AC 2010-1427: DEVELOPMENT AND INITIAL ANALYSIS OF A MINI CNC RAPID DEVELOPMENT SYSTEM

Lie Tang, Missouri University of Science and Technology

Robert Landers, Missouri University of Science and Technology

Hong Sheng, Missouri University of Science and Technology

Richard Hall, Missouri University of Science and Technology

Development and Initial Analysis of a Mini CNC Rapid Development System

Abstract

This paper describes the development of a mini Computer Numerical Control (CNC) Rapid Development System (RDS). The mini CNC RDS, which is based on Matlab Simulink, provides the student, within the context of a semester long course, with a tool to automate the major components of a table top three–axis machine tool. The major components of the mini CNC include three linear axes and a spindle. The mini CNC RDS allows the student to model and analyze the dynamics of the major components, design and analyze controllers for these components, and design and analyze interpolators for the linear axes. The analyses are done both virtually and experimentally. With this tool the student is able to explore all phases of automation development (i.e., simulation, emulation, and implementation). The student encodes their dynamic models, controllers, and interpolators as subsystems in Matlab Simulink. The inputs and outputs of each subsystem, along with their engineering units, are carefully specified. The student then utilizes the mini CNC RDS to analyze the performance of their dynamic models, controllers, and interpolators. The mini CNC RDS has three modes: simulation, emulation, and implementation. In the simulation mode the student simulates the three linear axis and spindle system dynamic responses for a variety of command voltage signals. The student can specify the magnitude and frequency of square, triangle, and sinusoidal command voltage signals, or they can create their own command voltage signal. In this mode the student can check their dynamic model for obvious errors (e.g., instability). In the simulation mode the student can also simulate the equipment controllers and interpolators with their dynamic models. In this case, the command voltage signals are not specified. Rather, they are generated by the controllers based upon the reference signals sent from the interpolators and the simulated linear axis position and spindle velocity measurements from the dynamic models. In the emulation mode, the simulation is executed on the target computer to ensure timing requirements are met. In the implementation mode, the controllers and interpolators are executed on the real linear axis system, while the simulation is executed in parallel. Simulation and experimental data are gathered and compared. This paper describes the development of the mini CNC RDS.

Introduction

A Rapid Development System (RDS) for a Linear Axis was developed in [1]. A RDS is a software environment that allows students to rapidly integrate their controller and analyze it via simulation, emulation, and implementation. In the simulation mode the student simulates a linear axis system that includes their controller and detailed models of the interface hardware and linear axis. In the emulation mode, the simulation is performed on the computer hardware that will implement the controller. In this mode the student can ensure their algorithm will run in real time (i.e., the algorithm's execution time is less than the sample period). In the implementation mode, the controller is deployed on the hardware system and experimental data is gathered. The Linear Axis RDS aided the students in the implementation of their controllers, even if the student had very little knowledge of control system hardware. The Linear Axis RDS also allowed the students to spend more time on controller analysis than if they had to also program the hardware.

Building on the results obtained in [1], this paper develops a RDS for a mini CNC. The mini CNC RDS was implemented in a graduate level controls course and informal feedback was given.

Mini CNC Rapid Development System

A Rapid Development System was developed and applied to a mini CNC (Figure 1). The mini CNC consists of three orthogonal linear axes and a spindle. A host computer, which has a non real–time operating system, runs the mini CNC RDS Graphical User Interface (GUI), as well as the models, controllers, and interpolators in simulation mode. A target computer, which has a real–time operating system, runs the models, controllers, and interpolators in emulation and implementation modes. The target computer has a National Instruments (NI) 6711 digital to analog (D/A) output board and a NI 6602 counter–time (C/T) board. The target computer, which utilizes the xPC real–time operating system, sends voltage commands to the motor amplifiers through the output board and receives motor encoder measurements through the C/T board, both at a sampling rate of 1 kHz. The mini CNC RDS provides three functions, i.e., Equipment Model, Equipment Control, and Interpolation, which cover three of the major issues during CNC machine development. These three functions are described below.

The Equipment Model function is used to calibrate the students' models of the mini CNC components (i.e., x, y, and z linear axes, and spindle). Three modes are provided for the student to create and test their dynamic models: System Test, Simulation, and Model Validation. The System Test mode allows the student to collect the dynamic response data of mini CNC components for different command voltage signals. The dynamic response data can be used to create a dynamic model of the components. The System Test mode Simulink model is shown in Figure 2. The command voltage is generated in the Voltage Signal subsystem and sent to the physical system via the Input/Output (IO) Interface subsystem, which processes with analog command voltage and encoder signals. The command voltage and measured position are recorded for further analysis. The Simulation mode allows the student to run the Simulink model on the host computer with different command voltage signals, such as square, triangle, sinusoid, etc. The Simulation mode Simulink model is shown in Figure 3. The command voltage generated in the Voltage Signal subsystem is sent to the component model via the Computer–System Interface subsystem, which simulates the physical input/output system effects such as quantization, saturation, etc. The command voltage and measured position are recorded. The Model Validation mode allows the student to compare the modeled dynamic response with the measured dynamic response for the same command voltage signal. This is beneficial because the student can directly observe the differences between the modeled and measured dynamic responses. The Model Validation Simulink model is shown in Figure 4. The command voltage from the Voltage Signal subsystem is sent to both the physical system and system model simultaneously. The command voltage, model dynamic response, and measured dynamic response are recorded for further analysis.

Axis and spindle controller design and implementation are important steps in CNC machine design. The Equipment Control function allows the student to test the mini CNC component controllers in three different modes: Simulation, Emulation, and Implementation. The Simulation mode allows the student to run the controller Simulink models on the host computer with models simulating the component dynamic responses. The Simulation mode serves as the first step for controller test and parameter calibration. The Simulation mode Simulink model, shown in Figure 5, contains four subsystems: Reference Generator, Controller, Computer–System Interface, and Component Model. The Reference Generator subsystem contains code that generates the reference signal, which is sent to the controller. The Controller subsystem contains the student's controller, which receives the reference and measured signals from the Reference Generator and Component Model subsystems, respectively, and calculates the command voltage according to the controller algorithm. This signal is sent to the Computer–System Interface subsystem, which simulates the effects (i.e., quantization and saturation) of the physical input/output hardware (i.e., the counter timer and analog input data acquisition and control cards). The Component Model subsystem contains the Simulink code describing the component dynamics. The reference signal, measured signal, and commanded voltage are recorded and used for controller performance analysis. The Emulation mode allows the student to run their controller on the target computer in real–time without the physical system. The component dynamics is simulated on the target computer. The Emulation mode serves as an intermediate step that allows the student to determine if the target computer is able to perform all controller computations and data input/output tasks within the specified sample period. The Emulation mode Simulink model contains the same four subsystems as the Simulation mode Simulink model. However, the IO Interface subsystem is different. Instead of simulations of the physical input/output hardware, the IO Interface subsystem contains the actual software drivers to interface with the encoders and motor amplifiers. Although these drivers are implemented in the Emulation mode, the data is not used in the simulation of the physical components. The Implementation mode allows the students to test their controller on the physical system. The Implementation mode Simulink model, shown in Figure 6, contains three subsystems: Reference Generator, Controller, and IO Interface. The IO Interface subsystem is the same as that used in the Emulation mode. The Component Model subsystem is not utilized in the Implementation mode since the controller is operating on the physical components and not simulations of the physical components. Note that the same Controller subsystem is utilized in all three modes.

Interpolators are used to generate the reference positions for the linear axes. The Interpolator test function incorporated in the mini CNC RDS provides a convenient platform for the students to test their linear and circular interpolators. The Simulink model for the Interpolator test function is shown in Figure 7. The Interpolator test Simulink model consists of three subsystems: Executive Controller, Linear Interpolator, and Circular Interpolator. The Executive Controller subsystem manages the dispatch of new segment types and the corresponding information (e.g., reference velocity, end point, circle radius) to the linear and circular interpolators, depending on the commanded segment type. During the interpolator test, the Linear Interpolator and Circular Interpolator subsystems are replaced by the corresponding student modules.

Mini CNC RDS

The mini CNC RDS was developed using the GUIDE function in Matlab. Graphical User Interfaces (GUIs) are provided for Equipment Model, Equipment Control, and Interpolator functions. The operations of these GUIs are described in the follow section.

The Equipment Model GUI is used to build and test the Simulink model containing the linear axis and spindle models developed by the student. As shown in Figure 8, the Equipment Model GUI is separated into the following sections: Operation, Simulink Model Builder, Voltage Signal, Animation, Parameters, Execution, and Results. The Operation section includes two pushbuttons: Main Menu and Jog, which direct the user to the main menu and jog GUIs, respectively. The Simulink Model Builder section is used to build the Simulink model (i.e., integrate the student's code and selections into one Simulink file). It incorporates four categories: System, Mode, Voltage Signal, and Model. The System category has four options: X axis, Y axis, Z axis, and Spindle, which determines the model the student intends to test. The Mode category has three options: Simulation, System Test, and Model Validation, which are described above. The Voltage Signal category has four options: Square, Triangle, Sinusoid, and User Defined. The student can use the first three predefined signals or insert their own voltage signal by choosing the last option. The Model category has two options: Default and User Defined. The Default option provides a benchmark with which the student can compare the performance of their own model. After selecting all of the settings, the Build pushbutton is pressed to build the Simulink model. Once the model is built, it can be viewed by pressing the View Model button. The voltage amplitude and frequency can be set in the Voltage Signal section if the voltage signal category option is not User Defined. An animation of the mini CNC is shown at the center of GUI. The animation provides quasi real time motion of the components. The simulation final time and sample period can be designated in the Parameter category. Once all of the settings are properly made, the student can run the model using the pushbutton Simulate/Run, depending on the mode option. If the System Test or Model Validation mode is selected, the corresponding Simulink model is downloaded to the target processor by pressing Load Model. In the Results section, three options are provided for results processing: Plot, Save, and Email. The results can be directly plotted on the screen using the Plot option or saved in a file using the Save option. The saved results can be emailed to the designated recipients using the Email option.

The Equipment Control test GUI is used to build and test the Simulink model containing the axis and spindle controllers. The Equipment Control test GUI, as shown in Figure 9, consists of the following sections: Operation, Simulink Model Builder, Reference, Animation, Parameters, Execution, and Results. Similar to the Equipment Model test GUI, the Operation section of the Equipment Control test GUI consists of two pushbuttons: Main Menu and Jog, which directs the user to the main menu and jog GUI, respectively. The Simulink Model Builder section is used to build the Simulink model and consists of five categories: System, Mode, Reference, Model, and Controller. The System category has four options: X axis, Y axis, Z axis, and Spindle, which specifies the component to be controlled. The Mode category consists of three options: Simulation, Emulation and Implementation, which are described above. The Reference category provides three reference signals: Square, Triangle, and Sinusoidal. The Model category is used to select the component model, which can be Default or User Defined. The Controller category contains five controller options: Proportional, Proportional plus Integral plus Derivative (PID), Modified PID, General Tracking Controller, and User Defined Controller. The student can use the first four controllers, Proportional, PID, modified PID, and General Tracking Controller, to operate the RDS without having to create their own controller. The controller gains for these four controllers have been tuned using trial and error methods. These controllers provide a benchmark with which the students can compare the performance of their own controller. The User Defined Controller option allows the student to test their own controller. When the proper settings are selected for these five categories of settings, a Simulink model is created using the Build pushbutton. The model can be viewed using the View Model pushbutton once the model is created. The reference amplitude and frequency can be designated in the Reference section. The Animation section, shown at the GUI center, is again used to show the motion of the mini CNC components. The simulation final time and sample period can be set in the Parameter section. Once the settings are made properly, the student can run the Simulink model using the pushbutton Simulate/Run depending on the Mode option. If System Test or Model Validation is selected, the model is downloaded to the target processor by pressing the Load Model button. The results can be plotted or saved to a file using the Plot and Save options, respectively, provided in the results section. The saved results can be emailed to the designated recipients using the Email option.

The Interpolator test GUI is used to test linear and circular interpolators designed by the student. The Interpolator test GUI, shown in Figure 10, consists of the following sections: Operation, Simulink Model Builder, Linear Interpolator Settings, Feedrate Setting, Simulation Parameter Settings, Animation, Circular Interpolation Settings, Execution, and Results. The Operation section provides two pushbuttons, Main Menu and Help, which return the user to the Main Menu GUI or Matlab help browser, respectively. Help documents are formatted as html files and integrated within the Matlab help system. The Simulink Model Builder section consists of four drop down lists (i.e., Mode, Linear Interpolator, Circular Interpolator, and Test) for Simulink model component settings and two pushbuttons for Simulink Model Build and View. The Mode list offers three options: Linear Interpolator Test, Circular Interpolator Test, and Linear&Circular Interpolator Test. The Linear Interpolator category contains two options: Default and User Defined. A default constant velocity linear interpolator is provided in the program, and the user can use the User Defined option to designate the linear interpolator. The Circular Interpolator category offers two options: Default and User Defined. The Test category determines the type of test used in the Simulink model to be built. Linear single segment, Linear multiple segments, and Default linear motion G code file are options for the linear interpolator test. Circular single segment, Circular multiple segments, and Default circular motion G code file are options for the circular interpolator test. Default G code file and User defined G code file are options for the linear and circular interpolator test. Upon completion of these settings, the Simulink model for the interpolator test can be initiated using the Build Simulink Model pushbutton. The Simulink model can be viewed using the View Model pushbutton. The Linear Interpolation section is used to set parameters for a single segment linear interpolation test, while the Circular Interpolation section is used to set parameters for a single segment circular interpolation test. For a linear or circular multiple segments interpolator test, a text file contains the corresponding G code is used. The Simulation Parameter section allows settings of the simulation final time and sample period. The simulation can be initiated or stopped using the pushbutton Simulate or Stop, respectively, in the Execution section. The outputs from the interpolator include reference position, reference velocities, and reference accelerations, and can be plotted, saved, and emailed using the options offered in the Results section.

The Help system is an important part of the mini CNC RDS program. As shown in Figure 11, the help files for mini CNC RDS are formatted as html files and integrated within the Matlab help browser.

Results

The mini CNC RDS was used by fifteen students in a graduate level controls course. This course covers state space control and is populated by first year graduate students and senior level undergraduate students as an elective. As an example, one students' implementation is described below. Following this, the results of all fifteen students are presented.

The students are required to design general tracking controllers and reduced order observers for the X, Y, and Z axes. The axis dynamics are described by the differential equation

$$\tau\ddot{x}(t)+\dot{x}(t)=Ku(t)-F_C \tag{1}$$

where τ is the time constant (s), $x(t)$ is the axis position (mm), K is the gain ((mm/s)/V), and F_C is the Coulomb friction (mm/s), which is modeled by

$$F_C=\begin{cases} F_{C+} & \text{if} \quad v(t)>0 \\ 0 & \text{if} \quad v(t)=0 \\ F_{C-} & \text{if} \quad v(t)<0 \end{cases} \tag{2}$$

where $v(t)$ is the axis velocity (mm/s). The X, Y, and Z axis parameters are listed Table 1. Ignoring Coulomb friction and using position and velocity as the system states, the linear axis state variable model is

$$\begin{aligned} \dot{x}(t)&=v(t) \\ \dot{v}(t)&=-\frac{1}{\tau}v(t)+\frac{K}{\tau}u(t) \end{aligned} \tag{3}$$

Letting $e_x(t)=x_r(t)-x(t)$ and $e_v(t)=\dot{x}_r(t)-v(t)$, the error dynamics can be written as

$$\begin{aligned} \dot{e}_x(t)&=e_v(t) \\ \dot{e}_v(t)&=-\frac{1}{\tau}e_v(t)+\ddot{x}_r(t)+\frac{1}{\tau}\dot{x}_r(t)-\frac{K}{\tau}u(t) \end{aligned} \tag{4}$$

where $x_r(t)$ is the reference position (mm). The pseudo control voltage is

$$\mu(t) = \ddot{x}_r(t) + \frac{1}{\tau}\dot{x}_r(t) - \frac{K}{\tau}u(t) \tag{5}$$

Equation (4) is rewritten in state space format as

$$\begin{bmatrix} \dot{e}_x(t) \\ \dot{e}_v(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} e_x(t) \\ e_v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu(t) \tag{6}$$

The pseudo control voltage is

$$\mu(t) = -g_1 e_x(t) - g_2 e_v(t) \tag{7}$$

where g_1 and g_2 are controller gains. The gains are selected to shape the closed–loop dynamic response of the linear axis. The physical control voltage is

$$u(t) = \frac{\tau}{K}\left(\ddot{x}_r(t) + \frac{1}{\tau}\dot{x}_r(t) + g_1 e_x(t) + g_2 e_y(t) \right) \tag{8}$$

Since axis speed cannot be measured directly, a reduced order observer is designed for axis speed estimation. The speed estimation is determined by

$$\begin{aligned} \hat{v}(t) &= Lx_m(t) + z(t) \\ \dot{z}(t) &= Fz(t) + \bar{G}x_m(t) + Hu(t) \end{aligned} \tag{9}$$

where L, F, $\bar{G}$, and H are observer parameters, $z(t)$ is an auxiliary state, $\hat{v}(t)$ is the estimated speed (mm/s), and $x_m(t)$ is the measured position (mm). The closed–loop controller poles are placed at –32±24i, providing a 0.125 s settling time. The observer pole is placed at –320, providing a 0.0125 s settling time. The general tracking controller and observer parameters for the X, Y, and Z axes are listed in Table 2.

The Mini CNC RDS is used to implement the student's controller and observer when tracking a sinusoidal reference with an amplitude of 5 mm and a frequency of 0.2 Hz. The student's Simulink model used in the implementation, as shown in Figure 12, contains two subsystems, controller and observer. The controller and observer models are shown in Figures 13 and 14, respectively. The tracking results for X, Y, and Z axis are shown in Figure 15, 16, and 17, respectively. The mean value and standard deviation of the tracking error are listed in Table 3. The results are very good for a linear controller that does not take into account the inherent Coulomb friction in the linear axes.

The students provided informal feedback on the utility of the mini CNC RDS. Most students indicted the mini CNC RDS was a valuable learning tool and helped to reinforce the concepts they learned in class.

Summary and Conclusions

A mini CNC Rapid Development System (RDS) was developed in this paper. The mini CNC RDS allows students to model and control three linear axes and a spindle, as well as design and analyze linear and circular interpolators. The mini CNC RDS was implemented by fifteen graduate students in a graduate controls course at the Missouri University of Science and Technology. The mini CNC RDS allowed the students to spend more time on the controller design and tuning, than if they had to also program the hardware. Informal feedback provided by the students indicted the mini CNC RDS was a valuable learning tool and helped to reinforce the concepts they learned in class.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant Number 0736731.

References

[1] Fleming, M., Jain, V., Landers, R.G., Sheng, H., and Hall, R., 2009, "Implementation and Evaluation of a Linear Axis Rapid Development System," *ASEE Annual Conference and Exhibition*, Austin, Texas, June 14–17.

Table 1: X, Y, and Z axis model parameters.

Axis	τ (s)	K ((mm/s)/V)	F_+ (mm/s)	F_- (mm/s)
X	$9.943 \cdot 10^{-3}$	1.764	0.539	–0.222
Y	$1.044 \cdot 10^{-2}$	1.882	0.631	–0.325
Z	$1.163 \cdot 10^{-2}$	1.594	0.220	–0.802

Table 2: X, Y, and Z axis general tracking controller and observer parameters.

Axis	Controller gains		Reduced order observer parameters			
	g_1	g_2	L	F	$\bar{G}$	H
X	$1.60 \cdot 10^{3}$	–36.6	219.4	–320	$-7.02 \cdot 10^{4}$	177.4
Y	$1.60 \cdot 10^{3}$	–31.8	224.2	–320	$-7.17 \cdot 10^{4}$	180.3
Z	$1.60 \cdot 10^{3}$	22.0	234.0	–320	$-7.49 \cdot 10^{4}$	137.1

Table 3: General tracking controller tracking error.

Axis	Mean error (mm)	Error standard deviation (mm)
X	$2.42 \cdot 10^{-2}$	$1.73 \cdot 10^{-2}$
Y	$2.92 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$
Z	$1.63 \cdot 10^{-2}$	$1.38 \cdot 10^{-2}$

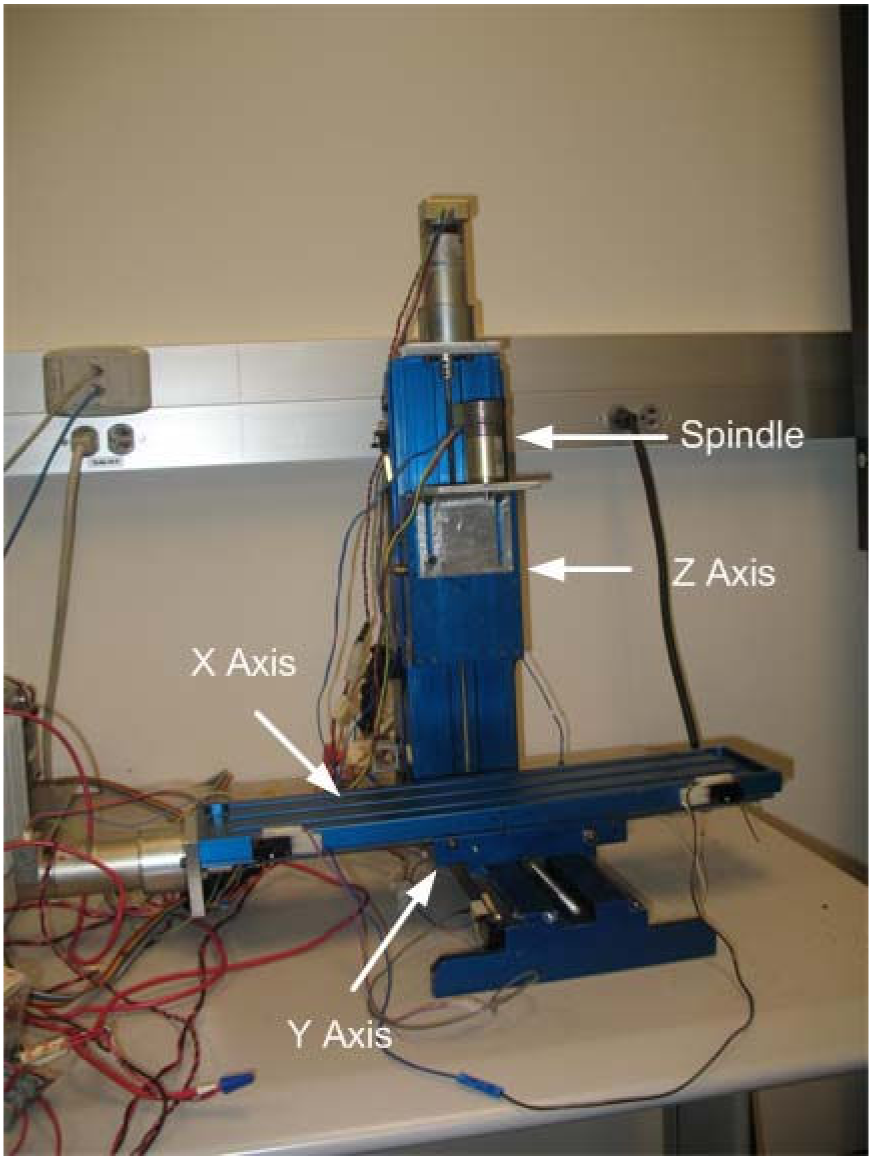

Figure 1: Mini–CNC System Used in RDS.

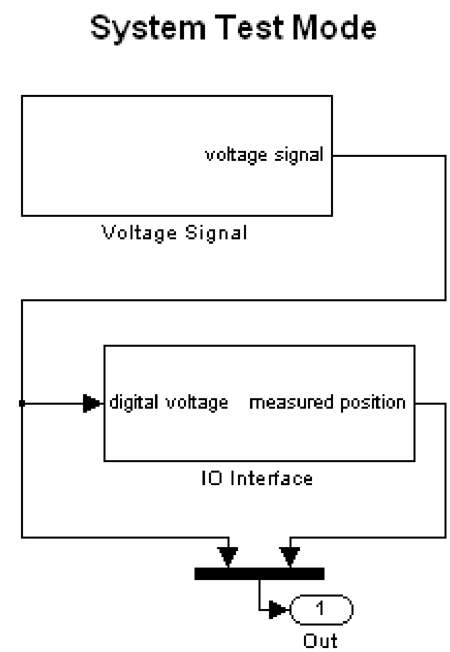

Figure 2: System Test mode Simulink model for Equipment Model function.

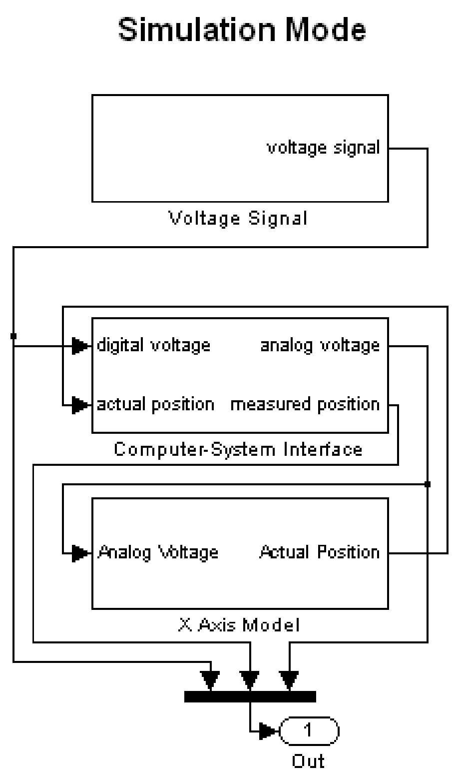

Figure 3: Simulation mode Simulink model for Equipment Model function.

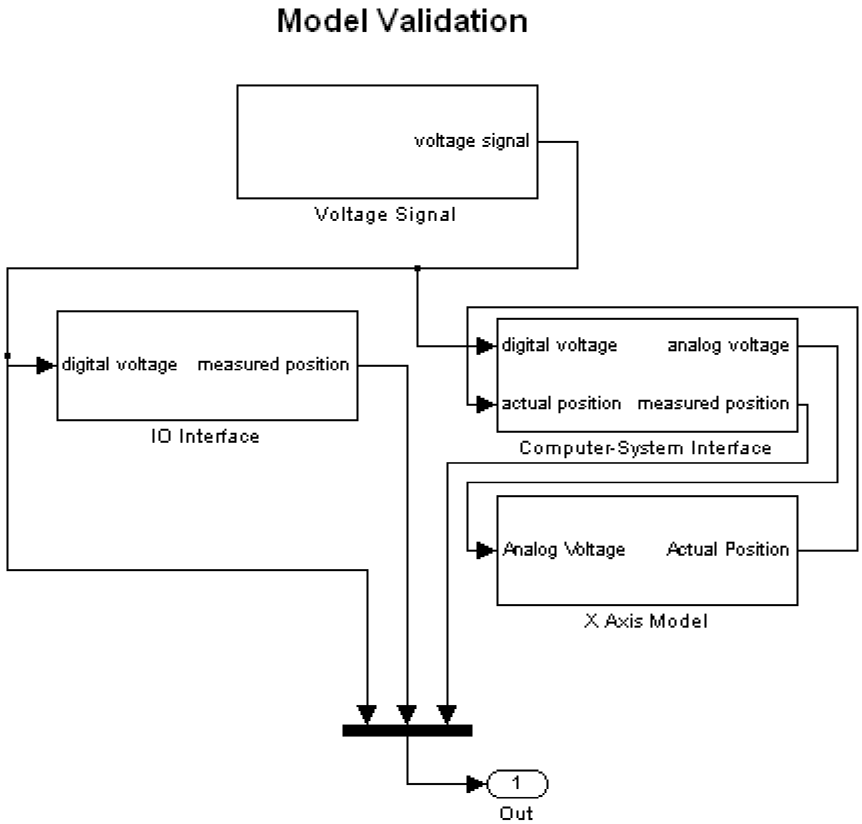

Figure 4: Model Validation mode Simulink model for Equipment Model function.

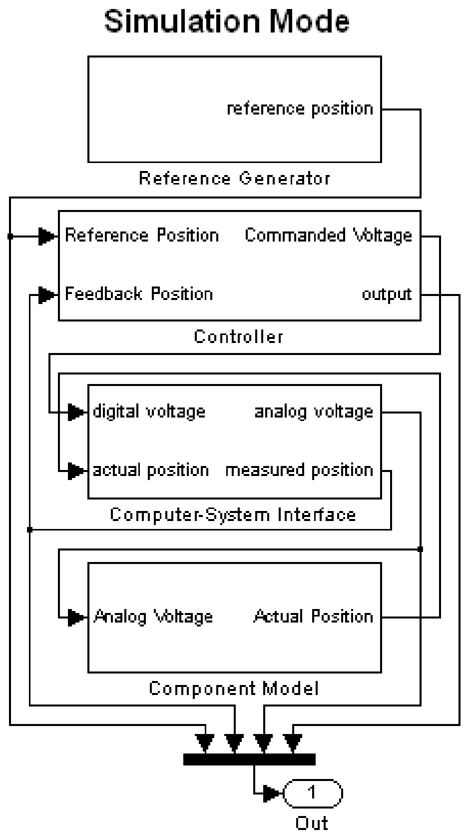

Figure 5: Simulation mode Simulink model for Equipment Control function.

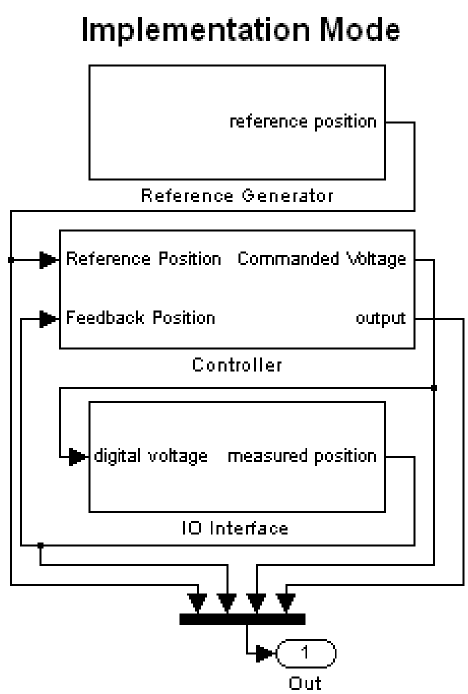

Figure 6: Implementation mode Simulink model for Equipment Control function.

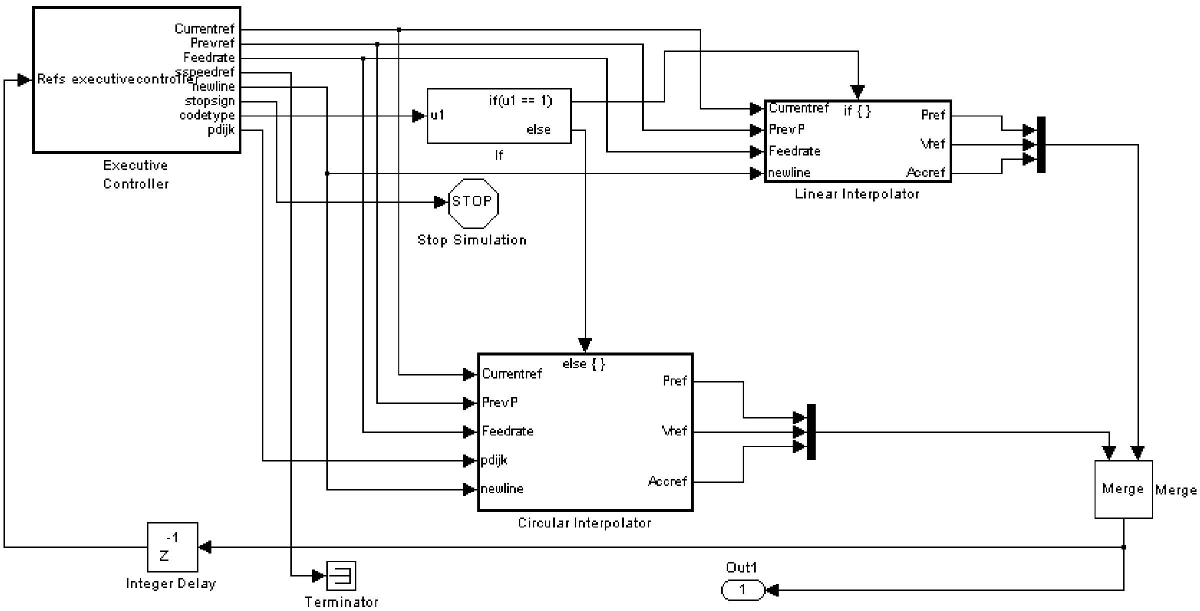

Figure 7: Interpolator test Simulink model.

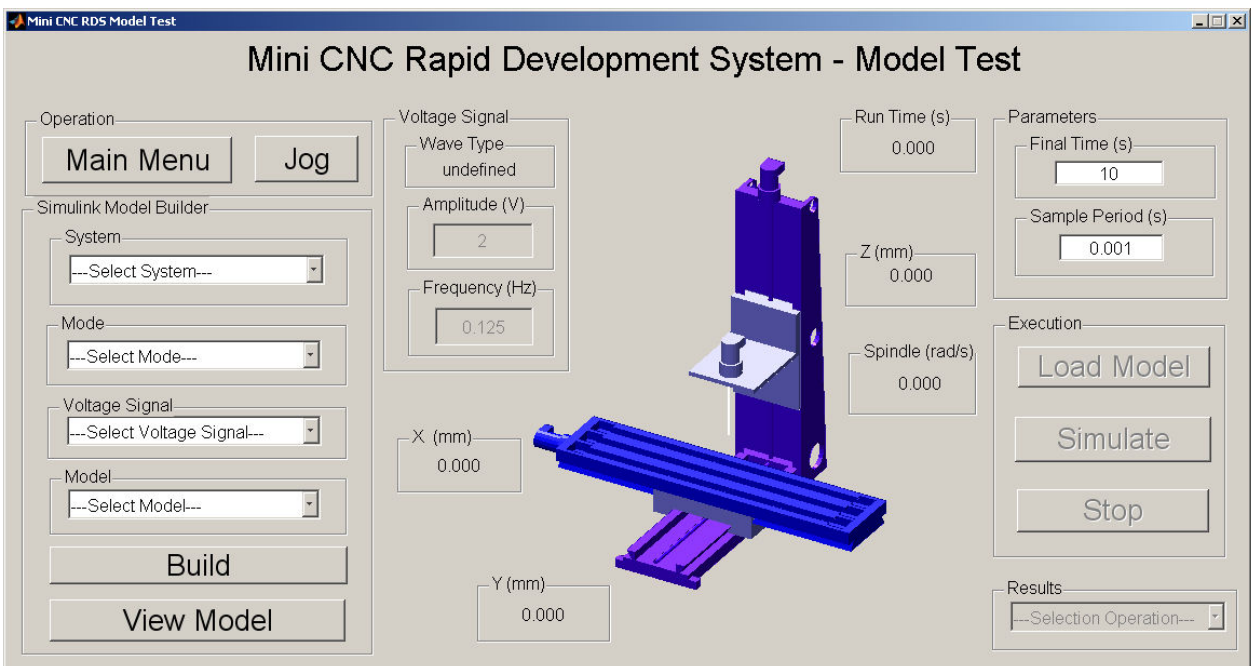

Figure 8: Mini CNC RDS Equipment Model GUI.

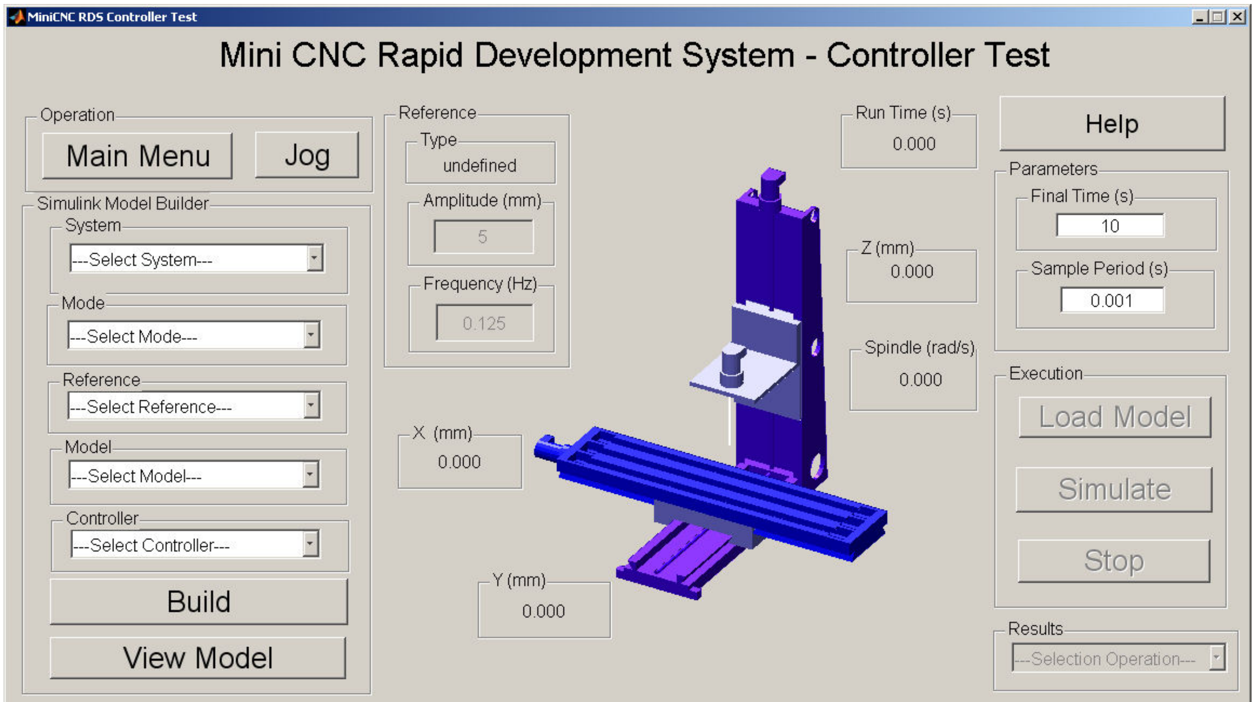

Figure 9: Mini CNC RDS Equipment Control GUI.

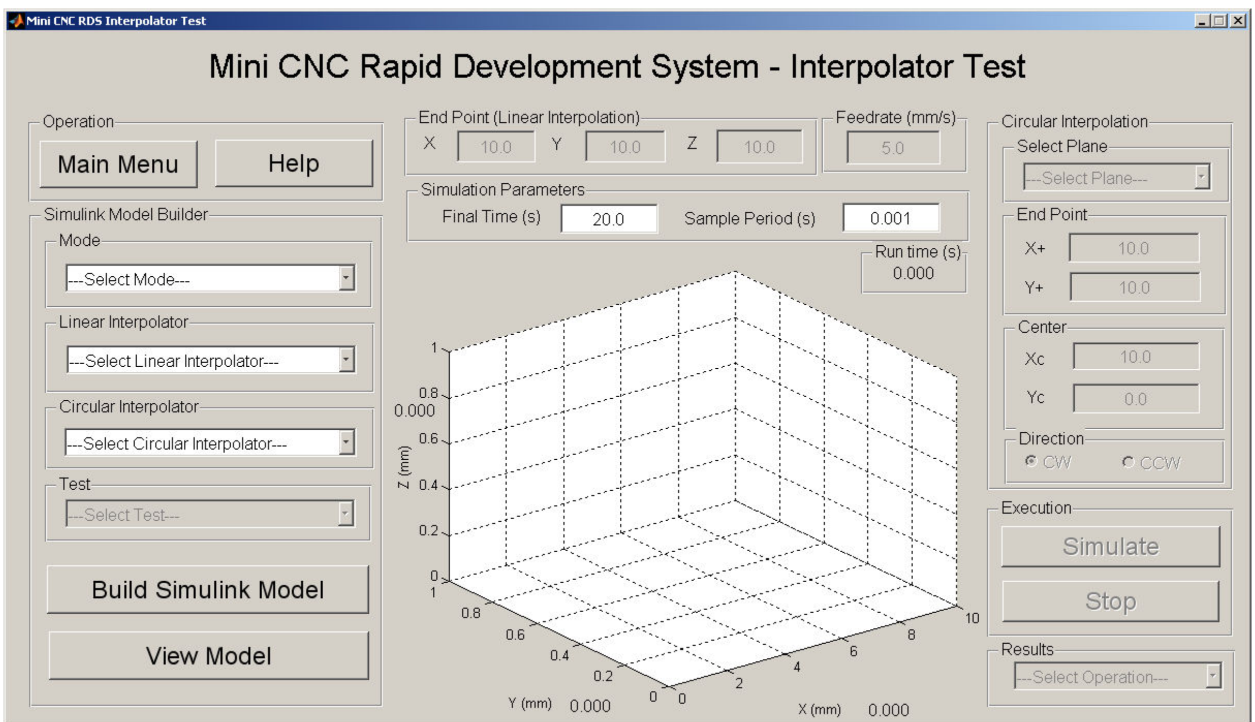

Figure 10: Mini CNC RDS Interpolator GUI.

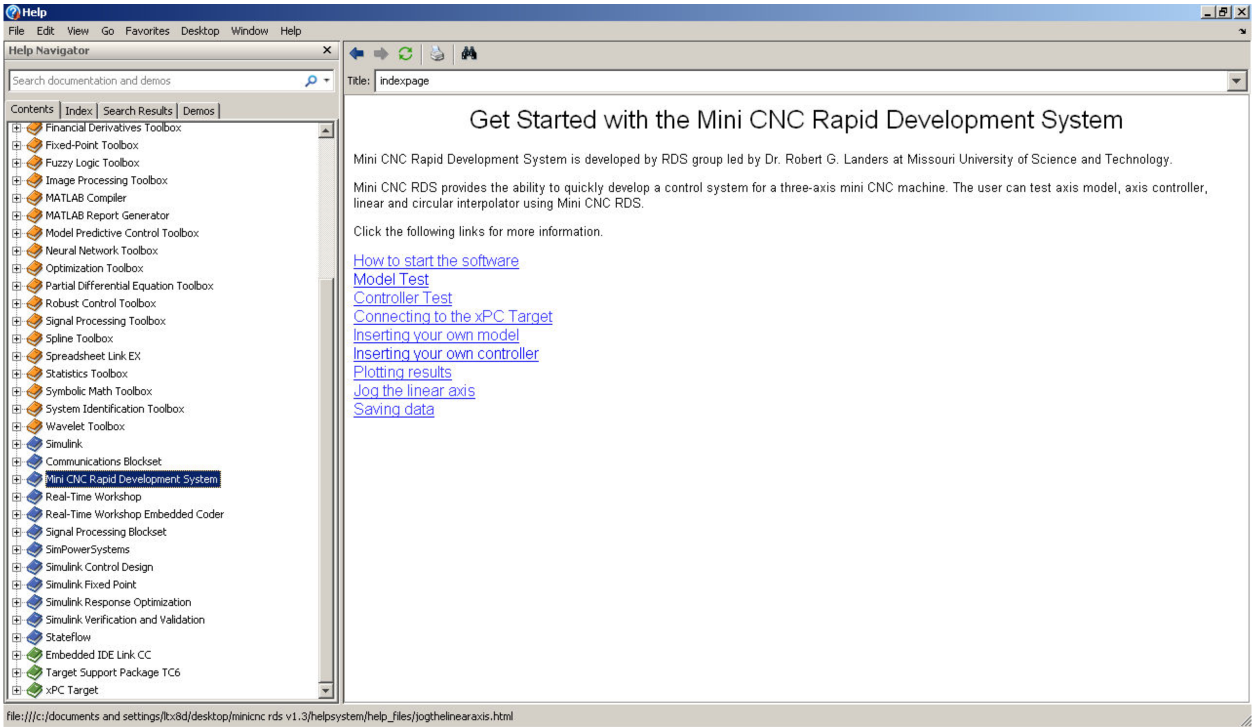

Figure 11: Mini CNC RDS help browser.

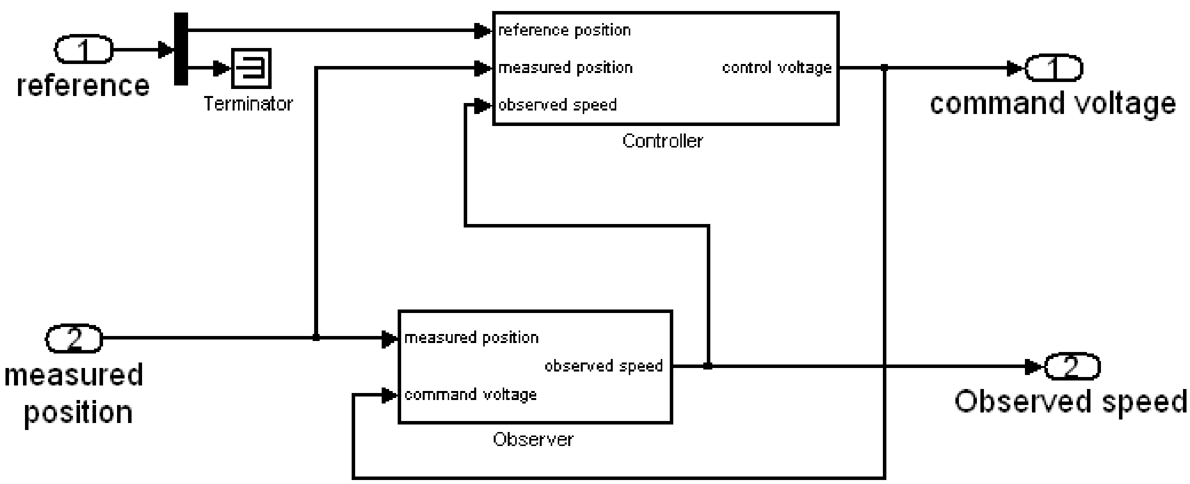

Figure 12: Student controller and observer Simulink model.

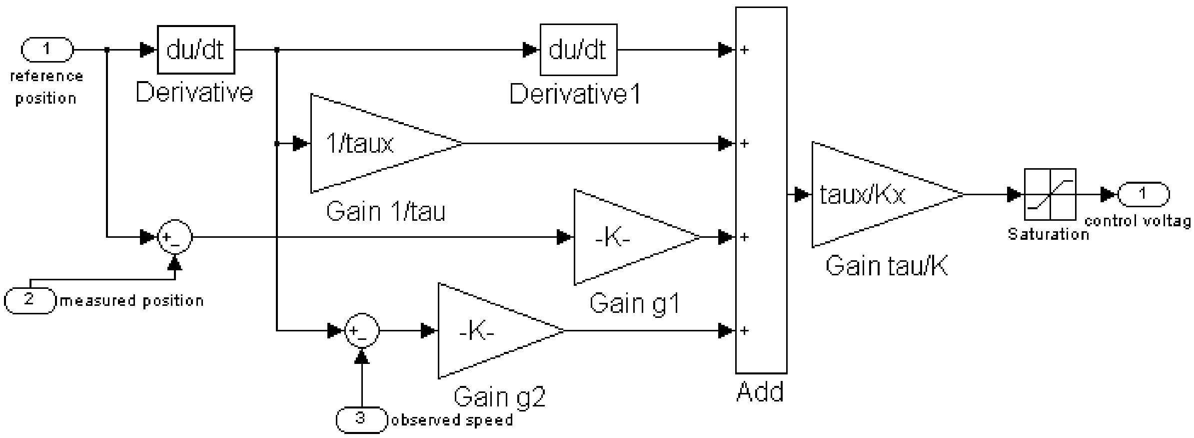

Figure 13: Student general tracking controller Simulink model.

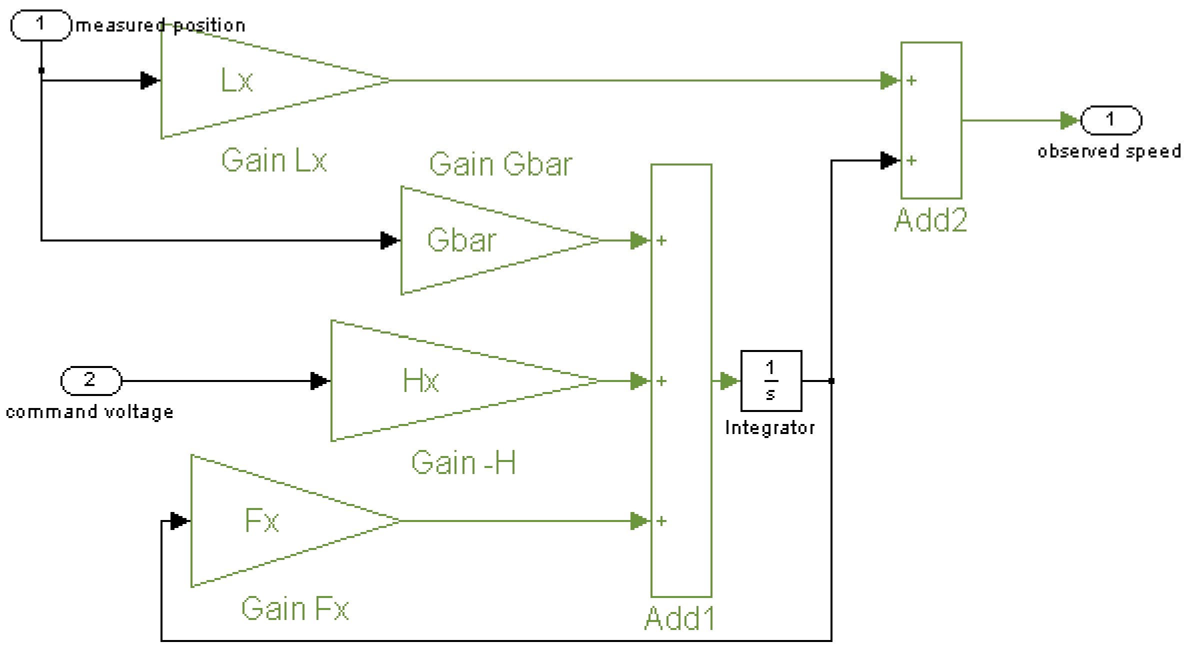

Figure 14: Student reduced order observer Simlink model.

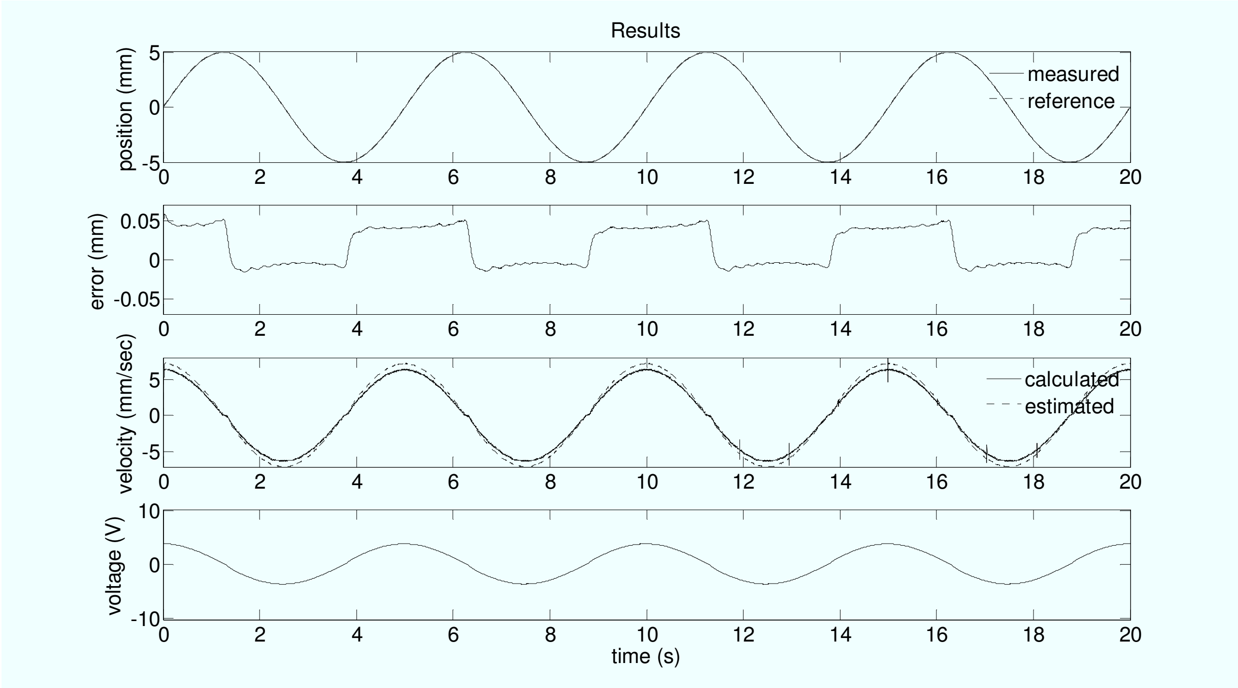

Figure 15: X axis tracking and velocity estimation results.

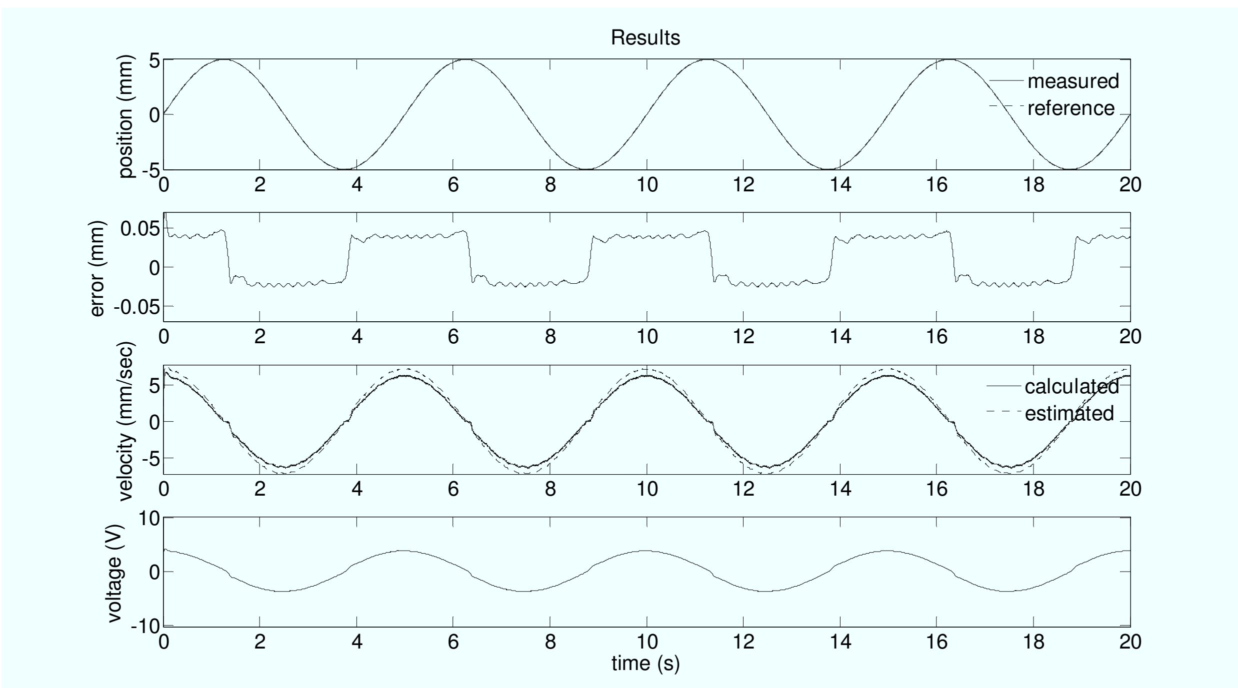

Figure 16: Y axis tracking and velocity estimation results.

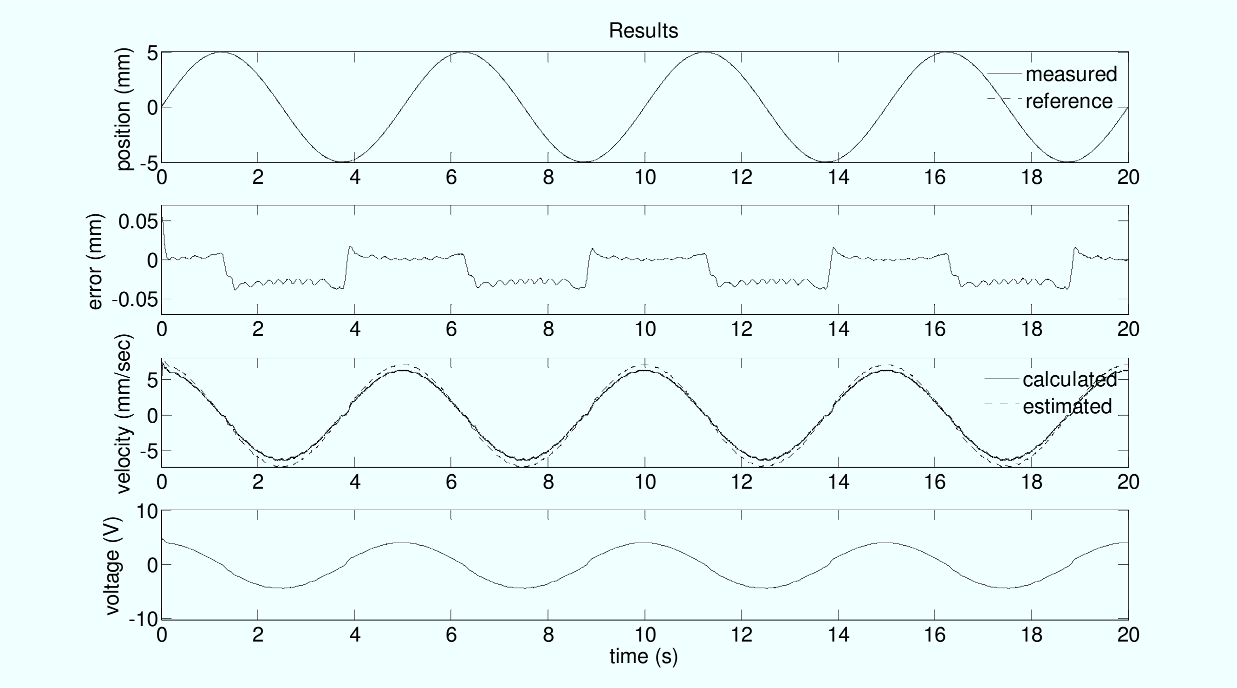

Figure 17: Z axis tracking and velocity estimation results.