Development of MATLAB GUI Application for System Identification (SID) of Beam Structure

Rickey Ting Pek Eek\textsuperscript{1}, Intan Z. Mat Darus\textsuperscript{1} and Shafishuhaza Sahlan\textsuperscript{2}
\textsuperscript{1}Department of Applied Mechanics, Fakulti Kejuruteraan Mekanikal
\textsuperscript{2}Department of Control and Mechatronics Engineering, Fakulti Kejuruteraan Elektrik
Universiti Teknologi Malaysia
81310, Johor Bahru
MALAYSIA

Abstract- This paper describes a System Identification (SID) Graphical User Interface using Least Squares, Recursive Least Squares and Particle Swarm Optimization methods whereby a mathematical model of a beam structure model is obtained. Using these System Identifications, the behavior and characteristics of the given system can be observed. A user friendly application, the proposed GUI is created using \textit{guide} MATLAB, which runs the System Identification method algorithms and chooses the mathematical model with the best performance. In the GUI, the results obtained are displayed in graphical plots and numerical values in the axes. The GUI application allows users to save the parameters and the results obtained into mat file hence ready for the next step application, i.e. the controller design stage.

Key-Words: - Graphical user interface, flexible beam, particle swarm optimization, system identification, recursive least squares.

1 Introduction

System Identification is a modeling technique to obtain a mathematical model or more precisely a transfer function for a given system. Through the obtained mathematical model, the behaviors and characteristics of the particular system can be observed. At current, there exists several types of System Identification methods; however, this paper focuses on the Least Squares (LS), Recursive Least Squares (RLS) and Particle Swarm Optimization (PSO) as the System Identification method.

In this paper, a beam structure model is the system that will be manipulated, as shown in Figure 1. The data input and output is obtained from the experimental or simulation of the beam’s vibration. The beam structure is a common flexible element that is widely applied in mechanical structures such as airplanes, ships, submarine, etc. However, the beam structure is easily influenced by unwanted vibration which is considered as noise to the system. The vibration may lead to problems such as fatigue, instability and performance reduction. \cite{2} Therefore, the vibration must be reduced by controlling the vibration of the beam structure.

System Identification is an essential part of this study where the mathematical model of the beam structure is obtained. From the obtained model, the vibration can then be controlled by the existing controllers which include PID, fuzzy logic, etc.

![Figure 1: Fixed-Free Mode of Flexible Beam structure](image)

To run the algorithms of LS, RLS and PSO, a graphical user interface is required to explain and display the result of each algorithm, in a user-friendly manner. In this paper, the GUI is built using \textit{guide} MATLAB. \textit{Guide} MATLAB prepares a great environment for designing and creating a user friendly interface which can be applied in teaching as well. GUI is useful and convenient as the result of each System Identification method will be displayed with a click of the button without having to deal with the necessary algorithms in the command windows MATLAB.

This paper discusses on System Identification, particularly Least Squares, Recursive Least Squares, Particle Swarm Optimization, and Graphical User Interface in section II. In section III, description of tools on SID GUI is presented. Meanwhile in section
IV, the result and simulation for the GUI used is discussed. Finally, the conclusion and future work for the GUI application is presented in section V.

2 System Identification

System Identification is a method used to obtain the mathematical model which describes the characteristics or behaviors of a system or model. In technical terms, System Identification is defined by Zadeh (1962) as the determination on the basis of input and output of a system within a specified class system to which the system under test is equivalent (in terms of a criterion).[14]

System Identification allows us to build the mathematical model of a dynamic system based on the measured data input and output. The measured data can be collected from the experimental or simulation on the system. A good test is to take a close look at the estimated model’s output compared to the measured one on a data set based on the best fit value [13]. The estimated model is the model obtained from system identification.

Most common models are difference equations descriptions, such as ARX and ARMAX models, as well as all types of linear state-space models. Model describes the relationship between the measured signals. As shown in Figure 2, $u$ indicates the input of the system while $y$ is the output produced by the system based on the input. However, the output is also affected by immeasurable input which is called disturbance signal or noise, $e$.

![Figure 2: Block Diagram of a Model](image)

2.1 Least squares algorithm

The least squares principle was proposed by Karl Gauss at the end of eighteenth century for determining the orbits of planets. This method has become an easy and simple tool applied in parameter estimation using experimental data. Due to the existence of closed solution, the method is easy to comprehend and easy to be implemented. The least squares method is also known as linear regression and equation error method.

Consider the z-transform of a transfer function of a system as denoted in (1):

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_1z^{-1} + b_2z^{-2} + \cdots + b_nz^{-n}}{1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_mz^{-m}}$$

where $b$ and $a$ are the coefficients of the numerator and denominator of the transfer function, $H(z)$, respectively.

In least squares method, a signal input $u(k), k \in \{1,2,\ldots,N\}$ is applied to the system. The output is measured as $y(k), k \in \{1,2,\ldots,N\}$.

The parameter $\theta$ is used to minimize the discrepancy between the left and the right-hand-sides of the equation (2) in a least squares sense.

$$y(k) = \Phi(k) \cdot \theta$$

The parameter $\theta$ can be obtained from the equation as below.

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T y$$

where

$$\Phi = \begin{bmatrix} -y(0) & \ldots & -y(1-n) & -u(0) & \ldots & -u(1-n) \\ -y(1) & \ldots & -y(2-n) & -u(1) & \ldots & -u(2-n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y(N-1) & \ldots & -y(N-n) & -u(N-1) & \ldots & -u(N-n) \end{bmatrix}$$

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}$$

$$\hat{Y} = \Phi \cdot \theta$$

while $\hat{Y}$ is the estimated output using least squares method. The error $e(t)$ between the experimental and estimated model is denoted as:

$$e(t) = Y - \hat{Y} = y(k) - \Phi(k)$$

2.2 Recursive least squares algorithm

Recursive least squares (RLS) is another type of System Identification method which considers the effect of additional observation data denoted as $Y_{N+1}$.

The equation for RLS estimation is shown in equation (7).

$$\begin{bmatrix} Y_N \\ Y_{N+1} \end{bmatrix} = \begin{bmatrix} \Phi_N \\ \Phi_{N+1} \end{bmatrix} \theta + \begin{bmatrix} e_N \\ e_{N+1} \end{bmatrix}$$

where $e$ is the white noise.

Rearrange in a row of matrix shown in equation (8).
\[ X_{N+1}^T = [-y(N) \ldots -y(N+1-n) \ u(N) \ldots u(N+1-n)] \] (8)

From the equation (3) in least squares,
\[
\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y
\]
\[
\hat{\theta}_{N+1} = \left( \begin{bmatrix} \Phi_N^T & X_{N+1}^T \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi_N^T & X_{N+1}^T \end{bmatrix} \begin{bmatrix} Y_N & Y_{N+1} \end{bmatrix}
\]
\[
\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1} (Y_{N+1} - X_{N+1}^T \hat{\theta}_N)
\] (9)

In RLS, \( \lambda \) is the parameter called forgetting factor. The range of the \( \lambda \) value is between \( 0 < \lambda < 1 \). It is used to ensure the data in the past are forgotten in order to get the statistical variations of observable data when the filter operates in non-stationary environment. [1]

\[
P_{N+1} = \frac{P_N}{\lambda} \left( 1 - \frac{X_N^T X_{N+1} P_N}{\lambda + X_{N+1}^T P_N X_{N+1}} \right)
\]

2.3 Particle Swarm Optimization Algorithm

Particle Swarm Optimization is another type of System Identification method that is discussed in this paper. It is introduced by Kennedy and Eberhart in the mid of the 1990s while attempting to simulate the choreographed, graceful motion of a swarm of birds as part of a socio cognitive study investigating the notion of “collective intelligence” in biological populations. [15]

PSO involves the problem that needs to be solved which the cost function, \( f(x) \), are defined and to be minimized or maximized considering all parameters of the problem.

Swarm is the number of potential solutions to the problem where particle is the potential solution for each swarm. Each particle holds position as the candidate solution to the problem and velocity as the flying direction of the particle.

The aim of PSO is to find the particle’s position with the best evaluation of a given cost functions. At each of the iteration, best position by one particle (pbest) and best position by whole swarm (gbest) will be updated.

\[
pbest = p_{id}(t + 1) = \begin{cases} 
p_{id}(t) & \text{if } f(x_{id}(t + 1)) \geq f(p_{id}(t)) \\
x_{id}(t + 1) & \text{if } f(x_{id}(t + 1)) < f(p_{id}(t)) \end{cases}
\]
\[
gbest = p_gd(t) \in \{p_{1d}, p_{2d}, \ldots, p_{sd}\} = \min\{f(p_{1d}(t)), f(p_{2d}(t)), \ldots, f(p_{sd}(t))\}
\]

where \( x_{id} \) is the current position of the particle.

In PSO, there are two basic equations which govern the motion of particles. The first equation is the velocity update equation as shown in equation (10).

\[
velocity = v_{id}(t + 1) = v_{id}(t) + c_1 r_1(t)(p_{id}(t) - x_{id}(t)) + c_2 r_2(t)(p_{id}(t) - x_{id}(t)) \] (10)

The second equation is the movement equation that provides actual motion of particles using their specific velocity.

\[
x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \] (11)

where \( c_1 \) and \( c_2 \) are the acceleration constants and \( r_1 \), \( r_2 \) are the random numbers. For a better control exploitation and exploration, the inertia weight is introduced as in equation (12) below.

\[
v_{id}(t + 1) = w v_{id}(t) + c_1 r_1(t)(p_{id}(t) - x_{id}(t)) + c_2 r_2(t)(p_{id}(t) - x_{id}(t)) \] (12)

3 Graphical User Interface

A Graphical User Interface (GUI) is a user interface platform with graphical objects such as push button, text fields, sliders and menus. Guide is the MATLAB Graphical User Interface development environment which will generate a GUI and the m-file that contains the code to handle the initialization and launching of the GUI.

To create a GUI, a good design with meticulous arrangement of graphical objects must be done accordingly. This step is a very important step which will create a user friendly application. Figure 3 displays the example of GUI creation for the System Identification which consists of several components. After that, the GUI is programmed by entering the algorithms into the call function in the m-file MATLAB. The steps of creating a simple GUI are shown in the flowchart in Figure 4.
3.1 Tool Description

In this study, the graphical user interface (GUI) is created using MATLAB R2011b guide which is widely used by academicians. The GUI allows the user to import data, select types of System Identification, key in the necessary parameters and run the System Identification algorithms. The GUI is built with several objects such as push buttons, static texts, edit texts, pop-up menus, axes, checkboxes, etc.

The System Identification (SID) GUI is a user friendly interface application which helps users in obtaining a mathematical model of a given system by changing the parameters in the GUI. However, if the input is a non-numerical value, the ‘Run’ push button will be disabled and an error dialog will be appeared to warn the user.

In the proposed GUI, users are able to select the imported data either from a specific location or data extracted from the workspace. The types of System Identification can also be chosen by clicking the radio button as shown in Figure 5.

4 Results and Discussion

The proposed GUI is divided into two parts, which are parameters selection and graphical plotting. In parameters selection, first of all, the data must be imported into the GUI. Next, the types of System Identification are chosen as the method to obtain the mathematical model. Lastly, the variables are required to run the GUI.

There are two ways to import the required data, which are obtained and saved from the simulation of the vibration for the beam structure. The data can be imported by choosing the file in specific location, from the MATLAB workspace.

By pressing the import button, a dialog box will pop-out and the user can choose their data or mat file from a specific location. Next, the names of the input and output have to be specified according to the names in the data imported. The default name for input and output is ‘Force_detection’ and ‘Force_Observation’ respectively.

If users required data from the workspace, the import option from workspace can be selected by clicking the checkbox, and the data will be easily loaded into the GUI. The pop-up menu will display all the variables in workspace to be chosen as the input and output.

After importing the data into GUI, the next step is to choose the type of System Identification by selecting the radio button. This paper proposes three types of system Identification which are Least Squares (LS), Recursive Least Squares (RLS) and Particle Swarm Optimization (PSO).

After selecting the System Identification method, the parameters need to be filled in accordingly. For the LS method, the only required parameter is the model order number, and the default value is 4. For the RLS, the values for model order number and forgetting factor (default value 0.7) are required.

On the other hand, for PSO, there are several parameters required in the GUI. This include model...
order number, correction factor, numbers of parameters, number of iteration, size of particles (i x j) and range of inertia. In addition to that, the estimation points are set by users to observe the training and testing points in the graphical plotting.

To run the process of System Identification, the ‘run’ button is pressed. The results are shown in the axes by choosing the types of graphical plotting in the pop-up menu as in the Figure 6 below. The ‘Enlarge’ button is used to observe the result in a pop-up window which can be printed and saved, per requirement.

Figure 6: The results plotting in the SID GUI

The types of graphical plotting are time plot, bode plot, error plot, root locus and semi log. Besides graphical plotting, the transfer function and the mean square error of the estimated system or beam structure can be displayed at the axes too.

In the SID GUI, the ‘Save Parameters’ button are used to save every parameters used and results into mat file for controller implementation on the next stage.

5 Conclusions

Least Squares, Recursive Least Squares and Particle Swarm Optimization are useful types of System Identification with different algorithms to obtain the mathematical model of the beam structure. However, with these different types, the best mathematical model based on the performance and behavior is required to be chosen. Therefore, a user friendly GUI is designed and created to observe the result of each type of System Identification through the graphical plotting and then the best mathematical model is selected for controlling purposes. Selecting the best model is important in order for the vibration to be controlled easily.

Besides that, the GUI is created to help in education field which can be a teaching tool to explain and show students with the graphical plotting and mathematical model of the beam structure.

In the future, there are many types of System Identification which can be programmed and created for the GUI to run the algorithm such as genetic algorithm, gravitational searching algorithm, etc. In the next stage, the controller with PID will control the output in order to reduce the vibration of the beam based on the mathematical model obtained from the GUI.

Acknowledgement

The authors wish to thank the Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for providing the research grant and facilities. This research is supported using UTM Research University grant, Vote No. 04H17.

References:

[9] Bachir ELMAGROUD and Abdelhak ZIYYAT, Educational Software for wave
propagation phenomena using MATLAB GUIs and FDTD Method.


[16] ZhiXiang Hou, Hammerstein Model Identification Based on Adaptive Particle Swarm Optimization, Workshop on Intelligent Information Technology Application, 2007 IEEE.