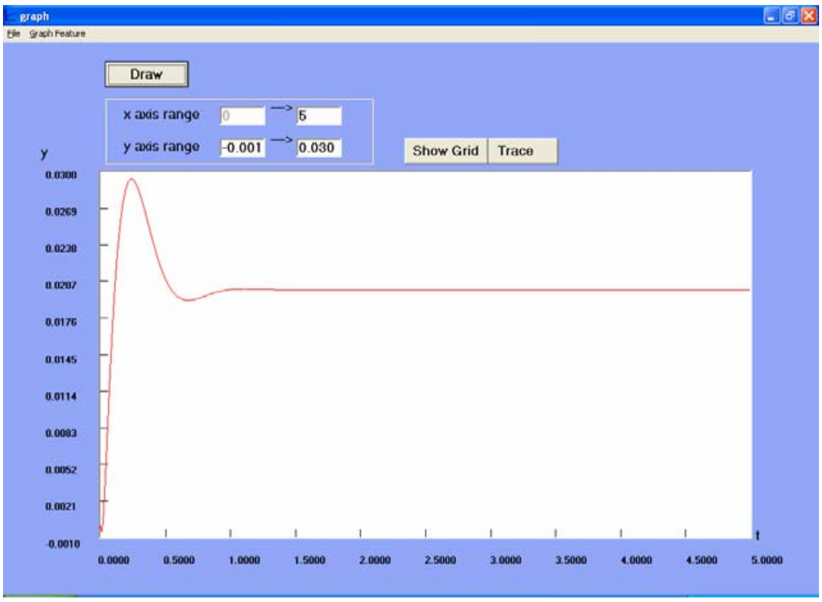

Figure 10. Graph response of new controlled loop from Controller Design module.

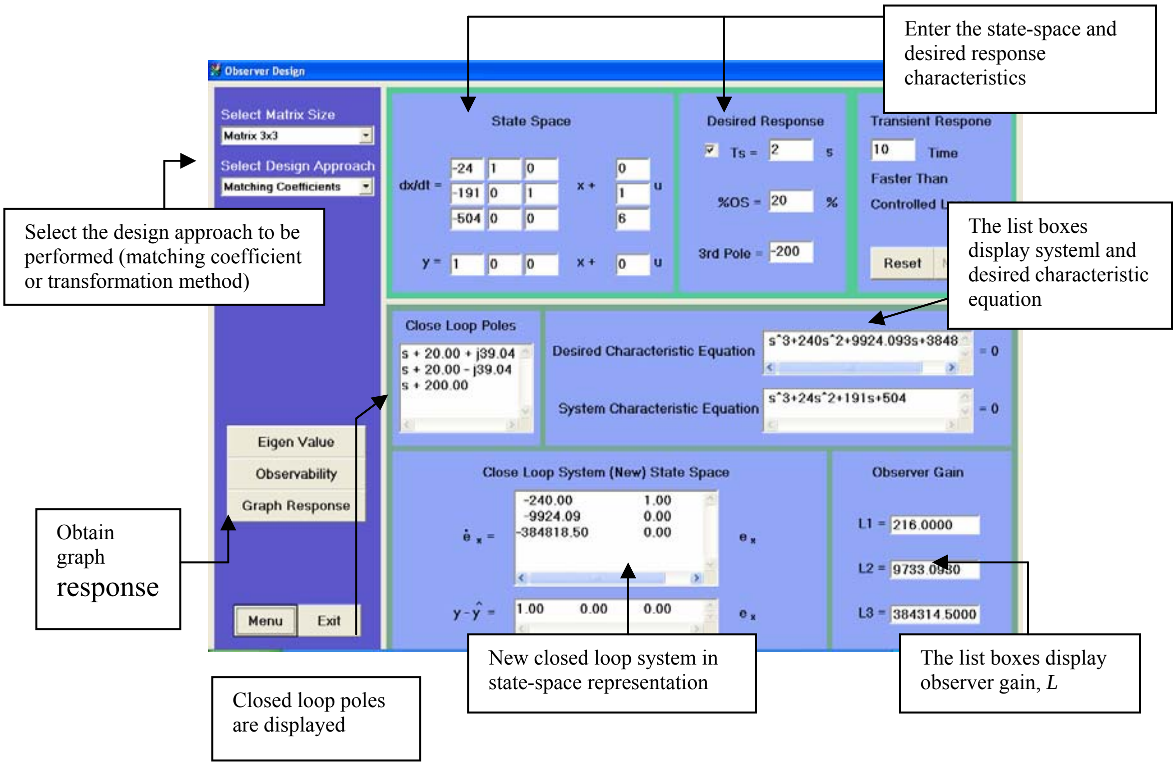

Figure 11. Observer Design module to get observer gains.

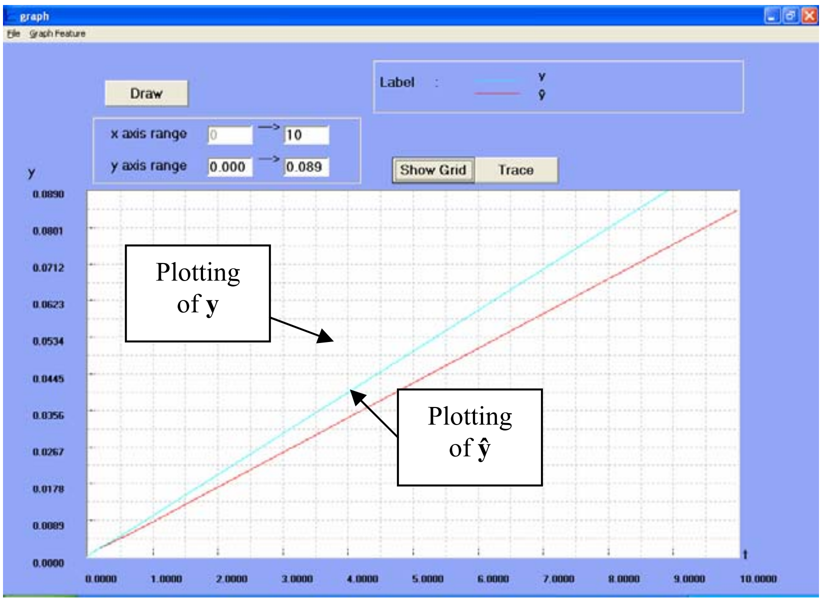

Figure 12. Graph response of y and ŷ from Observer Design module

5. CONCLUSION

This "Modern Control System Analysis Package" program, with user-friendly graphical user interface features besides interactive graph display can be useful to those who wish to learn modern control system theory in a way different from conventional classroom teaching method. This program will enhance the development of new tools and methodologies in the analysis of control system as well as open the opportunity for a computer integrated control engineering virtual environment.

REFERENCES

[1] P. N. Paraskevopoulos, *Modern Control Engineering*, New York: Marcel Dekker Inc., 2002.

[2] P. Hr. Petkov, N. D. Christov and M. M. Konstantinov, *Computational Methods for Linear Control Systems*, United Kingdom: Prentice Hall International Ltd., 1991.

[3] Stanley M. Shinners, *Modern Control System Theory and Design*, New York: John Wiley and Sons, 1992.

[4] Katsuhiko Ogata, *State Space Analysis of Control Systems*, USA: Prentice-Hall, 1967.

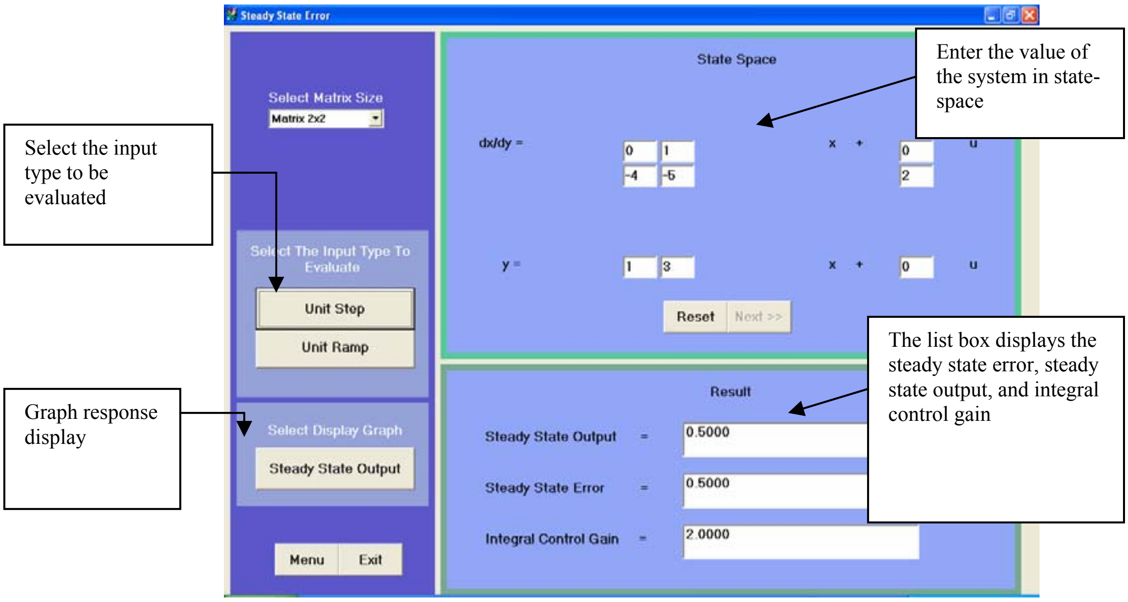

Figure 13. Steady state error evaluation in Steady State Error module.

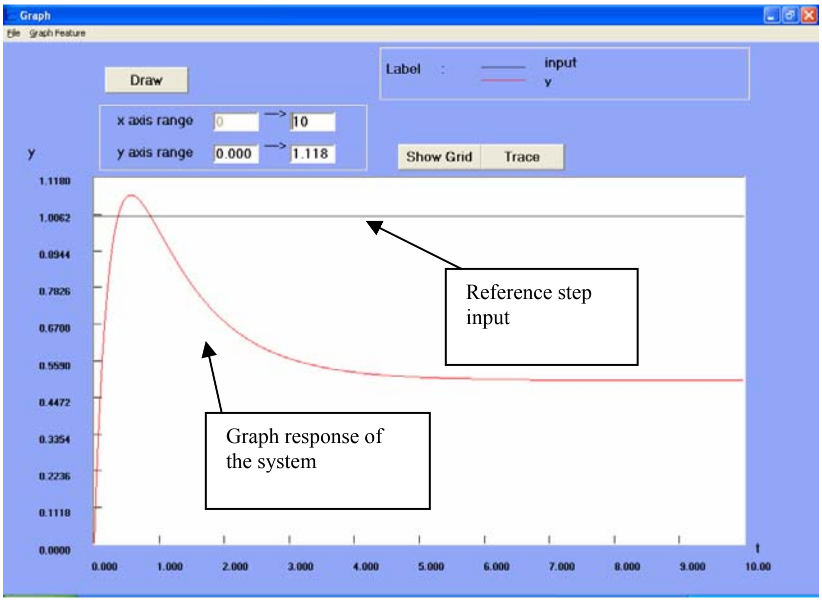

Figure 14. Graph response for Steady State Error module.

[5] Norman S. Nise, *Control System Engineering*, USA: John Wiley & Sons, 2000.

[6] John J. D'Azzo and Constatntine H. Houpis, *Linear Control System Analysis and Design, Conventional and Modern*, USA: McGraw-Hill, 1995.

[7] Z. Gajic and M. Lelic, *Modern Control Systems Engineering*, Great Britain: Prentice-Hall, 1996.

[8] Julia Case Bradley and Anita C. Millspaugh, *Programming in Visual Basic 6.0*, USA: McGraw-Hill, 2002.

[9] H. M. Deitel, P. J. Deitel and T. R. Nieto, *Visual Basic 6 How to Program*, USA: Prentice-Hall, 1999.

[10] Medya Riantama Syafri, "Penyelesaian Domain Masa Bagi Analisis Keadaan Ruang Menggunakan Visual Basic," *Tesis Sarjana Muda*, Universiti Teknologi Malaysia, 2003.

[11] Yap Lee Lin, "*Computer Aided Control System Design For State Space Analysis*", *Bachelor Degree Thesis*, Universiti Teknologi Malaysia, 2001.