Installation and user manual of ECP Plant (Model 210) for use with MATLAB® Release 14

Using

Real-Time Windows Target® (RTWT®)

For Use with MATLAB®, Simulink® and Real-Time Workshop®
A User's Guide

Version 3.0- 2004

1 Buckskin Court Bell Canyon, CA 91307 , USA ph: (818) 703-0802 fax: (818) 703-6040

Table of Contents

1 Buckskin Court Bell Canyon, CA 91307 , USA ph: (818) 703-0802 fax: (818) 703-6040

System Requirements

Hardware Requirements

The hardware required for implementing controllers discussed in this manual includes, (i) a host computer (*i.e.*, a Pentium computer with 128 MB+ main memory), (ii) data acquisition board (*i.e.*, a ECPDSP board), (iii) the ECP electromechanical plant, and (iv) the Model 210 Input/Output Electronics Control Box. The ECPDSP PCI board has the following features:

1. digital-to-analog (Dac) converter channels, +/- 10 volts range (2 Channels)
2. quadrature encoder inputs (4 Channels)

Note that the ECPDSP board is included with all ECP equipment purchased as "Complete Systems."

Software Requirements

The software required for implementing control algorithms using Version 2.5 of the RTWT includes: (i) Windows 2000/XP operating system, (ii) MATLAB Version 7.0, Release 14, (iii) Simulink Version 6.0, (iv) Real-Time Workshop 6.0, (v) RTWT Version 2.5, and (vi) Microsoft Visual C++ Professional Edition Version 6.0 (vii) the ECP Extension Files

About this Manual

This manual assumes that the user has correctly installed the Windows operating system, MATLAB, the MATLAB products, and the MS Visual C/C++ Professional compiler. It is also assumed that the standard ECP hardware is working correctly, and that the user has performed the experiments given in the ECP manual using the appropriate ECP Executive program (instructions and troubleshooting information about the ECP hardware and software systems are available in the ECP manuals). Furthermore, this manual assumes that the user is familiar with Simulink and Real-Time Workshop, and has previously constructed, built and executed real-time models in the RTWT environment (detailed procedures are outlined in the RTWT User's Manual).

Safety Instructions

Section 2.3 in the ECP Manual contains vital information about safety issues associated with the system.

All users must read and understand the safety guidelines in Section 2.3 of the ECP Manual prior to operating the system.

If any material is unclear, the user must contact ECP for clarification before operating the system.

In the event of an emergency, the control effort should be immediately discontinued, by pressing the red "OFF" button on the front of the control box.

Installation

Installation with a CD

1. Start MATLAB
2. Insert the CD into the CD ROM
3. In the MATLAB command window, go to the drive on which the CDROM is installed for example **cd D:**
4. Type **installExt** at the MATLAB command prompt

Installation without a CD

1. Copy all the files from the CDROM and put them into a local directory on the machine for example in directory, **C:\temp**
2. Start MATLAB
3. Go to the directory that you copied the files into from the MATLAB command prompt,
   cd C:\temp
4. Type **installExt** at the MATLAB command prompt

After the installation has been completed as described above, the following files should now be present on your computer.

- The following file should now be included in the user's `C:\ECP\Model210\` directory:
  `ECPDSP.mdl ECPDSPDriver.c, ECPDSPDriver.dll, ECPDSPDriver.h , Model210Default.mdl,`
- The following path will be added to the MATLAB Path Browser window.
  `C:\ECP\Model210\`

Uninstall

To uninstall all of the files created using this package

1. Start MATLAB
2. Insert the CD into the CD ROM
3. In the MATLAB command window, go to the drive on which the CDROM is installed for example **cd D:**
4. Type **uninstallExt** at the MATLAB command prompt

Implementation

Prior to implementation of any control algorithm through Simulink/Real-Time Workshop/ RTWT, you must "**Down Load**" the correct "**Controller Personality File"** via the **ECP Controls Executive** program, or **ECPUSR Executive** program. Either or both of these programs should have been install already on your computer. For the Model 210 the correct Personality File is **M210_rtwt_x.PMC** (x is the version number). This **Personality file** would be found in the directory where the ECP Executive is installed. To download the **M210_rtwt_x.PMC** file, enter the **ECP Executive** program, select the **Utility** Menu, and then select **Download Controller Personality**.

Note: To revert back to the use of ECPDSP Board for standard **ECP Executive** program(s) at a later time, you must first use the same **Utility** Menu option and download the file **M210Vxx.PMC** (xx is the version number).

Once the **M210_rtwt_x.PMC** file is downloaded **exit** the **ECP Executive** program. **Do not** have the **ECP Executive** program open with running RTWT experiments.

To load and execute the RTWT block diagram, the user follows the procedure outlined below:

Open MATLAB and change directories to

```
>> cd c:\ECP\Model210
```

To load the RTWT block diagram in the MATLAB environment, the user types

```
>> Model210Default
```

Figure 1 is the Model210default.mdl simulink model:

Warning: **Do not run the model until you have successfully completed ECPDSP Driver setup, and Running the model sections.**

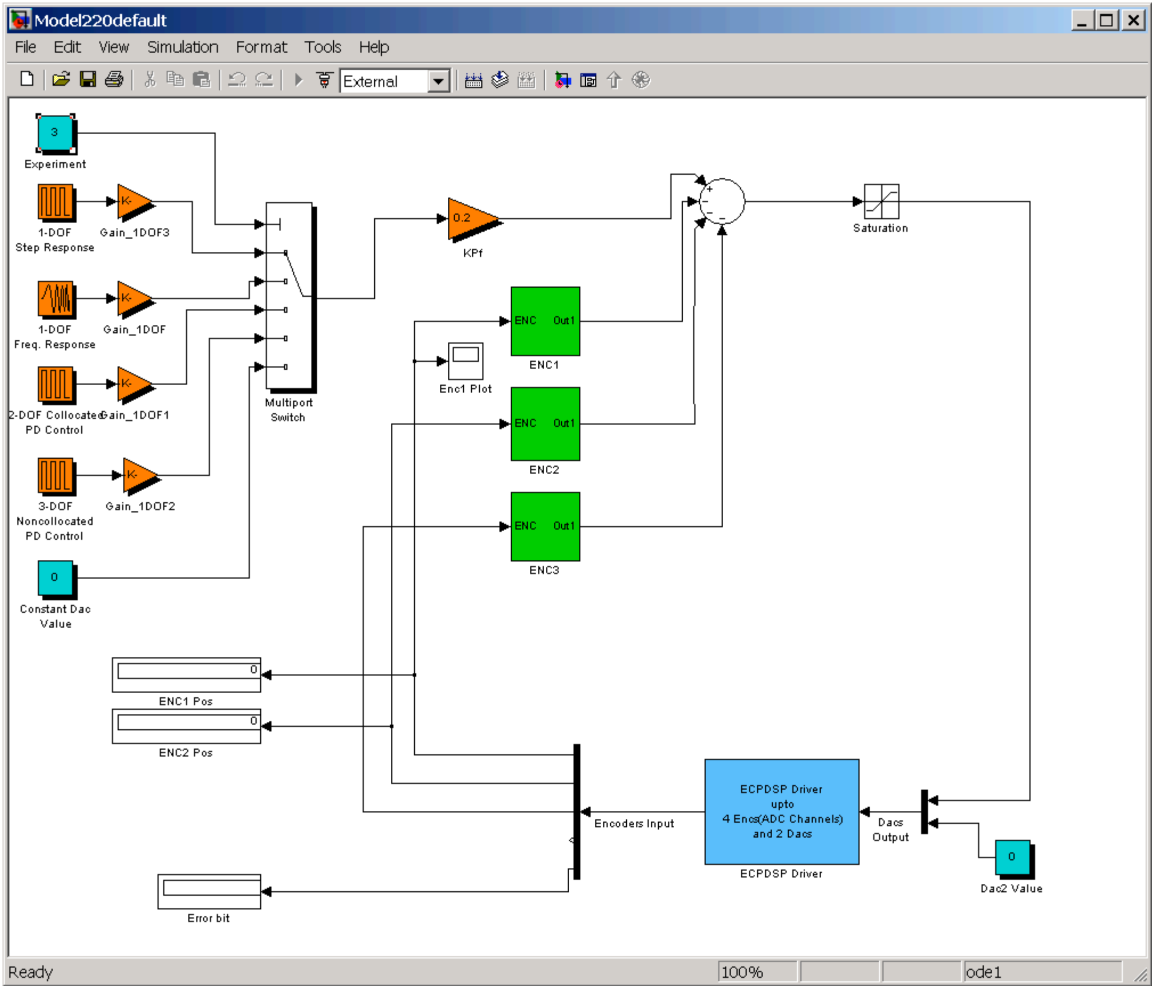


Figure 1.

ECPDSP Driver Setup

In Figure 2, the blue block labeled, “ECPDSP Driver” allows the users to communicate with an ECPDSP board. ECPDSP Driver block accepts 4 encoder inputs (ADC Channels) and 2 DAC outputs and it allows the users to set the **Base I/O Address,** and the **sample-time**. It also creates a **Limit Condition** flag, and reports back as a 1 or 0 value.

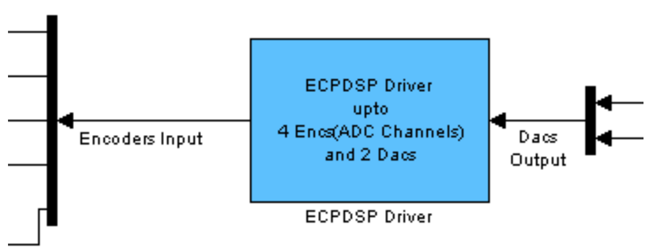


Figure 2.

To set the parameters of the ECPDSP Driver Module, the user double clicks on the "ECPDSP Driver" block shown in Figure 2. The user then enters the **Base I/O Address** and the **sample Time**. For example, in the following example (Figure 3), the **Base I/O Address** is set to '0xD000', and the **Sample Time** is set to 0.008, and **Access Hardware** is enabled as shown in Figure 3. If the **Access Hardware** checkbox is not checked, the encoder information will not be retrieved during program execution. Unless you are using older versions of ECP DSP Boards, leave the **Timeout Period** at it default value of 500.

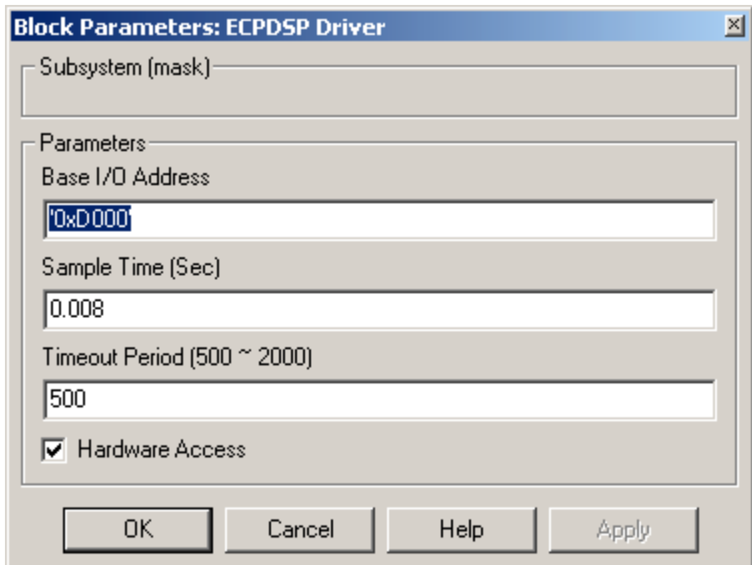


Figure 3

How to get the Base I/O Address:

To get the address the following procedure must be carried out initially:

1. Go to the control panel
2. Double click on the system
3. System Properties window will open up. Click on the hardware tab and then click on the device manager button.
4. The device manager window will open up Figure 4. In the device manager window, find the motion controllers option and expand the tree. You would see "PMAC PCI Motion Controller Card" device. Double click on it.
5. In the "PMAC PCI Motion Controller Card properties" window, select the resource tab. From the resource settings, the beginning value for the input/output range is the Base I/O Address Figure 5. In figure 5, the base I/O address is 0xC800 (notice that the value is always a hex number). The parameter must always be entered as a single quotation text string. For example: '0xC800'.

Warning: Since windows manages the I/O port assignments, if the user changes the ECPDSP PCI slot location, or adds another PCI device, then

the user has to recheck the Base I/O address to make sure that the address haven't changed. In addition, if the user executes the program with a wrong base I/O address, the computer will freeze up.

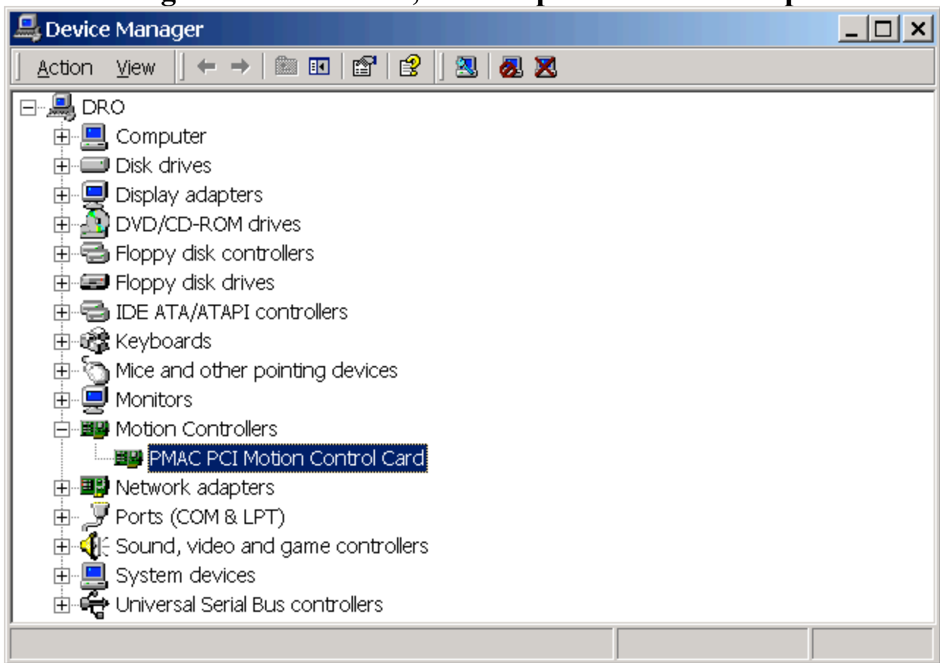


Figure 4

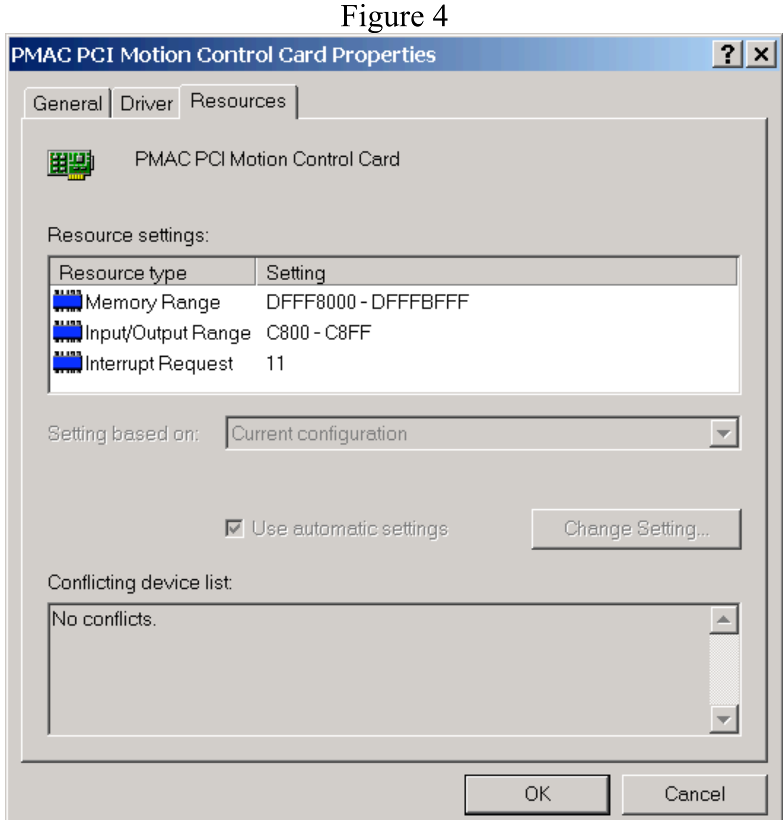


Figure 5

Input/Output signals of the ECPDSP Driver Block:

Input and output signals of the ECPDSP Driver block are vectors. So the user needs "Mux" multiplexer to combine single inputs (Dac Outputs) into one vector and "Demux" demultiplexers to extract Output lines (Encode Inputs) from the output vector.

- **Input Signal (Dacs Output):**
  ECPDSP Driver block Input is a 2 dimensional array. It accepts two Dac output values. The user should use a "Mux" to combine two individual dac values into one vector figure 6. In a "Mux" the top most signal is the Dac1 and the second signal is the Dac2. It is a good habit to assign Zero value to the Dac that we are not going to use.

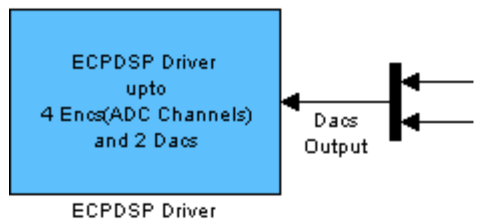


Figure 6

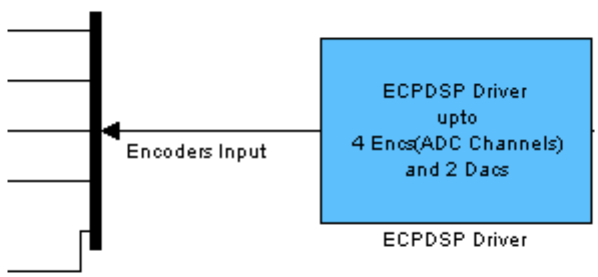
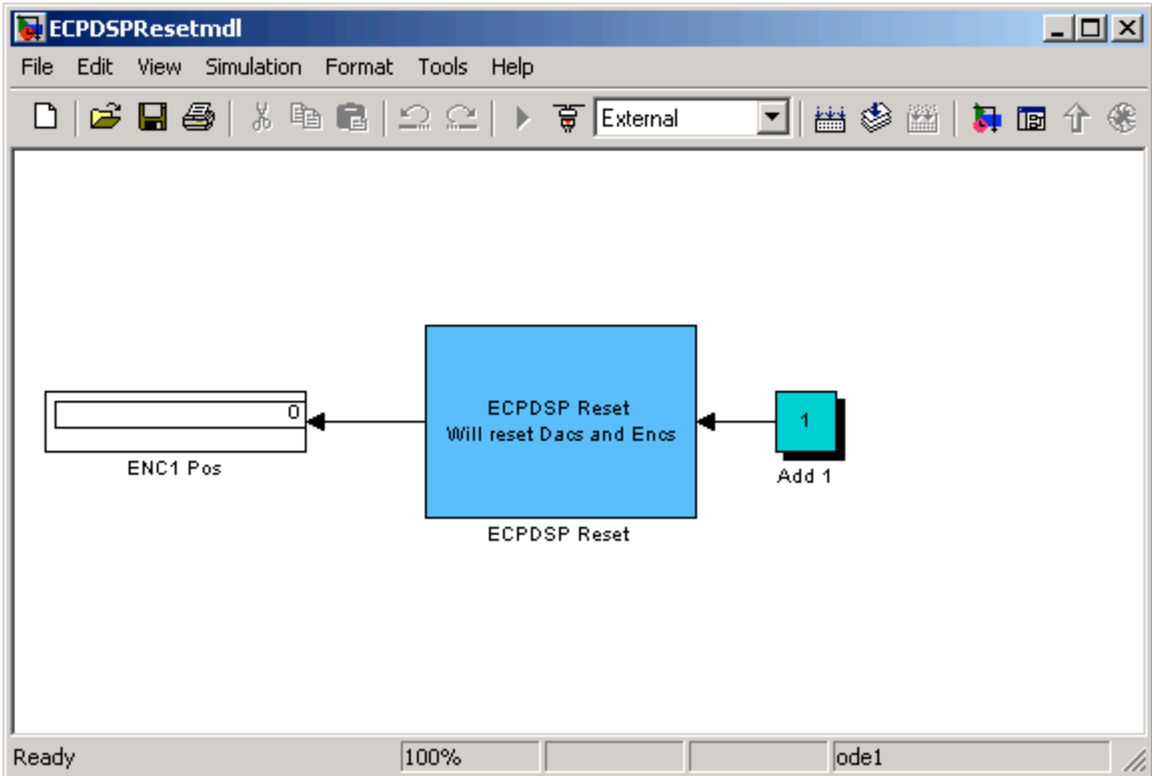


Figure 7

- **Output Signal (Encoders Input and Limit Condition Flag)**
  ECPDSP Driver block output is a 5 dimensional array. It accepts 4 Encoders Encoder or ADC input values. The user should use a "Demux" to extract 4 individual Encoder or ADC values and also one **Limit Condition** flag (figure 7). In a "Demux" the top most signal is the Enc1, the second signal is the Enc2, the third signal is the Enc3, the fourth signal is the Enc4, and the fifth signal is the **Limit Condition** Flag. This flag is for monitoring the safety features. If **the Limit Condition** flag value is set to one by the DSP firmware, it means that one or more safety limit violations have occurred in the run process. If the value is zero, it means that there are no limit violations. Once the flag is set, the DSP Board must be reset in order to run subsequent Simulink models. To reset (clear) the limit condition flag there are two alternatives. **Alternative 1:** the user may clear **the Limit Condition** Flag (set it to zero) by entering the **ECP Executive** program, then select the **Utility** Menu and click on **Reset.** Next, the user must **exit** the **ECP Executive** program in order to resume activities in the

Matlab/Simulink/RTWT environment. **Alternative 2:** The user may run the **ECPDSPResetmdl** model just once. This Simulink model (shown below) is included in the installation and should be setup just once for the correct **Base I/O Address** prior to the original run.



Running the model:

Before the user initiates the subsequent automatic build process, it is necessary to ensure that the build options are set correctly. The user must have installed RTWT on the system. To install RTWT, go to the Matlab prompt and type "rtwintgt –setup", this will install the RTWT. In addition, the user must have selected the VC++ compiler as the default compiler. To set the compiler, in the Matlab Prompt type "mex –setup" and follow the instructions. Now the user must set the environment variables for the simulink model. To this end, select **Tools** from the Simulink model menu bar and then select **Real-Time Workshop** followed by **Options** from the pull-down menu. From the **Configuration Parameters** window, select the **Real-Time Workshop** option as shown in Figure 8.

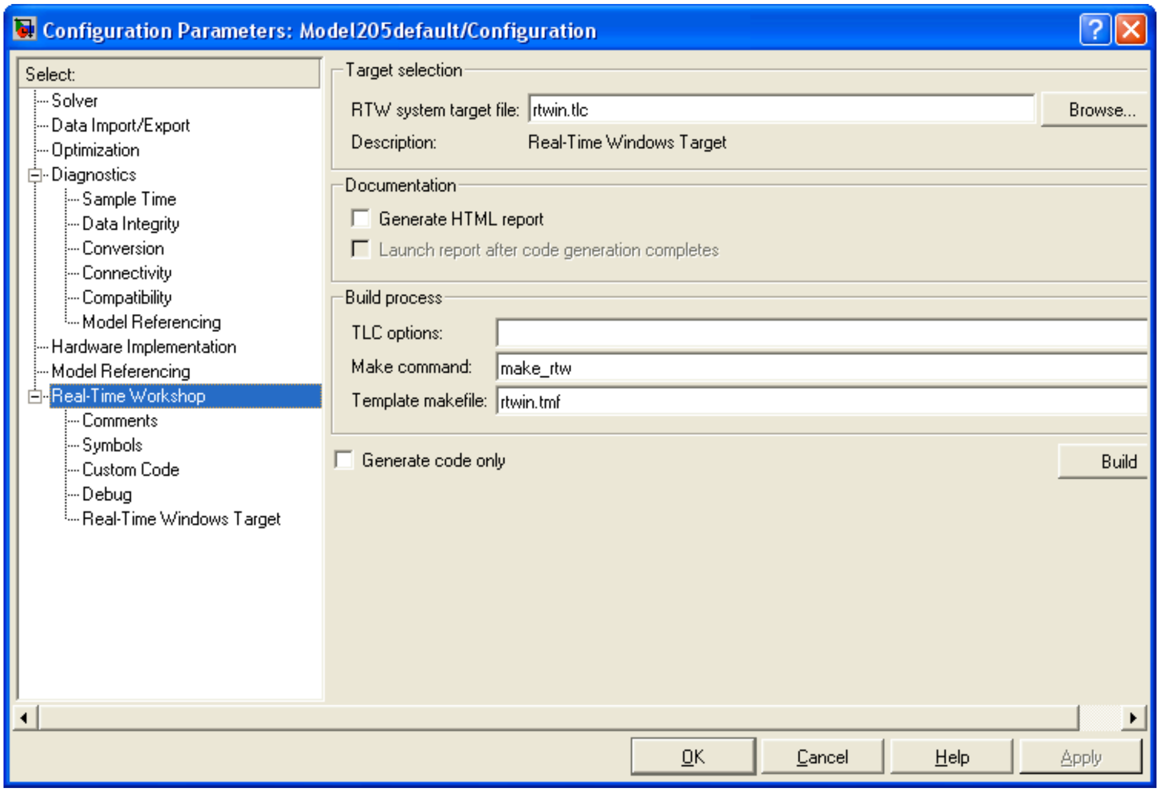


Figure 8

If the text in the **RTW System target file**, **Template makefile**, and **Make command** edit boxes is not the same as shown in Figure 8, the user must click on the **Browse** button. From the resulting **System Target File Browser** window, the user should select **Real-Time Windows Target rtwin.tlc** as shown in Figure 9 and click on OK button.

Next, select the **Real-Time Windows target** Figure 10. Make sure that the “External Mode” and “Compiler Optimization” check boxes are set. It is not necessary, but user can also check the build all option. If the user changes any value then press on **apply** button.

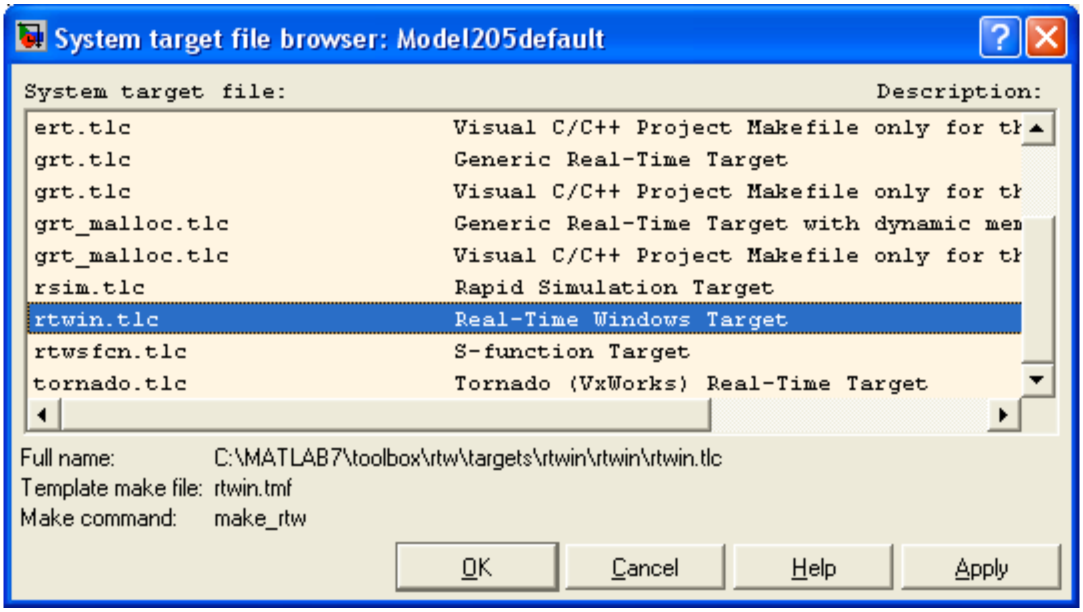


Figure 9

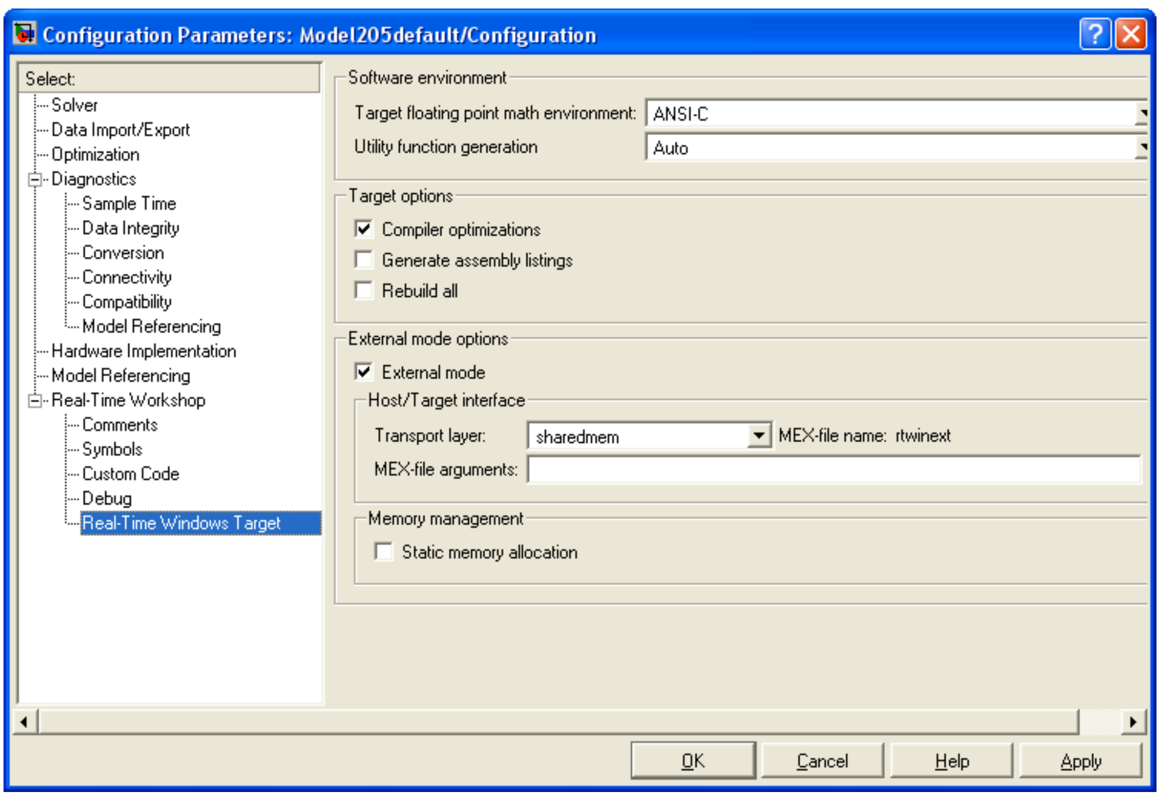


Figure 10

Finally select the **Solver** option from configuration parameters window. Make sure that the Solver options type is set to "Fixed-Step" and to "Ode1 (Euler)" figure 11. Press ok to exit this window.

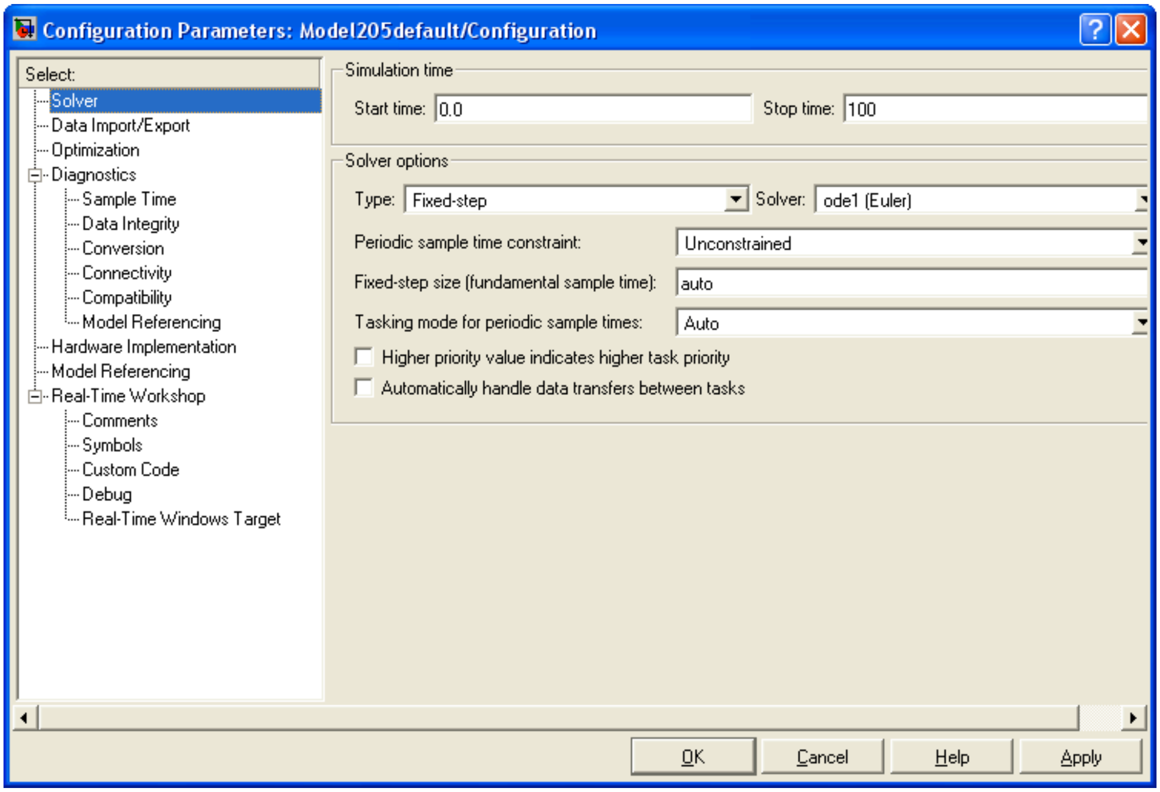


Figure 11

If the **Real-Time Workshop / Options** are set correctly the user should initiate the automatic build process. The automatic build process is initiated after selecting **Tools** from the Simulink model menu bar and then selecting **Real-Time Workshop / Build Model** from the pull-down menu. Before the control can be executed, the user must perform the above build operation for the first time.

After selecting **Real-Time Workshop / Build**, the following files will be automatically generated, and saved in the directory `c:\ECP\Model210\Model210default_rtwin`
`Model210Default.`*bat,* `Model210Default.`*c,* `Model210Default.`*dt,* `Model210Default.`*h,* `Model210Default.`*mk,* `Model210Default.`*obj,* `Model210Default.`*prm,* `Model210Default.`*reg,* and `Model210Default.`*rwd.*

To execute the real-time target in the External Mode, the user connects to the real-time target by selecting **Tools** from the Simulink model menu bar and then selecting **External Mode Control Panel** from the pull-down menu. In the **External Model Control Panel**, the user clicks on the **Connect** button (see Figure 12).

To begin execution, the user then clicks on the **Start real-time code** button (see Figure 13).

To stop execution of the real-time target, the user can click either the **Stop real-time code** button or the **Disconnect** button (see Figure 14).

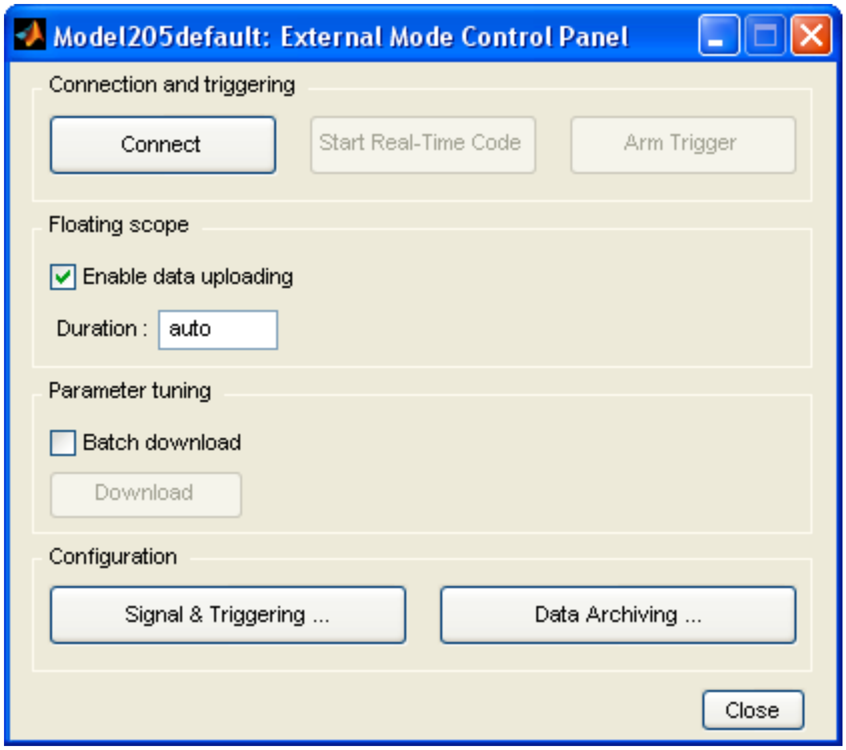


Figure 12

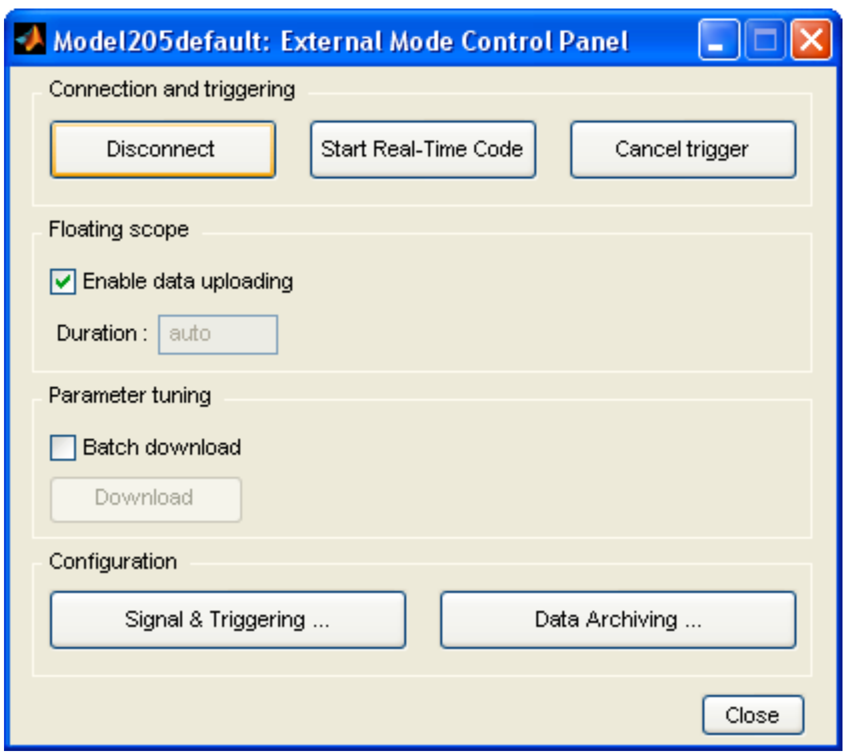


Figure 13

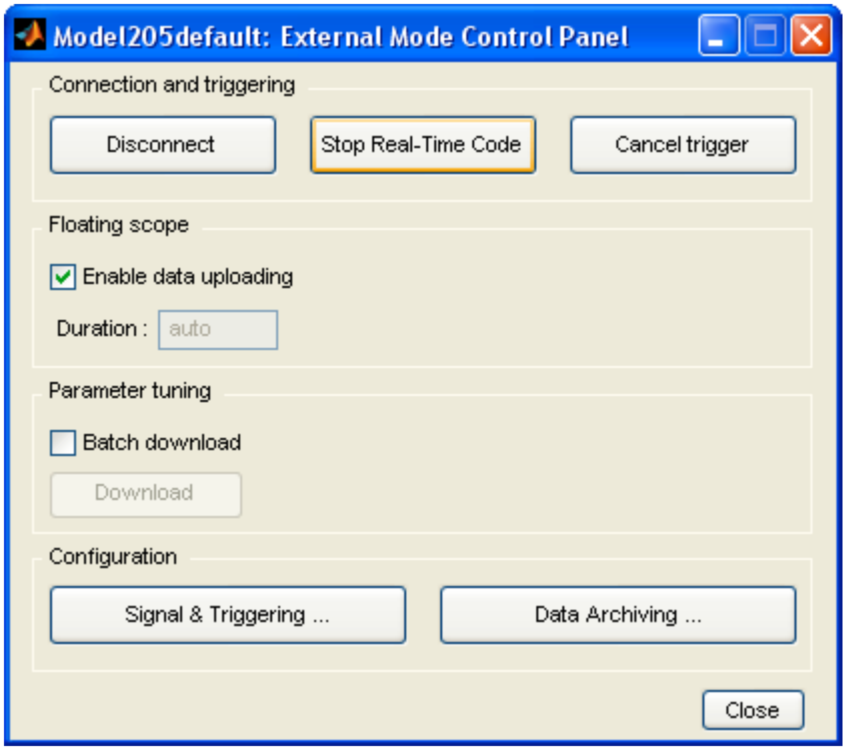


Figure 14.

Control Experiment

The following experiment was performed on the ECP Model 210 Control System using an ECPDSP board and the Real Time Windows Target software.

User Notes

The mechanism configuration details are provided in the "Self-Guide Demo" section of the chapter 3 of the main Model 210 Manual. Essentially, the Model 210 mechanism must have the same mechanical configuration requested for the "Self-Guide Demo" section (all three carriages connected, with 2 weights in each carriage, 1 soft spring between the base and the first carriage, 2 hard springs between carriage 1 to 2 and carriage 2 to 3).

Important Note: For dynamic and static (constant gain) behavior differences between running the same control algorithm in the Matlab/Simulink/RTWT environment relative to the **ECP Executive** program environment, please refer to the **Hidden Behavior** section of this Manual.

Experiment : Self Guided Demo

This experiment uses the same control structure and gains as in the "Self-Guide Demo" section of the Model 210 Manual.

In **Experiment** (cyan) block in the **Model210Default** block diagram, the user types 3 in the **Constant Value** edit box (see Figure 15)

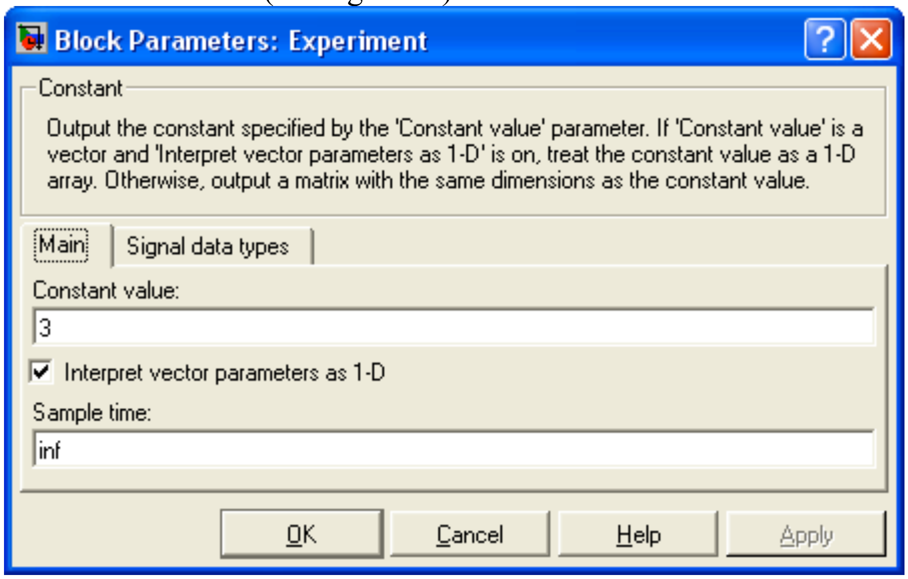


Figure 15

The reference (desired) step position is set by double clicking on the **2-DOF collocated PD control** block as follows figure16:

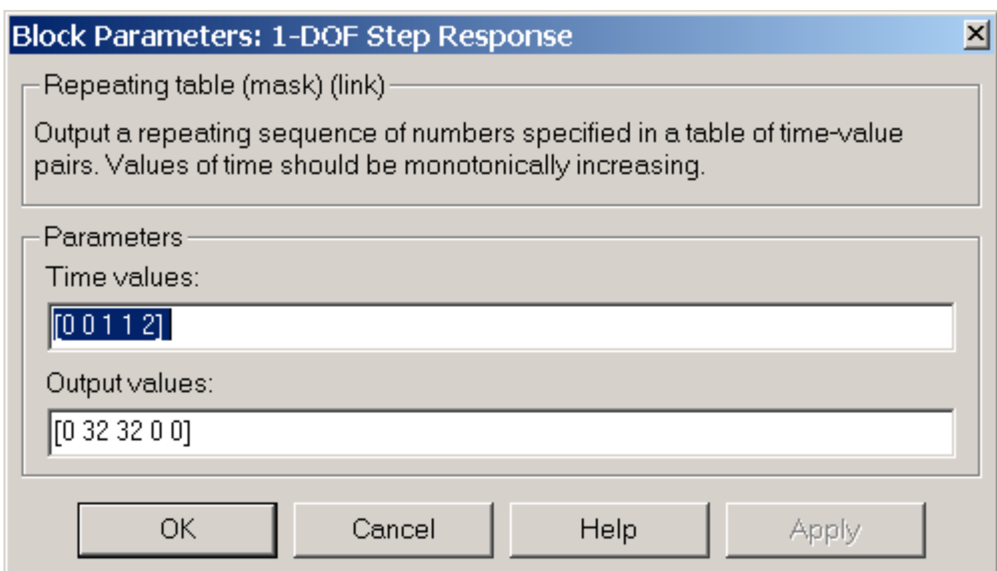


Figure 16

The user also needs to set the gain value for the 2-DOF experiment. Double click on the **Gain_1DOF1** block and set the value to 1000 figure 17.

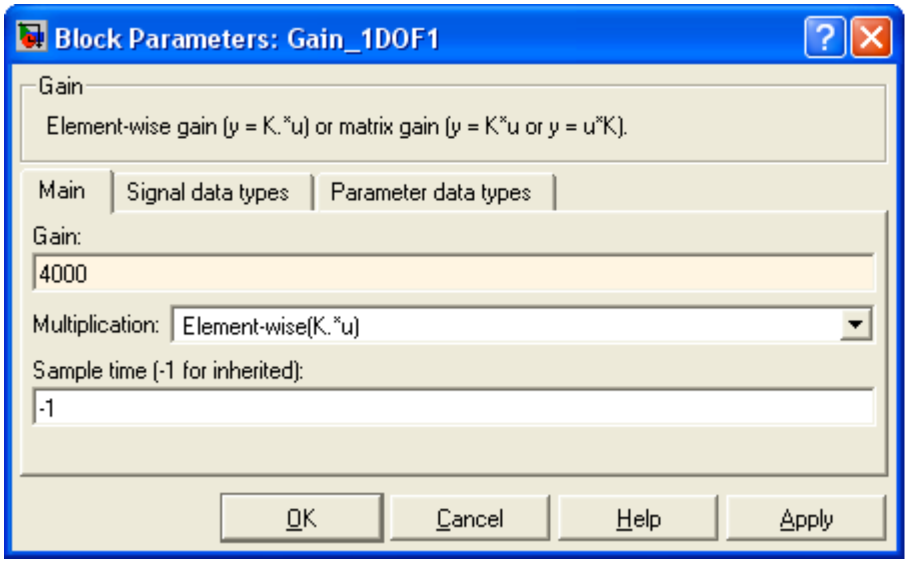


Figure 17

In addition, the users need to check and, if not already set, set the gain values for **Kpf** and **ENC1, ENC2 and ENC3**. To set the **Kpf** value, double click on the block and make sure its value to is set 0.937. Other gains should be: **ENC1** gains: 0.661, 0.0138 (proportional and derivative), **ENC2** gains 0.064, 0.031 (proportional and derivative), **ENC3** gains 0.213, 0.0282 (proportional and derivative). Figures 18 and 19 show the block formats.

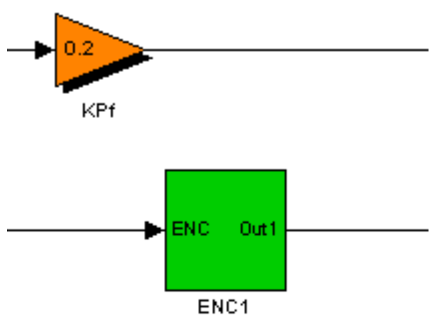


Figure 18

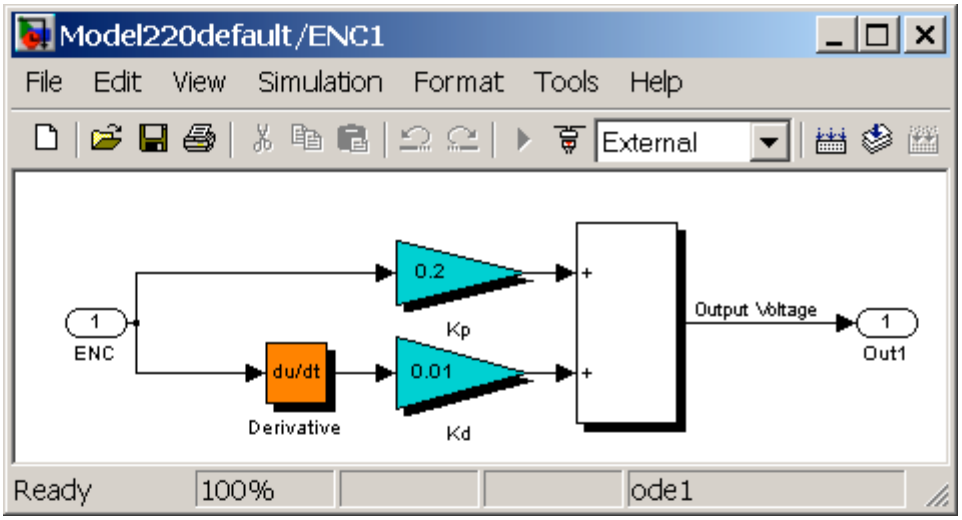


Figure 19

Figure 20 shows the step response associated with the above set of gains

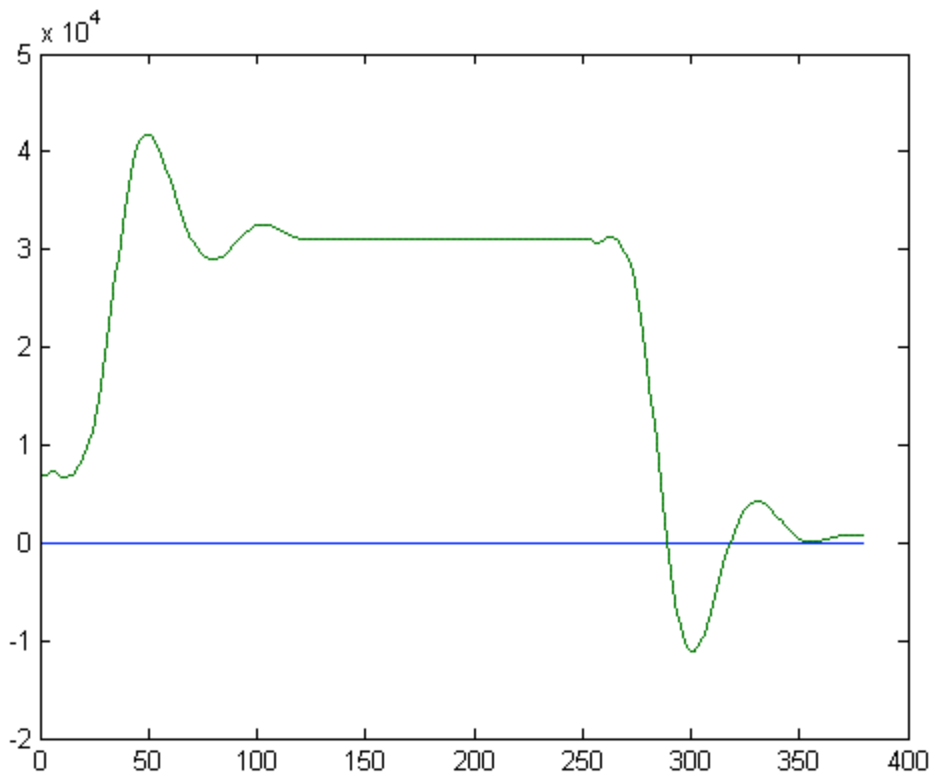


Figure 20

ECPDSP Reset

If **the Limit Condition** flag value is set to one by the DSP firmware, it means that one or more safety limit violations have occurred in the run process. If the value is zero, it means that there are no limit violations. To reset the limit condition flag the user has two choices. **Alternative1:** The user must enter the **ECP Executive**

program and **Reset** the ECPDSP board. The clear **the Limit Condition** Flag (set it back to zero), the user enter the **ECP Executive** program, select the **Utility** Menu, and click on Reset. Next, the user must **exit** the **ECP Executive** program in order to resume activities in the Matlab/Simulink/RTWT environment.
Alternative 2: The user may run the **ECPDSPResetmdl** model just once. This Simulink model (shown below in Figure 21) is included in the installation and should be setup just once for the correct **Base I/O Address** prior to its first time run.

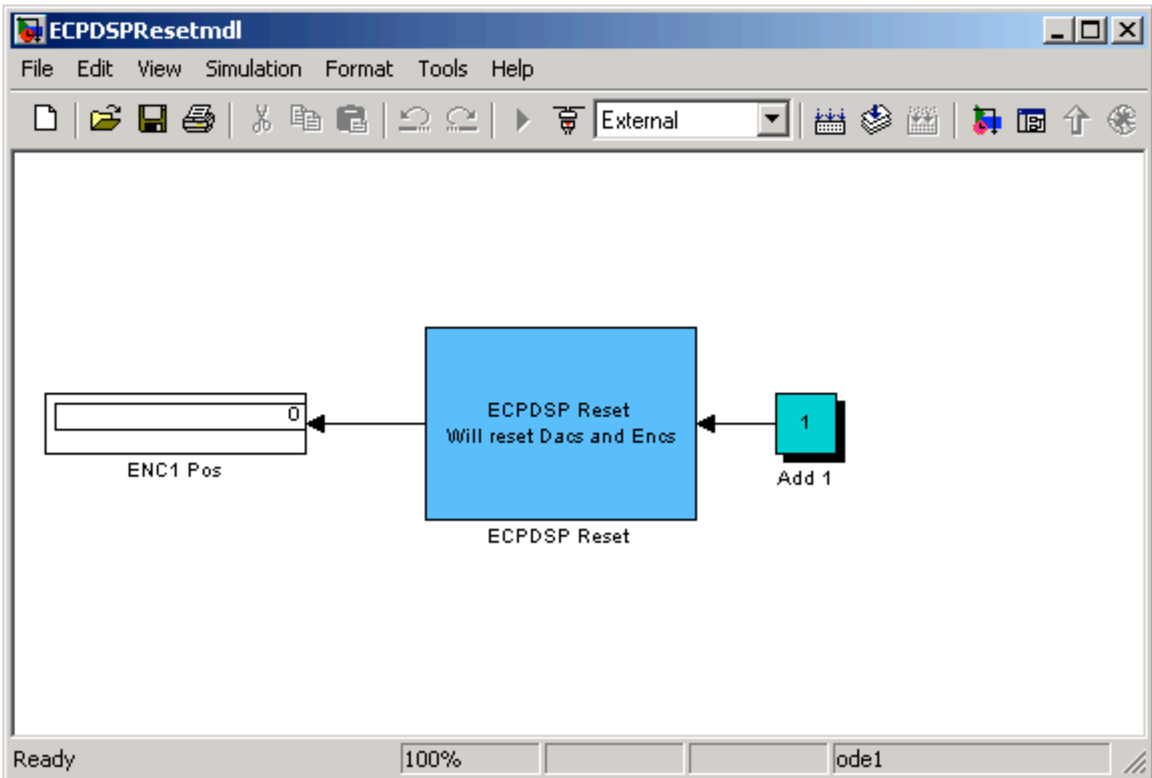


Figure 21

Important Note: Any time that you are running the RTWT under Matlab/Simulink environment you must make sure that the **ECP Executive** is positively exited (the background screen is not just minimized).

Hidden Behavior

Encoder Feedback Gain Factor

The ECPDSP Board encoder counting logic provides for advance sub-count interpolation in order to enhance the resolution of the feedback signals. Five bits of fractional counts are measured by high speed timers and then added with a 5-bit shift to the Encoder Counters. As a result the position feedback for ALL encoders have a 32 (5-bit) multiplication factor inherent in their value (e.g.1000 counts is multiplied by 32 giving the value of 32000 in the counter.) This means that in any control design procedure a gain of 32 must be included in series to the encoder resolution scale factor (gain). This behavior also exists in the ECP Executive program for the models 205, 210, 220, 505, 750 and the Accessory A51.

Dac Output Gain Factor

In the Matlab/Simulink/RTWT environment, the ECPDSP Board uses 16-bit Dacs where 32767 Dac output is the equivalent of maximum *positive* torque (force) command and –32768 is the is equivalent of maximum *negative* torque (force) command.
In the **ECP Executive** program environment, the firmware provides a **2x** multiplier for the Models 205, 210 and 220. As a result, the same control gains within the same control structure have an overall gain **twice** as much as in the Matlab/Simulink/RTWT environment. This **2x** factor explains most of **differences** between the responses of the mechanism (when using the same gains and the same control structure) in the Matlab/Simulink/RTWT environment relative to the **ECP Executive** program environment.

Unit Sample Time Delay

Due to the nature of the operation of the RTWT it essentially always suffers from a **unit sample time delay (1/z)**. This is because it updates the Dacs (control efforts) at the same time it reads the feedback. However, the **ECP Executive** program's firmware first reads the feedback signals and then, rapidly computes the control law, and updates the Dacs within the same sample period. As a result even though there is a short "aperture" time, or computation time, there is no **unit sample time delay (1/z)** in the **ECP Executive** program's firmware. This delay explains most of **differences** between the responses of the mechanism (when using the same gains and the same control structure) in the Matlab/Simulink/RTWT environment relative to the **ECP Executive** program environment at larger sampling periods (greater than 2 ms).