

# Virtual Measurement System MATLAB GUI Documentation

## INTRODUCTION

When taking real-world measurements on a dynamic system with an accelerometer and LVDT, these transducers will not always produce “clean” output, like that shown in Fig. 1.

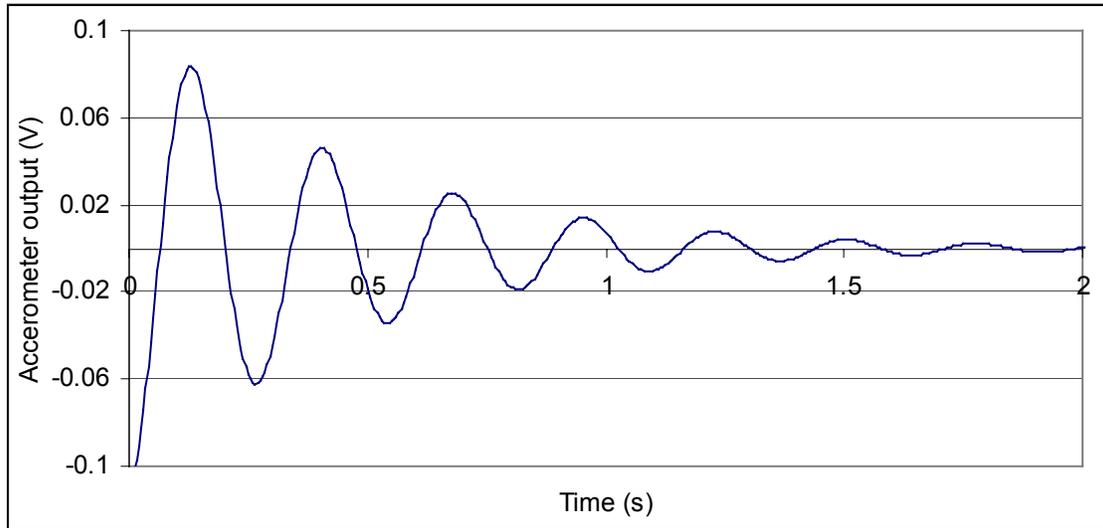


Fig. 1. Ideal “clean” accelerometer output

Factors such as noise (random or sinusoidal), drift, and bias can corrupt the transducer output. The results obtained from a test may look more like the data shown in Fig. 2.

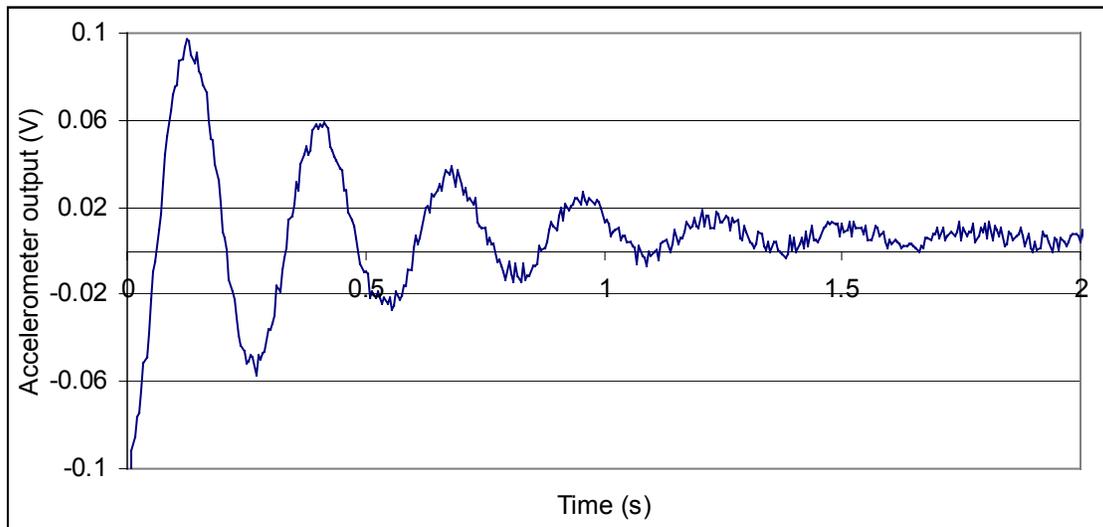


Fig. 2. Replicated real-world “messy” accelerometer output.

The Virtual Measurement System GUI attempts to analytically replicate the real-life problems seen with data from these transducers. These problems can then be more easily identified and understood.

The dynamic system modeled is a simple single-degree-of-freedom (SDOF) second-order system like the mass-spring-dashpot shown in Fig. 3.

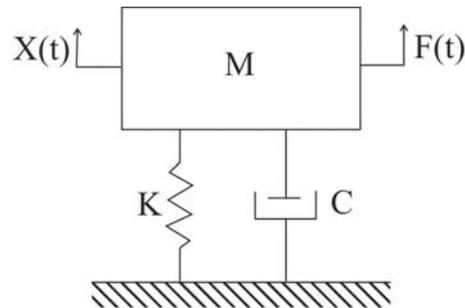


Fig. 3. Basic mass-spring-dashpot system.

To this basic model components are added which simulate the output from an LVDT and accelerometer placed on this system. Various corrupting factors can then be added and varied to observe the resulting effect on the output. The use of a simple RC circuit low-pass filter to reduce sinusoidal noise can also be explored.

## FILES NEEDED TO USE VIRTUAL MEASUREMENT SYSTEM GUI

basic\_system.fig  
basic\_system.p  
mass\_spring.mdl  
mass-spring-dashpot.jpg

Place these four files into the same directory, and make this the working directory in MATLAB.

## RUNNING THE SIMULINK MODEL AND GUI

With the proper working directory active, type 'basic\_system' in the MATLAB command window.

## SIMULINK MODEL

The GUI is based on a Simulink model. The controls of the GUI are used to change the properties of this model. The entire Simulink model is shown in Fig. 4.

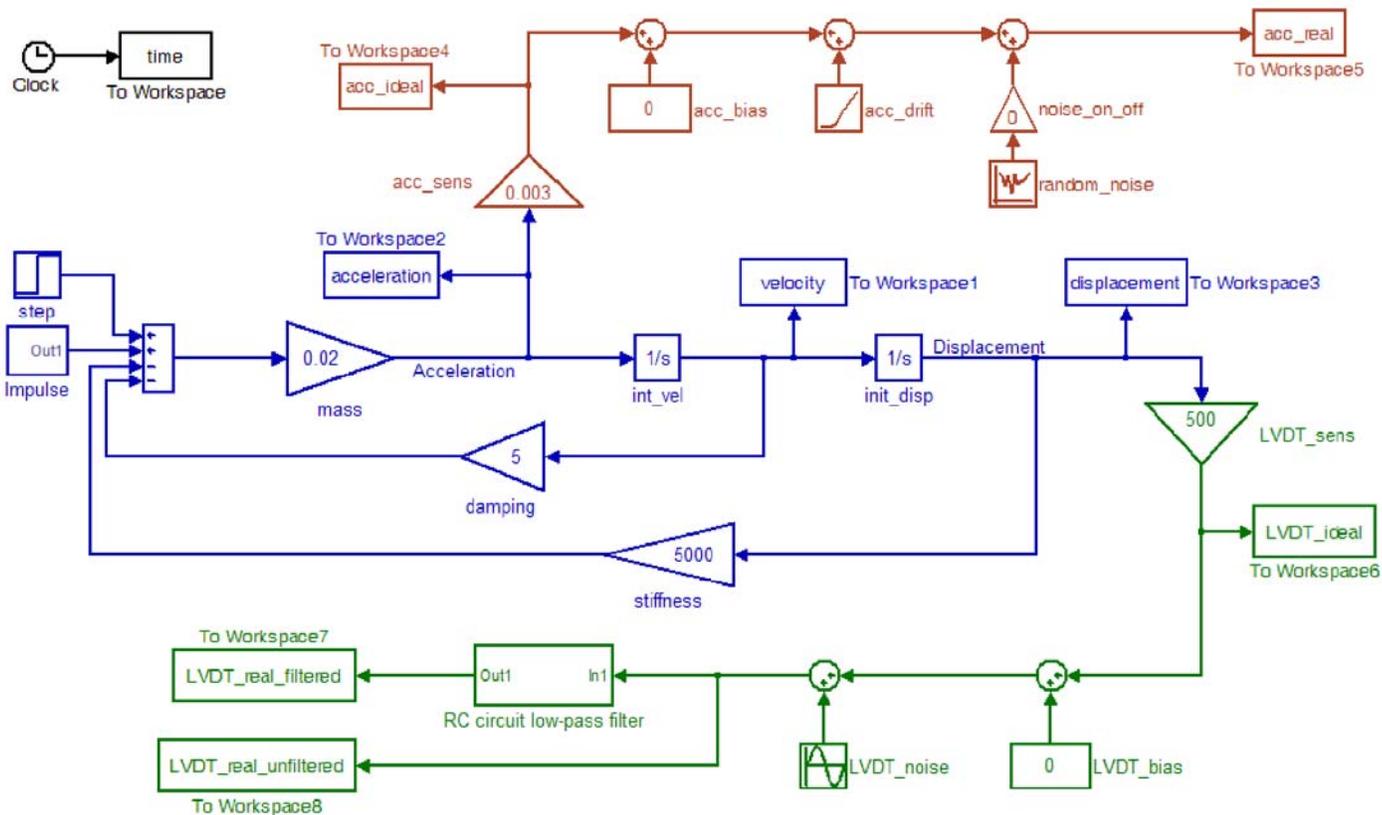


Fig. 4. Complete Simulink model.

It consists of three basic sections: the central portion which describes the SDOF second-order dynamic system (in blue), the simulated accelerometer output (in red), and the simulated LVDT output (in green). These sections will now be discussed in more detail. The properties of these various components can be altered using the GUI, which will be explained in the next section.

### Central System

The main part of the Simulink model is shown in Fig. 5.

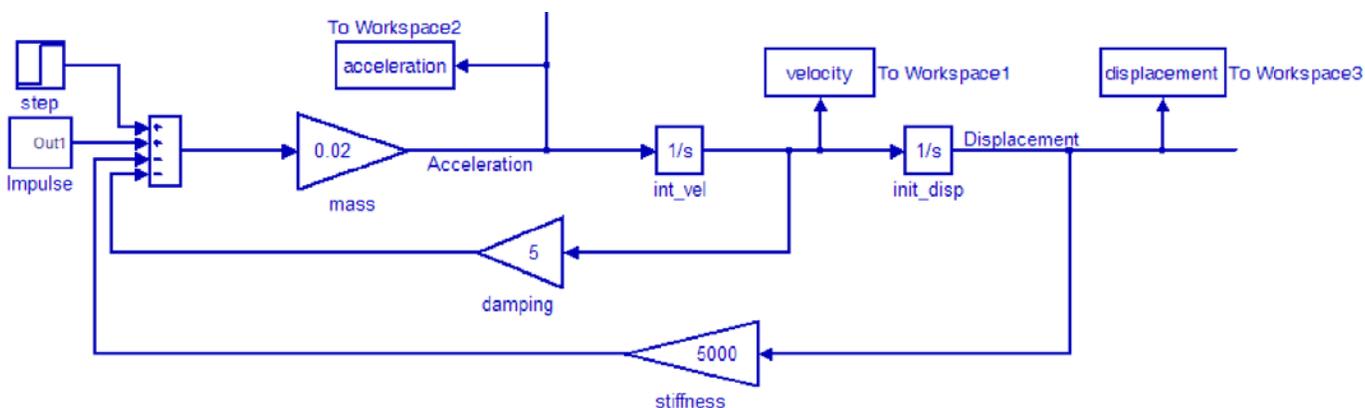


Fig. 5. Portion of Simulink model which models the basic SDOF system.

This is the foundation of the Simulink model: a single-degree-of-freedom, second-order system. For more information on modeling second-order systems in Simulink, see the tutorials which address block diagram modeling in Simulink and second-order systems in general.

This system has three characteristics which determine its response: its mass ( $m$ ), damping ( $c$ ), and stiffness ( $k$ ). Three different types of inputs can be applied to the system: a step function, an impulse, or a displacement input. The step function is straightforward; it is a standard Simulink source block. In this model, the impulse is modeled by combining two step functions. For more information this and other ways to model impulses in Simulink, see the tutorial specific to impulse responses. The displacement input is applied by adding an initial condition in the “init\_disp” integrator block.

When the model is run, it outputs to the workspace the displacement, velocity, and acceleration of the system.

### Accelerometer

The section of the model which imitates the behavior of an accelerometer is shown in Fig. 6.

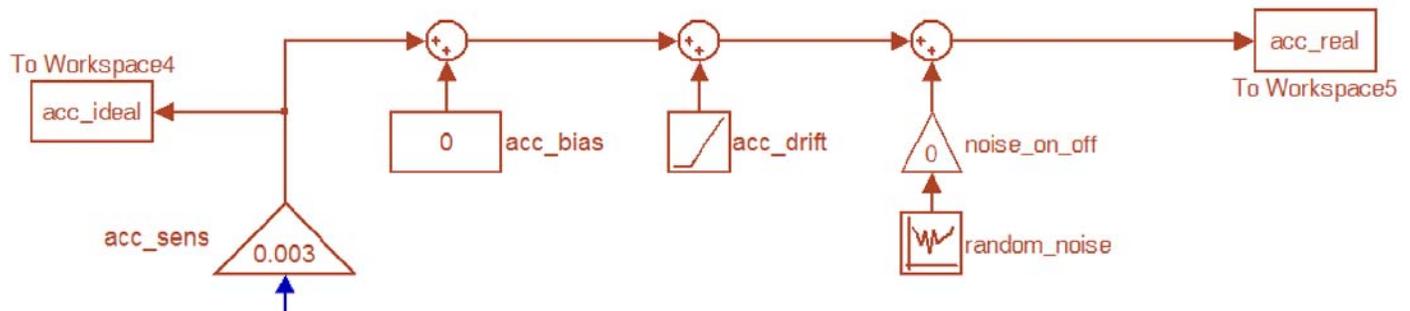


Fig. 6. Portion of Simulink model which models the accelerometer.

The acceleration of the system is converted to an accelerometer output by multiplying by a sensitivity (in volts per  $m/sec^2$ ) to produce an “ideal” accelerometer output in volts. This is how the data shown in Fig. 1 was produced. This accelerometer output is considered ideal because it assumes that the accelerometer perfectly measures the acceleration; if the accelerometer voltage is divided by the sensitivity then the exact acceleration of the system is determined. In practice, however, the accelerometer output could be corrupted by any of several problems. The problems which can be modeled here are bias, drift, and random noise.

*Bias:* A DC offset of the signal can be caused by the accelerometer’s signal conditioning circuitry. The signal conditioner converts the output charge from the piezoelectric material in the accelerometer to a voltage. This DC bias could be eliminated by AC coupling the signal, but for measurements on systems with low frequencies this is often not practical.

*Drift:* Drift occurs when the energy contained in the signal is not balanced about the x-axis. The capacitance in the signal conditioning system tends to acquire a net positive or negative charge (depending on whether the signal is more positive or more negative) so the signal will drift positive or negative.

*Random noise:* In this Simulink model, the level of the random noise can be specified. The most likely source of random noise in a laboratory setting is nearby electrical devices such as electrical wiring or computers.

These factors are combined, and the resulting accelerometer output is referred to as the “real-world” accelerometer output, because it simulates issues that are seen in actual measurements. The plot shown in Fig. 2 is an example of this.

### ***LVDT***

Fig. 7 shows the section of the Simulink model which models the LVDT.

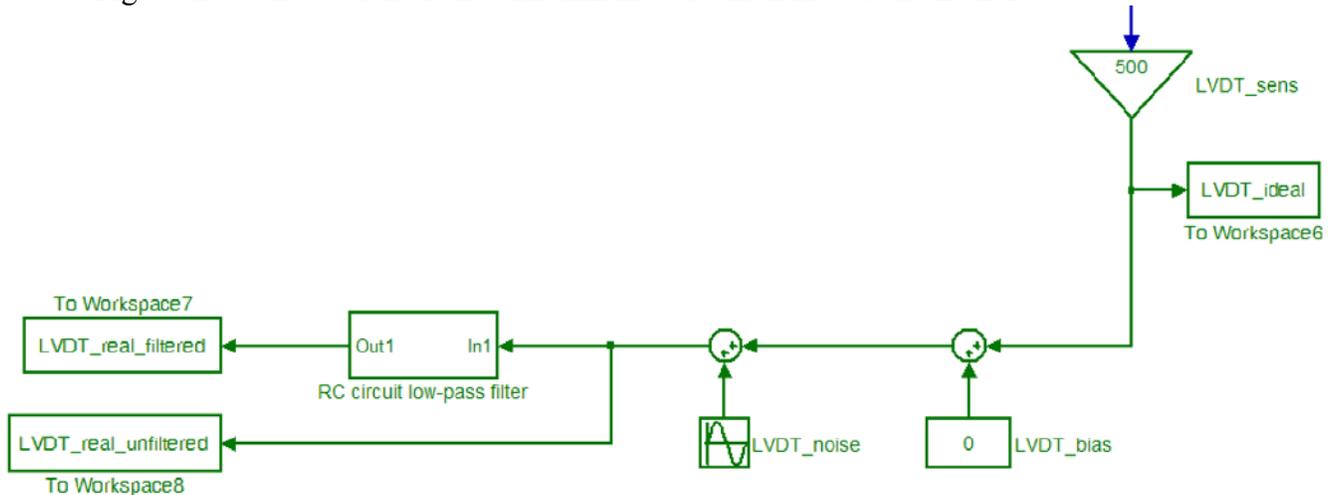


Fig. 7. Portion of Simulink model which models the LVDT.

Similar to the accelerometer, an “ideal” LVDT output is produced by multiplying the displacement of the system by a sensitivity in V/m.

*Bias:* A DC offset can be added. This occurs in real measurements when the LVDT is not perfectly zeroed when the system is at a position of zero displacement.

*Noise:* Sinusoidal noise can also be added to the LVDT signal. 60 Hz noise is most commonly seen, this is generally caused by nearby electrical wiring

These signals are added into the LVDT signal, and the resulting “messy” signal is referred to as the “real-world” LVDT output. This signal is output to the workspace, but it is also put through a simple filter. The filter appears as the “RC circuit low-pass filter” subsystem in Fig. 7 above.

### ***Low-Pass Filter***

Fig. 8 shows the contents of this subsystem.

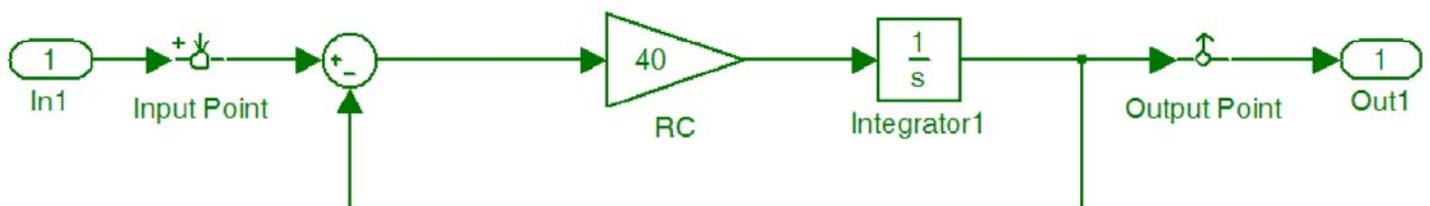


Fig. 8. Contents of RC circuit low-pass filter subsystem.

This is a simple first-order system, an RC circuit, which acts as a low-pass filter. For more information on RC circuits and filtering, see the tutorials specific to first-order systems. By changing the RC value of this model, the cut-off frequency of the filter is changed. The input and output points before and after the filter allow you to view the frequency response of the filter.

## GRAPHICAL USER INTERFACE (GUI)

Although the properties of the Simulink model can be changed directly by modifying the properties of the various blocks, they can be varied more easily using the GUI.

When the GUI is first opened, it will appear as shown in Fig. 9.

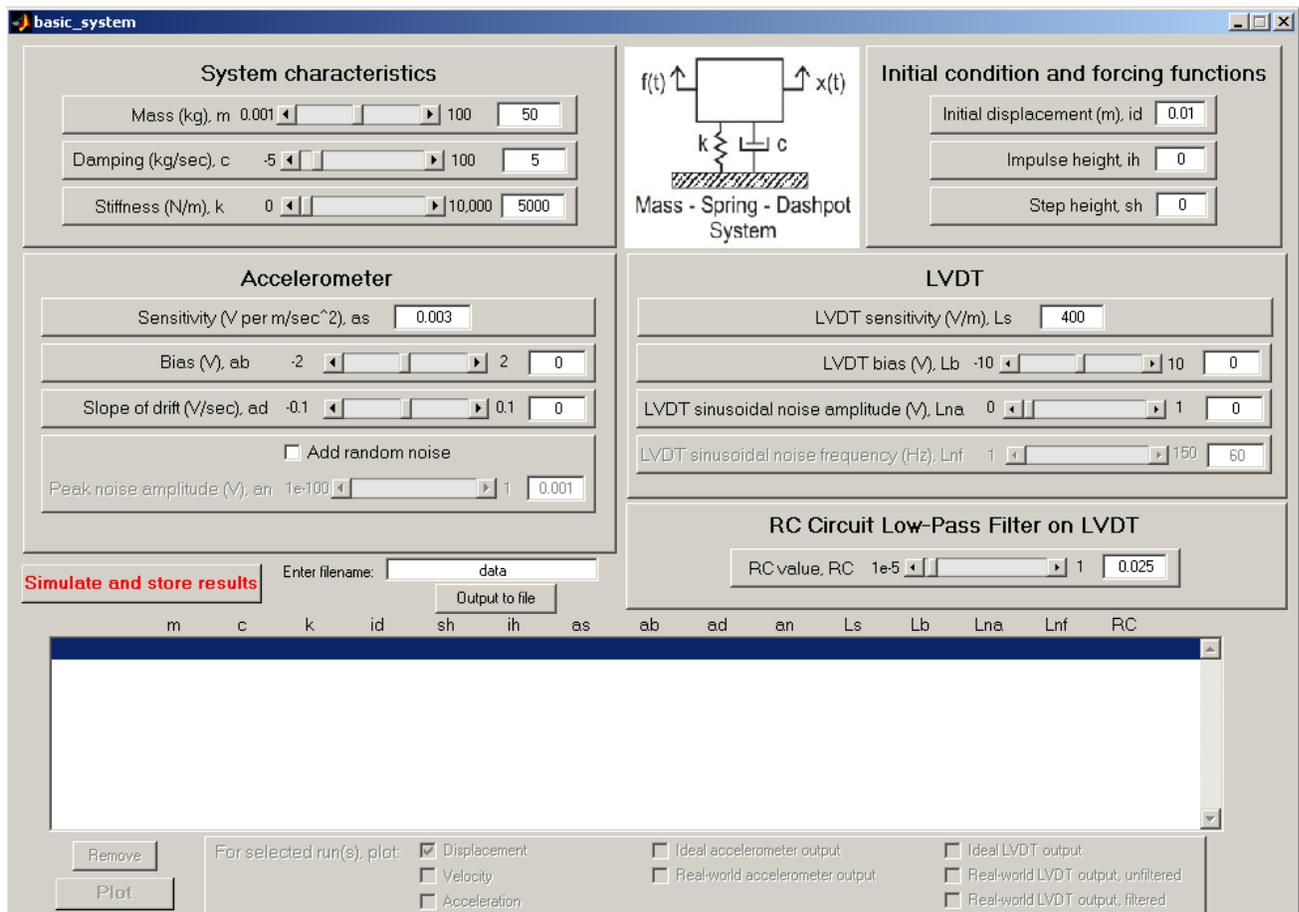


Fig. 9. Graphical User Interface.

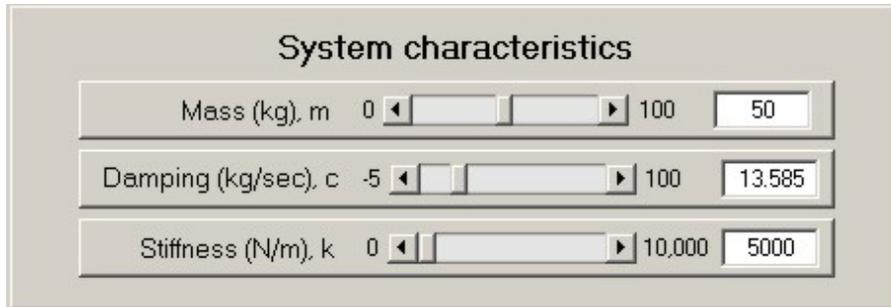
The interface consists primarily of slider bars and text boxes which can be used to change the parameters of the model. If a value is entered which is outside the range of the slider, the slider and the textbox will reset to the last valid value.

The label for each different parameter (mass, LVDT bias, etc.) is followed by a one or two letter abbreviation (m, Lb) by which it is referred to in the list box.

The following subsections discuss the different controls available on the GUI.

### *System Characteristics*

This section of controls, shown in Fig. 10, allows the user to set the mass, damping, and stiffness properties of the mass-spring-dashpot system.



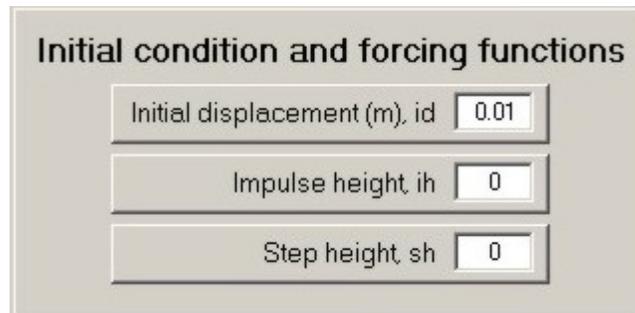
The screenshot shows a control panel titled "System characteristics". It contains three rows of controls, each with a label, a unit, a variable name, a range, and a value field. The first row is for Mass (kg),  $m$ , with a range from 0 to 100 and a value of 50. The second row is for Damping (kg/sec),  $c$ , with a range from -5 to 100 and a value of 13.585. The third row is for Stiffness (N/m),  $k$ , with a range from 0 to 10,000 and a value of 5000.

Fig. 10. Controls for setting the system characteristics.

It should be noted that the damping can be set to a negative value, which will result in an unstable system.

### *Initial Condition and Forcing Functions*

The initial displacement of the system and the forcing functions (impulse or step) can be set using the controls shown in Fig. 11. The impulse height and step height values are in Newtons.



The screenshot shows a control panel titled "Initial condition and forcing functions". It contains three rows of controls, each with a label, a unit, a variable name, and a value field. The first row is for Initial displacement (m),  $id$ , with a value of 0.01. The second row is for Impulse height,  $ih$ , with a value of 0. The third row is for Step height,  $sh$ , with a value of 0.

Fig. 11. Controls for setting initial condition and forcing functions.

### *Accelerometer*

The properties of the accelerometer can be set using the controls shown in Fig. 12.

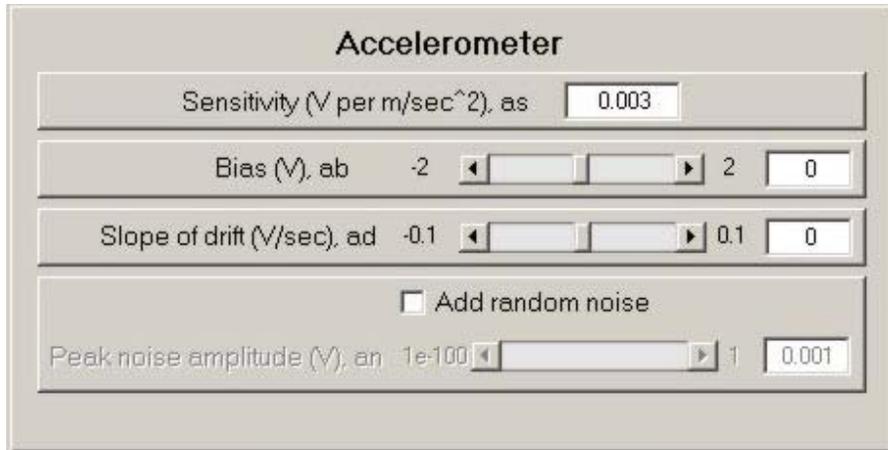


Fig. 12. Controls for setting the properties of the accelerometer.

The user can set the sensitivity of the accelerometer, in V per  $m/s^2$ .

A bias, or DC offset in volts, can be added using the second control. A drift may also be added by setting a positive or negative slope in V per second. To add random noise to the accelerometer signal, click the checkbox and then set the level using the slider bar or textbox.

### ***LVDT***

The controls for the LVDT portion of the model are shown in Fig. 13.

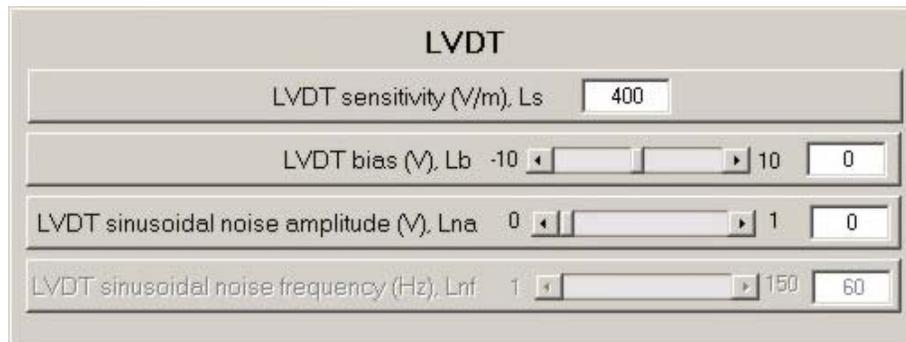


Fig. 13. Controls for setting the properties of the LVDT.

Similar to the accelerometer, the sensitivity of the LVDT can also be set. The second control sets a DC offset or bias. The third control changes the amplitude of sinusoidal noise on the signal, and the fourth control (which will become active if the noise amplitude is not equal to 0) allows the frequency of the sinusoidal noise to be changed.

### ***RC Circuit Low-Pass Filter on LVDT***

Fig. 14 shows the control to set the RC value of the low-pass filter circuit on the LVDT.

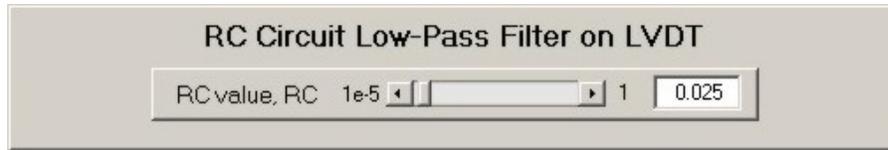


Fig. 14. Control for setting the RC value of the low-pass filter.

### ***Running Simulation and Plotting***

The bottom section of the GUI, shown in Fig. 15, contains the controls for running the simulation, storing the results, and plotting.

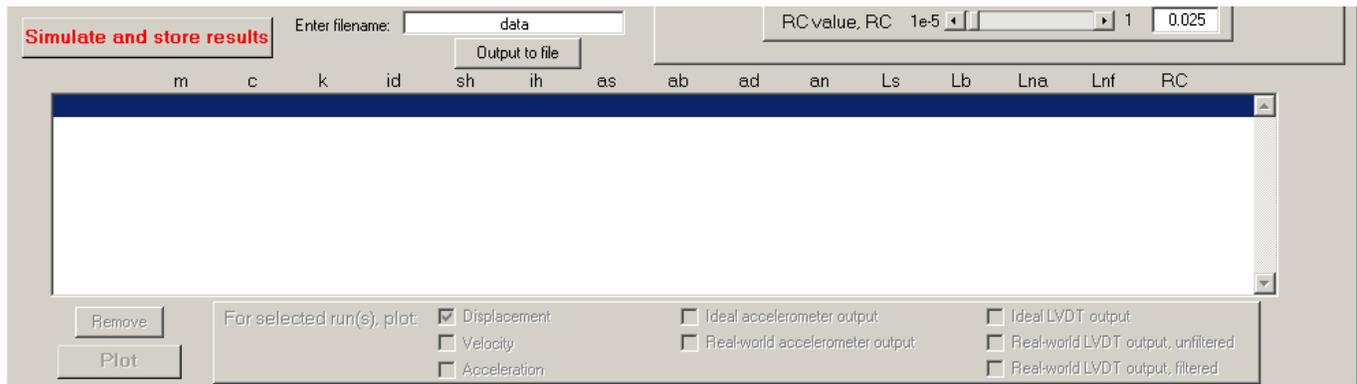


Fig. 15. Controls for simulating, storing results, and plotting.

Press the “Simulate and store results” button to run the Simulink model and store the results. Depending on the speed of the computer being used, this may take a few moments. When the simulation is complete, a label for the run such as “Run 1” will appear in the list box, followed by the parameter values used in that run. Above each column is the one or two letter abbreviation for the parameter listed. Each time the “Simulate...” button is pressed, a new row appears in the list box.

Clicking the “Remove” button will delete each selected row in the box. To select multiple rows hold down the Control key while clicking.

If there are data listed in the list box, the plotting controls will become active. To plot data, select a run or runs in the list box and then check the boxes for the desired output (displacement, velocity, etc). Clicking the “Plot” button will then cause a figure to appear with the desired plots, as shown in Fig. 16.

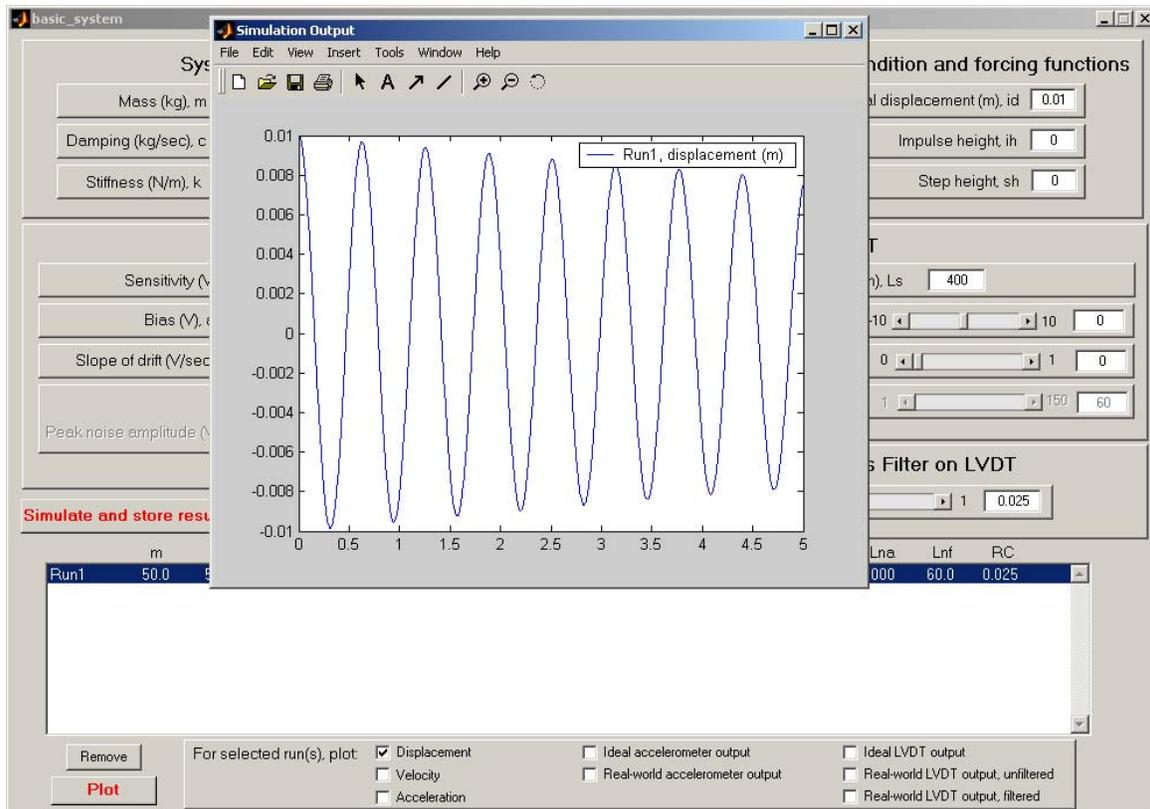


Fig. 16. Plotting results.

The same output can be plotted for multiple runs, different outputs from the same run can be plotted, or any combination of these.

### *Outputting Data to File*

The output from the Virtual Measurement System GUI can be output to a file, which can then be imported into Excel. To do this, enter the desired filename ("data" is the default) into the text box shown in Fig. 17.



Fig. 17. Control for outputting data to file.

Then click "Output to file." All of the available data (displacement, velocity, etc.) for the selected run, along with the model properties for that run, will be saved in a file in the current working directory.