Embedded Control and Mechatronics (ECE-456)

Farzad Pourboghrat

dsPIC Platform for PMDC Motor Modeling & Control using Matlab/Simulink & Other Mathworks Tools

User Guide

NARAYANAN RAMACHANDRAN (SAI) &
AISHWARYA VASU (HARINI)

MathWorks
Matlab / Simulink / Simscape
System Identification Toolbox
Control Toolbox
Control Design Toolbox

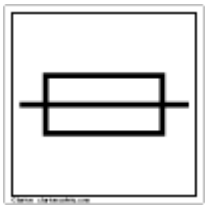
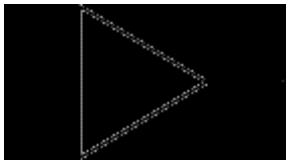
MICROCHIP
dsPIC
dsPIC Blockset by LK

RealTerm: Serial Capture Program

TABLE OF CONTENTS

Contents

Symbol Sheet

Caution 	Electric Shock 	Ground 	ESD 
Positive Polarity 	Negative Polarity 	DC 	AC 
ON 	OFF 	Output 	Variable 
MANUAL CTRL 	Fuse 	OPAMP 	Capacitor 
LED 	Motor 	Wire	Diode 
DISCONNECT BEFORE SERVICING 			

SAFETY SHEET

- Experiment needs to be done in the presence of authorized personnel (Teaching Assistant).
- Maximum AC voltage 110 to 120V, Maximum DC voltage 18V. Some terminals on the board are not shielded / insulated.

- All the Integrated circuits on the board are electrostatic sensitive devices. Make sure you discharge yourself before operating.

- If components are to be replaced or repaired on board, disconnect the power supply and wait for at least 30 seconds.

- This experiment consists of rotating equipment (DC motor, +/-4000 rpm), stay cautious while energized or ON/RUN condition.
- The board is naturally air cooled.
- IP 00 (No Dust/Solid/Liquid Protection).
- The setup is static and not portable when powered ON.

AGENDA

1. Estimation/tuning of the parameters of a permanent magnet DC motor (PMDC), modeled in Simscape, to match the measured I/O data.

2. Programming dsPIC microcontroller (from Microchip) using Lubin Kerhuel's dsPIC blockset, for motor control and data acquisition application.

3. Time-domain modeling: ARX model of the system, using system identification toolbox in Matlab.

4. Designing a "PID" controller for a closed-loop speed control application.

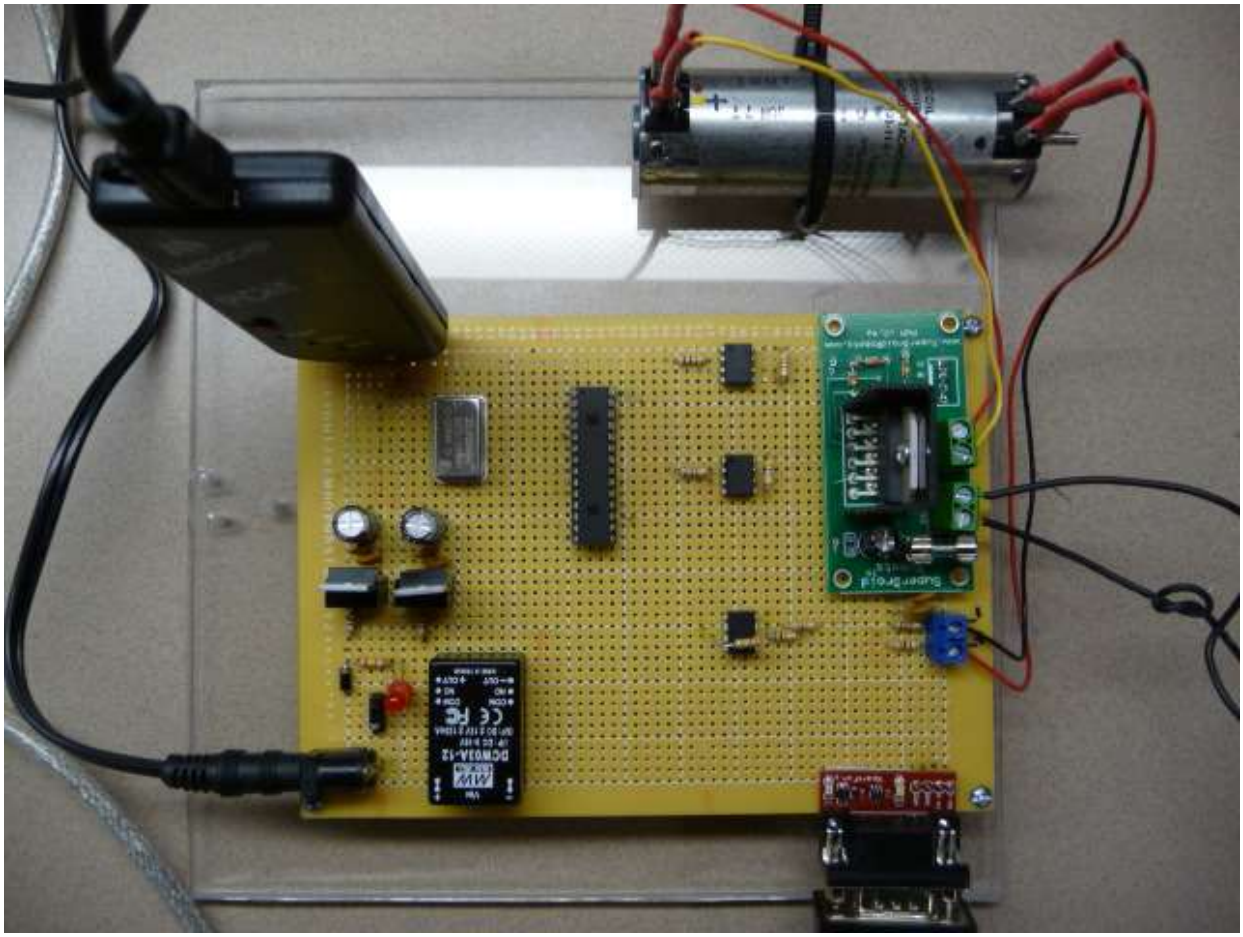
5. Rapid prototyping: redesigning controller for control implementation, target code generation, testing and verification in embedded hardware.

CHAPTER 1

HARDWARE MANUAL

PHOTOGRAPHIC PICTURE OF THE PLATFORM:

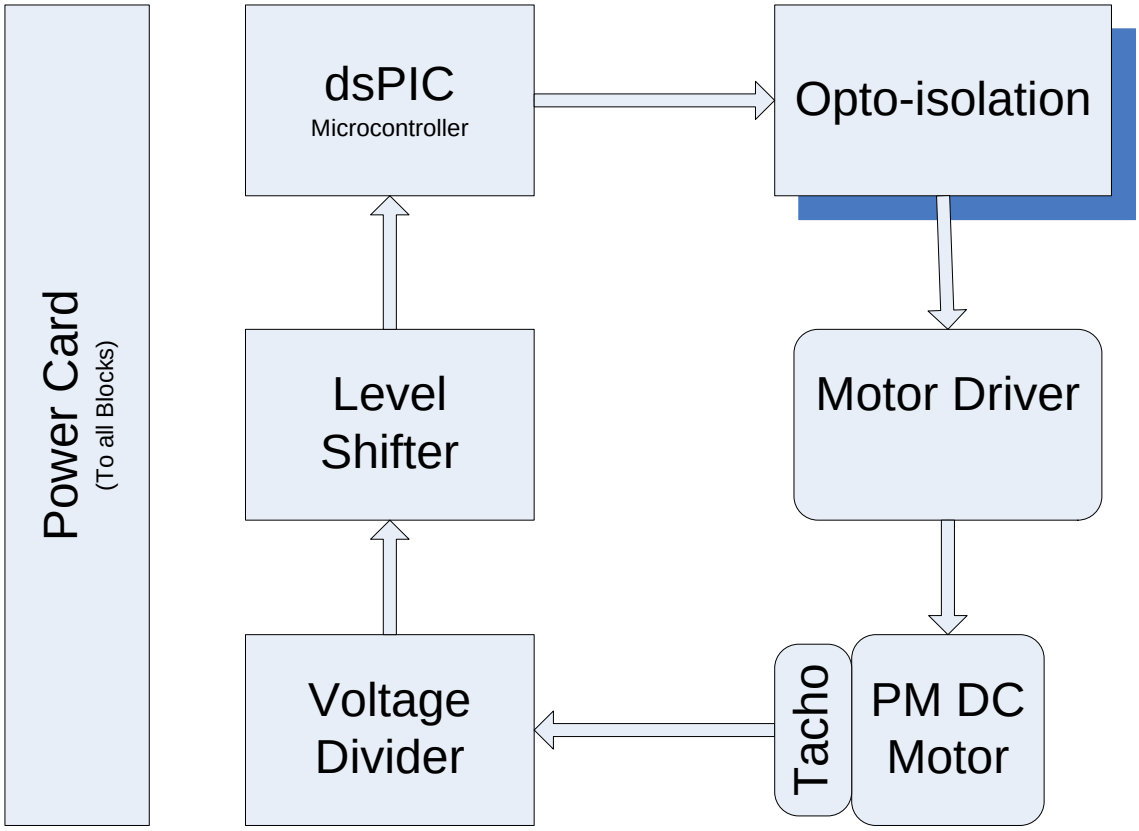
The platform consists of an integrated PMDC motor interfaced with a dsPIC30 microcontroller through a power drive. The platform is also interfaced with a host PC through USB serial port.



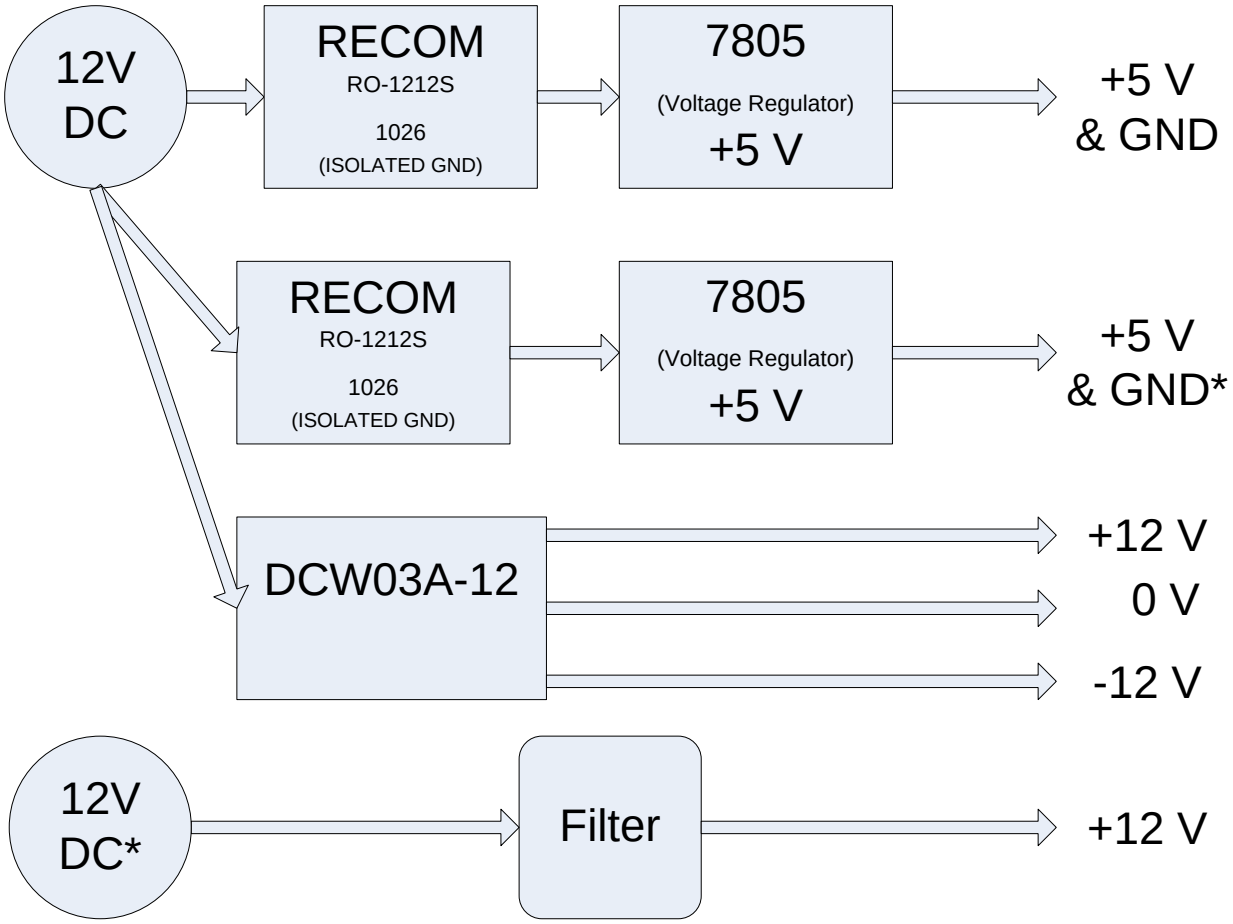
BLOCK DIAGRAM REPRESENTATION:

The system consists of the following components/blocks. These are categorized based on their functions.

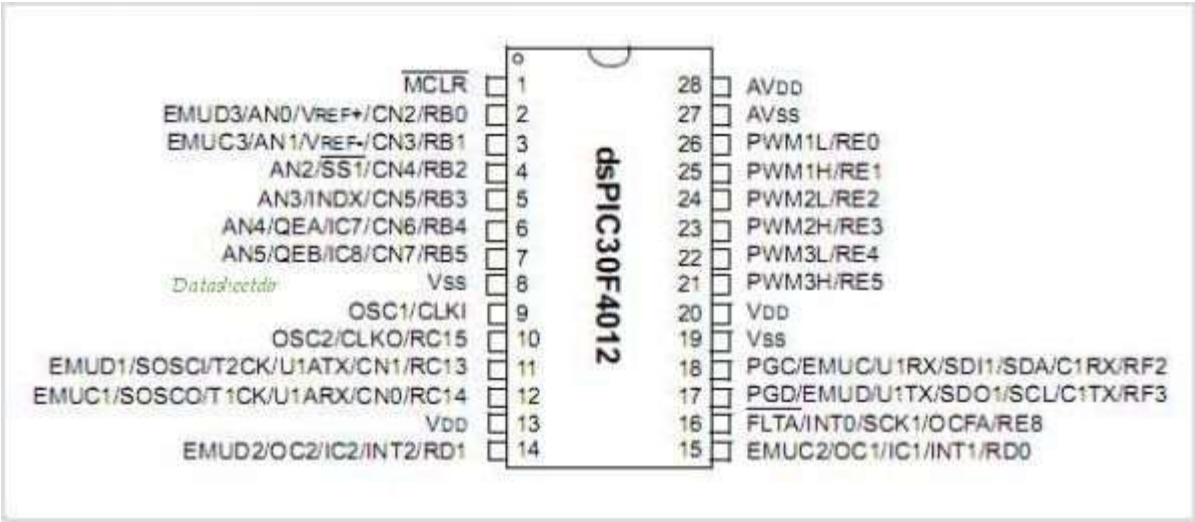
- Power Card (“Pwr card”)
- Microcontroller (dsPIC Family)
- Opto-isolator (6N137)
- Motor Driver (LMD18200T-H-Bridge or Wirz #203 RevB)
- PMDC Motor/Tacho (eBay Part No: 130103)
- Voltage Divider (resistor arm)
- Level Shifter (LM341)



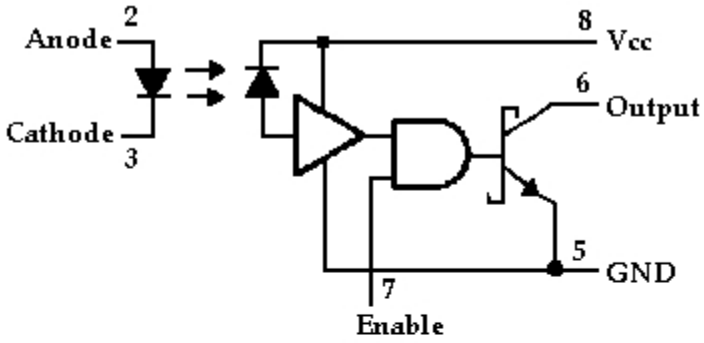
POWER CARD:

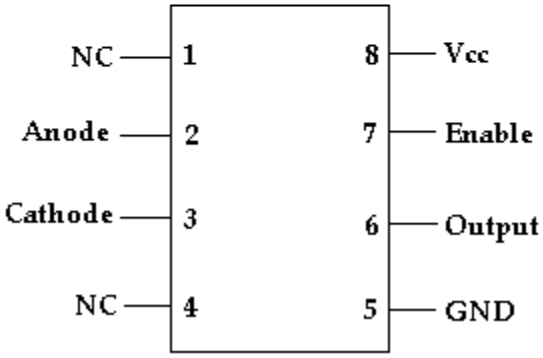


MICROCONTROLLER (dsPIC30f4012):



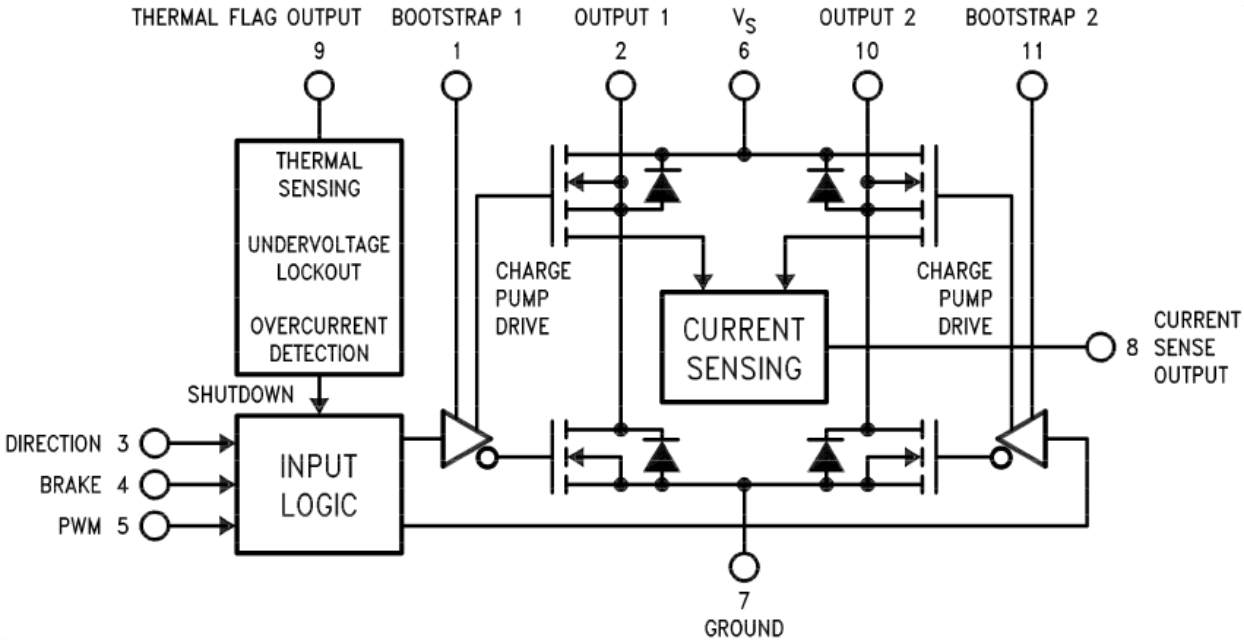
OPTO – ISOLATOR (6N137):





DC MOTOR DRIVER:

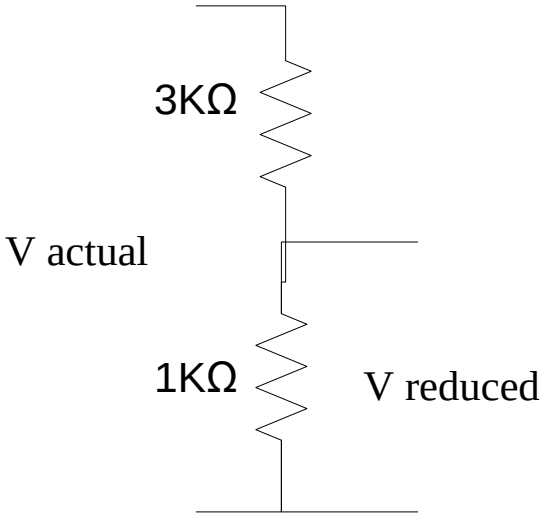
LMD18200T (3A, 55V)



PMDC MOTOR: +/- 4000 rpm, 12V, 7mH, 10Ω, 3.45 0z-in/amp

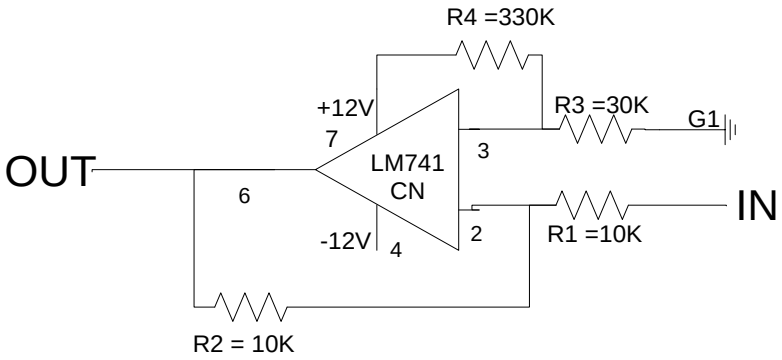
With “Tacho” as speed sensor: 1.9V/1000 rpm

VOLTAGE DIVIDER: (quarter watt resistor)



Maximum V actual is +/- 12V, to limit the input voltage to max of +/- 5V, the chosen resistor values are 3K and 1K .

LEVEL SHIFTER:



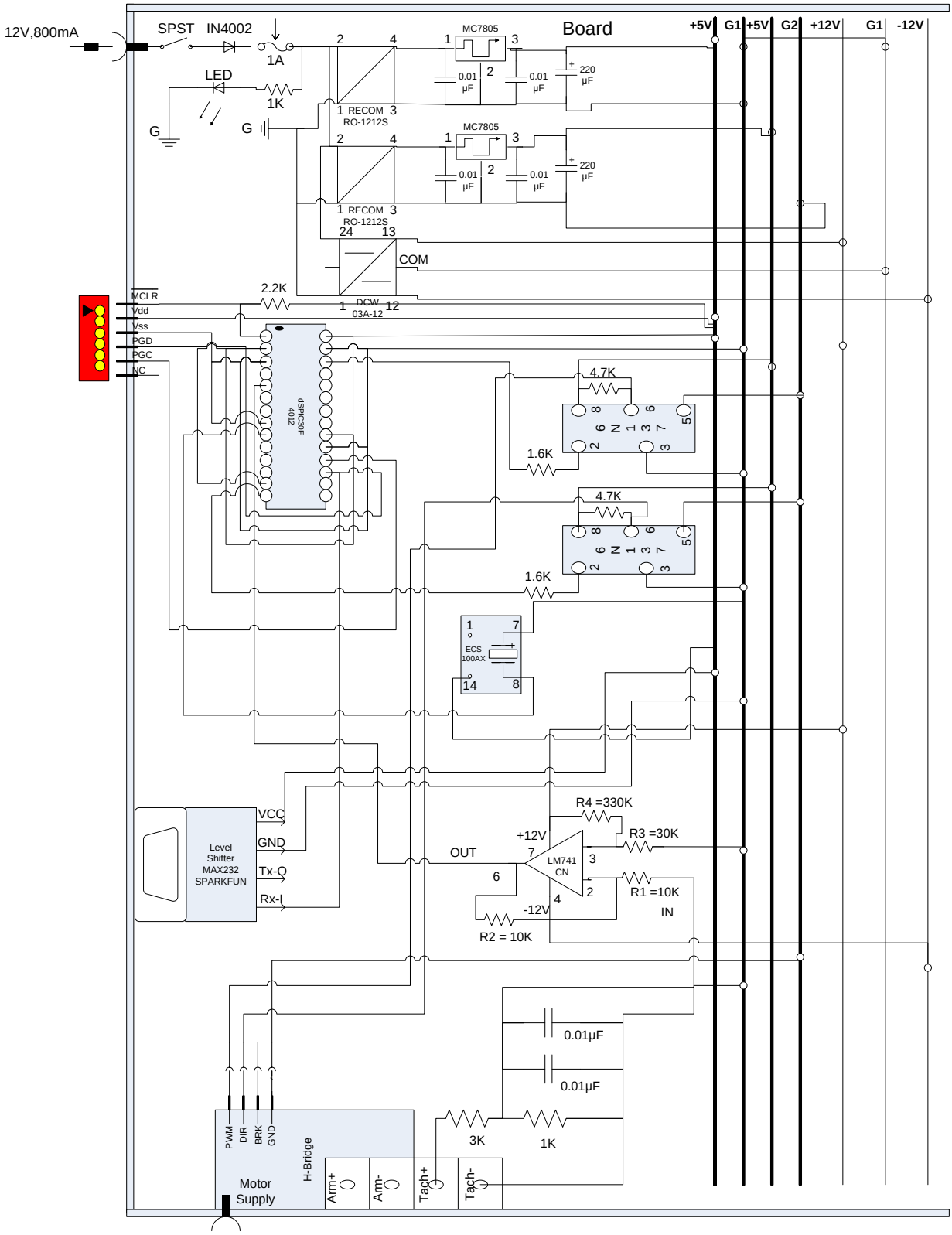
$$Gain = \frac{-R2}{R1} = -1 \Leftrightarrow R2 = R1 = 10K\Omega;$$

$$V_{+} = \frac{12 \times R3}{(R3 + R4)}; \qquad V_{OUT} = \frac{24 \times R3}{(R3 + R4)} - V_{IN};$$

$$24 \times R3 = 2 \times R3 + 2 \times R4;$$

$$11 \times R3 = R4; \quad R3 = 30K\Omega; \quad R4 = 330K\Omega$$

CIRCUIT/BOARD DIAGRAM:



CHAPTER 2

SOFTWARE MANUAL

2.1 OBJECTIVE:

- To identify the model of the underlying hardware (DC motor).
- To design a controller for the control of the speed of the DC motor.

2.2 SYSTEM REQUIREMENTS:

- Matlab R2010a with the following toolboxes installed.
  - Simulink Control Design,
  - Simulink Design Optimization,
  - System Identification,
  - Optimization toolbox,
  - Real-time Workshop and Real-Time Workshop Embedded Coder
  - Embedded Target for dsPIC by Lubin Kerhuel. This is downloadable from Lubin Kerhuel's website http://www.kerhuel.eu/
- MPLAB IDE for Microchip family of microcontrollers for downloading the generated hex file to the microcontroller.
- MPLAB C compiler for microchip. The student version can be downloaded from www.microchip.com. A user account with a valid university email-id is required to download the compiler.
- PICkit 2 debugger, which can be used to download the generated hex file to the microcontroller.
- RealTerm data capture tool for logging the received UART data to a text file.

2.3 DC MOTOR – PARAMETER ESTIMATION AND SYSTEM IDENTIFICATION:

The DC motor parameter estimation and system identification can be done in two ways by considering:

- DC motor model of the hardware excluding the nonlinear electronic components.
- DC motor model of the hardware including the nonlinear electronic components, exhibiting dead zone in the DC motor dynamics which may play a significant role in the estimation of the DC motor parameters.

The following steps are to be carried out for parameter estimation and system identification of the DC motor.

- Capture Input to the DC motor and Output data from the motor.
- Process the raw input and output data.
- Estimate and update the parameters of the DC motor Simscape model.
- Identify the model of the system using System Identification toolbox.
- Design a controller using Simulink compensator design.

2.4 PARAMETER ESTIMATION AND SYSTEM IDENTIFICATION:

The procedure to capture the input and the output data is the same for both linear and nonlinear models of the DC motor in Simscape.

2.4.1 CAPTURING THE INPUT AND OUTPUT DATA:

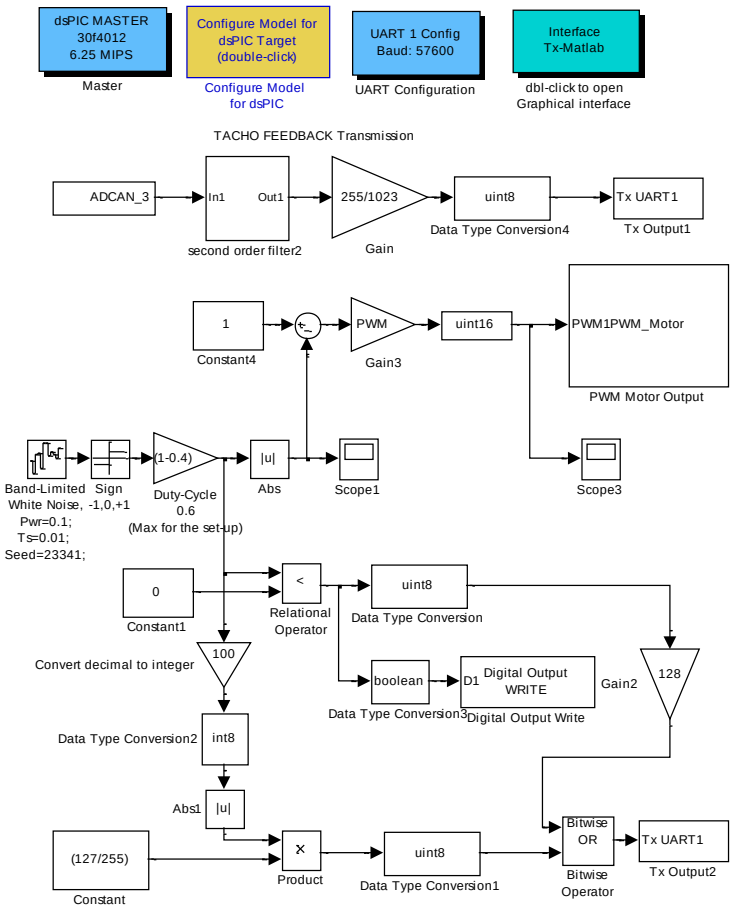
- Input - The dsPIC30f4012 microcontroller generates the PWM signals to drive the DC motor.
- Output - The output data from the Tachometer is the feedback to the A/D channel of the dspic30f4012 microcontroller.

The following procedure explains how to capture the input and output data.

- The dsPIC Target Toolbox contains the necessary blocks and components to construct the I/O interface in the Simulink model.

- The required blocks in the model are:
  - dsPIC Master block
  - Configure Model for dsPIC Target
  - UART Config
- The DC motor driver requires PWM signals and the direction to drive the DC motor. These are generated by the dsPIC.
- The output is fed back to the A/D channel of the dsPIC microcontroller. This is the speed of the DC motor which along with the input data is used for system model identification.
- The dsPIC microcontroller transmits the input and output data to the computer. RealTerm communication tool (a freeware) captures and records the incoming data from the dsPIC.

The Simulink model is given here.



2.4.2 PROCESSING THE RAW INPUT AND OUTPUT DATA:

- The captured data are written to the text file.
- The data file has to be parsed to get the input and output data. This is done in Matlab.
- The Matlab file “DATA_PROCESS.m” generates the input and output data.
- The ADC scaling is also incorporated in the source code. This is done to correctly interpret the values read in the A/D channel.

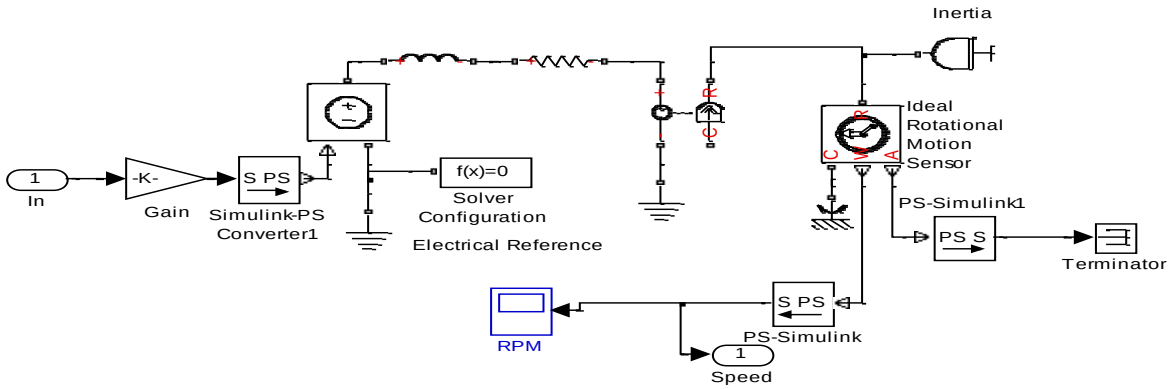
2.4.3 PARAMETER ESTIMATION:

To estimate the DC motor parameters and design a controller to control the speed of the DC motor, it becomes necessary to design the DC motor in Simulink using Simscape. The Simscape language is a subset of Simulink environment for modeling mechanical, electrical and other physical systems as physical networks.

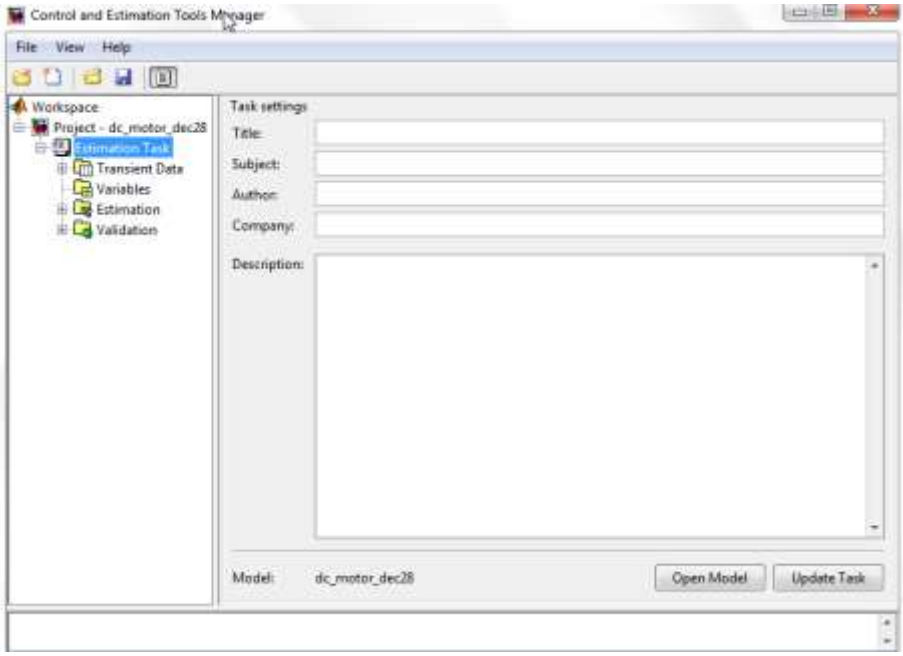
The procedure for the parameter estimation involves:

- Constructing the DC motor model in Simscape with or without any non-linear components as present in the actual hardware. In this section, we consider the linear case only.
- Using the control design toolbox, the parameters of the DC motor are estimated.

The Simulink model of the DC motor is shown here.



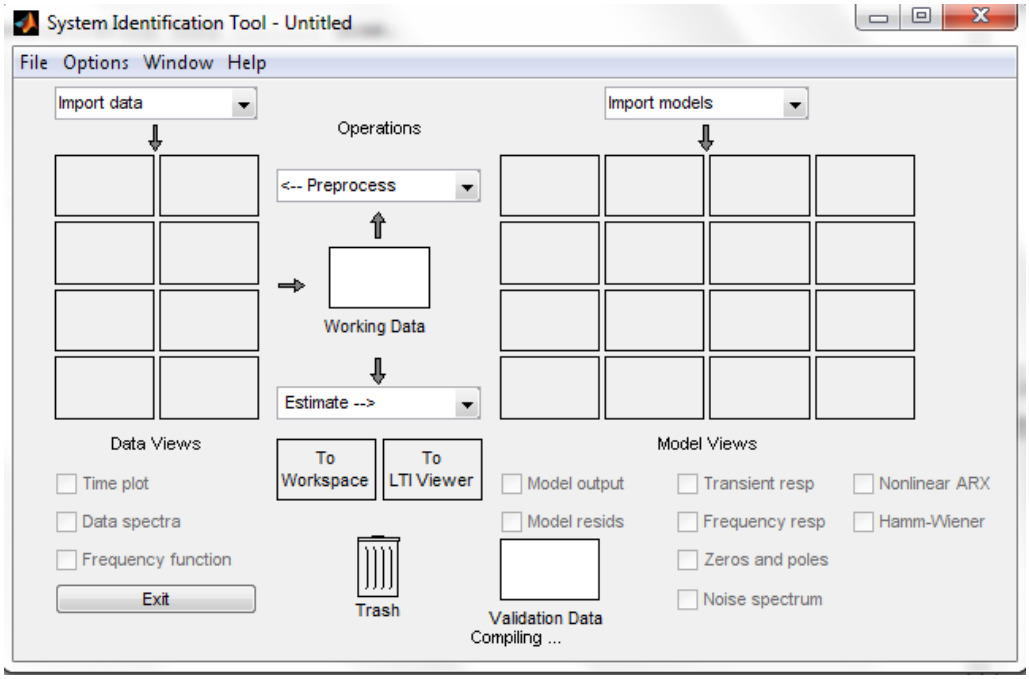
- The model is an equivalent of the DC motor. The parameters R, L, K and I are estimated to match closely with the actual motor parameters.
- The Simulink control design toolbox has an option “Parameter Estimation” under Tools menu.
- The control design toolbox analyzes the Simulink model and displays a dialog box to specify the input and output data. The variables to be estimated are also to be chosen in the dialog box before estimation is done. Detailed instructions on how to estimate the parameters are given in the next section.



- The toolbox offers various optimization algorithms to be selected for the parameter estimation.
- Once the parameters are updated, the DC motor model parameters are updated with the new estimated values and the Control design toolbox can be used for compensator design.

2.4.4 SYSTEM IDENTIFICATION:

The System Identification toolbox can be used to find the model of the system from the input and output data. Using this model, the controller can be designed to control the speed of the DC motor.



- Import the collected input and the output data from the workspace.
- Preprocessing:
  - Select range for the input and the output data, for estimation and validation.
  - Filter the data if required.
- Estimate -> Linear parametric models can be chosen to identify the model. The estimated model can be plotted to verify if a best fit is achieved.
- The model can be imported to workspace, by dragging the estimated model to the workspace field, and a controller can be designed for the identified model.

2.4.5 CONTROLLER DESIGN:

Simulink Control Design toolbox is used to design controller for the identified model. The controller parameters can be tuned based on the identified model.

The discrete PID controller block of Simulink can be automatically tuned for improving the response of the system to the reference command. This automatic tuning of the PID parameters is done using Simulink Control Design toolbox.

CHAPTER 3

INSTRUCTION MANUAL

3.1 OBJECTIVE:

This section provides all the necessary steps to be followed for all the sections identified in the previous chapter.

3.2 PROCEDURE:

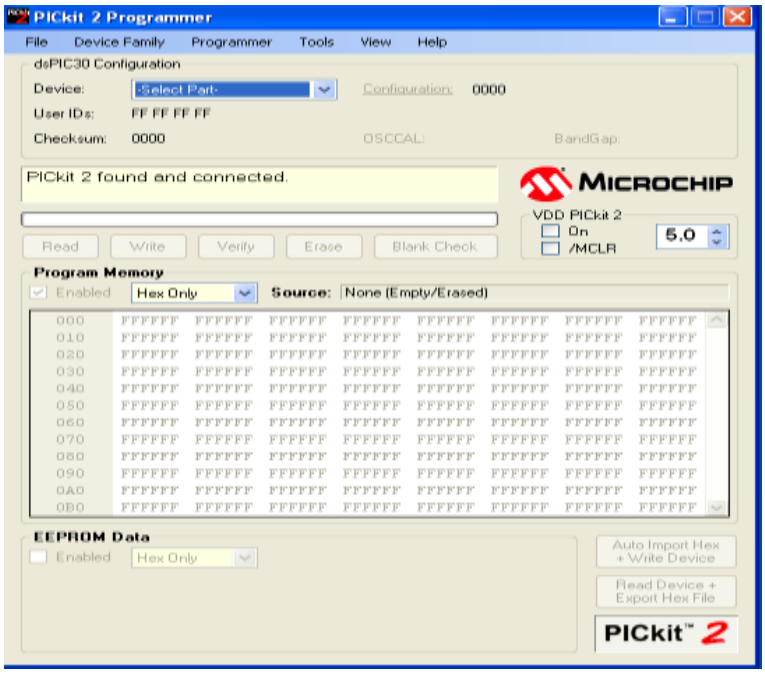
This section explains the procedure for data capture, system identification, controller design and verification.

3.2.1 DATA CAPTURE:

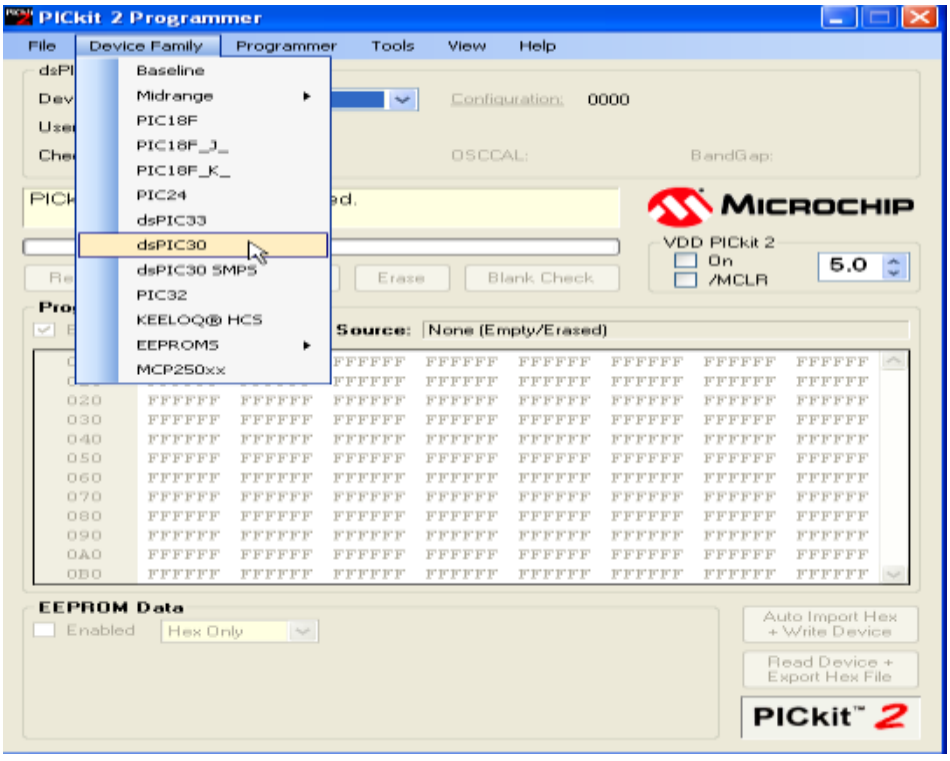
These steps are to be followed to capture the data.

1. Create a folder for your project (i.e., Desktop\your_folder\)
2. Copy the folder "**dsPIC_motor_control**" from the path "C:\Documents and Settings\ECE Lab\Desktop\ECE456_Files" to "your_folder"
3. Launch Matlab R2010a.
4. Set Matlab path to "Desktop\your_folder\"
5. Open the model **"dspic_dc_motor_data_capture.mdl"** from the path "Desktop\your_folder\dsPIC_motor_model\"**.**
6. Build the model. (The keyboard shortcut to build the model is Ctrl + B).
7. When the build process is complete, the hex file will be generated in the Matlab path (the current path).
8. Connect the PICKit programmer tool to the prototype board. The arrow mark on the programmer tool and the prototype board's first pin must align.

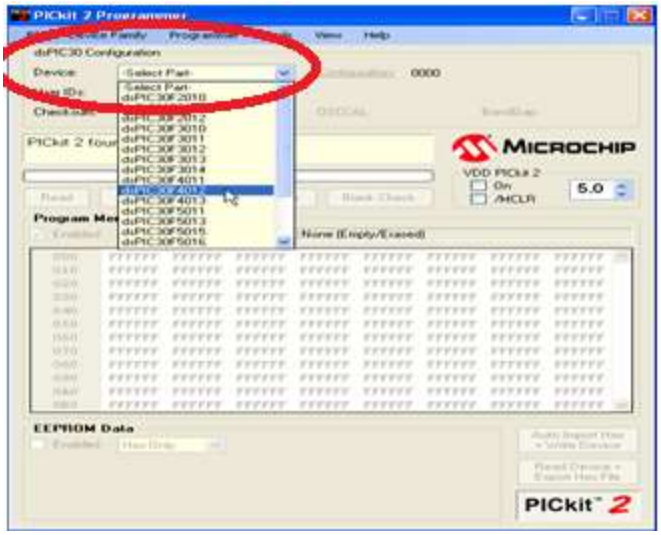
9. Launch Pickit2 application to download the hex file to the dsPIC30f4012. (Start -> Programs -> Microchip -> Pickit2).



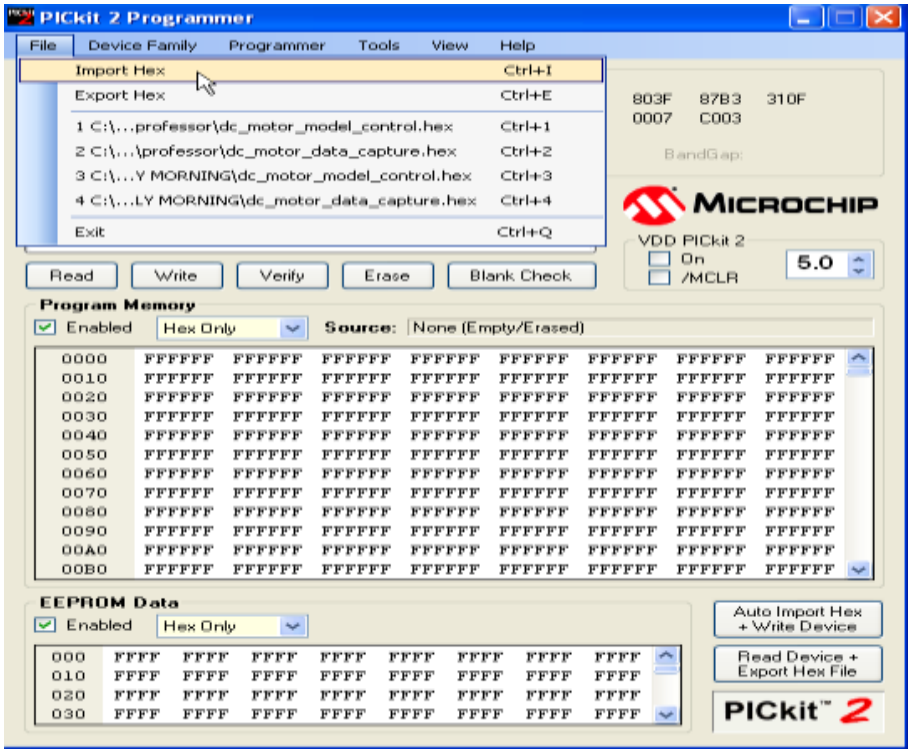
10. Click on “Device Family” tab. Choose dsPIC30 family.

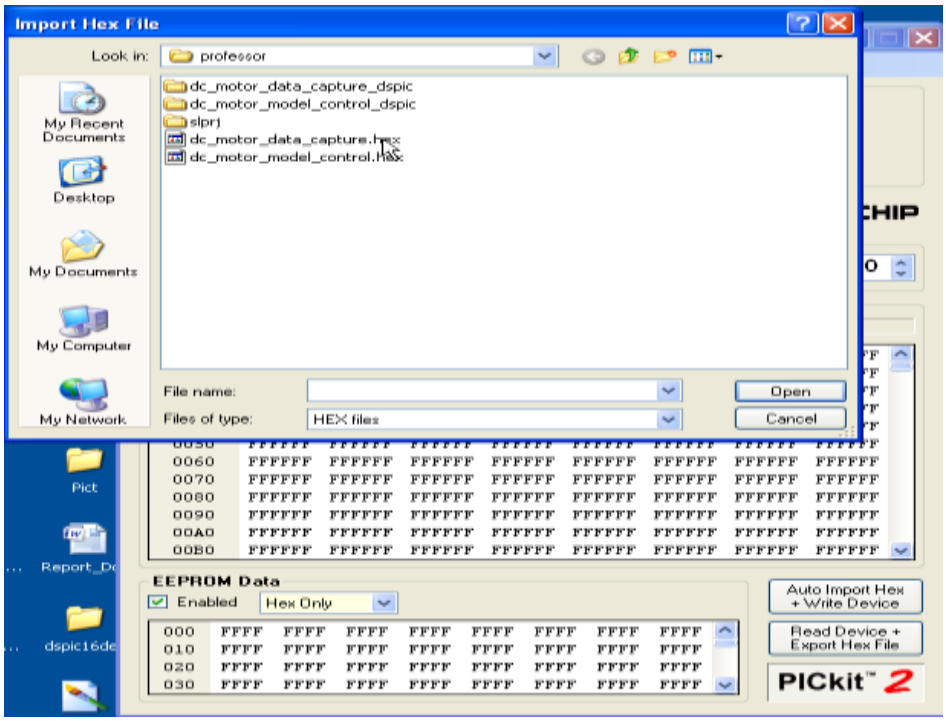


11. Choose the device “dsPIC30f4012”.

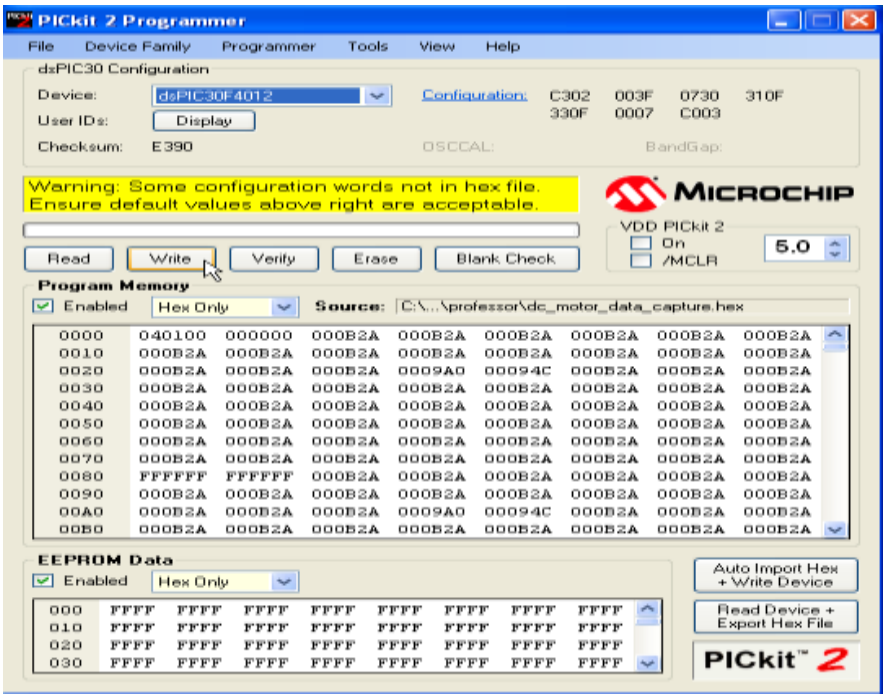


12. Click File -> Import to import the hex file. Choose the hex file from the path “Desktop\your_folder\ dsPIC_motor_control” to be downloaded to the target. Make sure the correct hex file is chosen before programming the target by checking the time the hex file is generated.

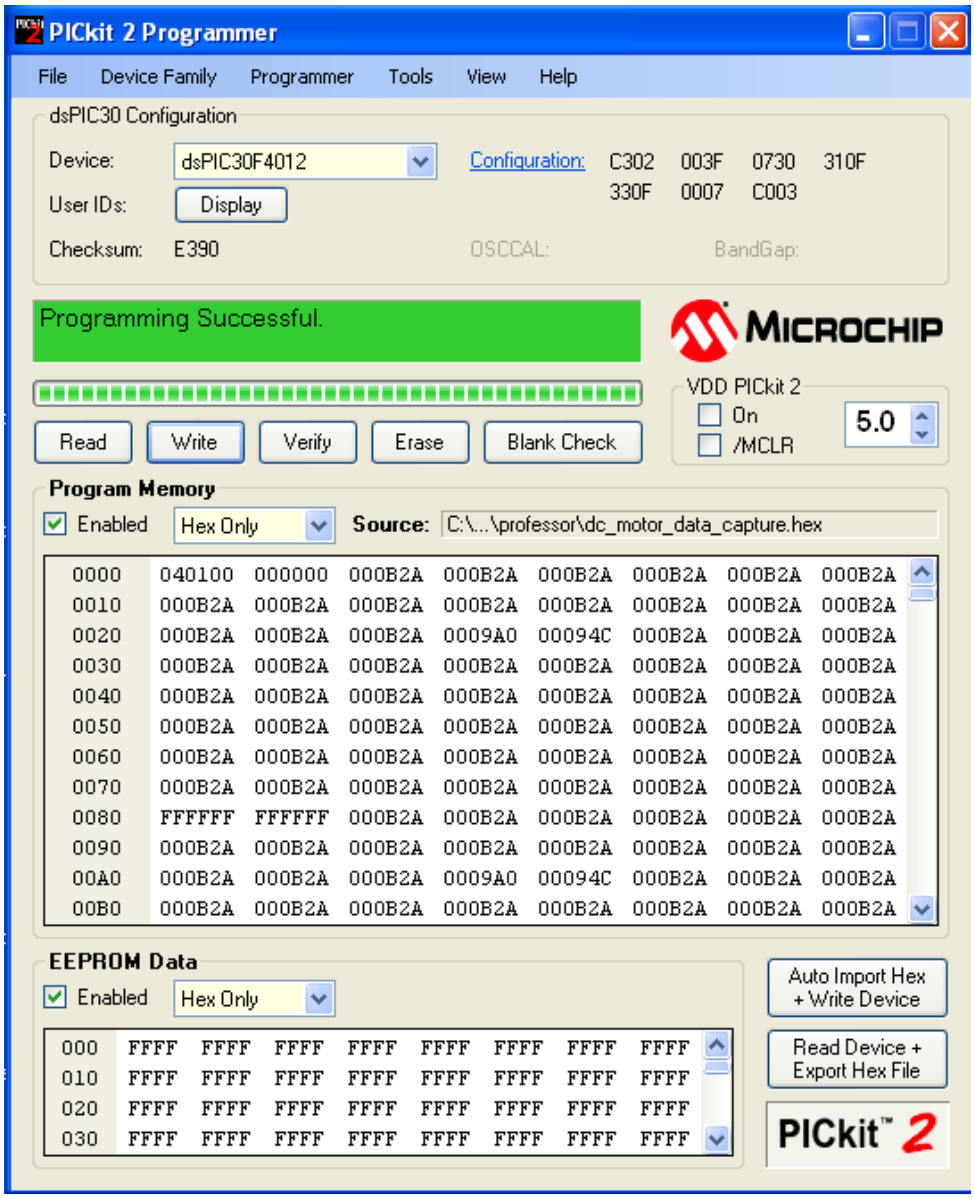




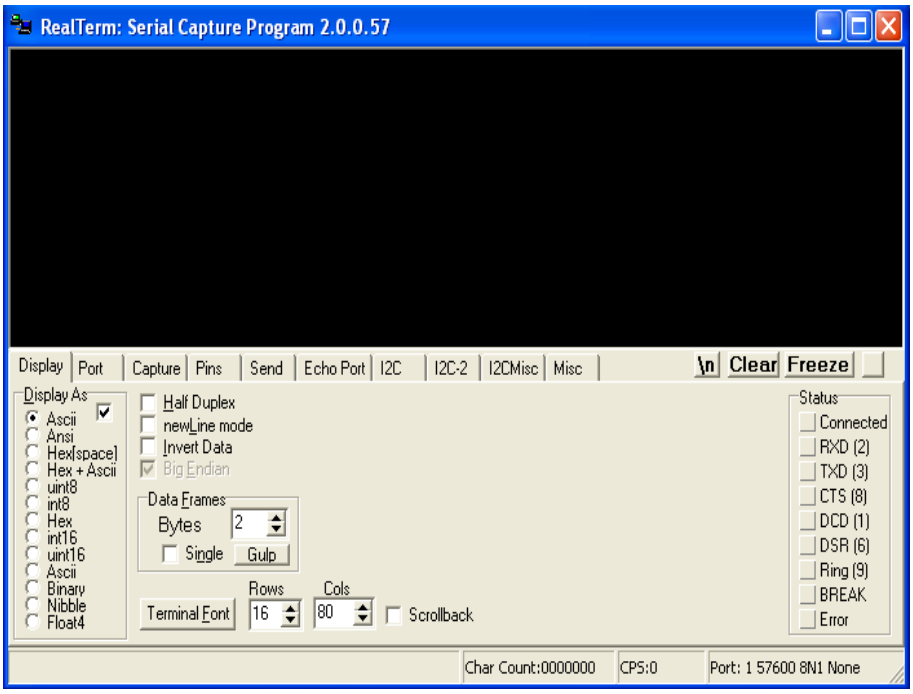
13. Click on “Write” button to program the target.



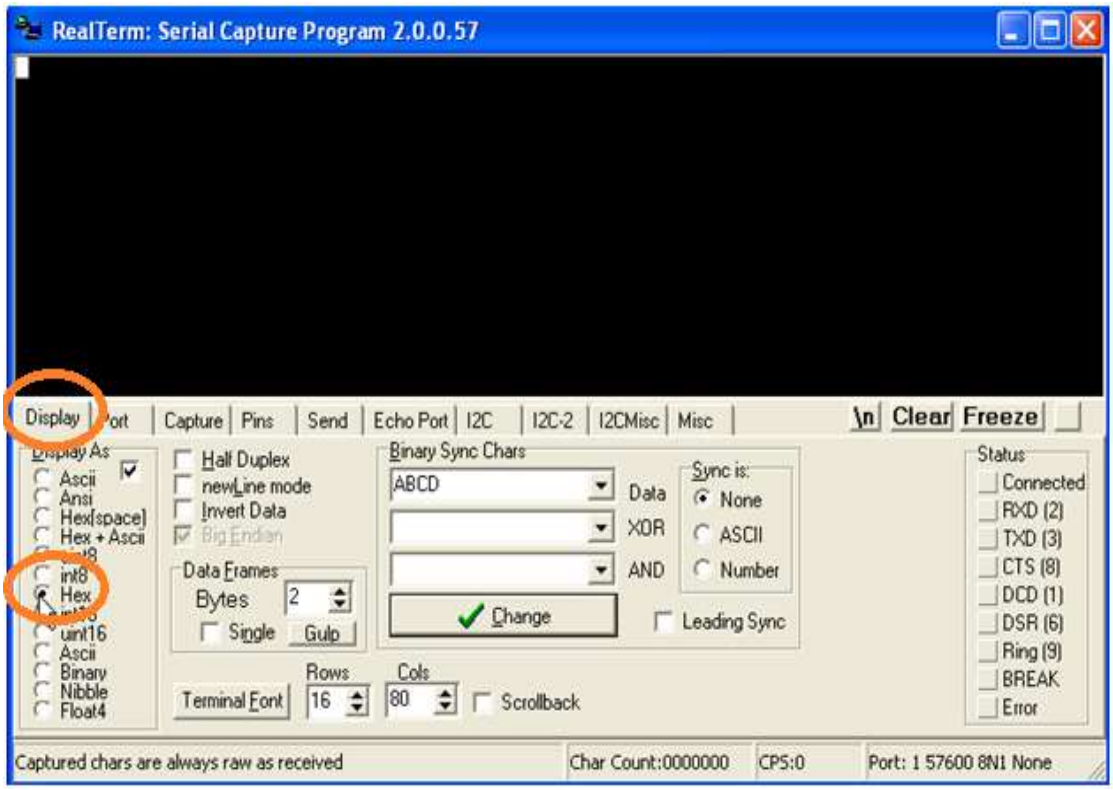
14. Check for “Programming Successful” message to ensure successful download of the program to target.



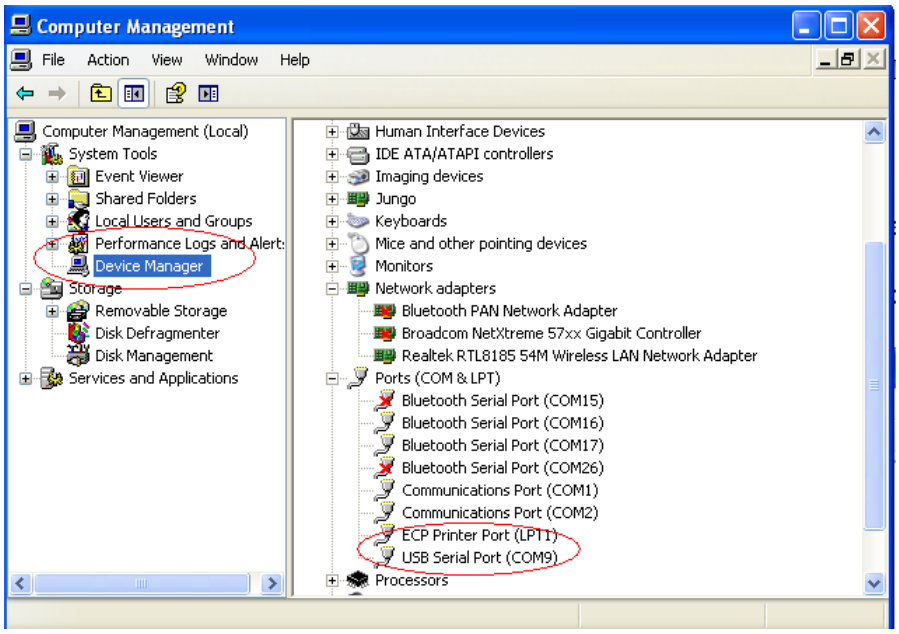
15. Remove the PICKit programmer tool from the prototype board.
16. Switch on the power supply to the board. The motor starts running.
17. Connect the USB to serial cable to the computer.
18. Launch “RealTerm” tool to capture the serial communication data. Start -> Programs ->RealTerm

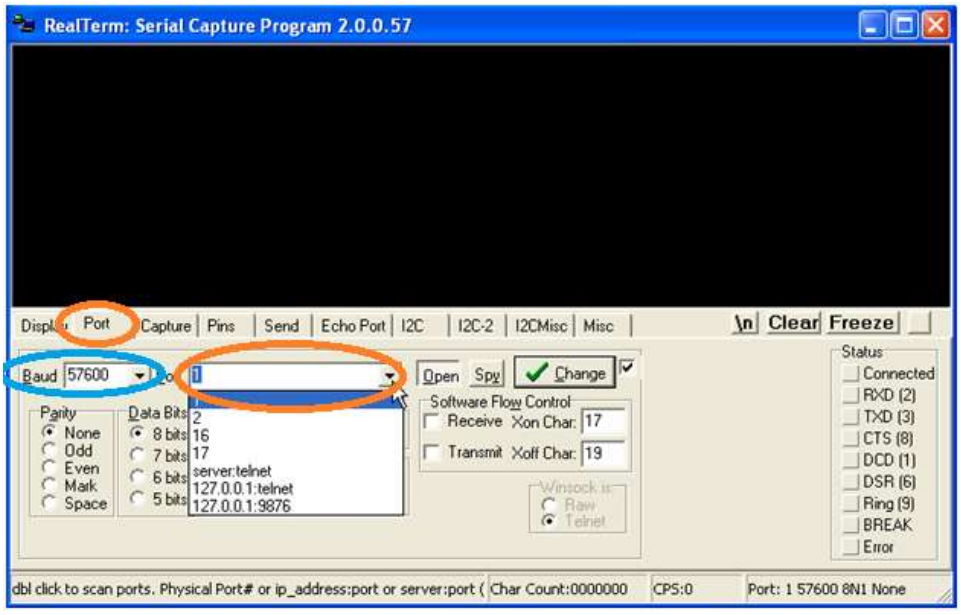


19. Under “Display” tab, choose “Hex” as the display format.



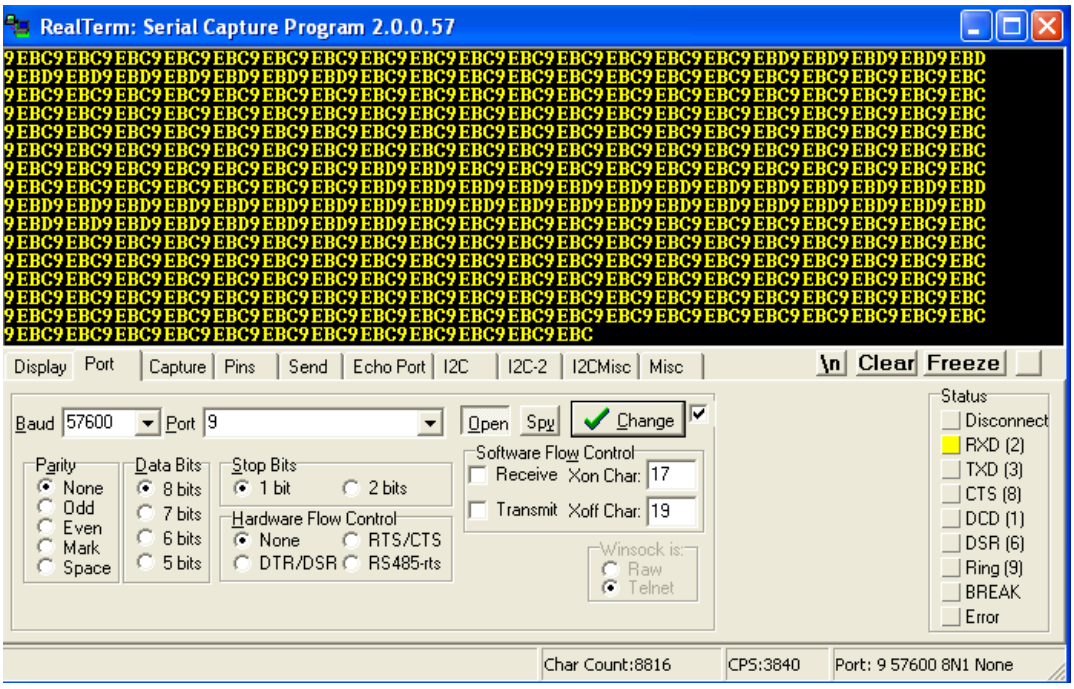
20. Under “Port” tab, choose the port number to which the USB to serial cable is connected to. To check the port number the USB to serial cable is connected to, click on Start. Right click on My Computer and choose Manage. Click on Device Manager to check for the port number. Refer to the options highlighted in red. Give this port number under the “Port” tab of the RealTerm tool. Check to see if the baud rate is 57600.



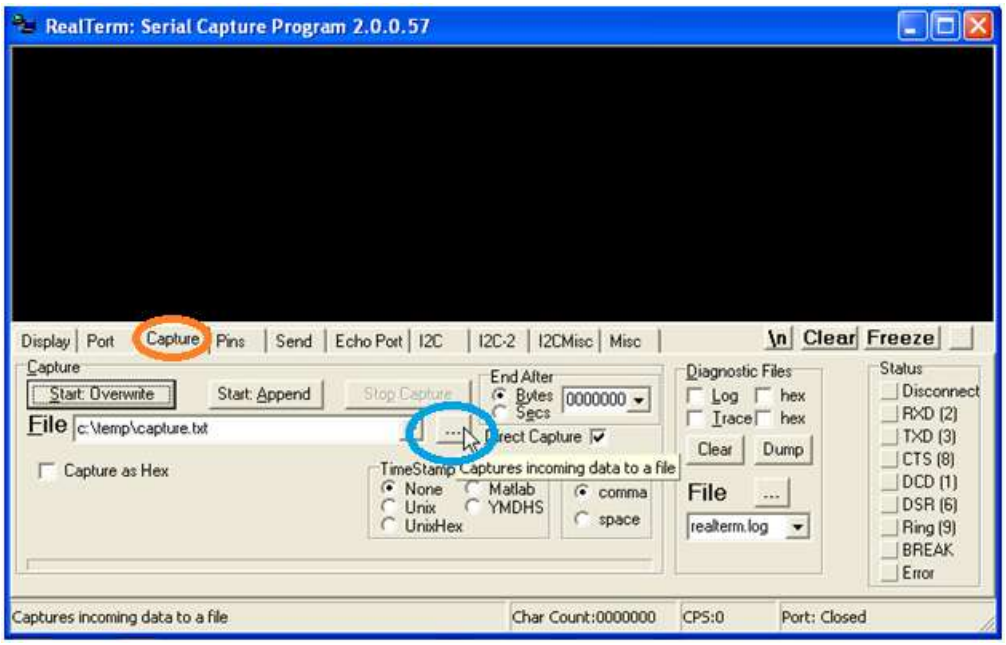


21. Click on “Open” button to open the com port.

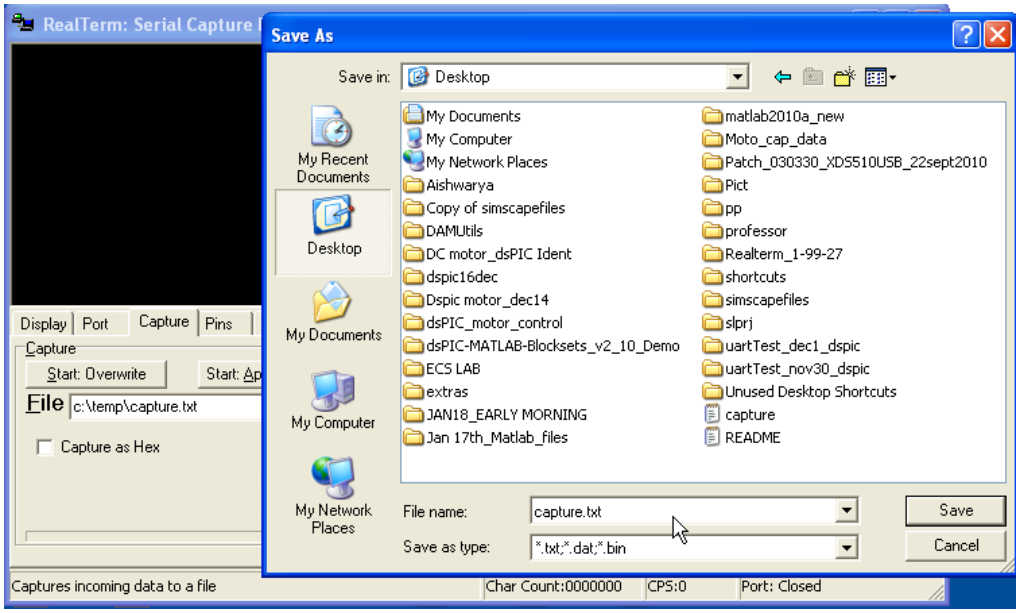
22. When the port is opened and if the motor is running, RealTerm tool receives the data and prints it on the black window. The data are displayed in hex format.



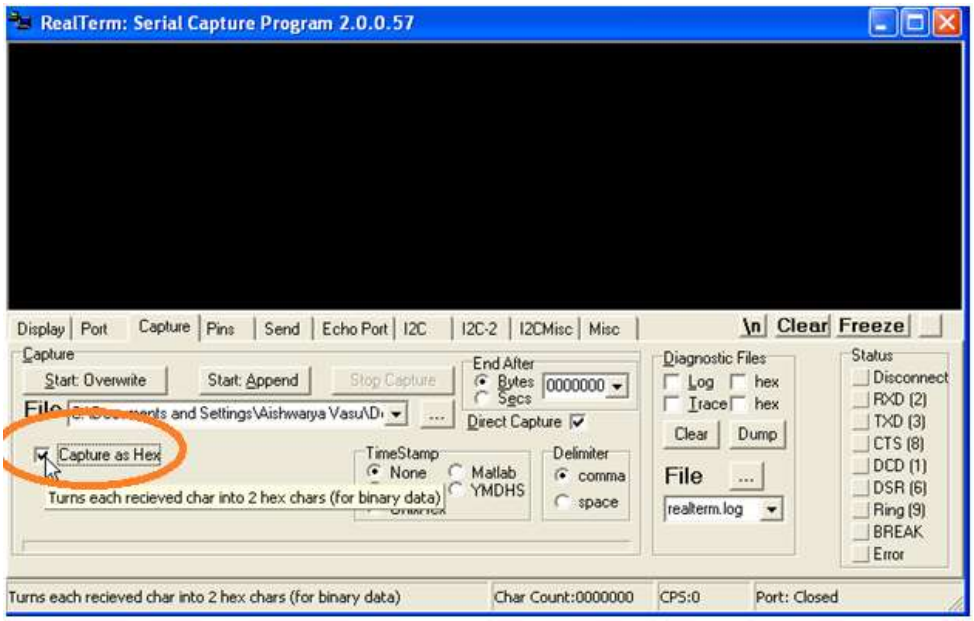
23. Under “Capture” tab, click on the browse button enclosed in a blue circle. This is done to capture and write the data to a text file.



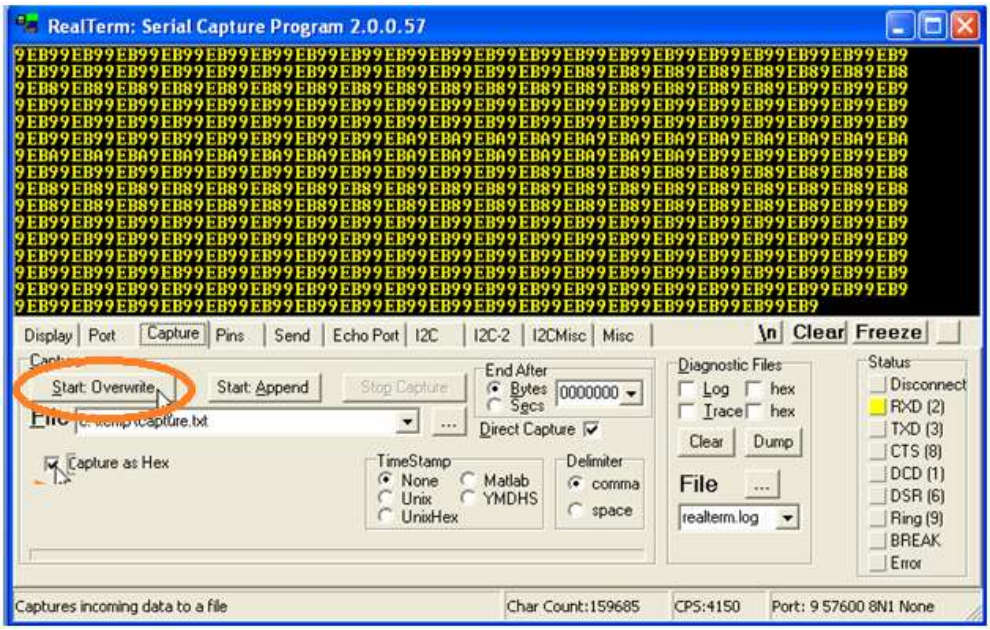
24. This opens up a dialog box to choose the path for saving the file. Specify name with the “**.txt**” extension. For example, the file name can be “capture.txt”. Click on “Save” button.

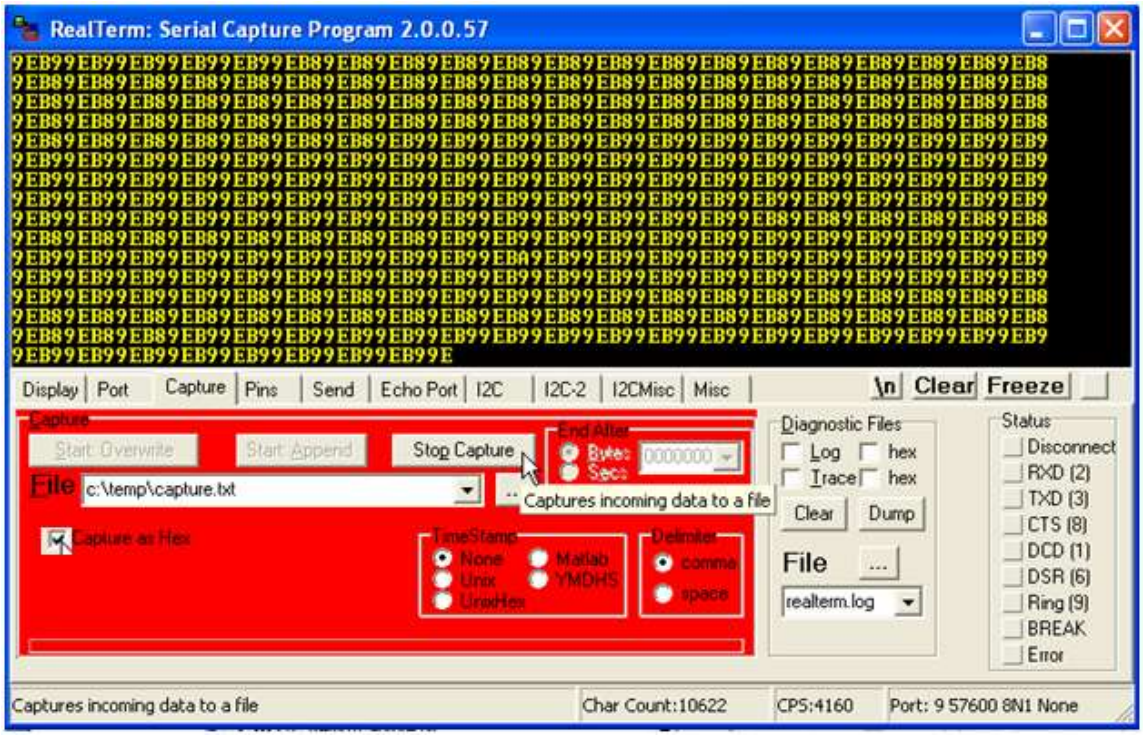


25. Check the “Capture as Hex” option. This writes the data to the file in hexadecimal format.

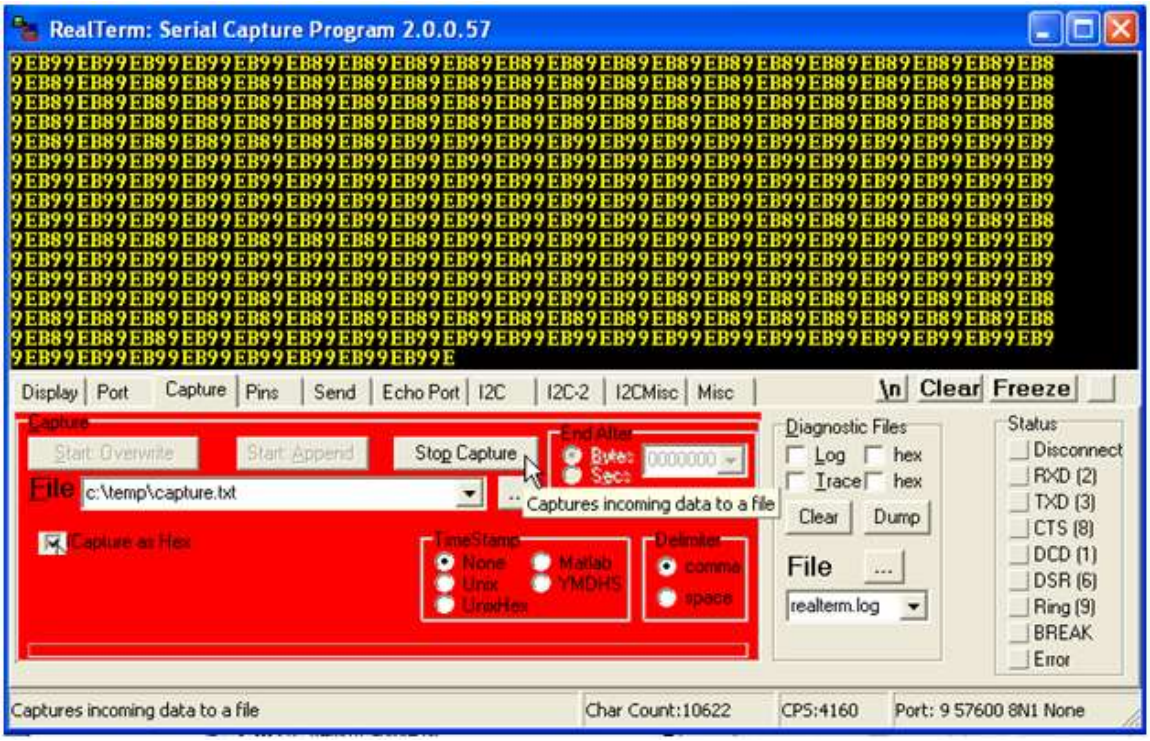


26. Click on “Start Overwrite” button to start writing to the file. During write operation, the data capture box turns red.





27. Wait for a few seconds and then click on “Stop Capture” button to stop writing to the file.

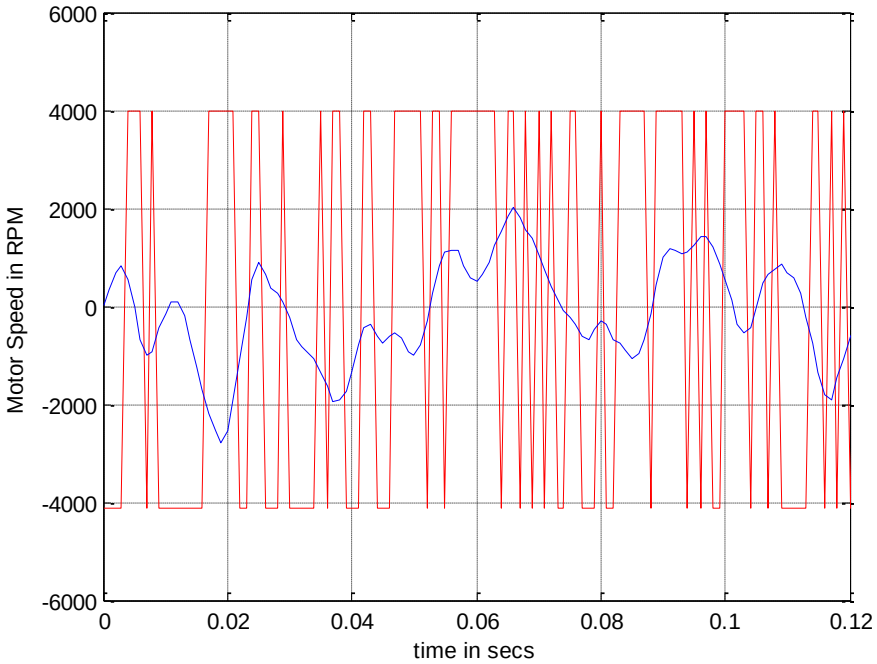


3.2.2 PROCESSING RAW INPUT AND OUTPUT DATA

Once the data are captured, it is necessary to read the data file to retrieve the input and output data. The input and the output data are in hexadecimal format. These are to be converted to decimal numbers. Proper scaling has to be done to retrieve the original input and output data. The “DATA_PROCESS.m” in the same directory as the “dspic_dc_motor_data_capture.mdl” processes and plots the input and output data.

These steps are to be followed to process the data.

1. Open “**DATA_PROCESS.m**” from the path “Desktop\your_folder\dsPIC_motor_model”.
2. Specify the data file name “capture.txt” as parameter to the “fopen” command. This is line 4 of the “DATA_PROCESS.m” file.
3. Debug -> Run the “DATA_PROCESS.m” file.
4. It takes a while to process the data. Once the data process is done, it plots the input and output data.
5. Check to see if the input and output data are similar to the one shown in the figure.



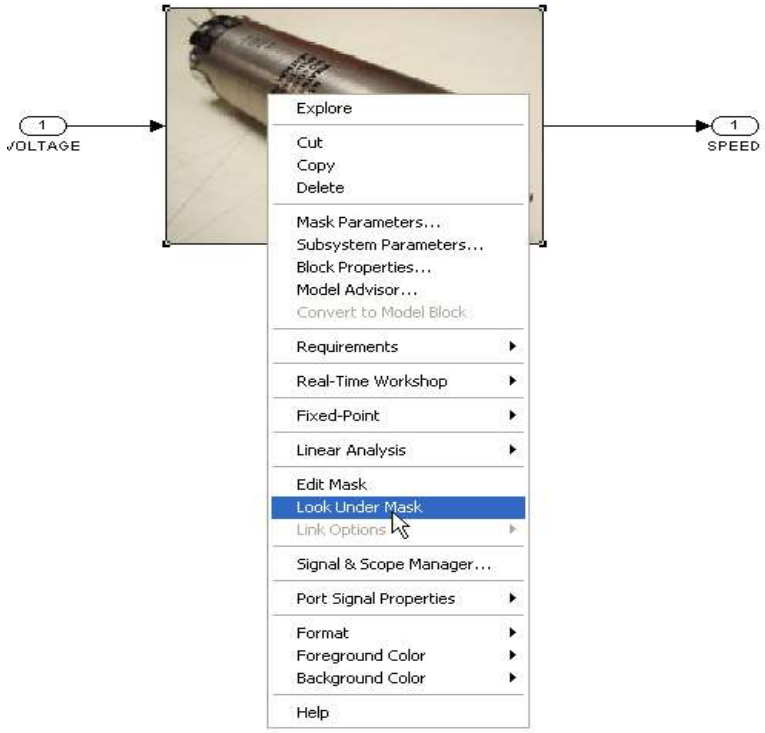
6. If the plot is not similar to the one shown above, run the "**DATA_PROCESS_swap.m**" file instead. Follow steps 1 through 3.
7. Once the motor input and motor output data are processed, these variables appear in workspace window. These are used for parameter estimation and model identification.

3.2.3 PARAMETER ESTIMATION USING SIMULINK CONTROL DESIGN TOOLBOX

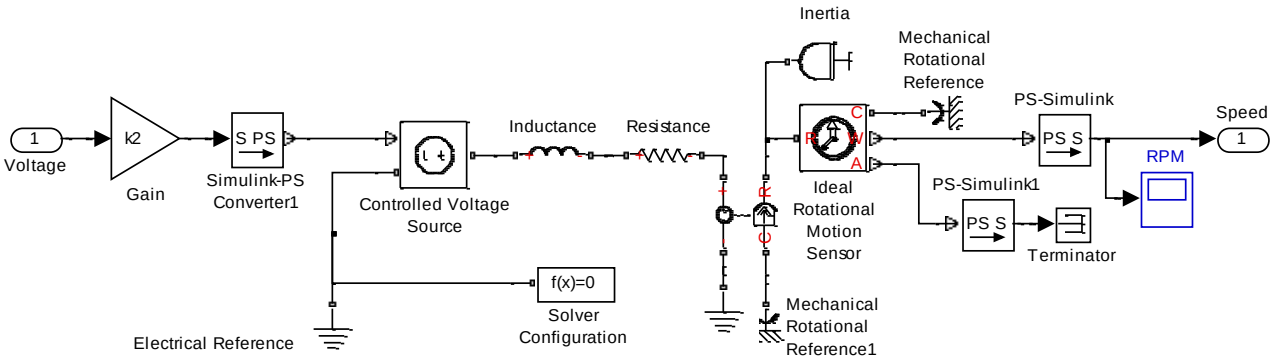
To design a controller and implement the same on the hardware, it is necessary to replicate the hardware model in Simulink and design a controller. Simscape component of Simulink can be used to design mechanical, electrical and other physical systems. The DC motor is built in Simulink using the Simscape components considering the linear components of the hardware. This section provides the necessary steps to estimate the parameters of the DC motor.

1. Open "INITIAL_PARAM.m" from the same path. Run the file to initialize the parameters used in the DC motor model.
2. Open "simscape_dc_motor_model.mdl" from the path "Desktop\your_folder\dsPIC_motor_model".

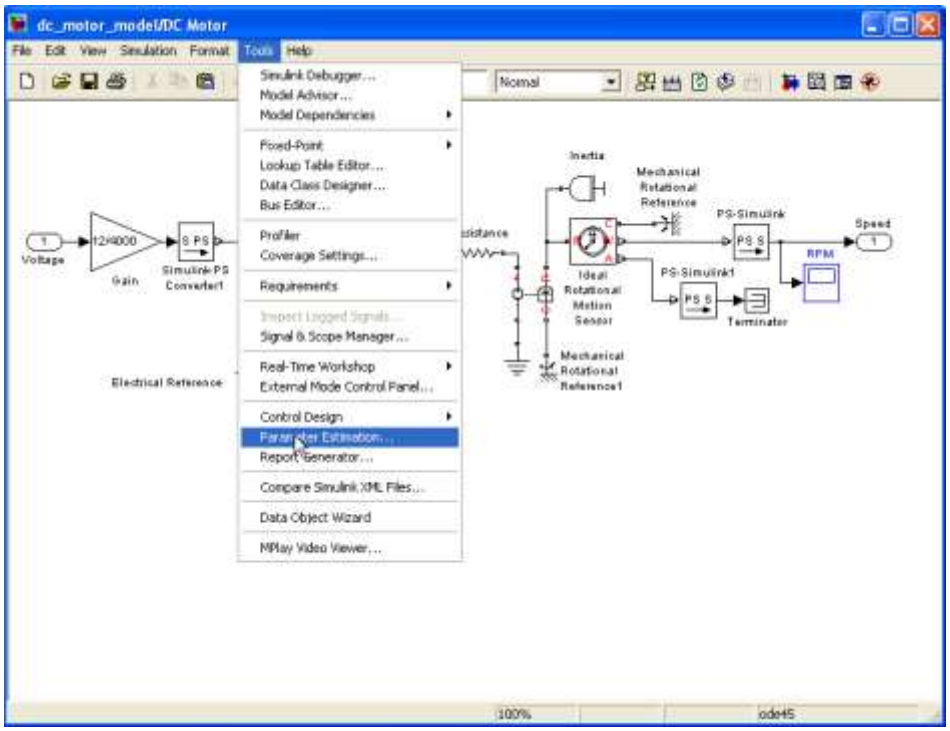
3. Right click on the DC motor picture. Select “Look Under Mask” option to see the DC motor model.



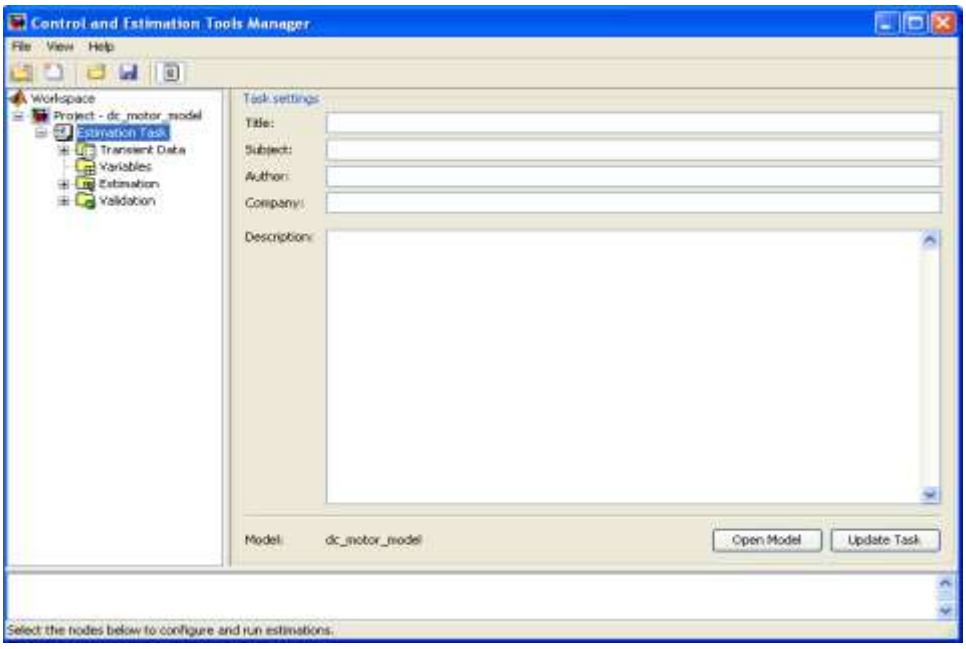
4. The DC motor model is shown here.



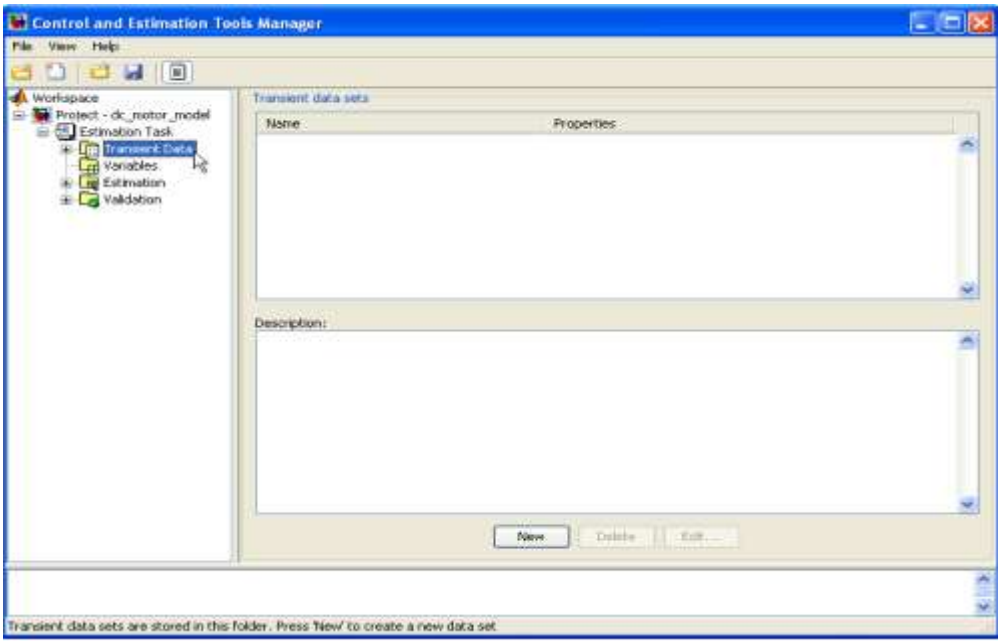
5. Click on Tools -> Parameter Estimation.



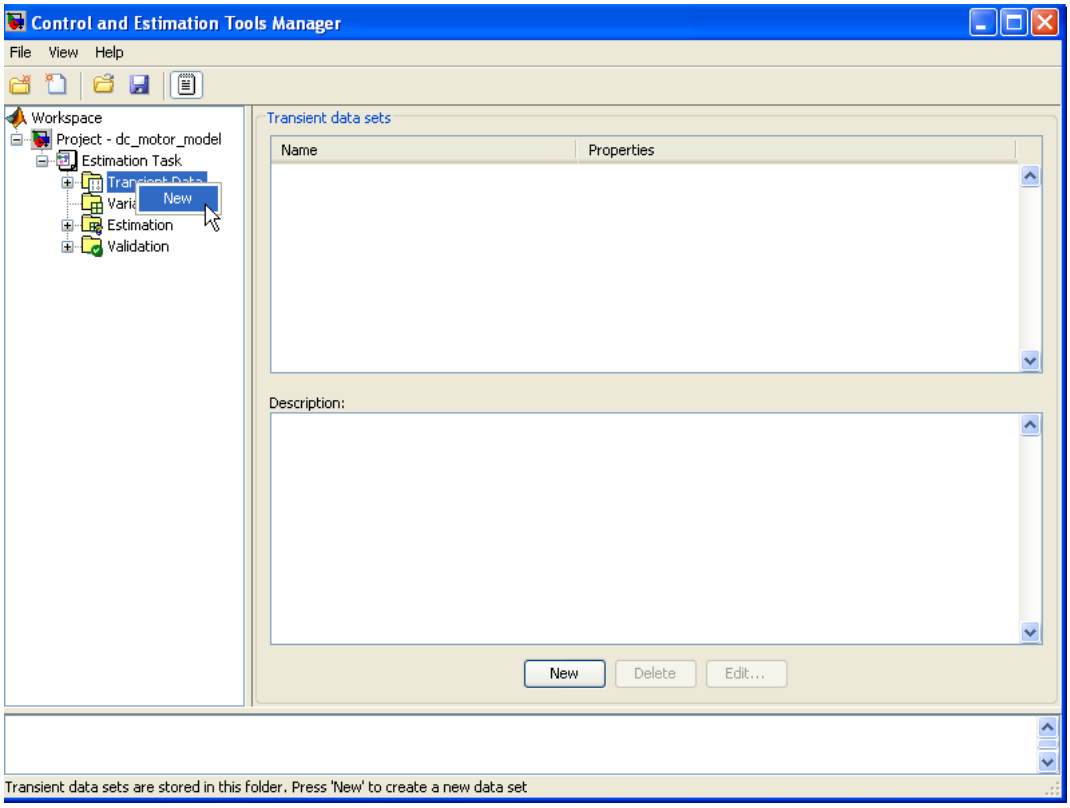
6. This opens up a dialog box.



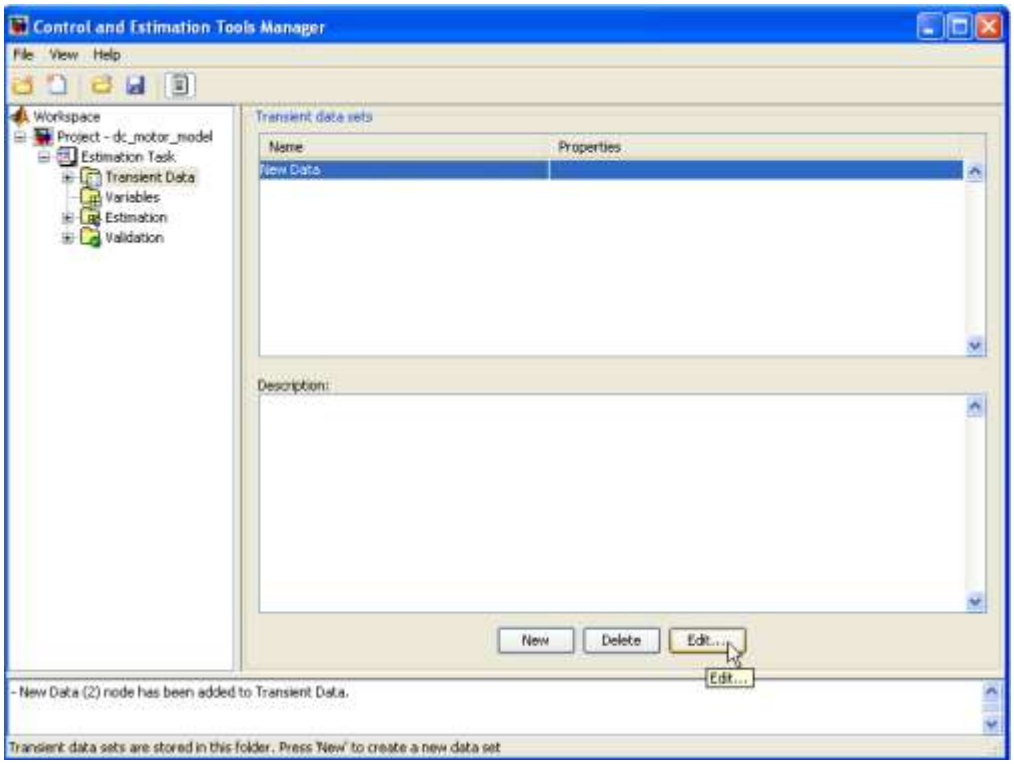
7. Click on “Transient Data” to import the input and output data.



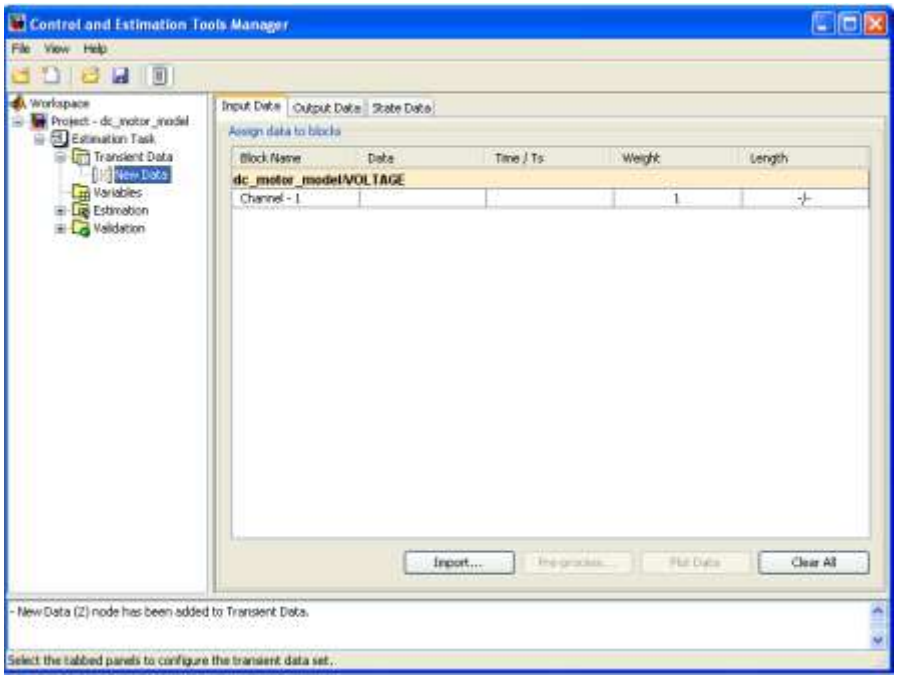
8. Right click on “Transient Data” and Click on “New”.



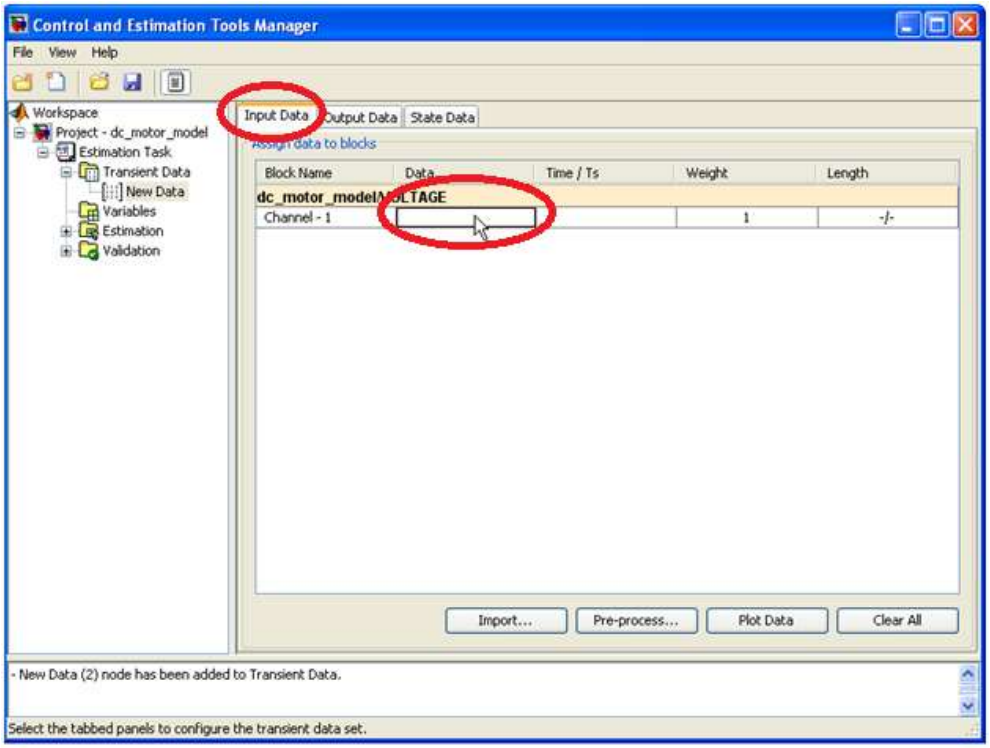
9. It adds a new data to the environment. Click on “Edit” button to edit the data.



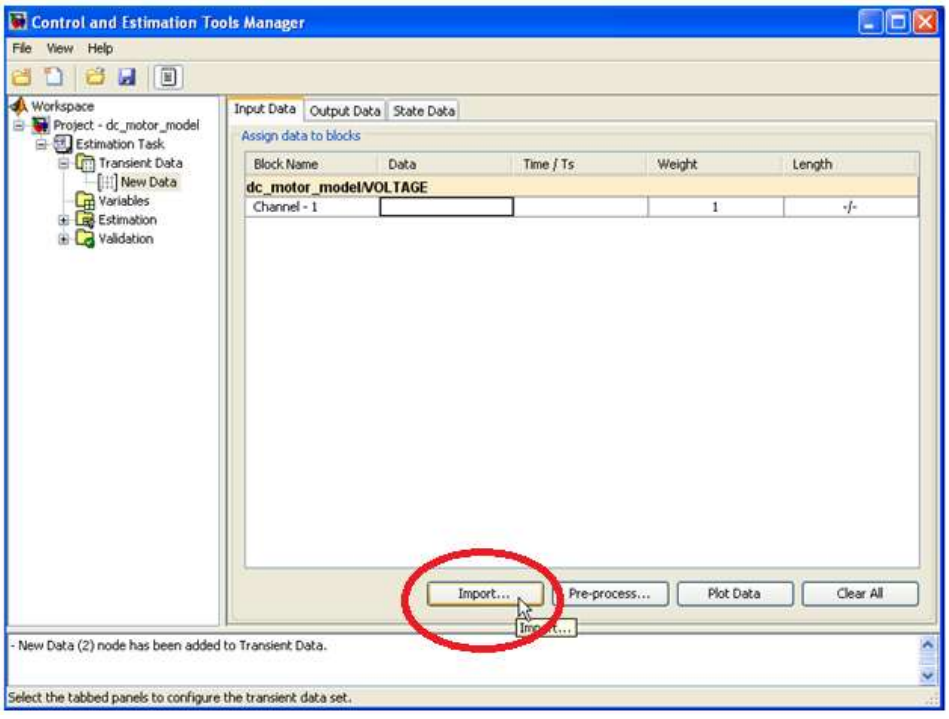
10. The window appears as given below.



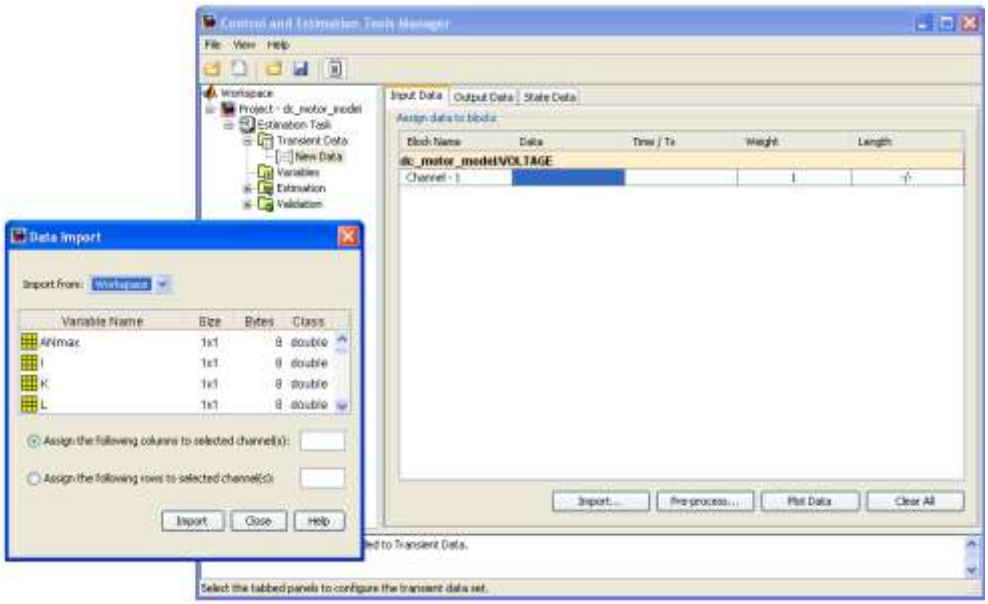
11. Under “Input Data” tab, click on the data cell.



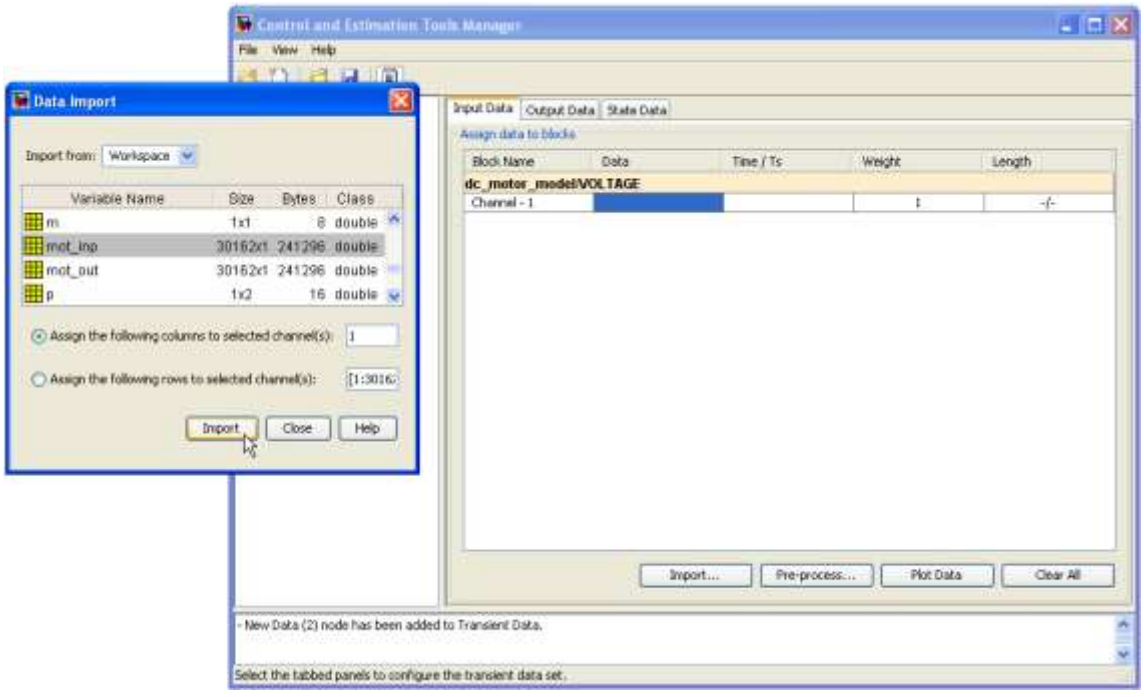
12. Click on “Import” button.



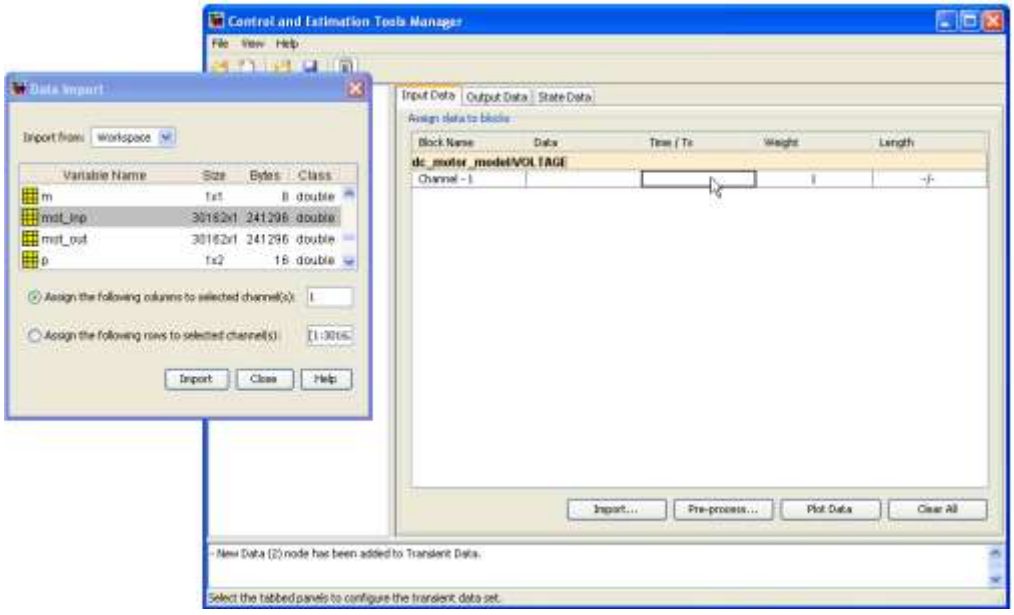
13. The “Import” dialog box opens.



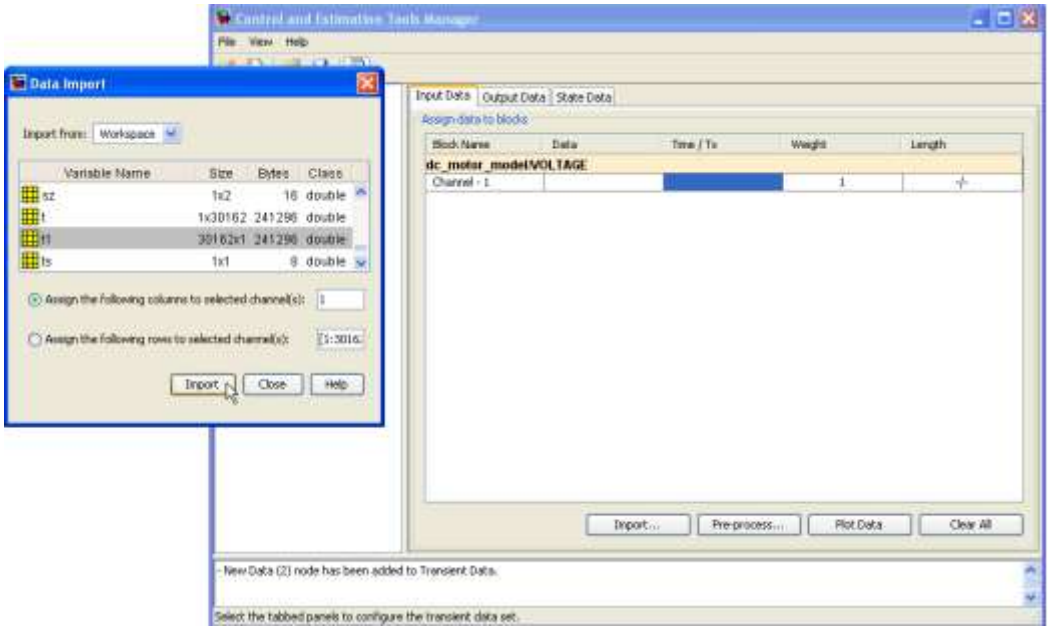
14. Scroll down to select the “mot_inp” data. Click on “Import” button.



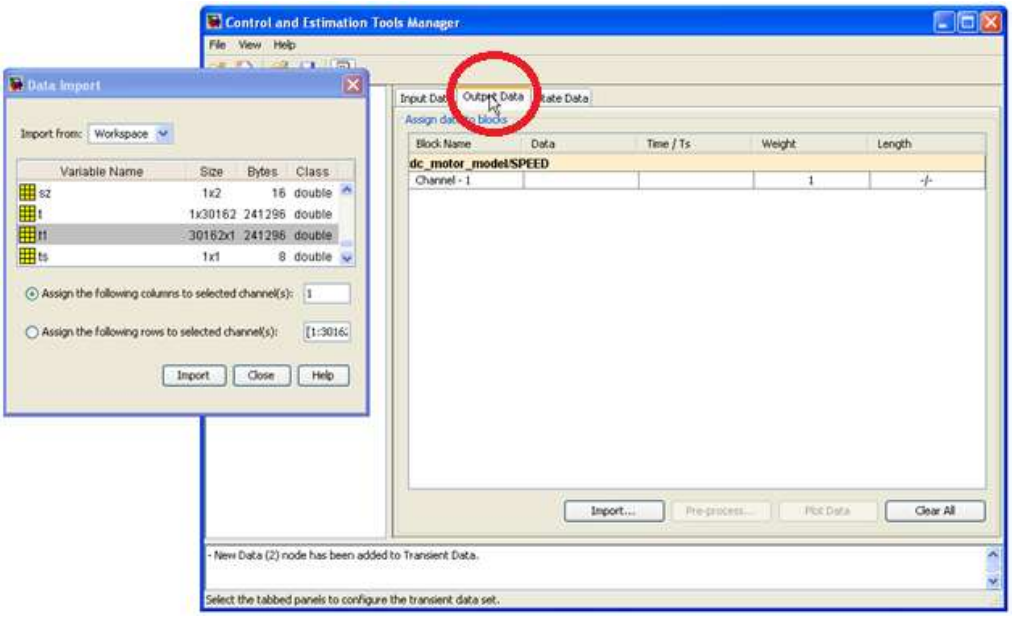
15. While the “Import” dialog box is open, click on the “Time/Ts” cell as shown in the figure below.



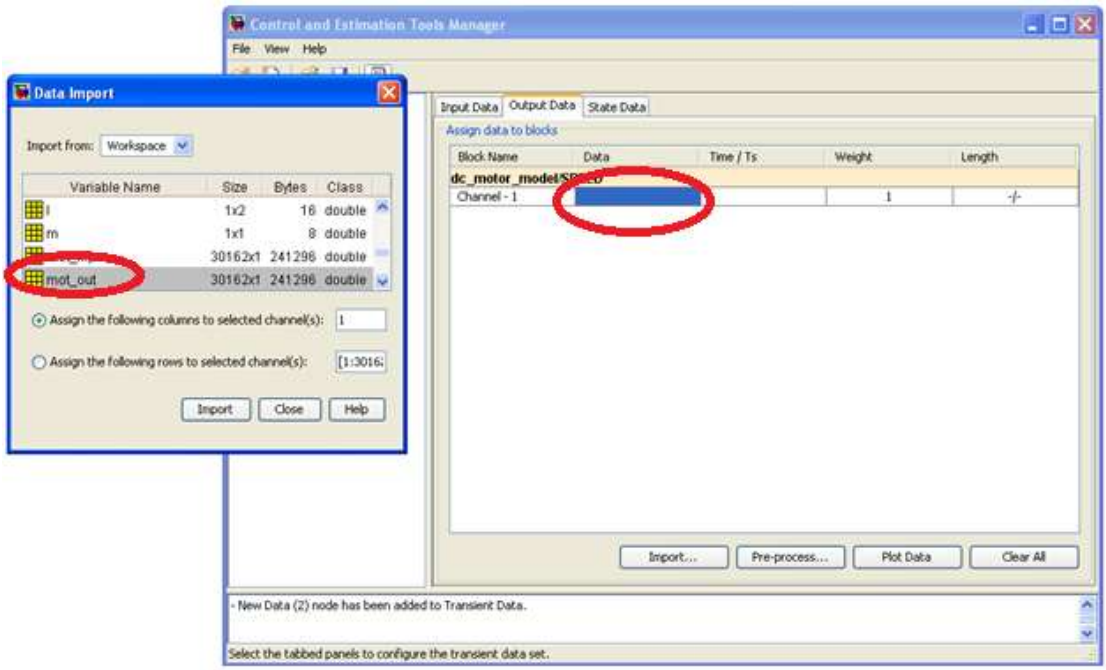
16. Scroll down to select the time vector “t1” from the “Import” dialog box. Click on “Import” button.



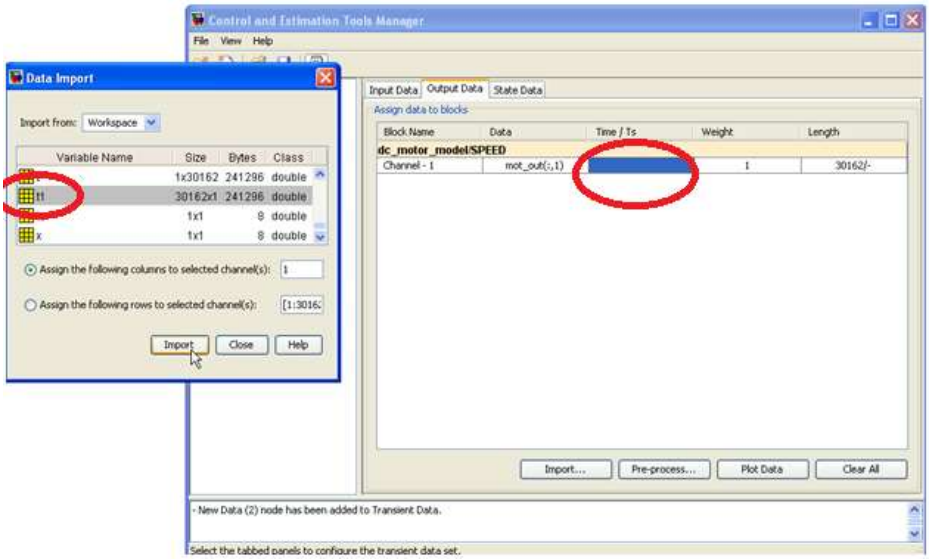
17. Click on “Output data” tab.



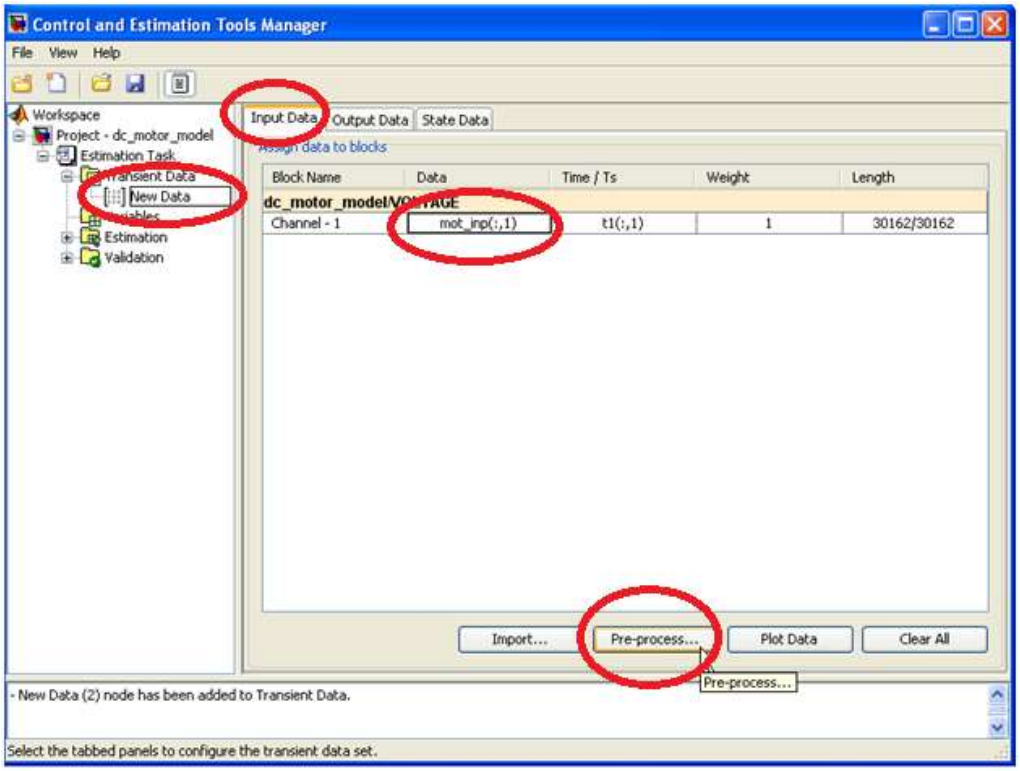
18. Click on the “Data” cell. Scroll up to select the “mot_out” data from the “Import” dialog box. Check the figure to do this.



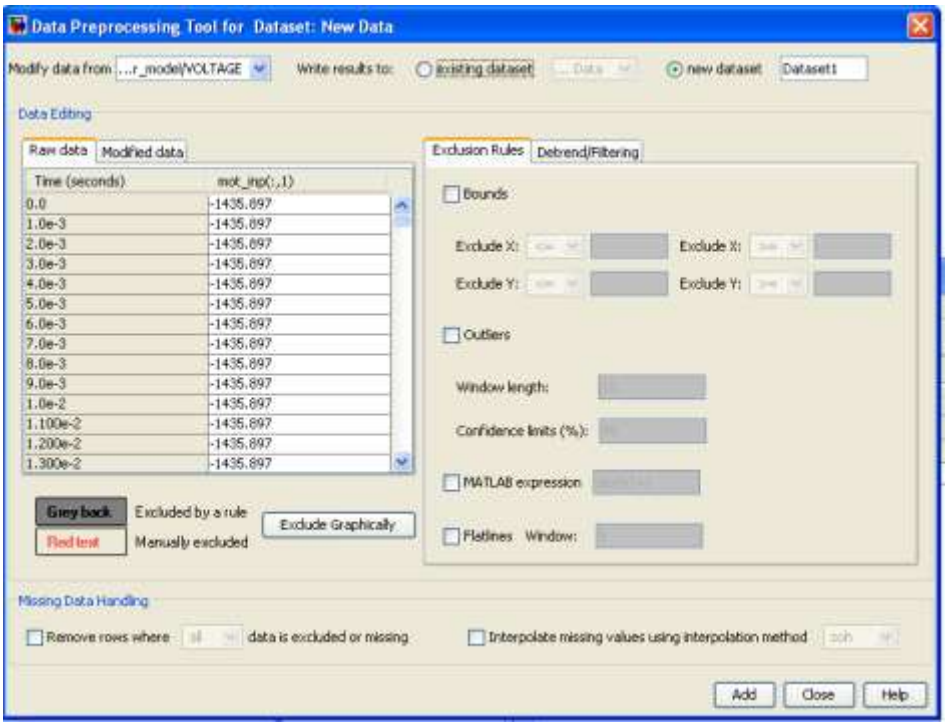
19. Click on the “time” data cell. Scroll down to select the time vector “t1”. Click on Import. Please follow the figure.



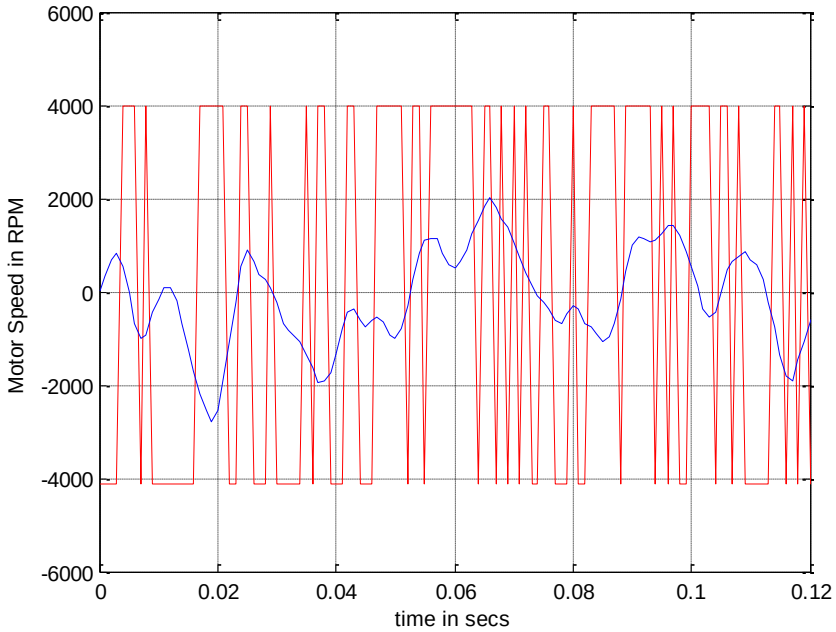
20. The input and the output data are to be processed. To do this, click on “New data” in the left pane. Under “Input data” tab, select the cell “mot_inp” and click on “Preprocess” button.



21. A new dialog window opens with various options to process the data.



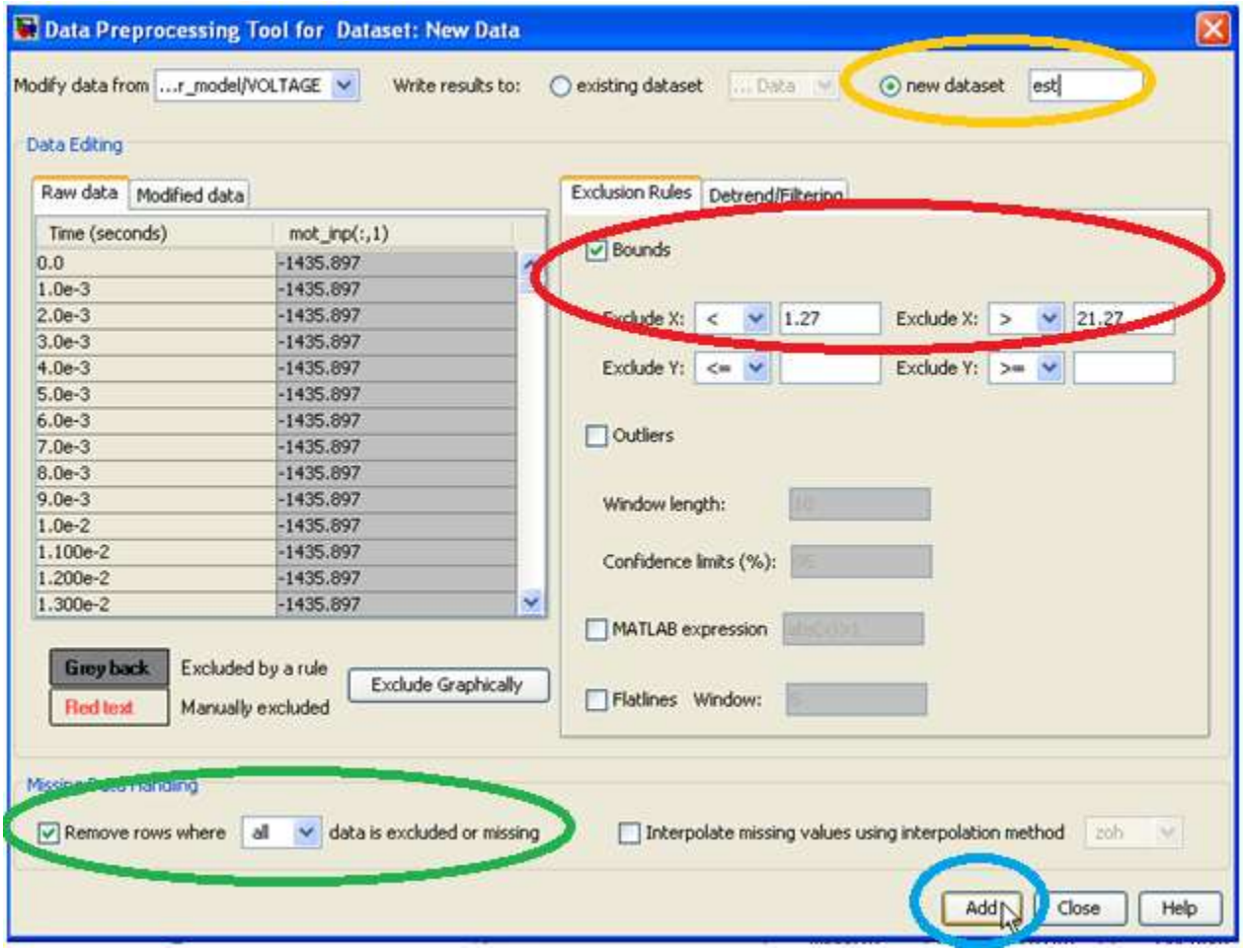
22. In this task, the entire captured input and output data are used to estimate the Simscape model parameters. To do this, refer to the input and output plot.



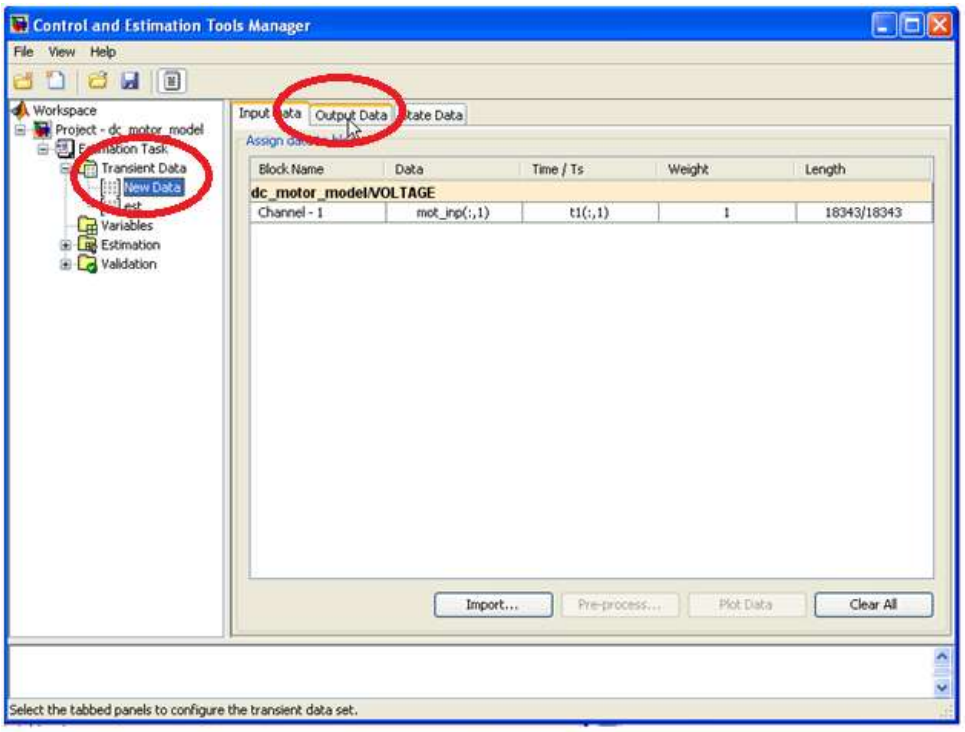
23. To use part of the data, enter the appropriate values as shown in the figure below.

    a. Select the “Bounds” check box. Specify the “X” and “Y” bounds. Refer to the block encircled in red.
    b. Select the option “Remove rows where all data is excluded or missing”. Refer to the block encircled in green.
    c. Make a new data set by specifying a new dataset. Refer to the block encircled in yellow.
    d. Once the steps a, b and c are done, click on “Add” button and then click “Close” button.

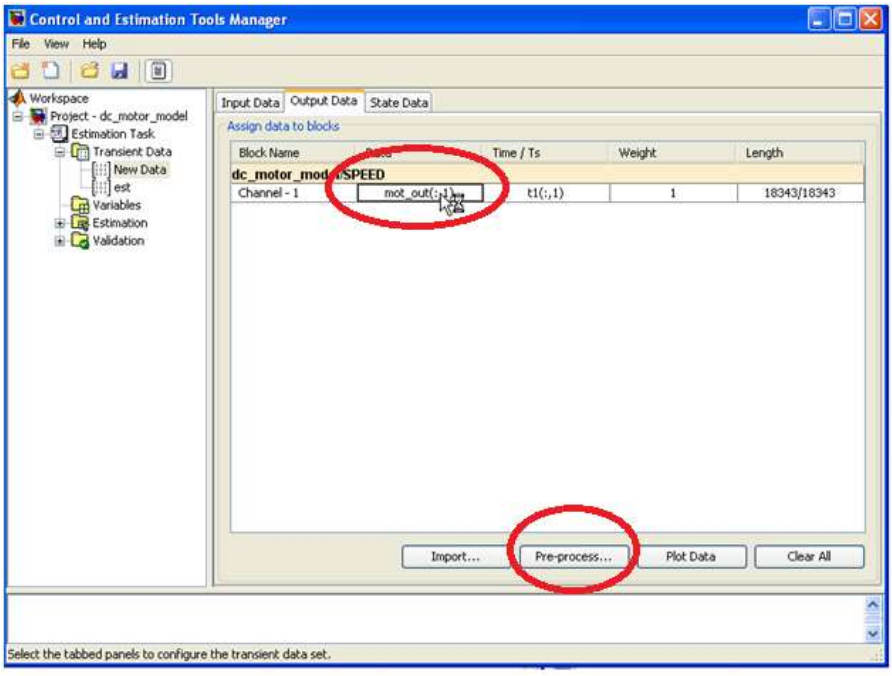
Then the new data set “est” will be added to the estimation task.



24. Click on “New Data” set to preprocess the output data. Click on “Output Data” tab to select the motor output data.



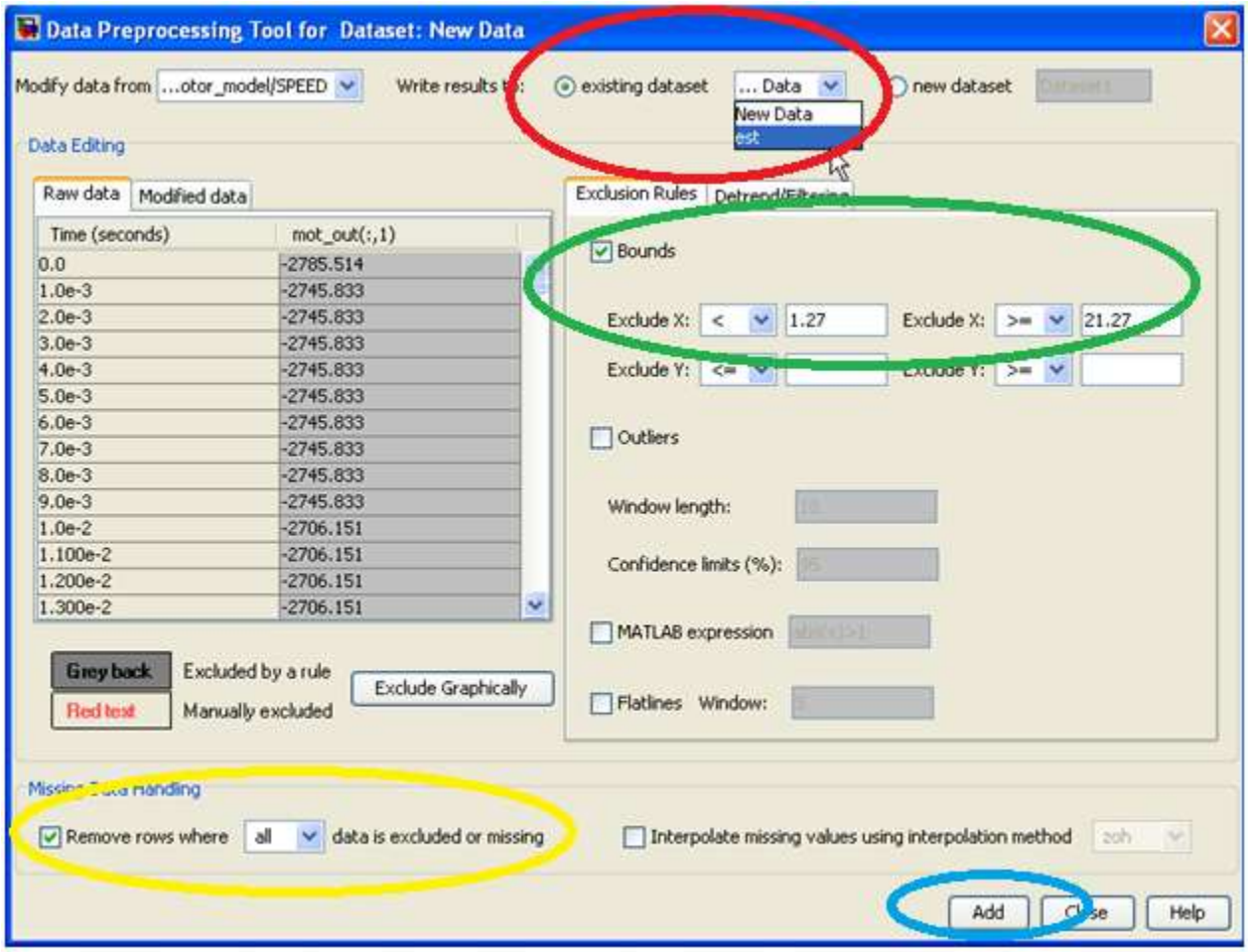
25. Click on “mot_out” data cell and then click on “Preprocess” button.



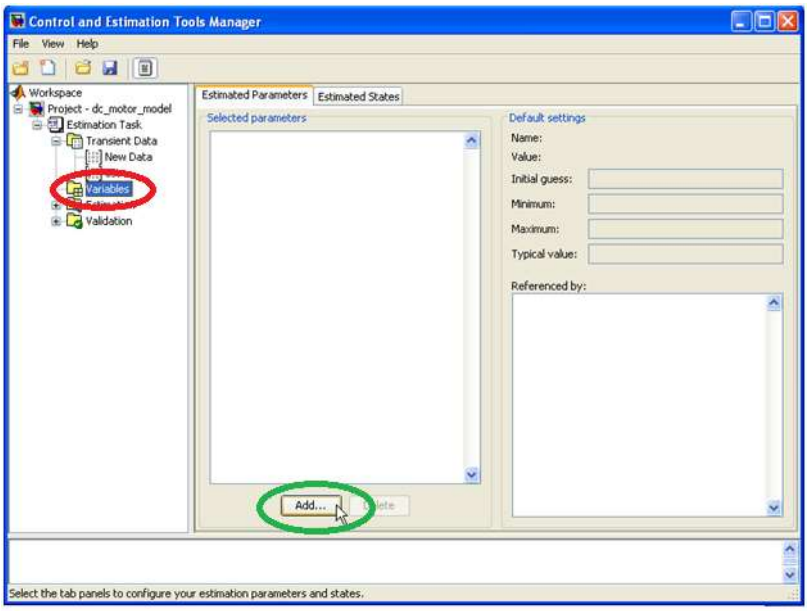
26. If the input data was preprocessed, then also preprocess the output data as shown in the figure below.

    a. Specify the bounds for the output data. Refer to the “Bounds” section encircled in green.
    b. Select the option “Remove rows where all data is excluded or missing” encircled in yellow.
    c. Select “Existing dataset” and choose “est” data. This is the data set which has the input data preprocessed.
    d. Click on “Add” button. This adds the processed output data to “est” dataset.

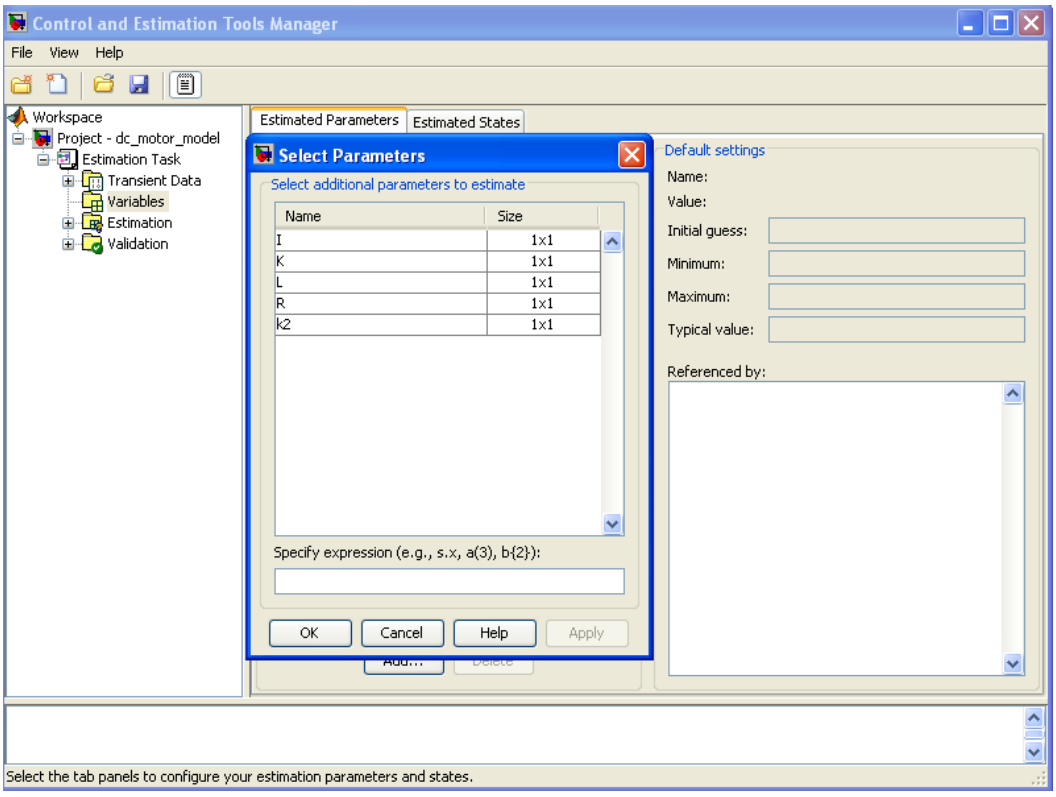
    Once the input and output data are preprocessed, the variables that are to be estimated must be selected.



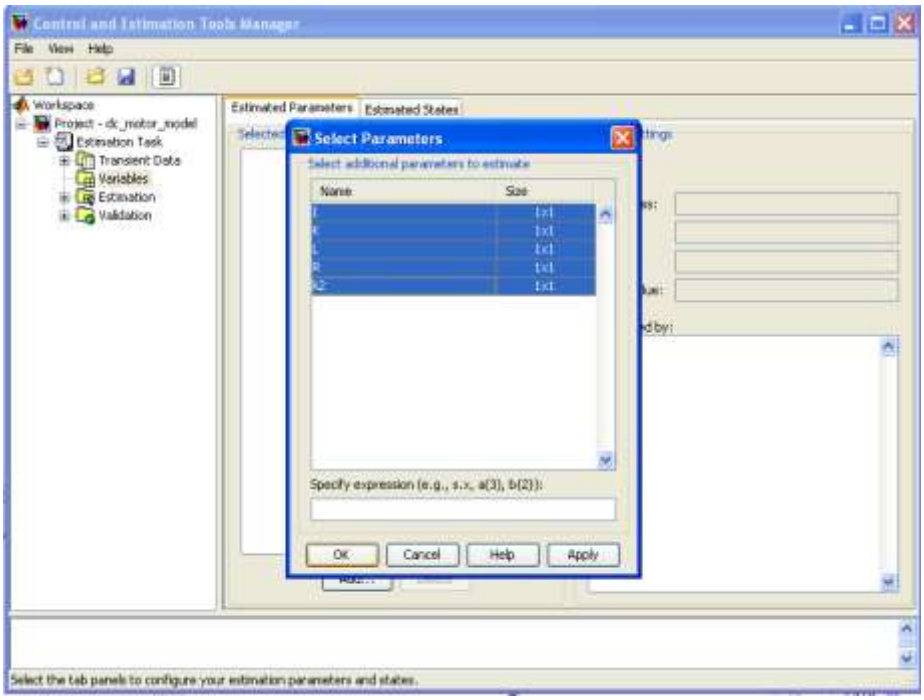
27. Click on “Variables” to select the variables to be estimated. Click on “Add” button to add the variables.



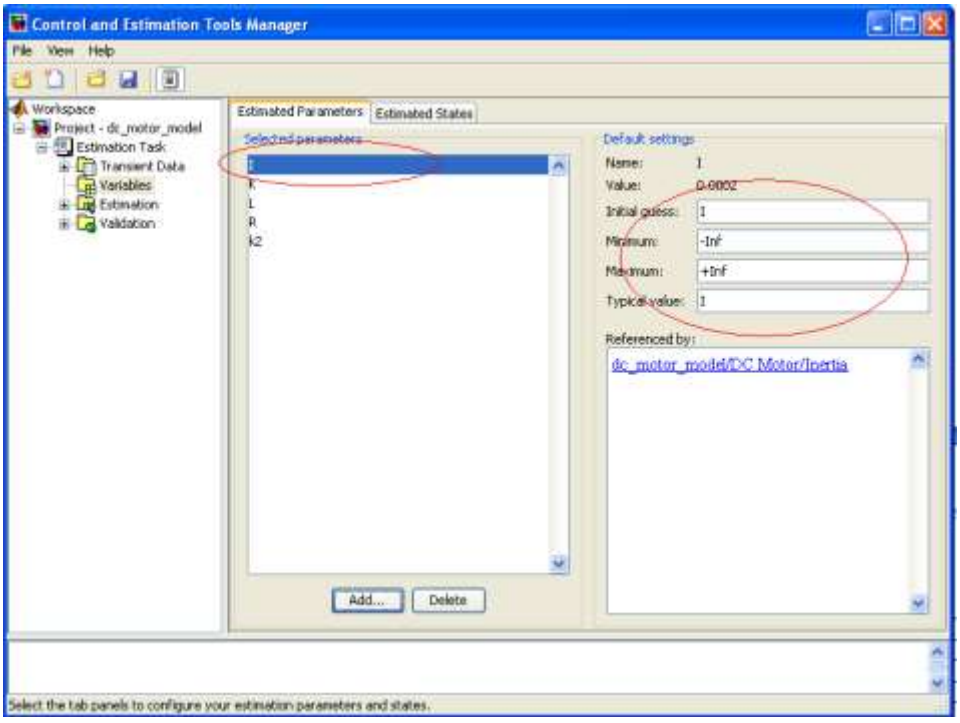
28. This opens a dialog box.



29. Select all the variables for estimation. Click on “OK” button.



30. Choose the parameter “I” from the list and then specify the Initial, minimum and maximum values.

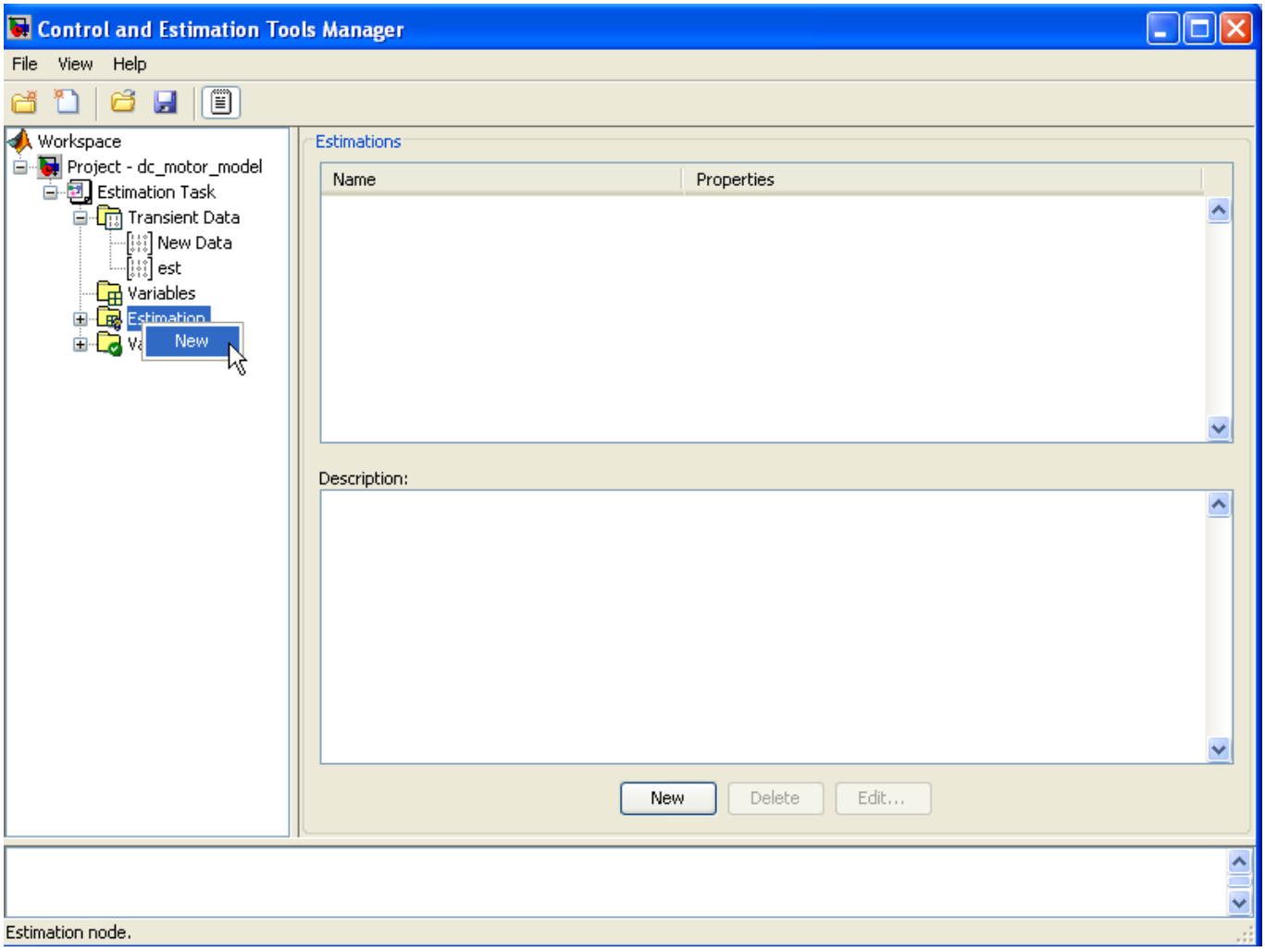


31. Typical values for all the motor parameters in the set-up are given here.

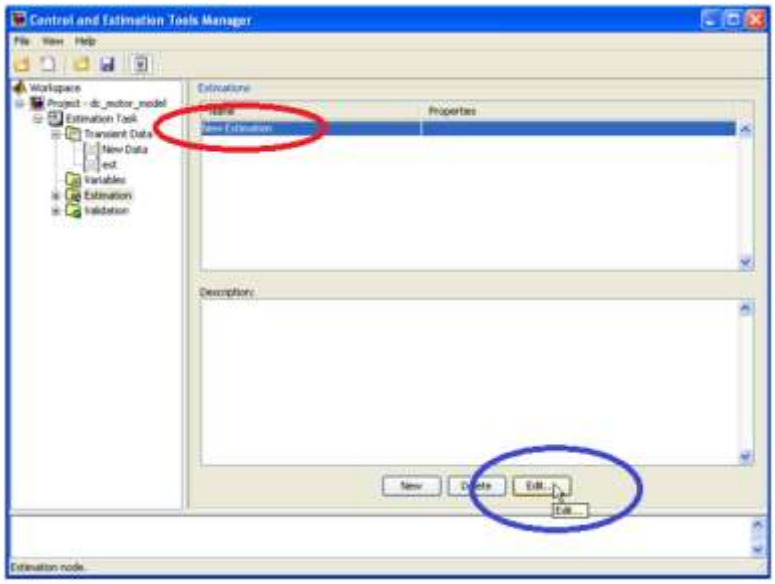
Parameter	Initial Value	Minimum Value	Maximum Value
I	0.0002 Kgm^2	0.0001	0.001
K	12/4000 V/rpm	0.1/4000	200/4000
L	7 mH	0	400
R	10 Ω	0	400
K2	12/4000 V/rpm	0.1/4000	200/4000

Once the minimum and maximum values are specified for all the variables, parameter estimation can be performed using the imported input and output data.

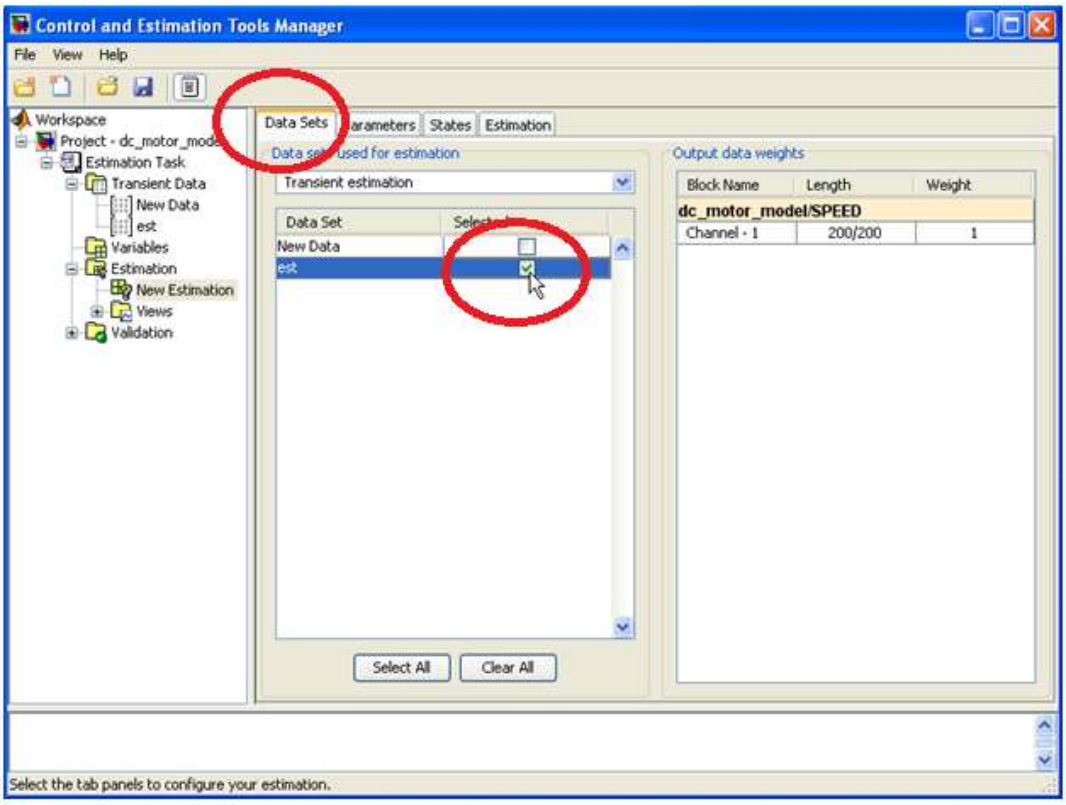
32. Right click on “Estimation” and then click on “New”. This adds a new estimation to the environment.



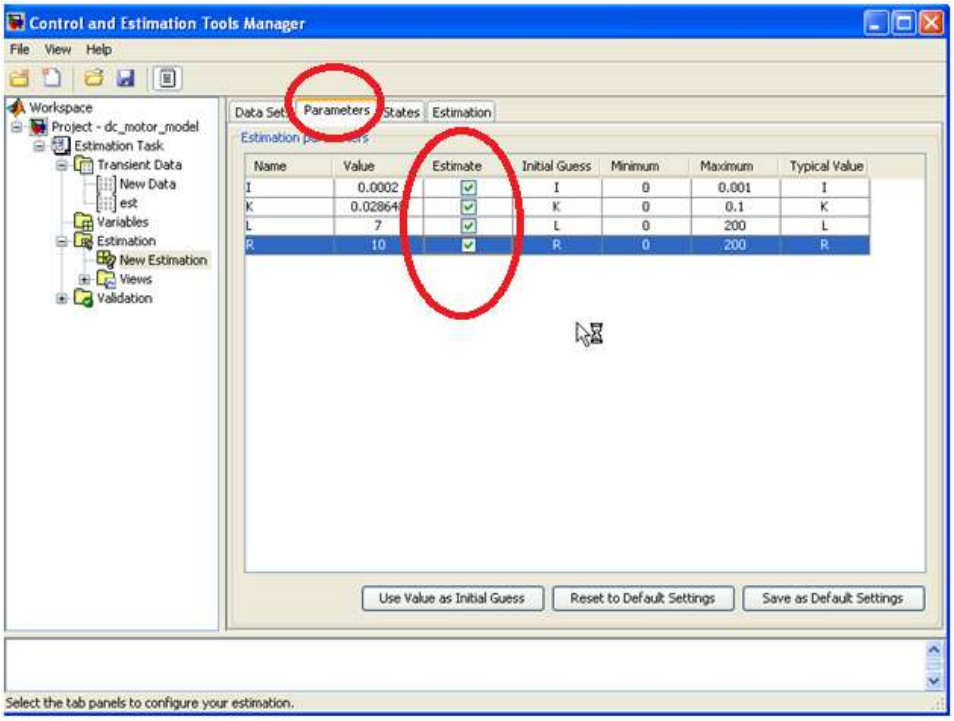
33. To specify the parameters to be estimated and the estimation options, select the “New Estimation” encircled in red and click on “Edit” button.



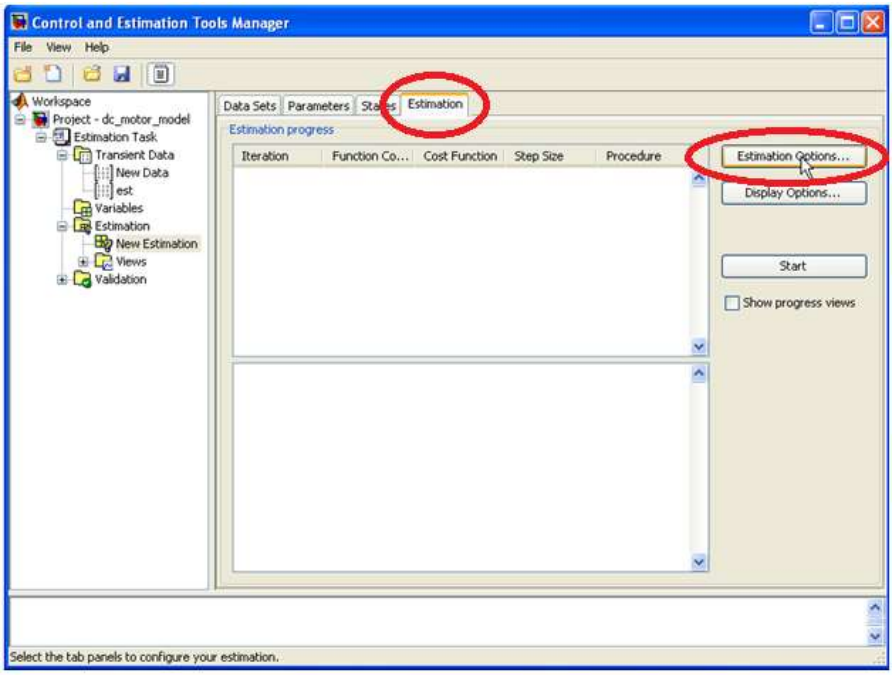
34. Select the “est” data for estimation under the “Data sets” tab.



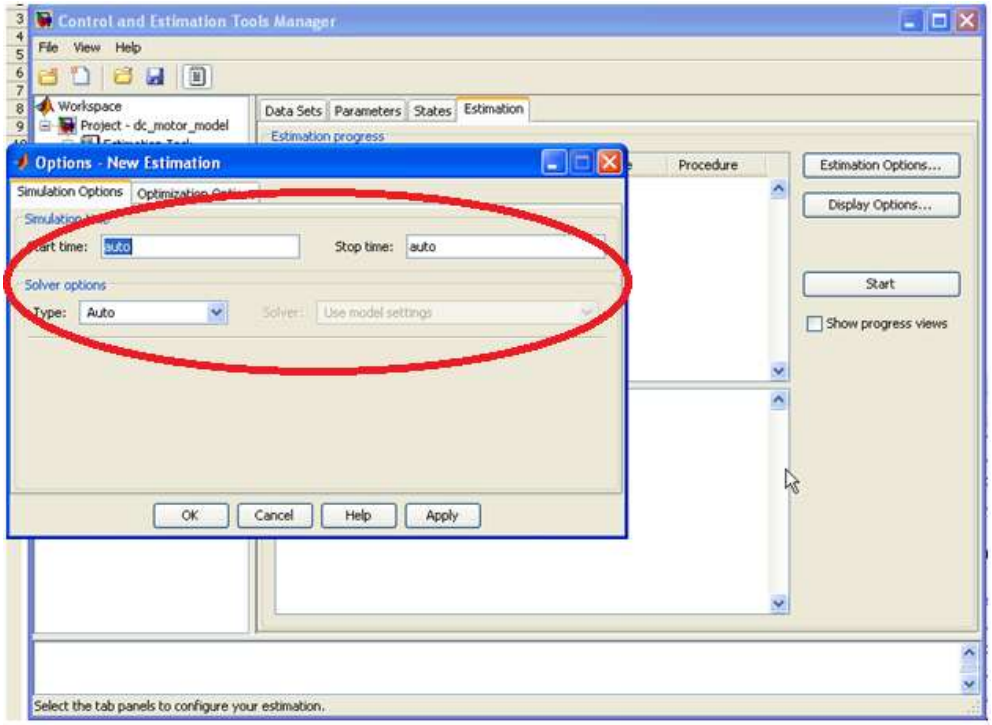
35. Under “Parameters” tab, select the parameters to be estimated.



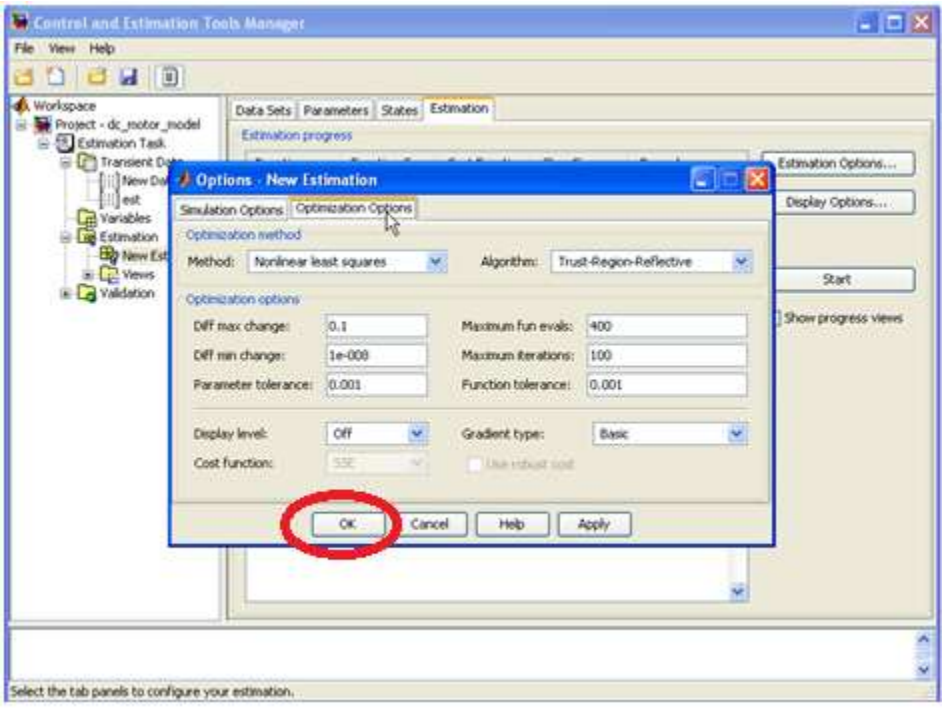
36. Under “Estimation” tab, click on “Estimation Options” button to select the estimation options.



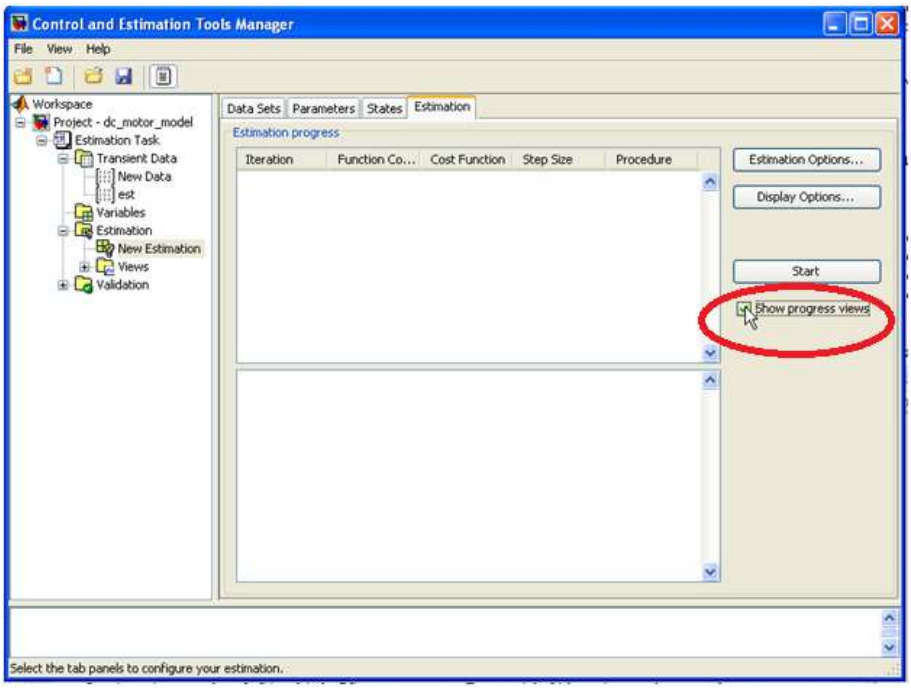
37. This opens a dialog box. Under the “Simulation options” tab of the options dialog, all the options can be chosen to be “auto”.



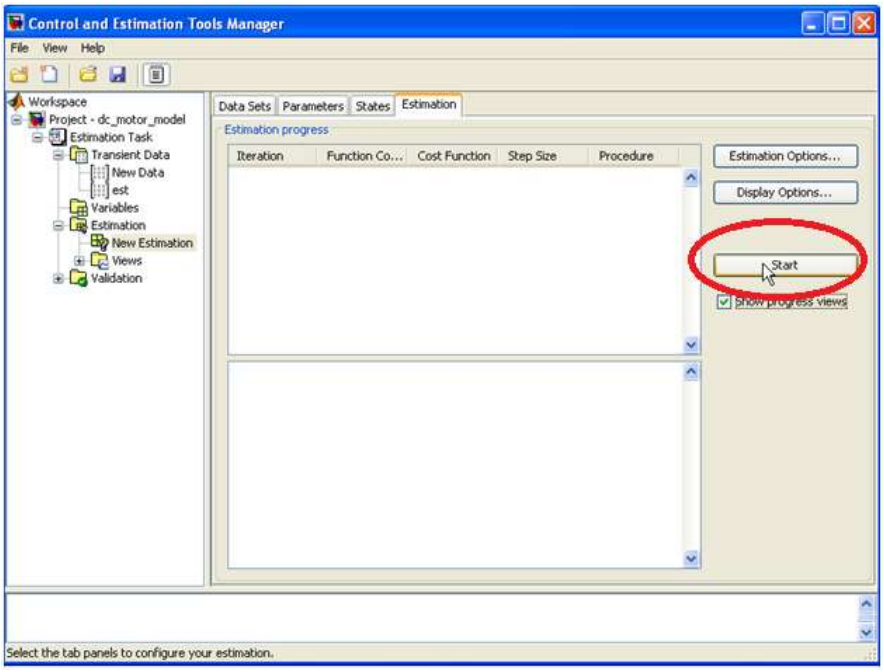
38. Click on “Optimization options” tab. Keep these as default values and try estimating. Click on “OK” button.



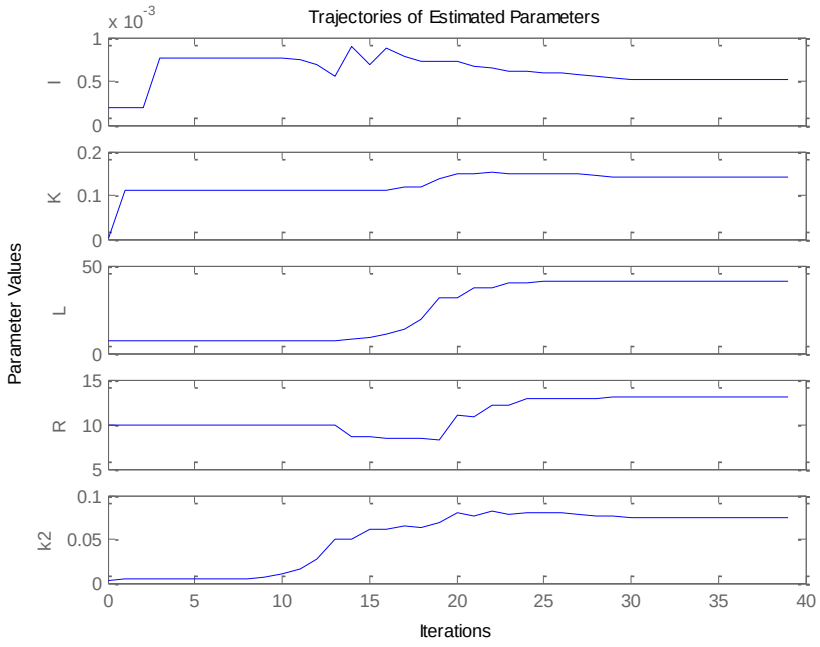
39. Select the option “Show progress Views”. This plots the trajectories of the variables and the estimated response during the estimation process

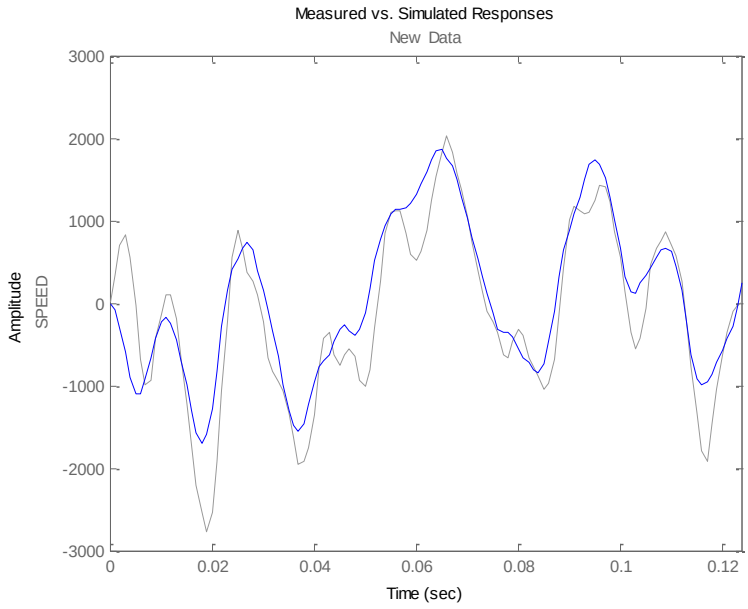


40. Click on “Start” to start the estimation process.



41. The estimation process is displayed. The parameter trajectories and the measured and simulated response are also plotted.



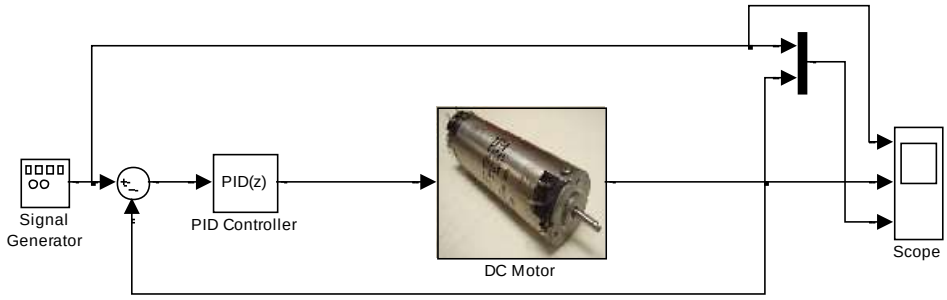


42. Once the estimation is complete, the final values of the variables are updated and the Simulink model is also updated. Do not clear the workspace in Matlab or exit Matlab. (For your reference write down the values).

3.2.4 PID CONTROLLER DESIGN & VERIFICATION

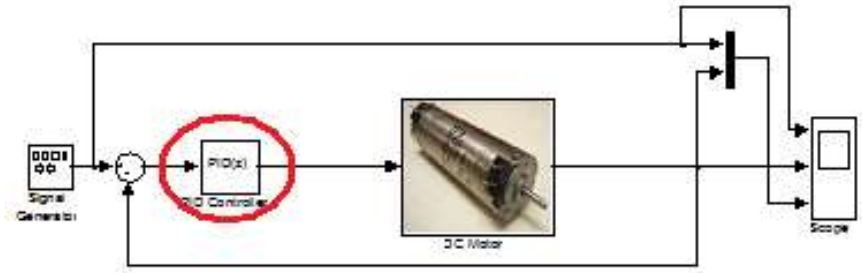
Follow these steps to design a PID controller for the DC motor model.

1. Motor speed controller (PID) Automatic design.
   Double click on the Simulink model file "simscape_dc_motor_PID_control.mdl". The values of the motor model parameters should be in the workspace.



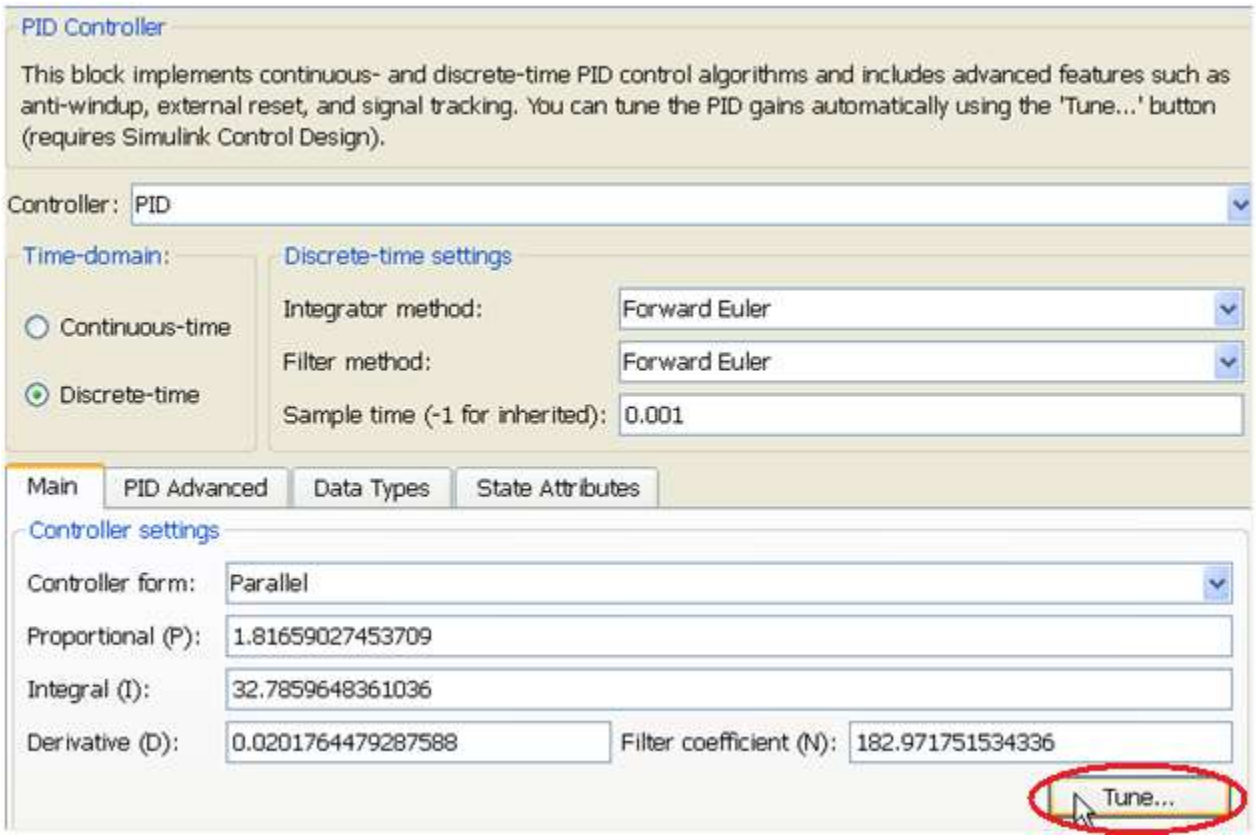

```
plot(linear_PID.time(1:20001,1), linear_PID.signals(1,1).values(1:20001,1), linear_PID.time(1:20001,1),linear_PID.signals(1,2).values(1:20001,1))
>> grid on;
>> title('Speed Response');
>> xlabel('Time in seconds');
>> ylabel('Speed in rpm');
>>
```

2. Double click on the block PID. This will open a new dialog box.

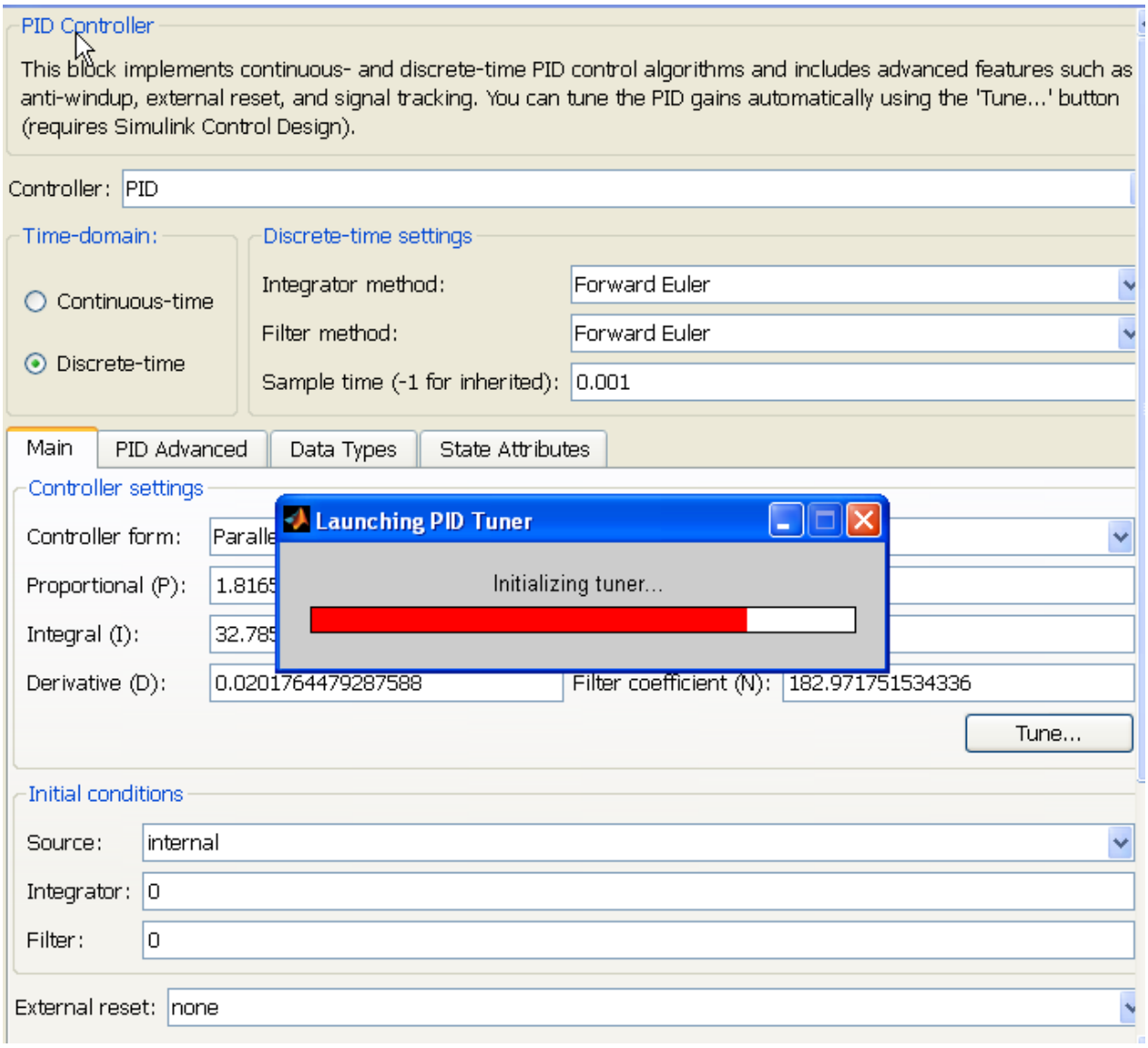



```
plot(linear_PID.time(1:20001,1), linear_PID.signals(1,1).values(1:20001,1), linear_PID.time(1:20001,1), linear_PID.signals(1,2).values(1:20001,1))
grid on;
title('Speed Response');
xlabel('Time in seconds');
ylabel('Speed in rpm');
```

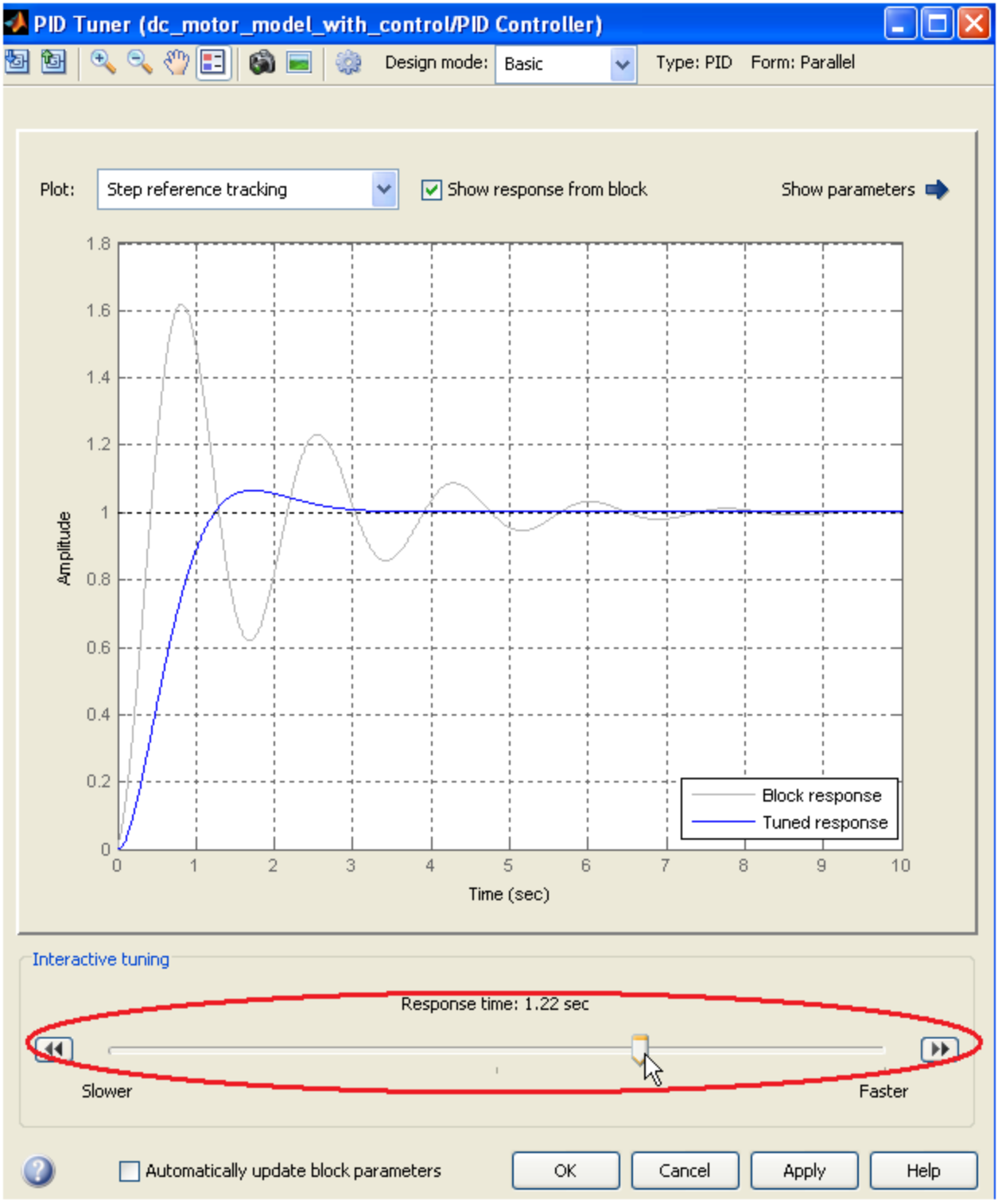
3. Click on the “Tune…” button. This will launch the PID tuner.



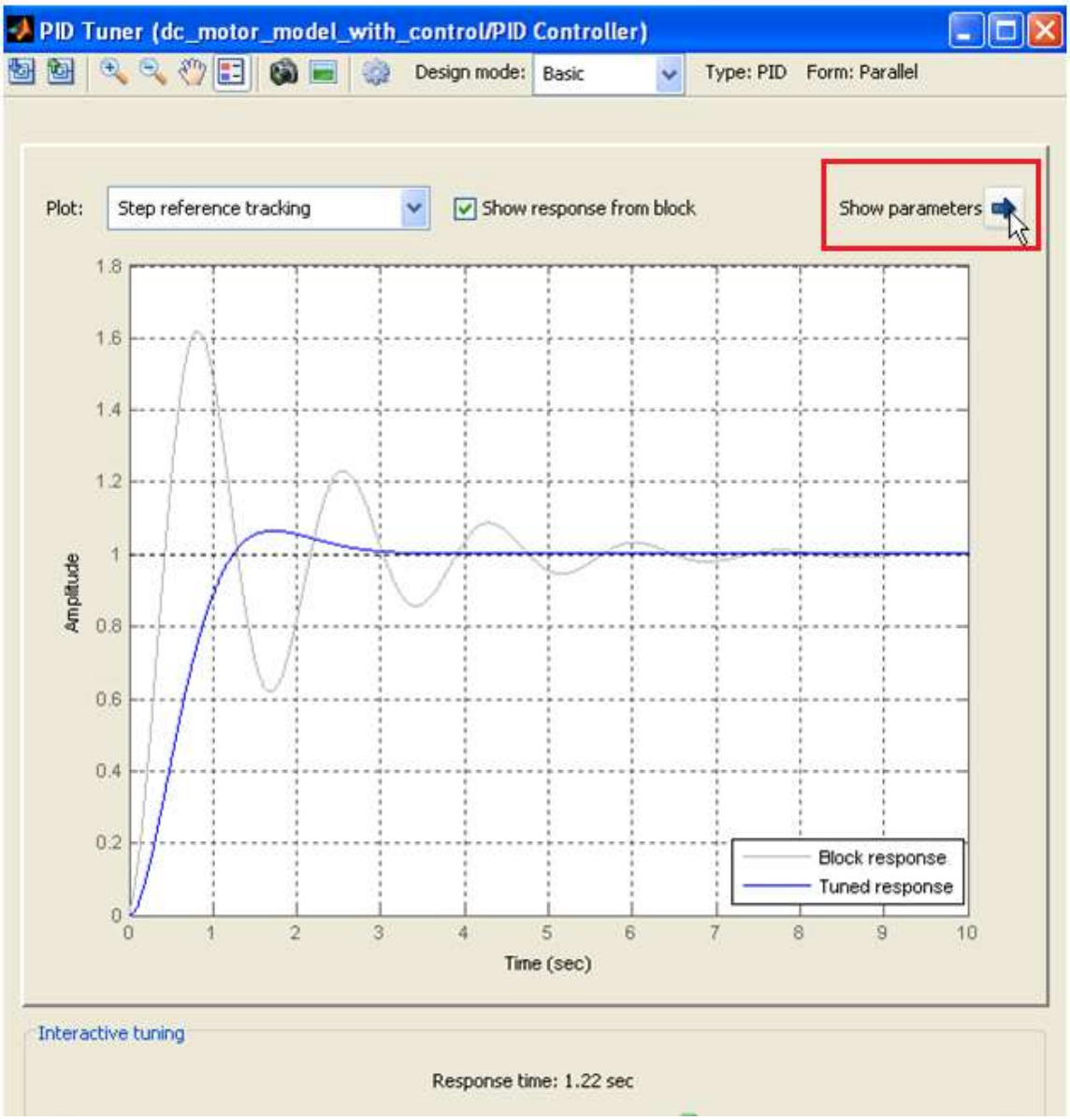
4. A window appears and displays the intermediate progress as shown below: It initializes the tuner and analyzes the model.



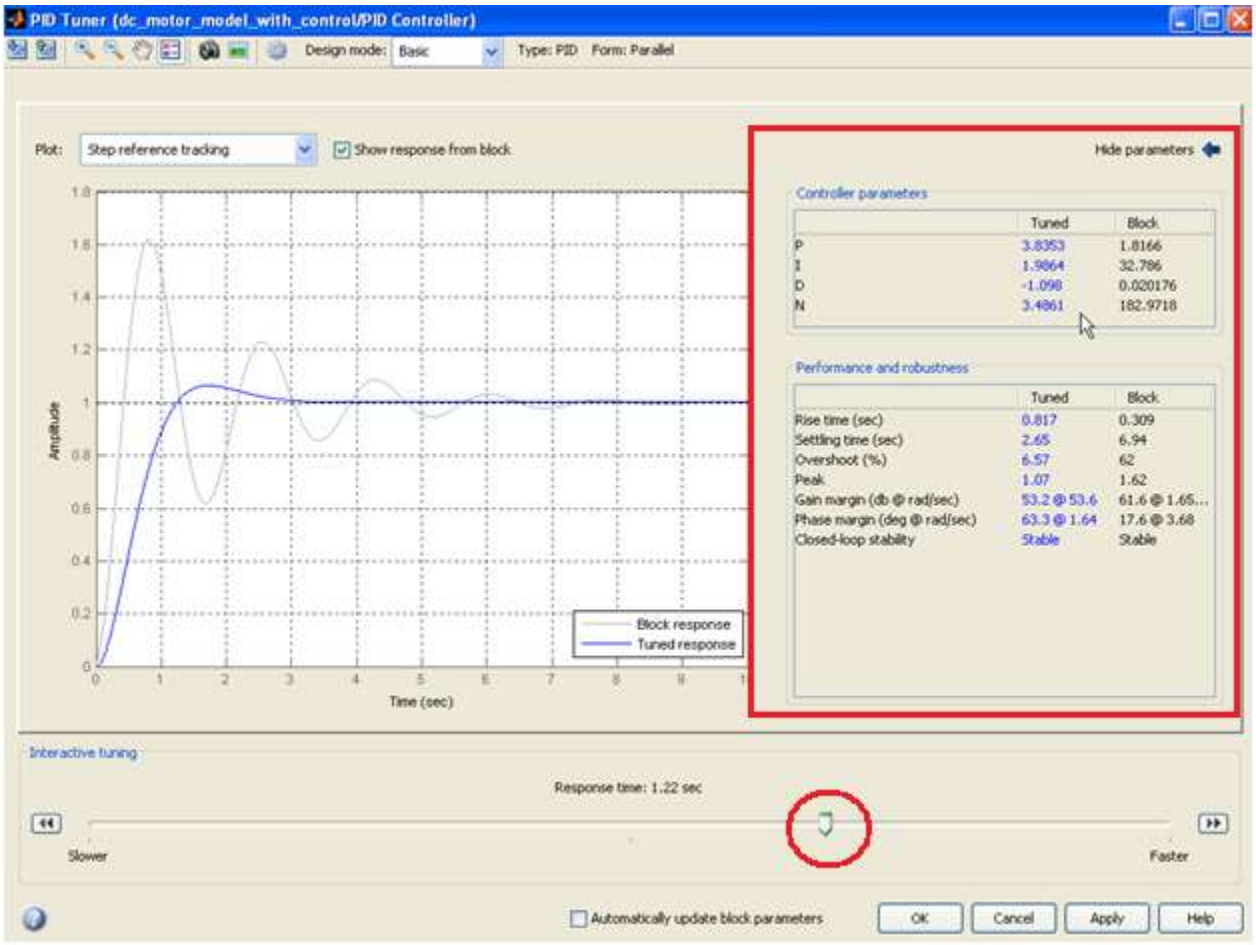
5. Once this is done successfully, a PID Tuner window displays the tuning progress of the PID control gains. When completed, one can re-adjust the gains to modify the response specifications.



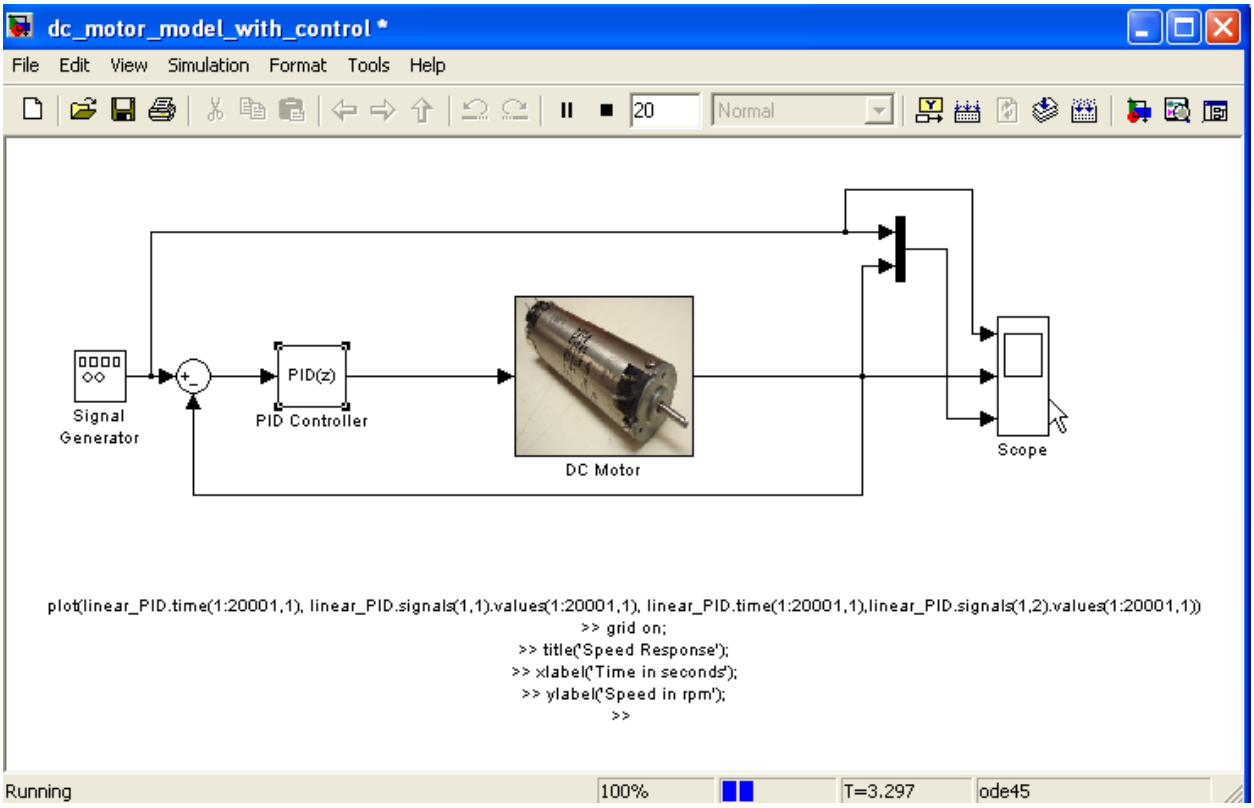
6. To see the PID and system performance parameters click on the button "Show parameters" as shown below.



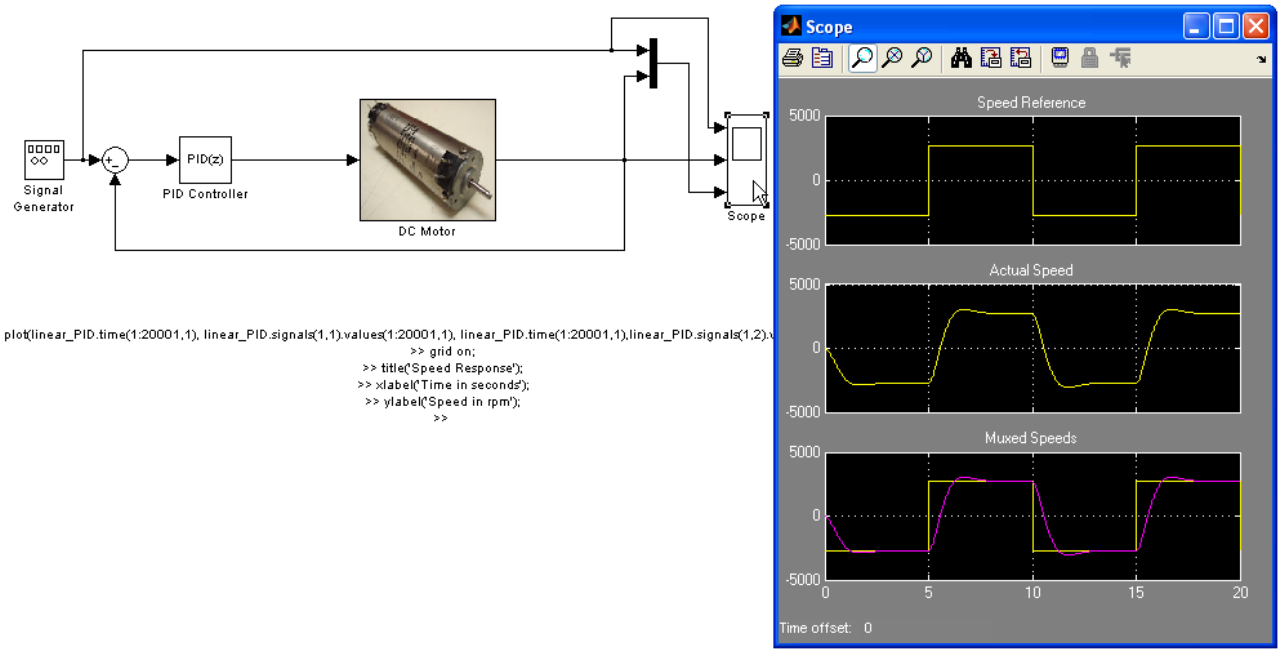
7. Click on “Show parameters” arrow button to view the parameters. One can see the change in the controller and performance parameters by varying the slide toolbar.



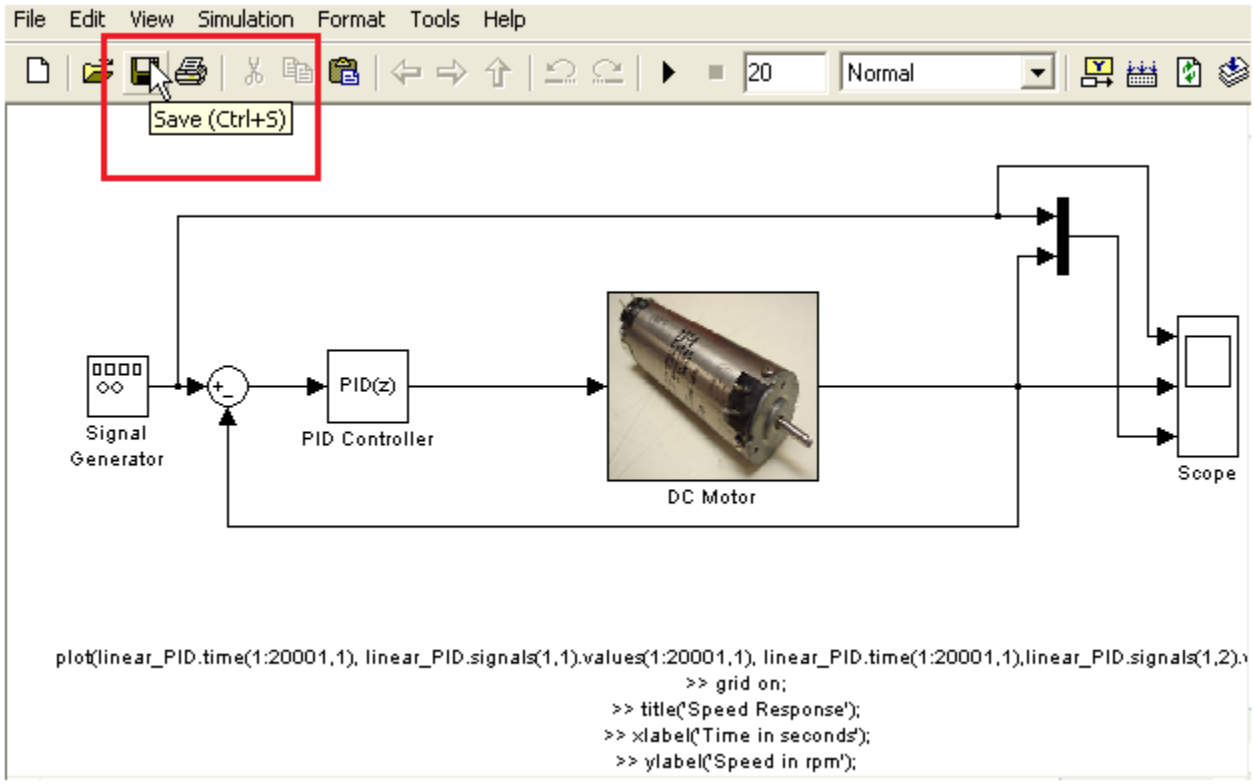
8. Once the desired performance is achieved, then click on “Apply” and “OK” button. This will update the new tuned PID and N values of the block. Run the model in simulation mode to observe the performance for a given input on the scope.



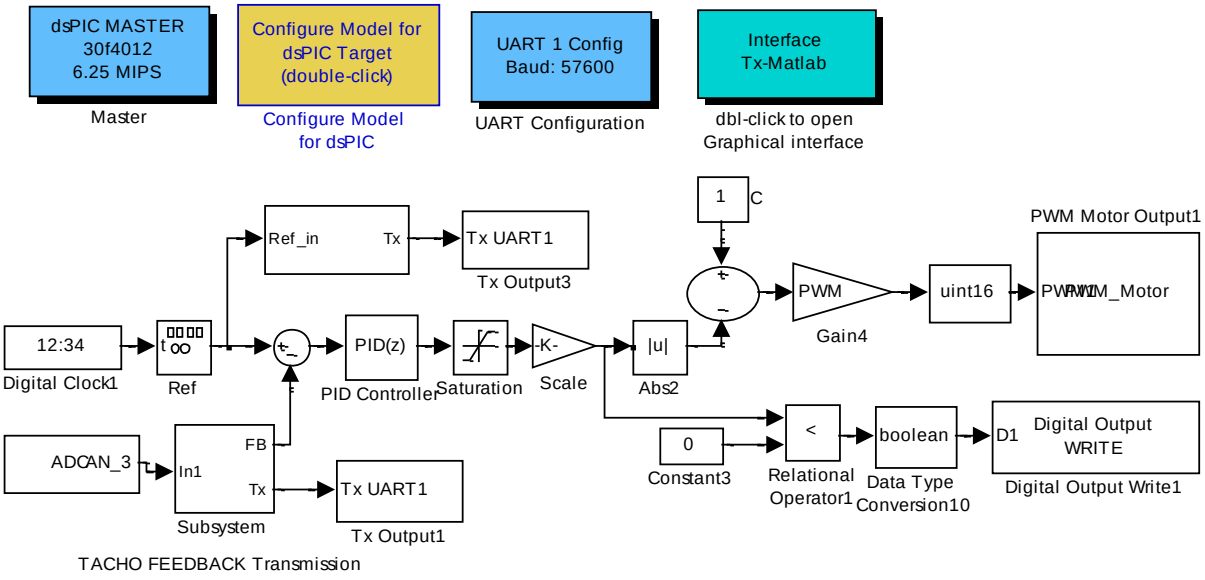
9. Double click on the scope when the simulation is completed. And check to confirm that the desired performance is achieved.



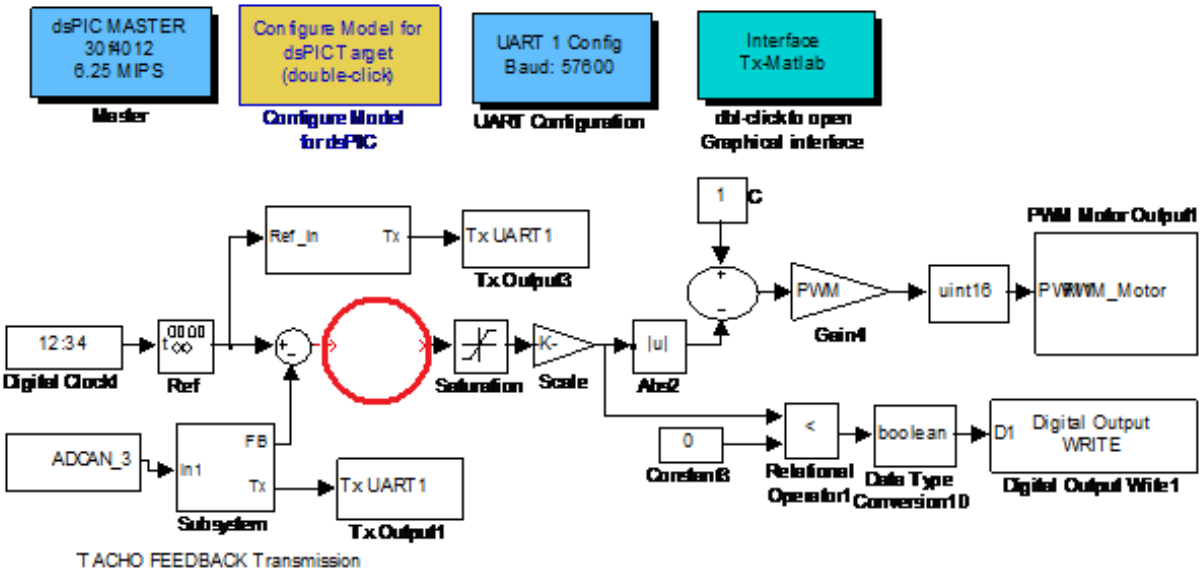
10. Make sure to save the changes done to the model. And do not close the model.



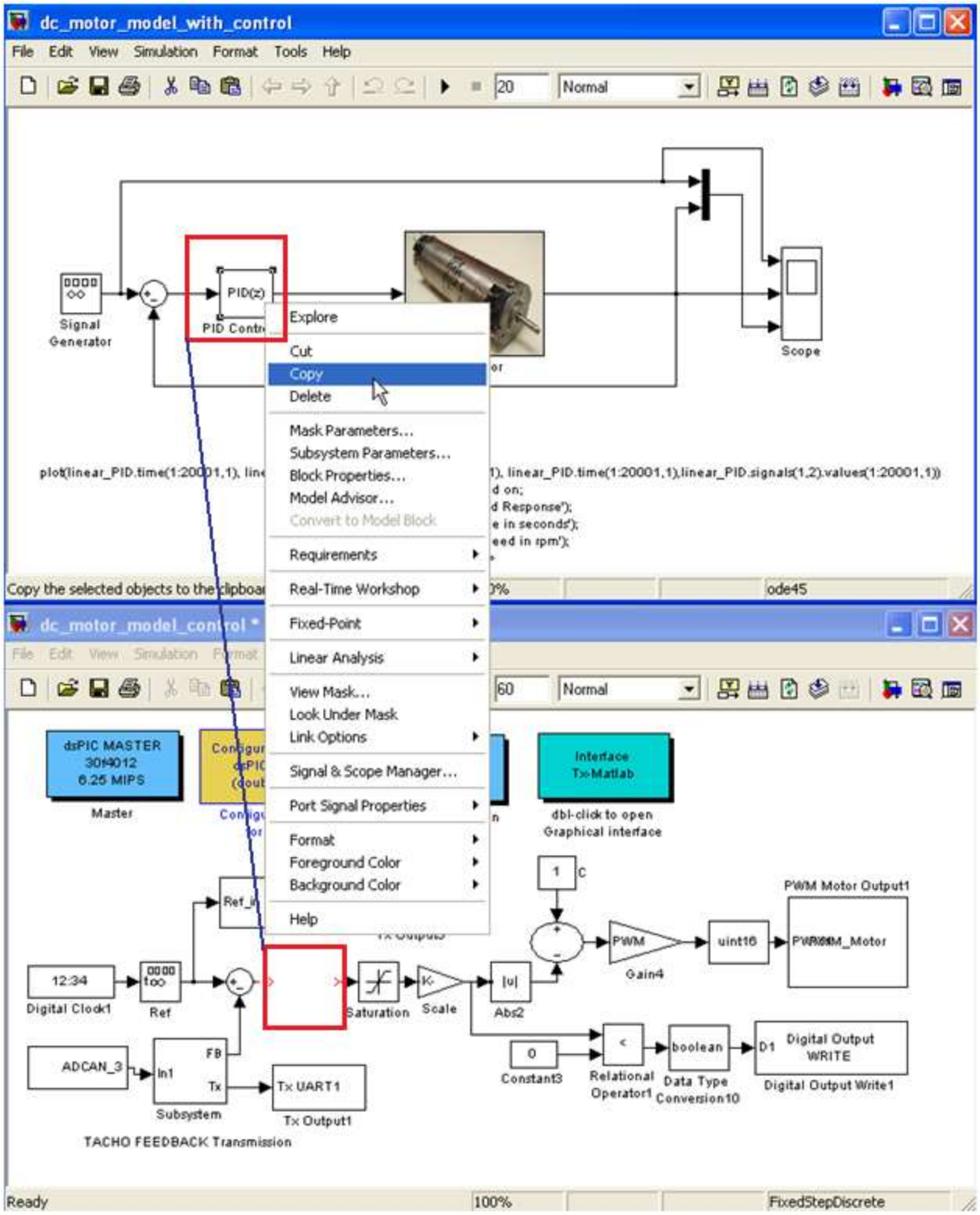
11. Open “dspic_dc_motor_control.mdl” Simulink file.



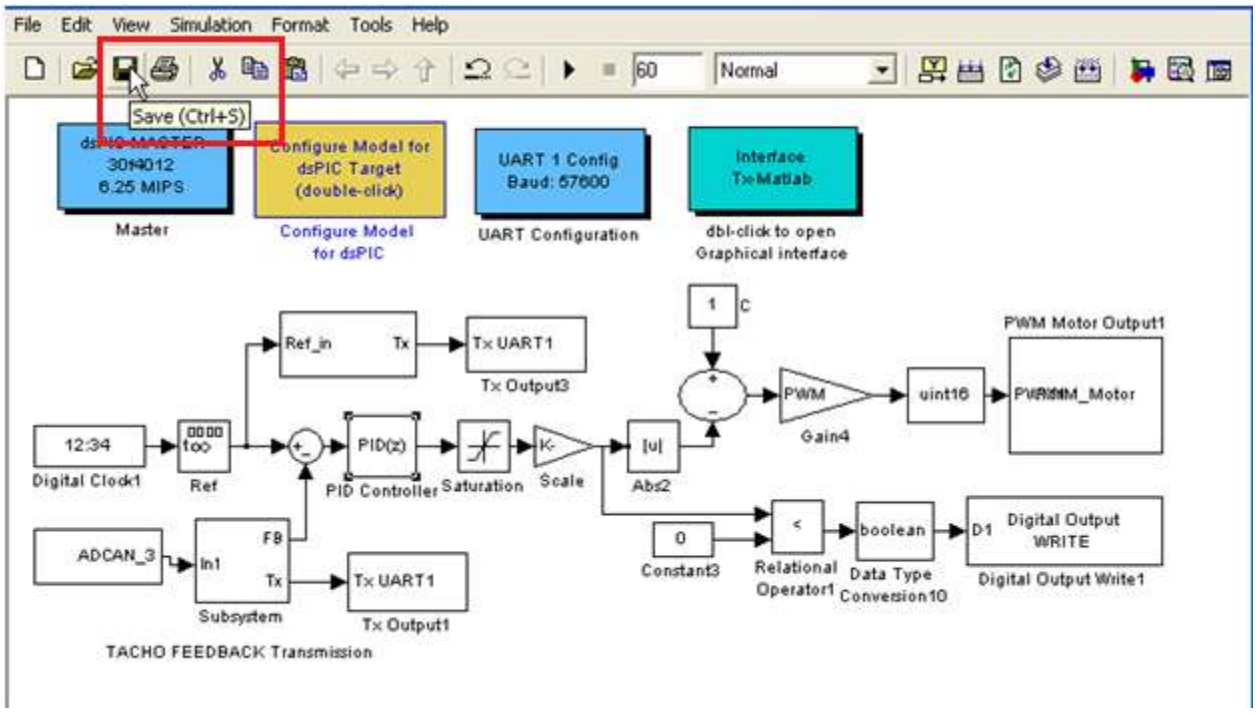
12. Click and delete the PID Controller block.



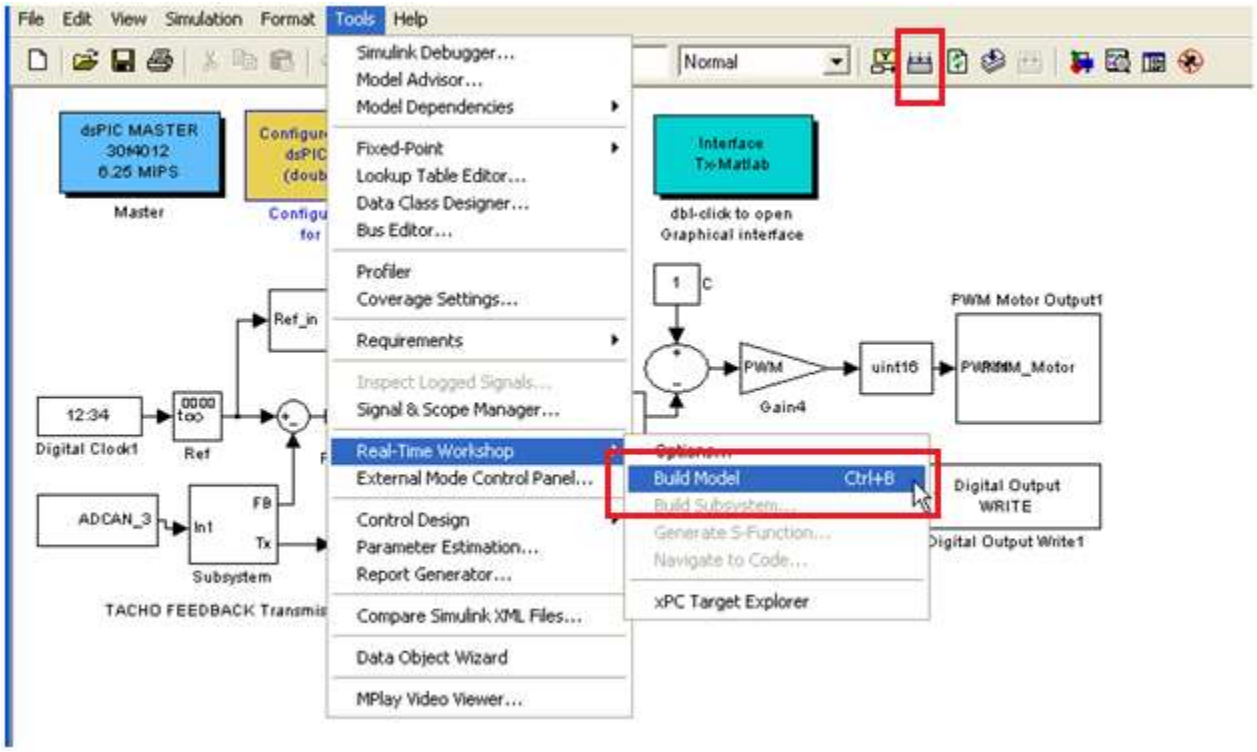
13. Copy the PID controller block from “simscape_dc_motor_PID_control.mdl” and paste it in the “dspic_dc_motor_control.mdl”.



14. Make sure to save the changes in “dspic_dc_motor_control.mdl”.



15. Incremental build the model to generate the hex file. This hex file can be downloaded to the dsPIC controller to verify the PID controller performance.

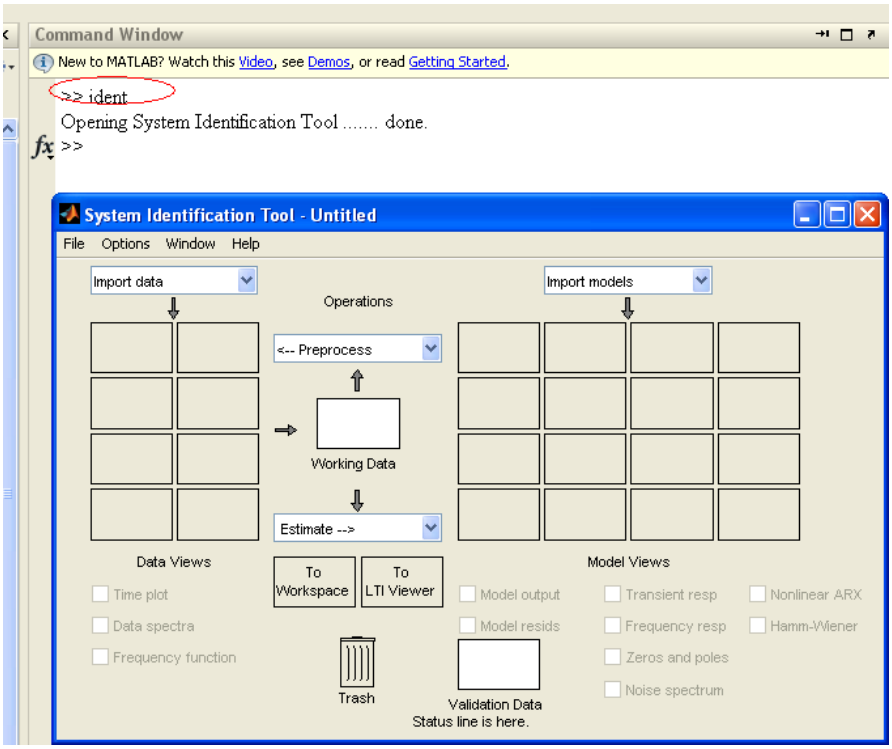


16. Once the hex file is successfully generated, repeat the procedures 6 to 13 of Section 3.2.1
17. Repeat procedures 14 to 24 of Section 3.2.1, but make sure to change the name of “xxxx.txt” in step 21 (not the name used during data capture).
18. Open “RESPONSE_PLOT_swap.m” from the path “Desktop\your_folder\dsPIC_motor_model”.
19. Specify the file name of the data file (text file) as parameter to the “fopen” command. This is line 4 of the “RESPONSE_PLOT_swap.m” file.
20. Debug -> Run the “RESPONSE_PLOT_swap.m” file.
21. It takes a while to complete the process and plot. Once the data process is done, it plots the speed response.
22. Make sure to check if the actual motor speed follows the reference speed in the plot with the desired specifications.

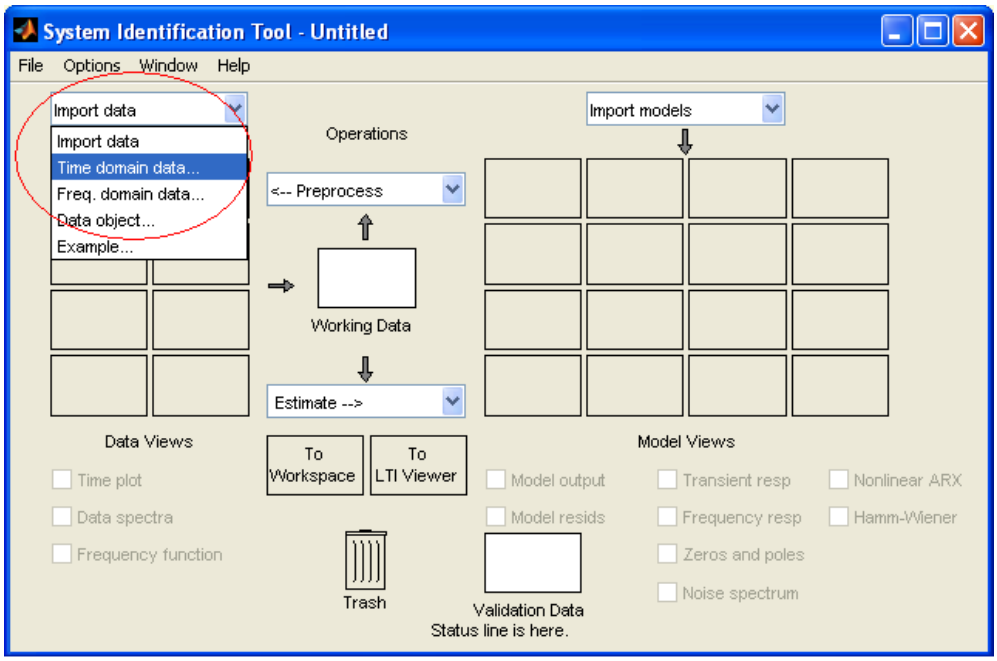
3.2.5 SYSTEM IDENTIFICATION USING SYSTEM IDENTIFICATION TOOLBOX

The sections 3.2.3 and 3.2.4 explain the procedure to model parameter estimation and controller design using SIMULINK Control Design toolbox. The same can be done using System Identification toolbox without developing the Simscape model of the DC motor. The procedure is explained in this section.

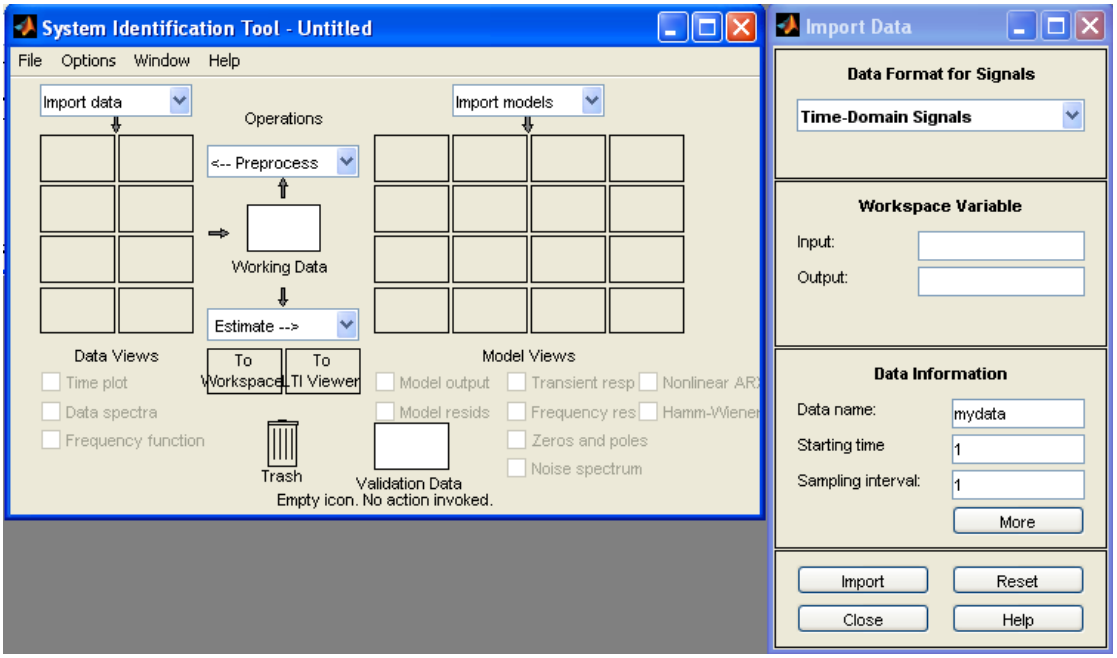
1. The procedure explained in sections 3.2.1 and 3.2.2 is to be followed to capture the I/O data and data processing.
2. The procedure explained in sections 3.2.3 and 3.2.4 can be ignored if procedure in this section is desired.
3. Type “Ident” in Matlab command window and press Enter. It opens the System Identification tool.



4. Click on “Import Data” and then select “Time domain data” to import the motor input and output data.

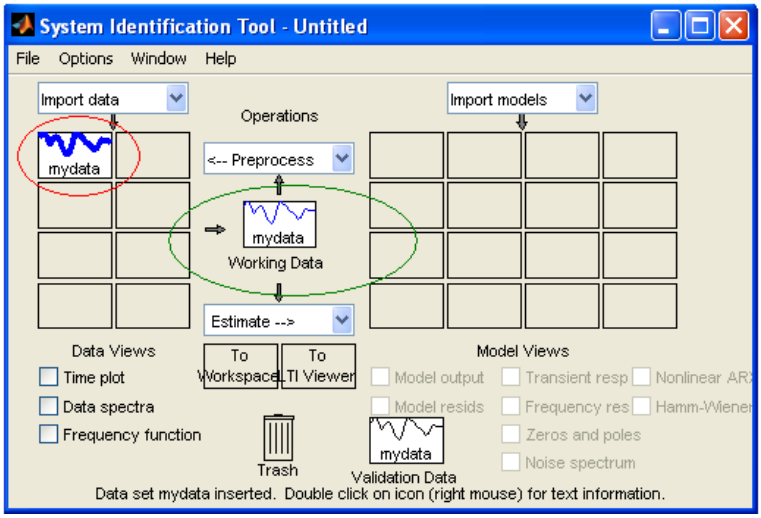


5. This opens up a new dialog box.

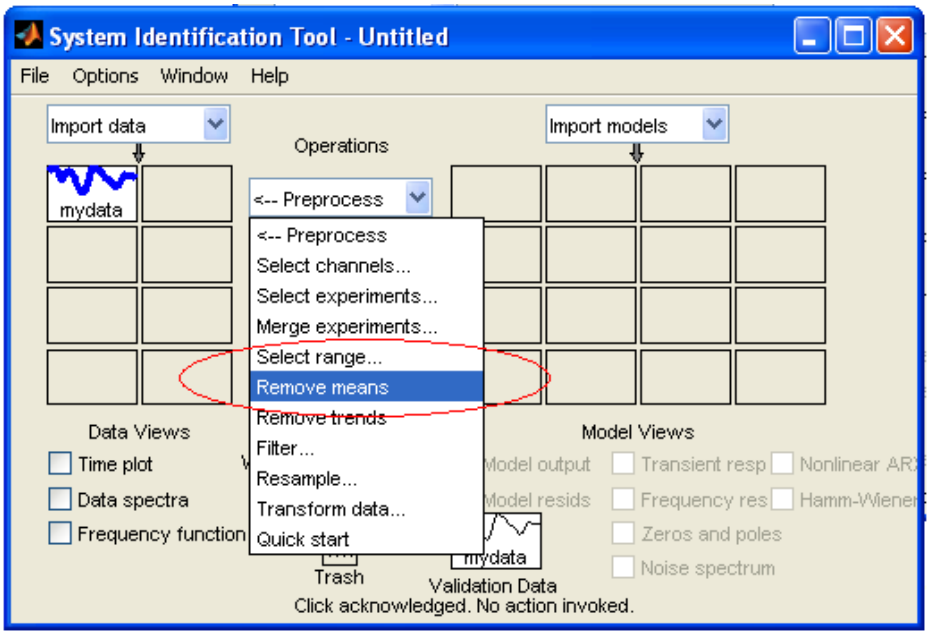


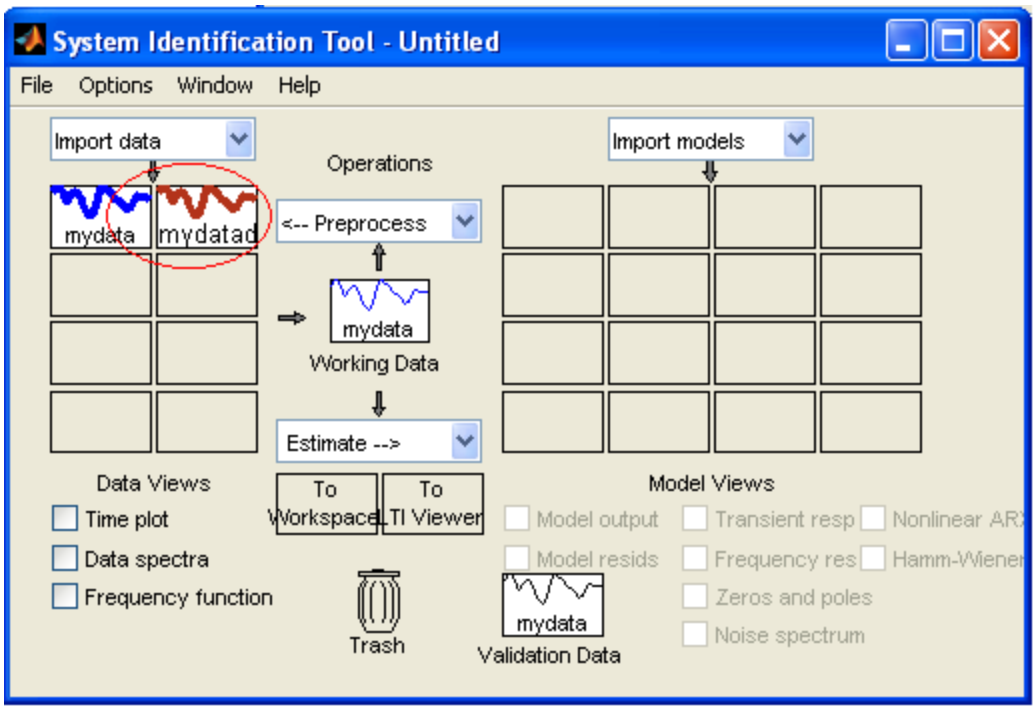
6. Specify the workspace variables in the dialog box. Specify the starting time and the sampling interval of the data. Click on “Import” data to import the data.

7. This opens a new dialog box with the data imported. The data “mydata” as given in the Import data dialog window gets imported. This is highlighted in red. Make sure the working data is “mydata”.

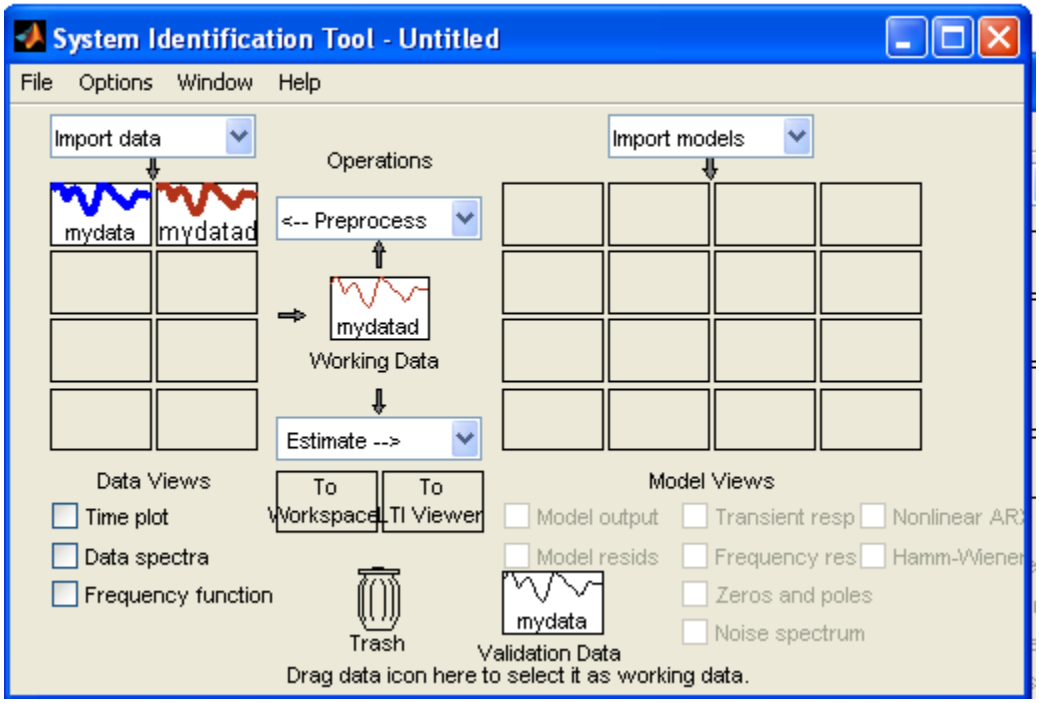


8. Click on “Operations -> Preprocess” drop down to preprocess the data. Click on “preprocess” and select “Remove means” to remove the means from the data. This adds a new data to the data set “mydatad”.

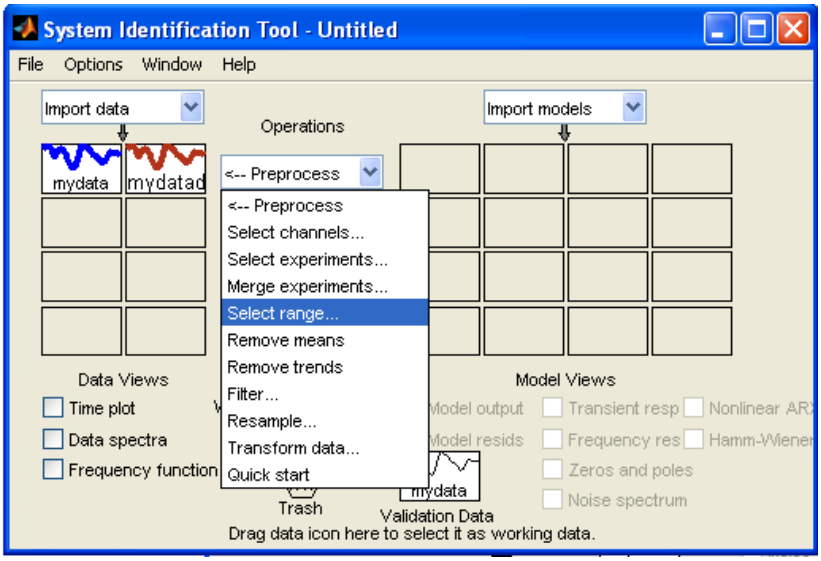




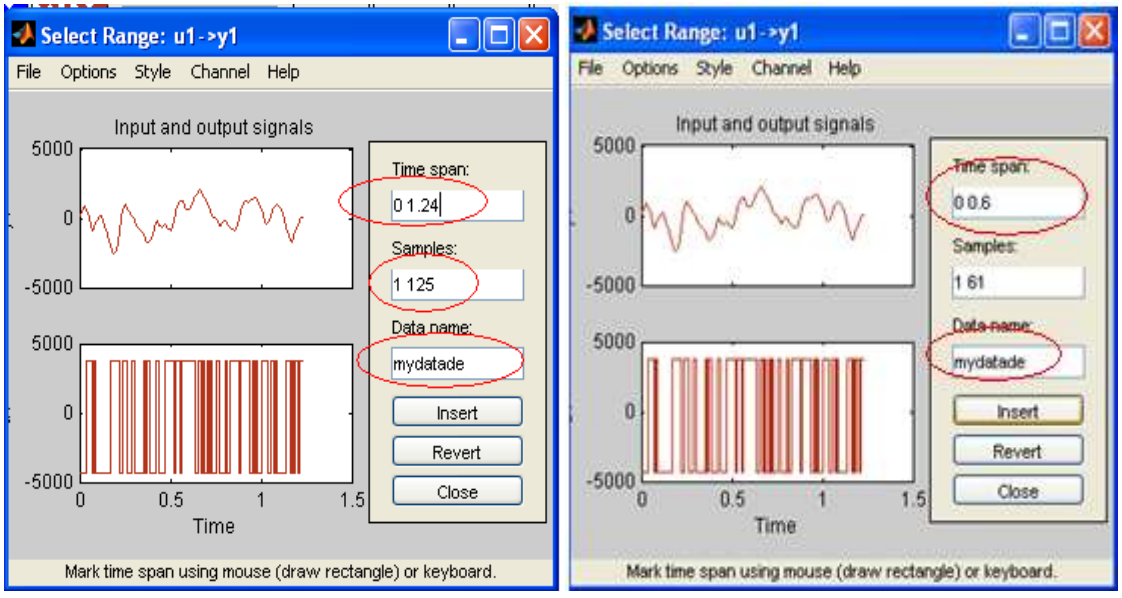
9. Click on “mydata” data, drag and drop it on the working data. After this operation, the window should look like this.



10. Click on “Preprocess” again to select the range of input and output to be used for estimation and validation purposes.

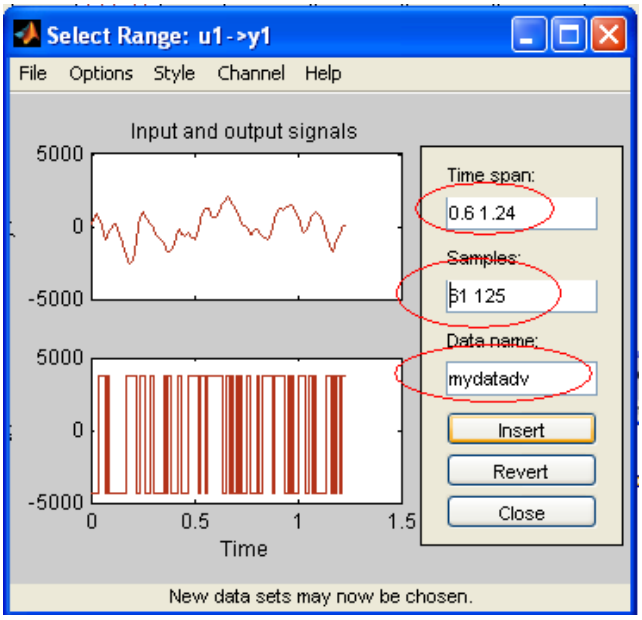


11. This opens another window to select the range. The entire range is given in the “Time span” text box. The number of samples is also given in the Samples text box.

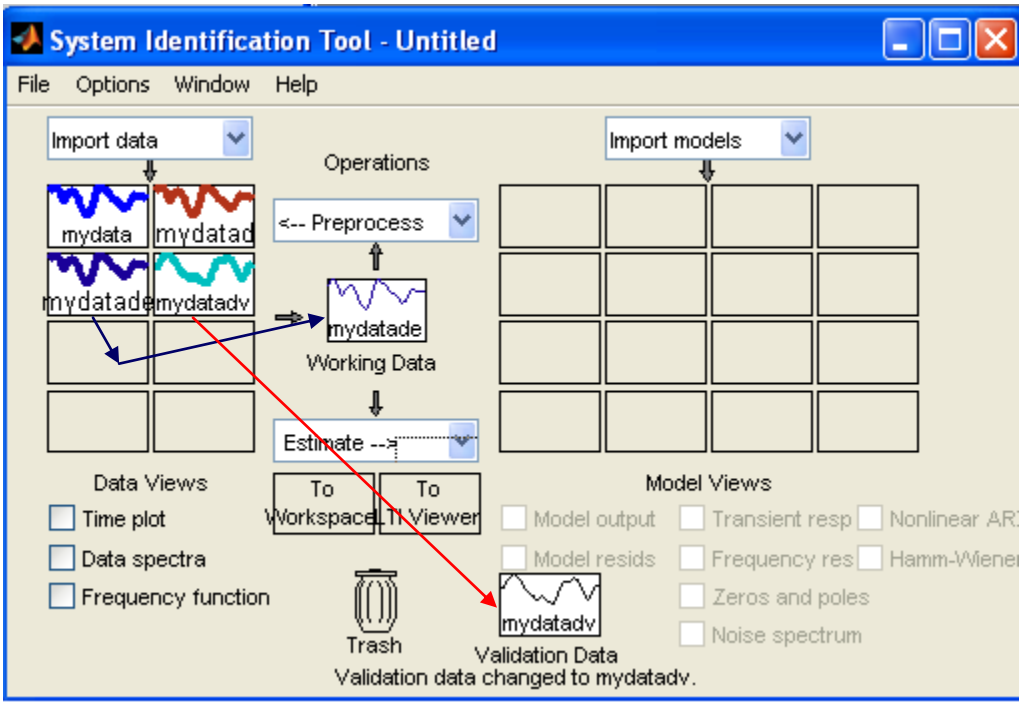


Specify the time frame as given in the figure. The data name can be “mydatade” for estimation. Click on “Insert” to insert the data for estimation.

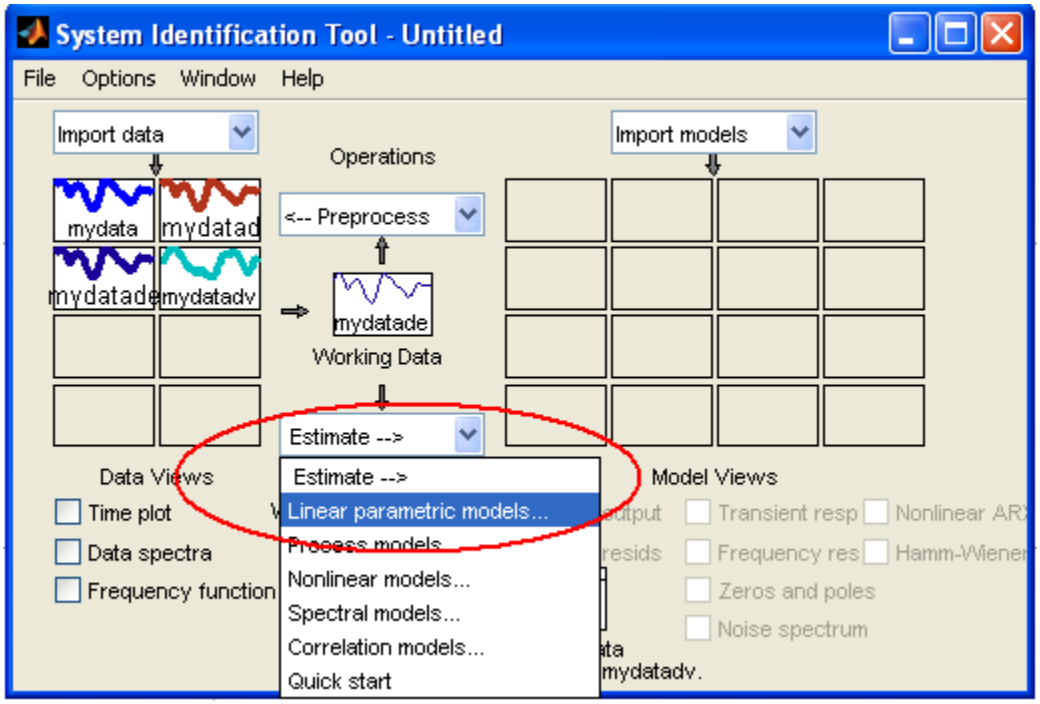
12. Once the range is chosen for the estimation, the data range for the validation data should be specified. Specify the time frame for the validation data as given in the figure. Click on “Insert” to insert the validation data.



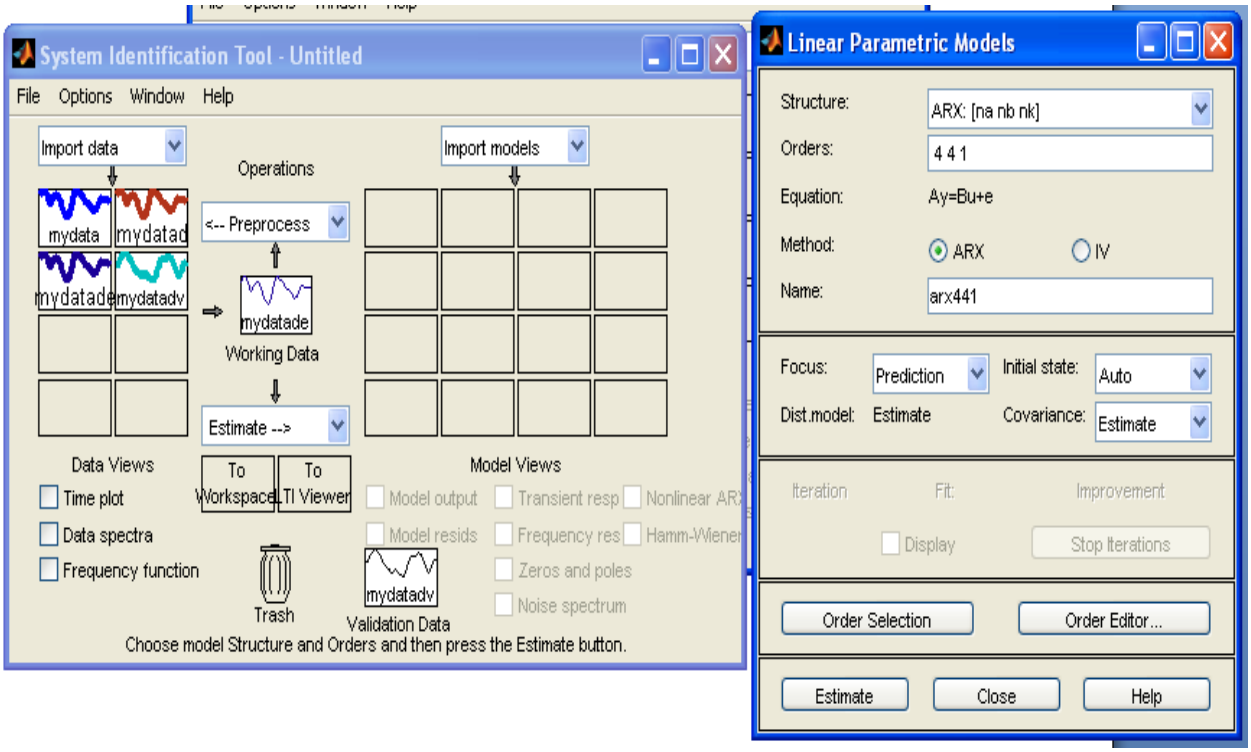
13. Move the “mydatade” estimation data to the “Working data “ box and the “mydatadv” validation data to the “Validation data” box.



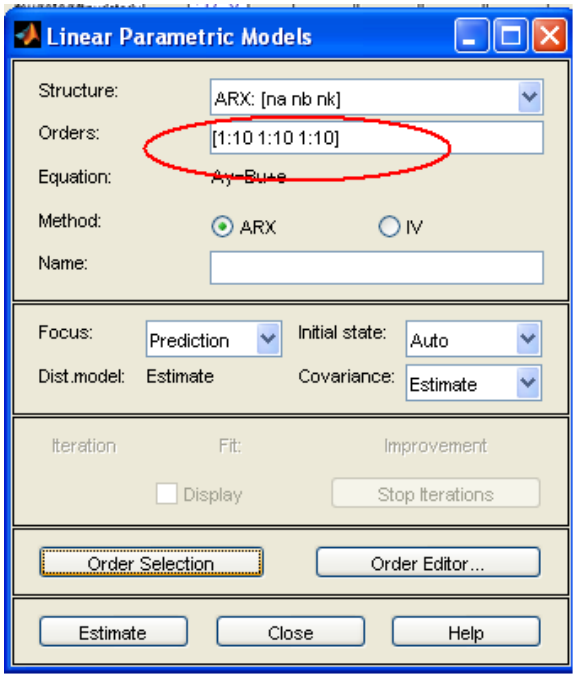
14. Once the data are selected for the estimation and validation, the model can be identified. Click on “Estimate -> Parametric models” as given in the figure.



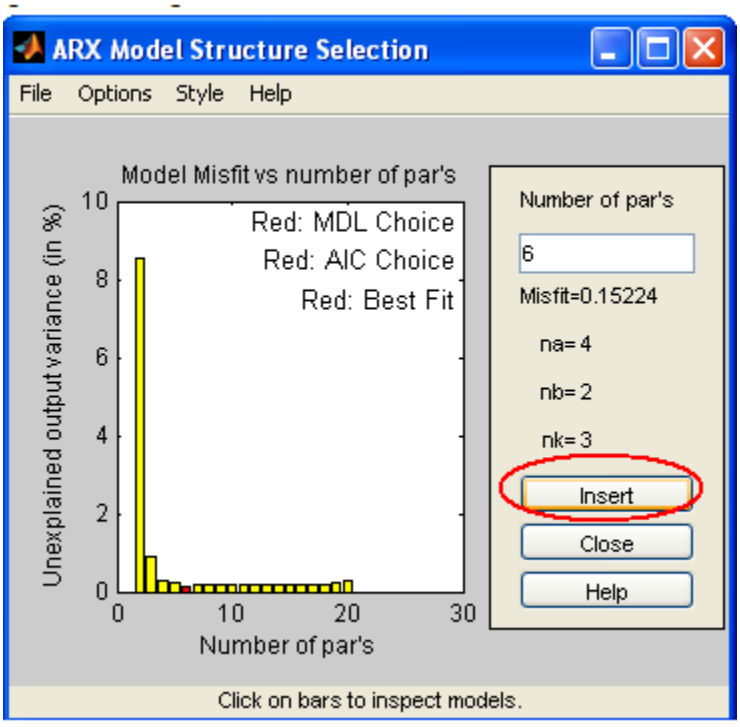
15. This opens another window.



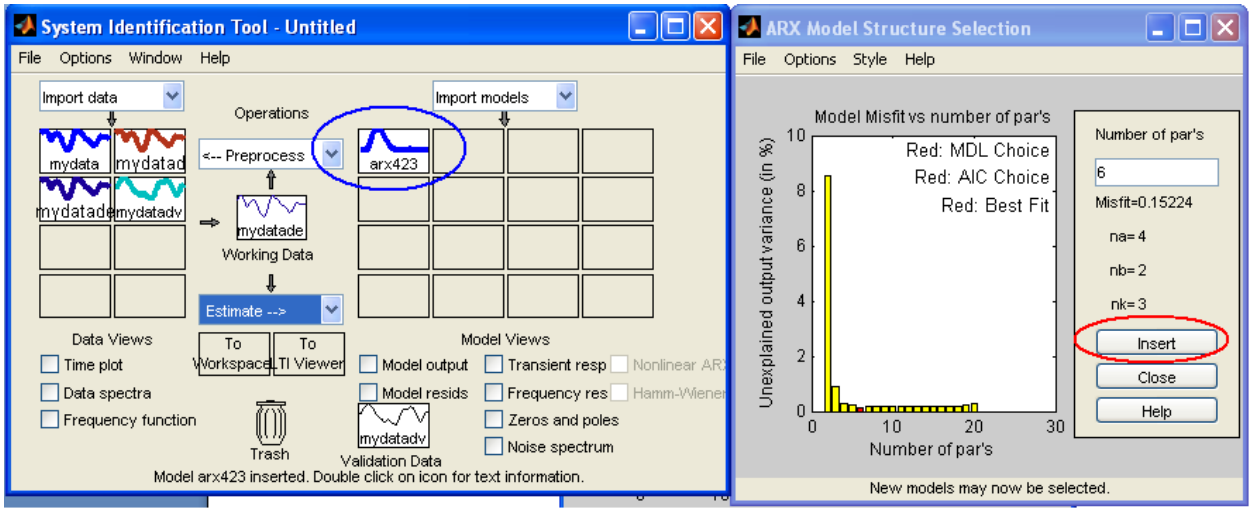
16. Click on “Order selection” to let the toolbox choose the order of the system. The “orders” text box is updated with the new values as given in the figure.



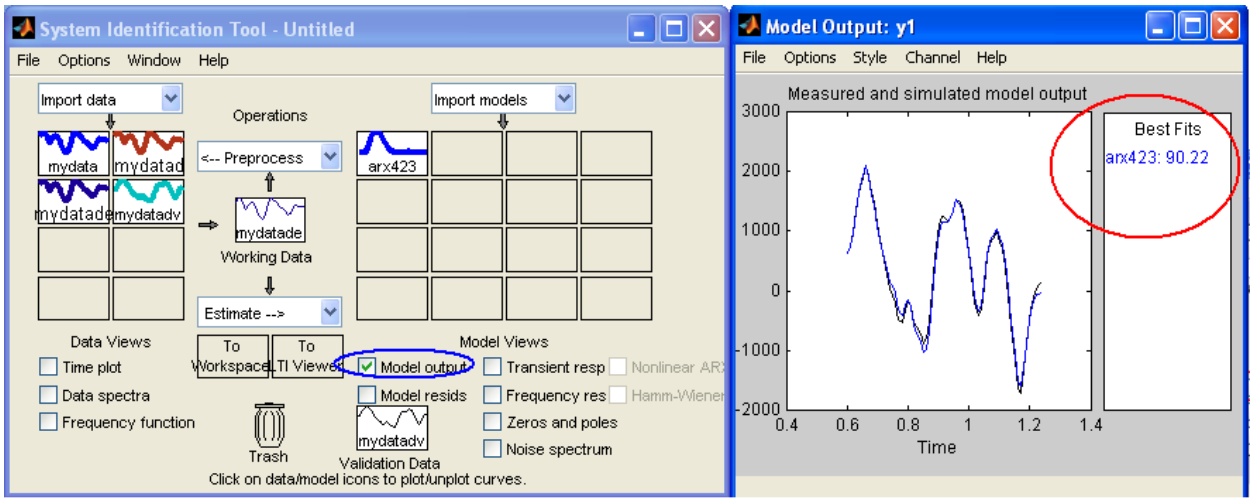
17. Click on “Estimate” button. It estimates and opens another window displaying different results for the identification. Click on “Insert” button.



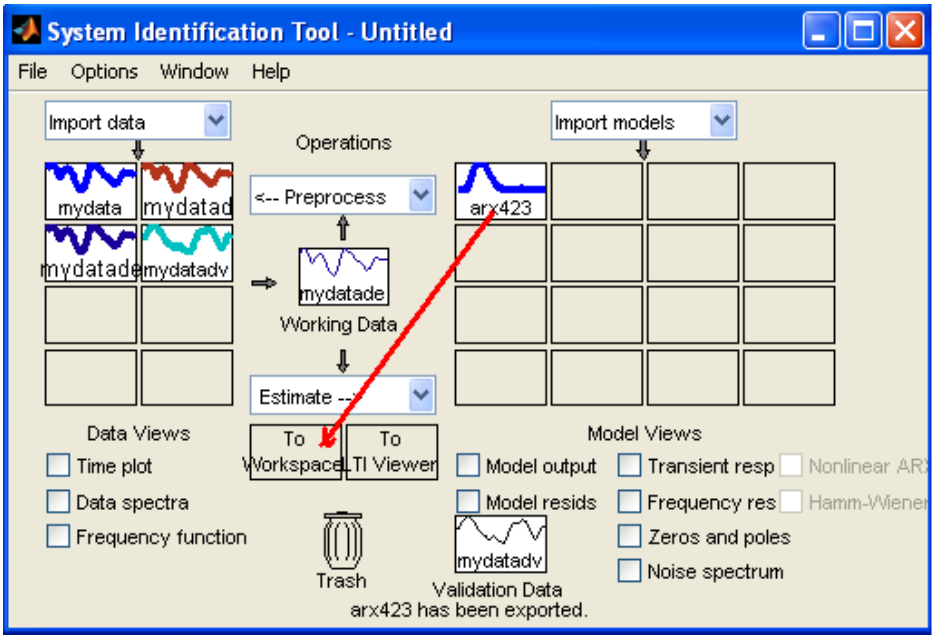
18. This inserts the model to the “System Identification” window. The model is circled in blue.



19. Click on “Model Output” check box circled in blue to display the model output. It opens another window displaying the percentage of fit (this is circled in red).



20. Click on “arx423” in the window, drag and drop it on “To workspace” box. This loads the model to workspace.



21. Enter these commands in Matlab prompt to get the discrete and continuous models of the identified model.
    a. tf_identified = tf(arx423, 'm'); % The arx423 is the model imported to workspace.

    The transfer function is

    0.03571 z + 0.02317

    ---

    z^4 - 1.455 z^3 + 0.2911 z^2 + 0.5251 z - 0.2626

    b. tf_identified = d2c(tf_identified, 'Ts', 0.01); % Find the model in s domain

    The transfer function is

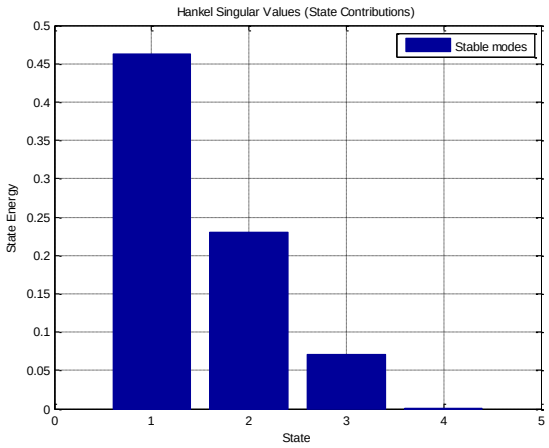
    -0.006401 s^4 - 2.172 s^3 + 2839 s^2 - 6.703e005 s + 4.81e007

    ---

    s^4 + 920.1 s^3 + 7.847e004 s^2 + 4.449e006 s + 8.03e007

    c. Find the energy measure of the system using Hankel Singular value decomposition.
        i. hsvd(tf_identified);

    This plots the energy levels of the system.



Ignore the low energy states. This is mainly done to reduce the identified model based on the energy of the system. From the figure, we can say that the system can be reduced to an order of 3 (since the energy of the system dies out after state 3).

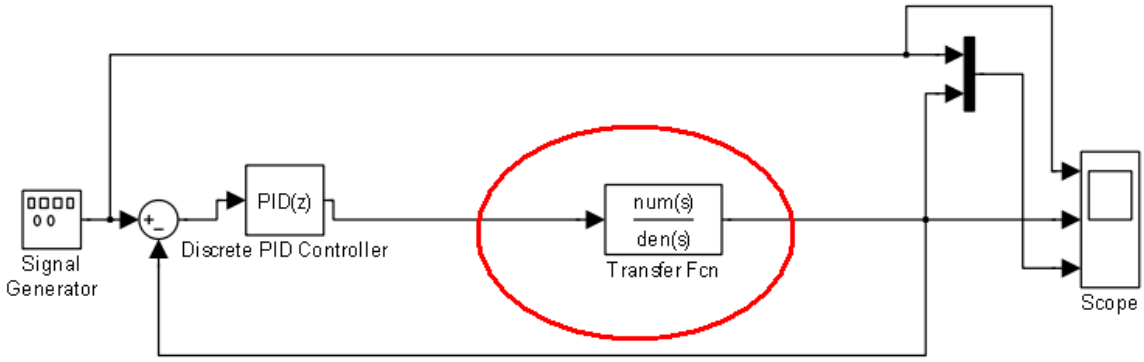
22. Reduce the order of the identified model. Enter the following command in Matlab.

```
tf_reduced = balred(tf_identified, 3);
```

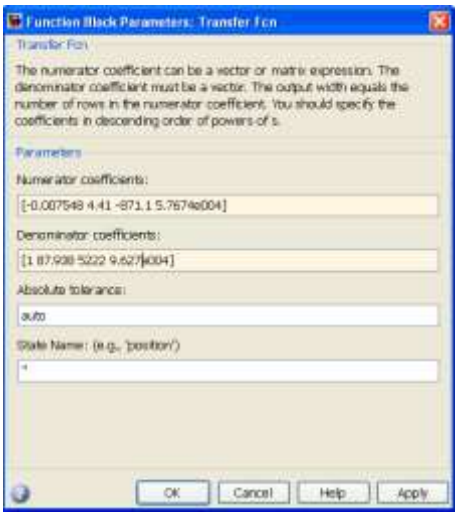
The transfer function is

```
-0.007548 s^3 + 4.41 s^2 - 871.1 s + 5.767e004
-----------------------------------------------
   s^3 + 87.93 s^2 + 5222 s + 9.627e004
```

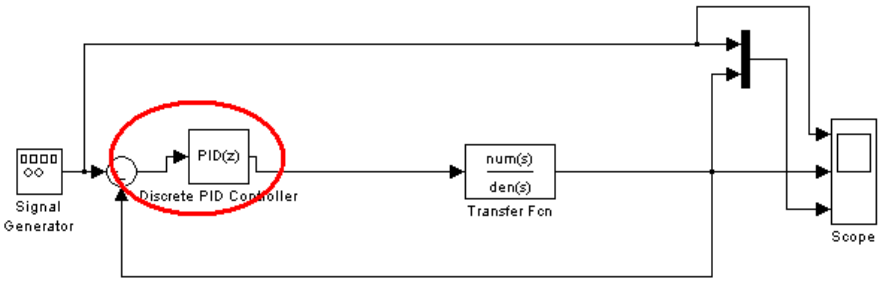
23. Open the model “sysID_dc_motor_model.mdl” to design a controller. Double click on the Transfer function block to input the transfer function **“tf_reduced”**.



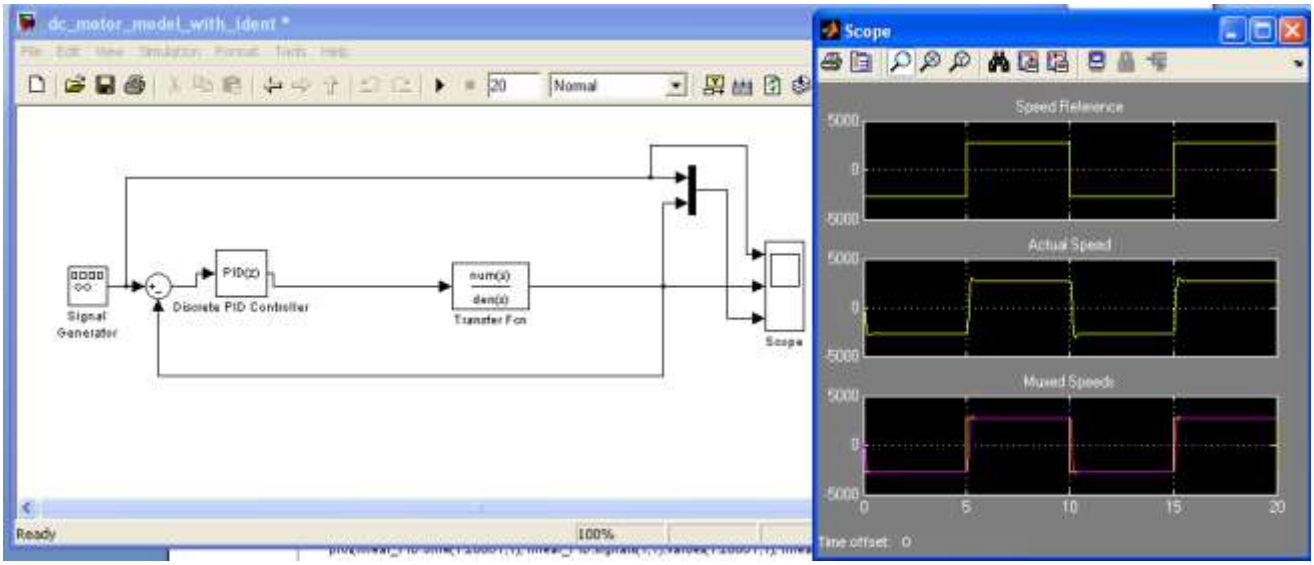
24. This opens a new dialog box. Specify the numerator and denominator coefficients. Click on OK when done.



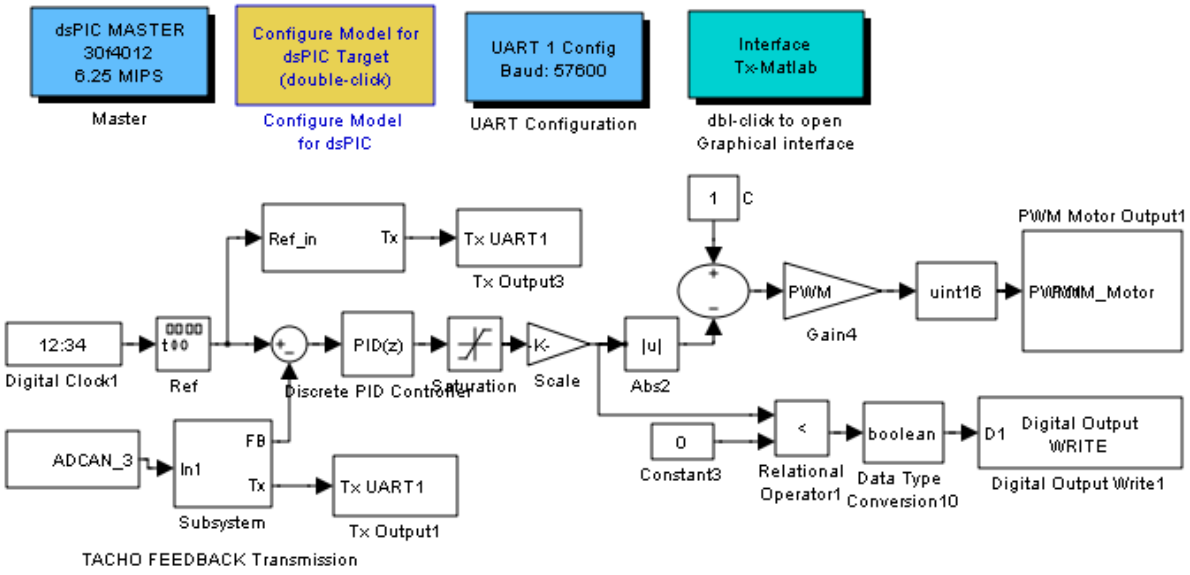
25. Double click on “PID” controller block to auto tune the P, I, D parameters.



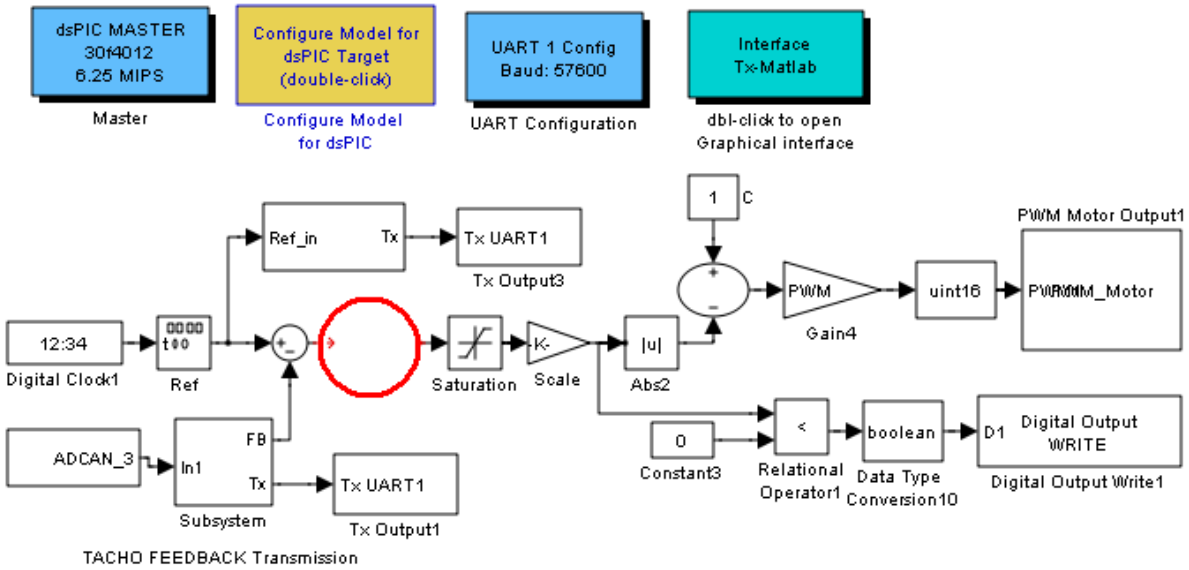
26. Please follow the steps 3 through 8 of section 3.2.4 for PID auto tuning.



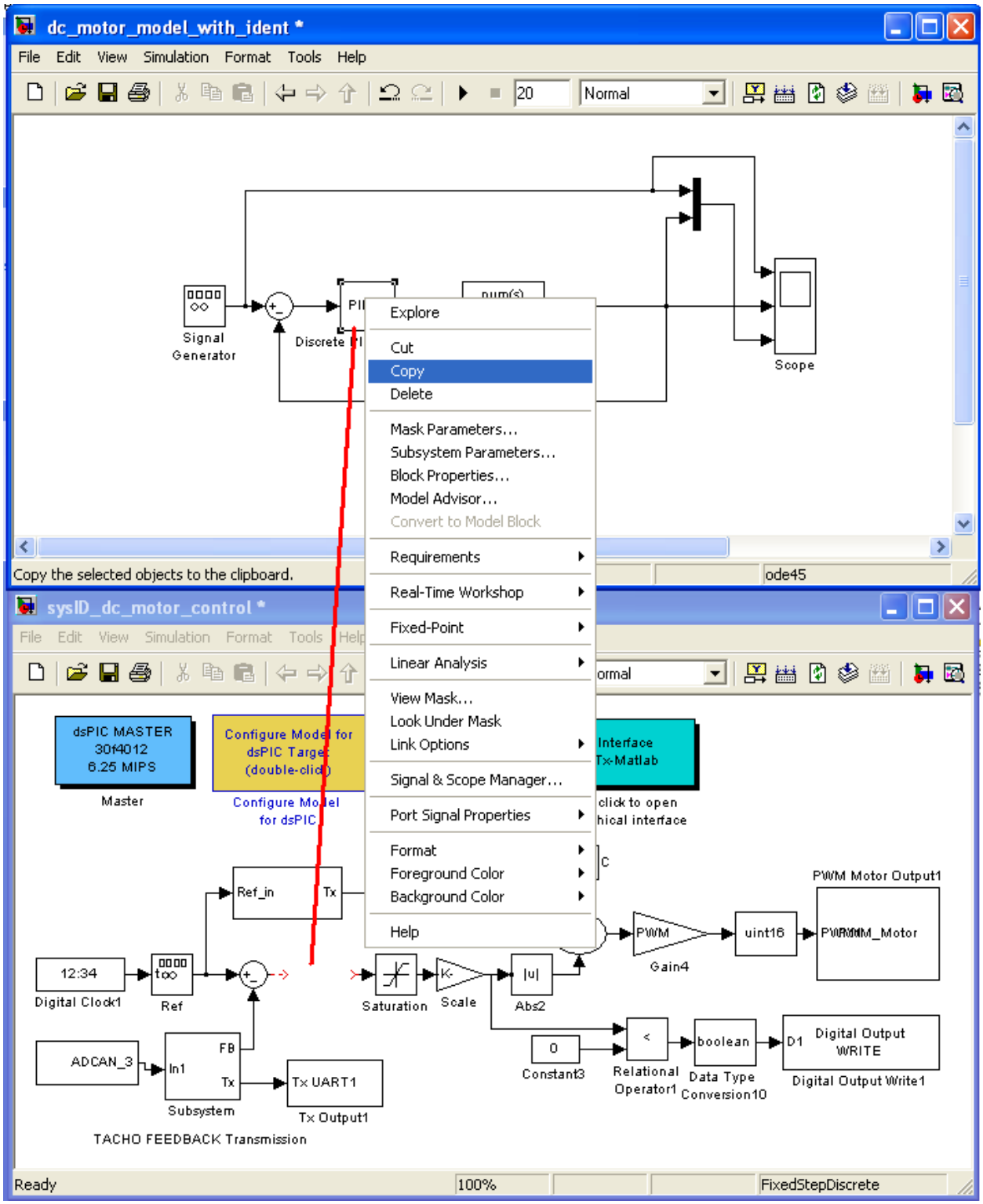
27. Open the model “sysID_dc_motor_control.mdl”.



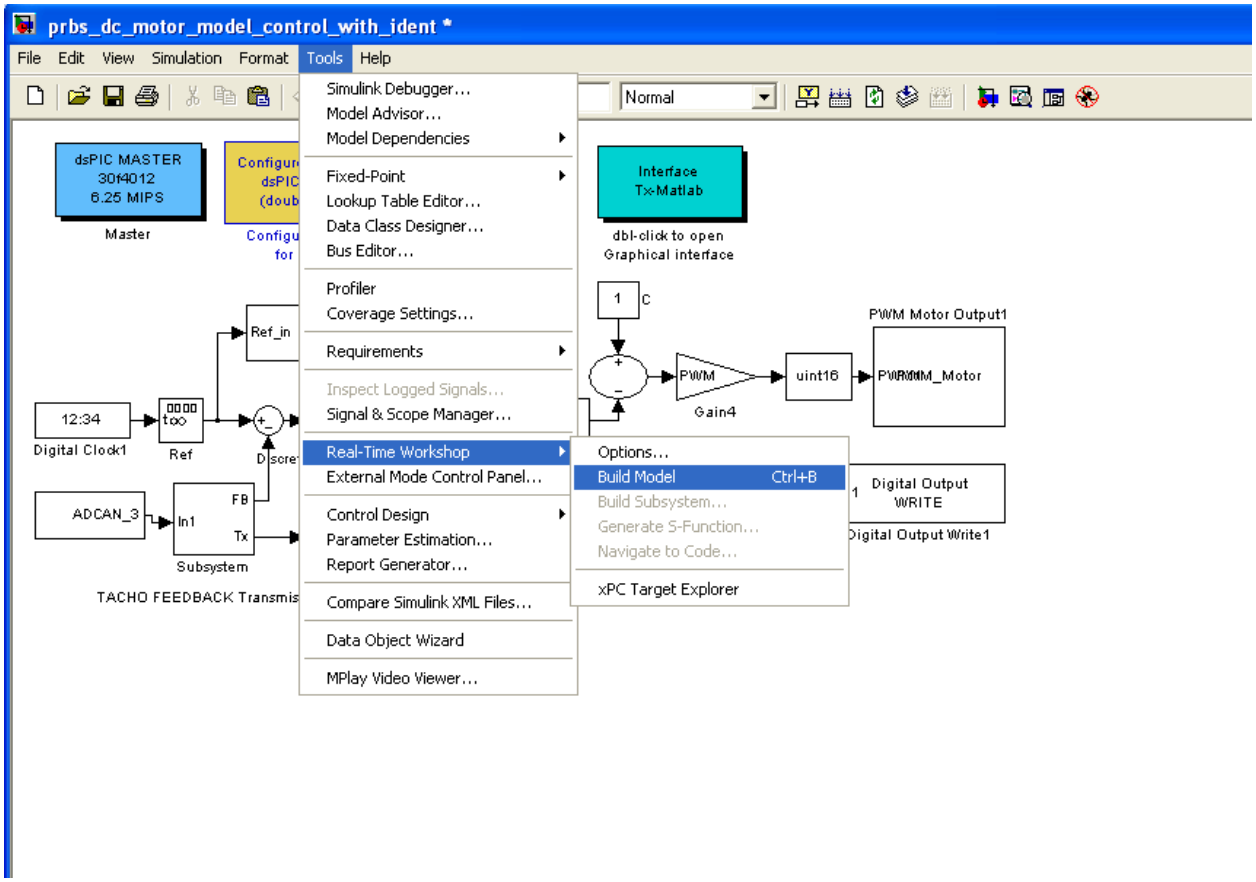
28. Delete the PID controller block from the model.



29. Copy the “PID” block from the “sysID_dc_motor_model.mdl” and paste it in the “sysID_dc_motor_control.mdl”.


dc_motor_model_with_ident *
File Edit View Simulation Format Tools Help
Normal
Signal Generator
Discrete PI
num(s)
Scope
Explore
Cut
Copy
Delete
Mask Parameters...
Subsystem Parameters...
Block Properties...
Model Advisor...
Convert to Model Block
Requirements
Real-Time Workshop
Fixed-Point
Linear Analysis
View Mask...
Look Under Mask
Link Options
Signal & Scope Manager...
Port Signal Properties
Format
Foreground Color
Background Color
Help
Copy the selected objects to the clipboard.
ode45
sysID_dc_motor_control *
dsPIC MASTER
30f4012
6.25 MIPS
Master
Configure Model for dsPIC Target (double-click)
Configure Model for dsPIC
Interface Tx-Matlab
click to open hical interface
Ref_in
Tx
PWM Motor Output1
PWM
uint16
Gain4
12:34
Digital Clock1
Ref
Saturation
Scale
Abs2
ADCAN_3
In1
FB
Tx
Subsystem
Tx UART1
Tx Output1
TACHO FEEDBACK Transmission
0
Constant3
Relational Operator1
boolean
Data Type Conversion10
D1
Digital Output WRITE
Digital Output Write1
Ready
100%
FixedStepDiscrete

30. Build the model to generate the hex file.



31. Repeat steps 16 through 22 of section 3.2.4. Verify if the system response is indeed satisfactory.