

ECP MANUAL FOR GYROSCOPE CONTROL SYSTEM

Prof. R.A. de Callafon, Dept. of MAE, UCSD, version 3.14159

ECP HARDWARE & SOFTWARE

Turning on Hardware

Turn on the ECP (model 750) control box by pushing the **BLACK ON** button. Make sure the **AXIS 3 AND AXIS 4 BRAKE ARE OFF** (switches on ECP box to top position) and the gyroscope is in its default initial position: put **ALL GIMBALS AT 90 DEGREE ANGLES**. The ECP control box also has a **RED OFF** button. Use the **RED OFF** button to turn of the hardware in case of emergency.

In Case of Emergency

In case you experience a wildly moving gyroscope system or when the system is vibrating strongly, make sure to push the **RED OFF** button on the front of the ECP control box. Turning of the ECP control box avoids damage to the experiment in case you specified the wrong control algorithm (unstable) and or reference signals that are too large. At all times, keep away from any moving parts. Make sure to push the **RED OFF** button also when someone is too close (less than 4 inches) to the experiment.

Start ECP software

Log in with your username and password and start the ECP software via the ECP icon on the desktop. Once ECP starts up, you will see a window similar to Figure 1.

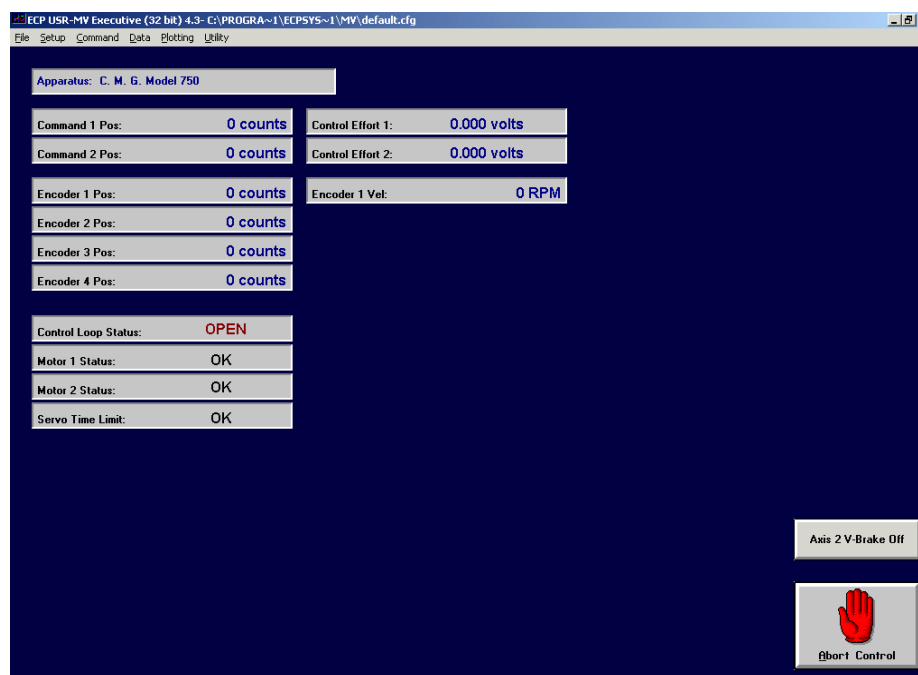


Figure 1: ECP main window

Open-loop versus Closed-loop experiments

Open-loop experiments (no feedback controller implemented) are used to study the dynamics of the uncontrolled gyroscope system. Closed-loop experiments require the *design of a feedback control algorithm* that continuously monitors the encoder positions and computes control signals for (one of the) Servo Motors to control the forces applied to the gyroscope system. In summary:

- Open-loop experiments consist of applying *input signals to (one of) the Servo Motors and measuring the outputs* (encoder position of the different rigid bodies) of the gyroscope via the encoders. These experiments are done typically during the *modeling phases of your lab*.
- Closed-loop experiments consist of applying *reference signals to (one of) the Servo Motor control loops and measuring the outputs* (encoder position of the different rigid bodies) of the gyroscope. These experiments are done typically during the *control implementation phases of your lab*.

OPEN-LOOP EXPERIMENTS (for modeling of the gyroscope)

An open-loop experiment for the ECP gyroscope system requires the implementation of a so-called *open-loop control algorithm* and the *specification of an open-loop trajectory*. To perform an open-loop experiment, follow these steps:

Implementation of open-loop control algorithm

1. First turn off any closed-loop control algorithm by clicking the large **Abort Control** button in the ECP Main Window (see Figure 1).
2. Select **Setup - Control Algorithm** from the main menu bar to implement an *open-loop control algorithm*.
3. When performing an **open-loop experiment using Servo Motor 1**:
Select **Load from Disk** and go to your directory under C:\labcourse\ to load the open-loop control algorithm **open-loop_motor1.alg**.
When performing an **open-loop experiment using Servo Motor 2**:
Select **Load from Disk** and go to your directory under C:\labcourse\ to load the open-loop control algorithm **open-loop_motor2.alg**. You should now see the algorithm in your Setup Control Algorithm similar to Figure 2. You can edit the algorithm by clicking on Edit in the Setup Control Algorithm Window, but this is not required for the open-loop control algorithms.

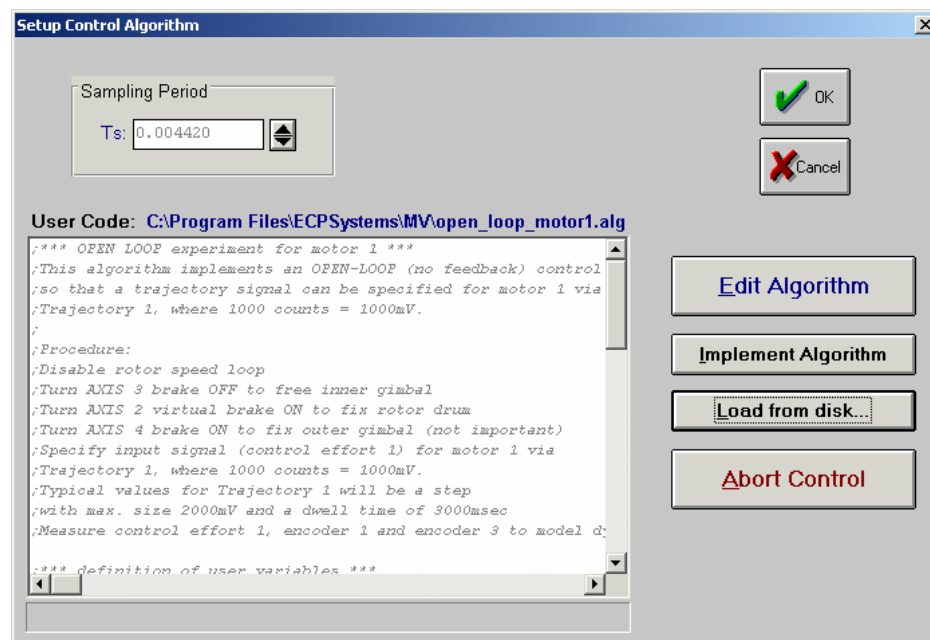


Figure 2: Setup Control Algorithm Window

4. Now click on **Implement Algorithm** button to actually compile and implement the algorithm on the ECP control box. You will see a small Download window with a progress bar, indicating the compilation and download progress.

NOTE: it is good practice to have someone keep his/her your finger close to the **RED OFF** button on the front of the ECP control box. In case the algorithm or the trajectory results in a violently moving Gyroscope system, push the **RED OFF** button on the front of the ECP control box to avoid damage to the ECP gyroscope system.

5. Close the Setup Control Algorithm Window by clicking the **OK** button and your main ECP window should now indicate that the (open-loop) control algorithm is implemented via the **Control Loop Status: CLOSED**.

Control Loop Status: **CLOSED**

Setup of open-loop trajectory

Make sure you have loaded the correct *open-loop control algorithm* for either Servo Motor 1 or Servo Motor 2 as described in the steps above. To perform an open-loop experiment, continue these steps:

- For an **open-loop experiment using Servo Motor 1**: select **Command - Trajectory 1** from the main menu bar in the ECP Main Window to specify the test signal (called trajectory) for Servo Motor 1. For an **open-loop experiment using Servo Motor 2**: select **Command - Trajectory 2** from the main menu bar in the ECP Main Window to specify the test signal (called trajectory) for Servo Motor 2. A window similar to Figure 3 will open, allowing you to specify various test signals (trajectories). Typically we will use Step inputs.

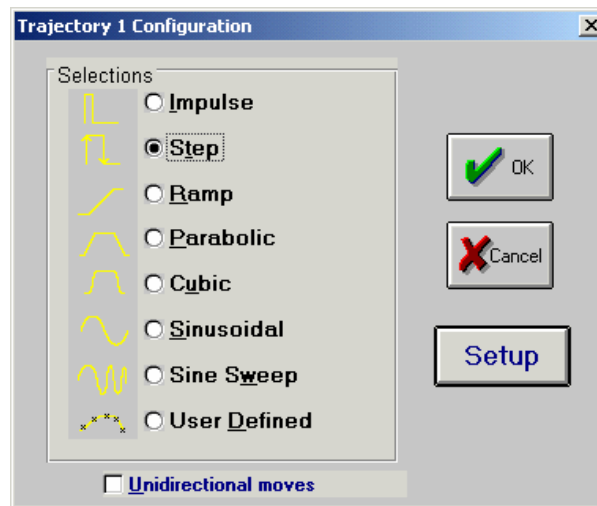


Figure 3: Trajectory 1 Configuration Window

- Select **Step** and click on **Setup** button to set up the step signal. A window similar to Figure 4 will open, allowing the specification of the Step Size, Dwell Time (how long is the step) and the number of repetitions.

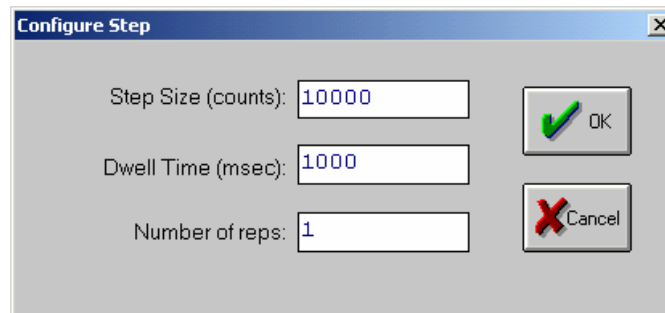


Figure 4: Configure Step Window

NOTE: with the Open-Loop control algorithms (`open-loop_motor1.alg` for Motor 1 or `open-loop_motor2.alg` for Motor 2) implemented in the control box, a Step Size of 1000 counts = Step Size of 1000mV on the motor. For the open-loop experiments, typically the step size is in the order of 2000 = 2000mV to overcome the friction of the motor. Do not make step size larger than 4000 = 4000mV to avoid equipment damage!

- Close the Configure Step window by a click on the **OK** button and close the Trajectory Configuration window also by a click on the **OK** button.

EXPERIMENTS & DATA ACQUISITION

Performing experiments (open- and/or closed-loop) requires setting up data acquisition parameters to indicate which signal should be measured during the experiment. Subsequently the experiment must be run to upload the experimental data. These steps are described in the following.

Set up data acquisition

Make sure you have set up the correct (open-loop or closed-loop) control algorithms and trajectory as described in the steps above.

1. Select **Data - Setup Data Acquisition** from the main menu bar (See Figure 1) to specify which signals to measure during your experiment. A window similar to Figure 5 should appear.

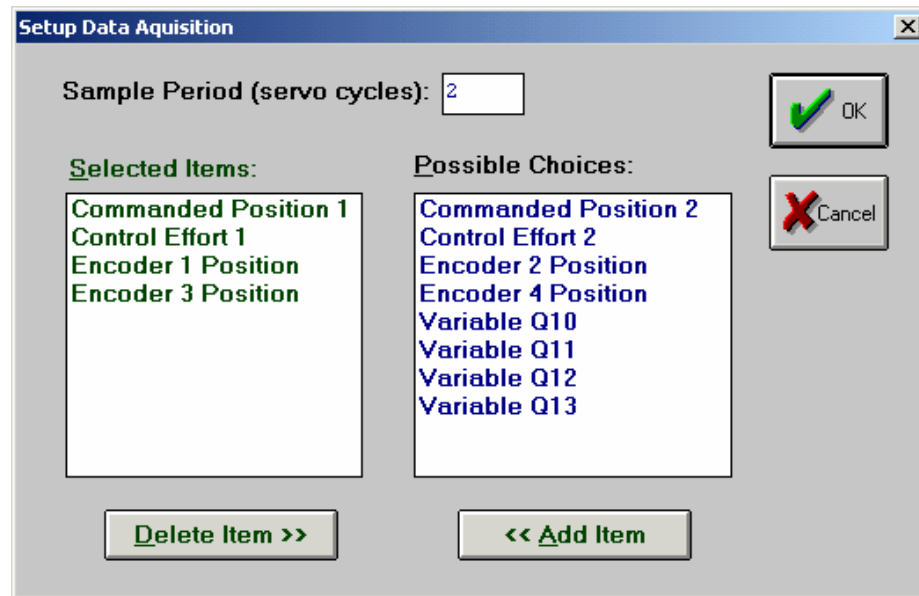


Figure 5: Setup Data Acquisition Window

2. Select the data you would like to measure by clicking on the following names:

Encoder 1 Position	= Output Position of Rotor
Encoder 2 Position	= Output Position of Rotor Drum
Encoder 3 Position	= Output Position of Inner Gimbal
Encoder 4 Position	= Output Position of Outer Gimbal
Control Effort 1	= Input Signal to Servo Motor 1
Control Effort 2	= Input Signal to Servo Motor 2
Commanded Position 1	= Reference signal specified under Setup Trajectory 1 (see above)
Commanded Position 2	= Reference signal specified under Setup Trajectory 2 (see above)

Click on the **Delete Item** or **Add Item** buttons to respectively exclude or include that variable.

3. Think which data you would like to measure when you do your experiment. Examples:
 - a. If you do an *open-loop experiment using Servo Motor 1*, you probably only want to measure **Control Effort 1**, **Encoder 1 Position** and/or **Encoder 3 Position**.
 - b. If you do an *open-loop experiment using Servo Motor 2*, you probably only want to measure **Control Effort 2**, **Encoder 2 Position** and/or **Encoder 4 Position**.
 - c. For **closed-loop experiments** it is worthwhile to also measure the **Commanded Position 1 or 2** (reference) signals to inspect the steady-state error. Click on **OK** button to close the Setup Data Acquisition Window.

Perform Experiment for Data Acquisition

Make sure you have set up the data acquisition according to the steps described above.

1. Select **Utility - Zero Position** from the main menu (see Figure 1) to reset all encoder values to 0. This gives nice plots that will start at 0. You might have to do this several times in case of closed-loop experiments.
2. **ONLY** in case you do an (**open- or closed-loop**) experiment on **Servo Motor 2** and the experiment requires you to have the **Rotor (Motor 1) spin at a contact speed** to model the gyroscopic effect, first follow these steps:
 - a. Read the file **info.txt** in your directory under C:\labcourse\ The file states the desired rotor speed in RPM (Rounds Per Minute).
 - b. **Enable the axis 3 break** (switch down on front of ECP control box). This avoids the rotation of the inner gimbal of the Gyroscope while the rotor is spinning up.
 - c. Select **Command - Initialize Rotor Speed** from the main menu. A window similar to Figure appears where you may enter your specific rotor speed in RPM (Rounds Per Minute).

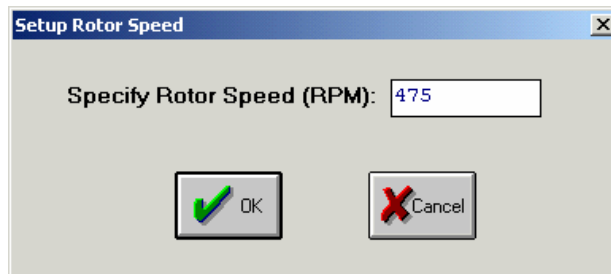


Figure 6: Setup Rotor Speed Window

- d. Now Click **OK** to initialize rotor speed and wait a few seconds till the rotor reaches its desired speed. After that you can **disable the Axis 3 break**.

NOTE: it is good practice to have someone keep his/her your finger close to the **RED OFF** button on the front of the ECP control box. In case the spinning up of the rotor results in a violently moving Gyroscope system, push the **RED OFF** button on the front of the ECP control box to avoid damage to the ECP gyroscope system.

3. Select **Command - Execute** from the main menu to execute your experiment. A window similar to Figure 6 will appear that allows you to specify how to execute the trajectory.

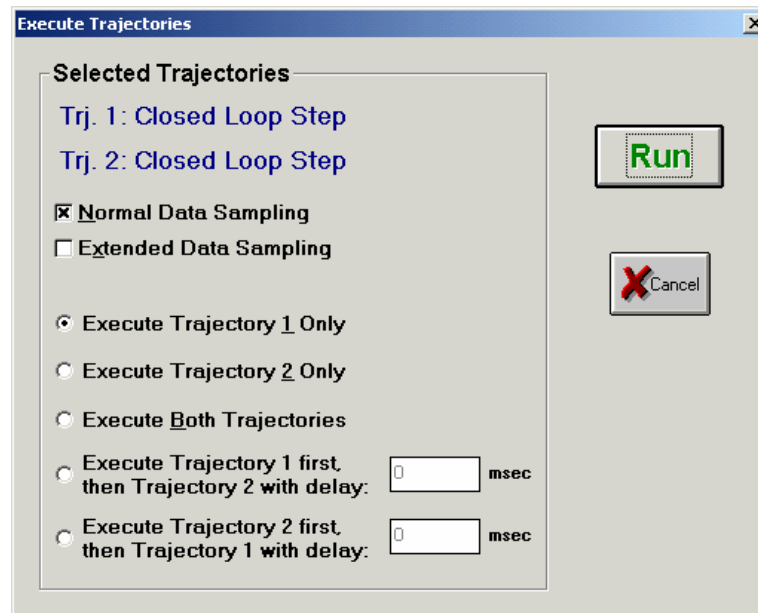


Figure 7: Execute Trajectories Window

In the Execute Trajectories Window (see Figure 7) select **Normal Data Sampling**. Depending on which Servo Motor you are trying to control, select the following:

- a. **Execute Trajectory 1 Only** if you are performing an open-loop experiment using Servo Motor 1 or if you performing a closed-loop experiment that involved the control of Servo Motor 1 (typically in week 1 of the 175a lab when the rotor should not be spinning!).
 - b. **Execute Trajectory 2 Only** if you are performing an open-loop experiment using Servo Motor 2 or if you performing a closed-loop experiment that involved the control of Servo Motor 2 (typically in week 2 & 3 of the 175a lab when the rotor is spinning at a desired RPM).
4. Think if you need to lock/unlock any of the axes before you run your experiment. Example:
 - a. In the **1st week of the 175a lab** you plan to control axis 3 via Servo Motor 1, while axis 4 is locked. In that case the **axis 4 brake should be engaged**. You can also lock axis 2 via the Axis 2 V-brake button in the ECP Main Window (see figure 1). This will implement a local feedback loop around motor 2 to keep axis 2 “virtually locked”.
 - b. In the **2nd and 3rd week of the 175a lab**, you either control axis 2 or axis 4 via Servo Motor 2. Hence, the **axis 3 brake should be engaged**, while the **rotor should be spinning at the desired RPM**.
 5. Click on **Run** in the Execute Trajectories Window (see Figure 7) to start your experiment.

NOTE: it is good practice to have someone keep his/her your finger close to the RED OFF button on the front of the ECP control box. In case the algorithm or the trajectory results in a violently moving Gyroscope system, push the RED OFF button on the front of the ECP control box to avoid damage to the ECP gyroscope system.

6. After the experiment ran successfully (no violent movement of Gyroscope), an Upload Successful window appears similar to Figure 8. You can then click on **O.K.** to finish your experiment.

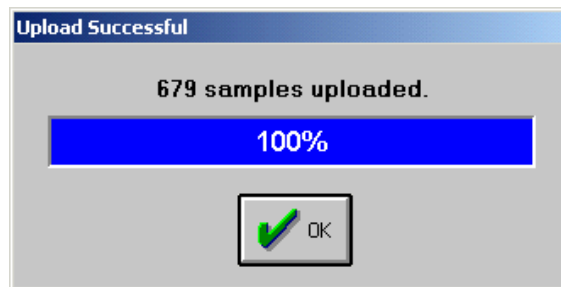


Figure 9: Upload Successful window after experiments completed

If an Error window occurs similar to Figure 10, and no data was uploaded, the Gyroscope system safety (inertial) switches were initiated due to violent movement. Try to reduce your step size or adjust the (PID) control parameters in case of closed-loop control to avoid this error message.

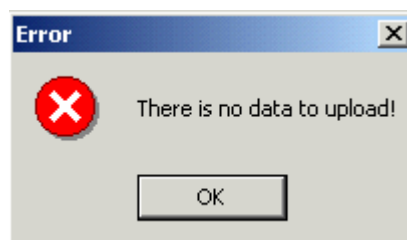


Figure 10: Error window after experiment failed due to violent Gyroscope movement

PLOTTING & SAVING DATA

Plotting Data

Make sure you were able to perform a successful experiment (no violent movement of the ECP Gyroscope system) and that you were able to upload the data according to the steps described above.

1. Select **Plotting - Setup Plot** from the main menu (see Figure 1) to plot and examine the data from your experiment and a window similar to Figure 11 will appear.

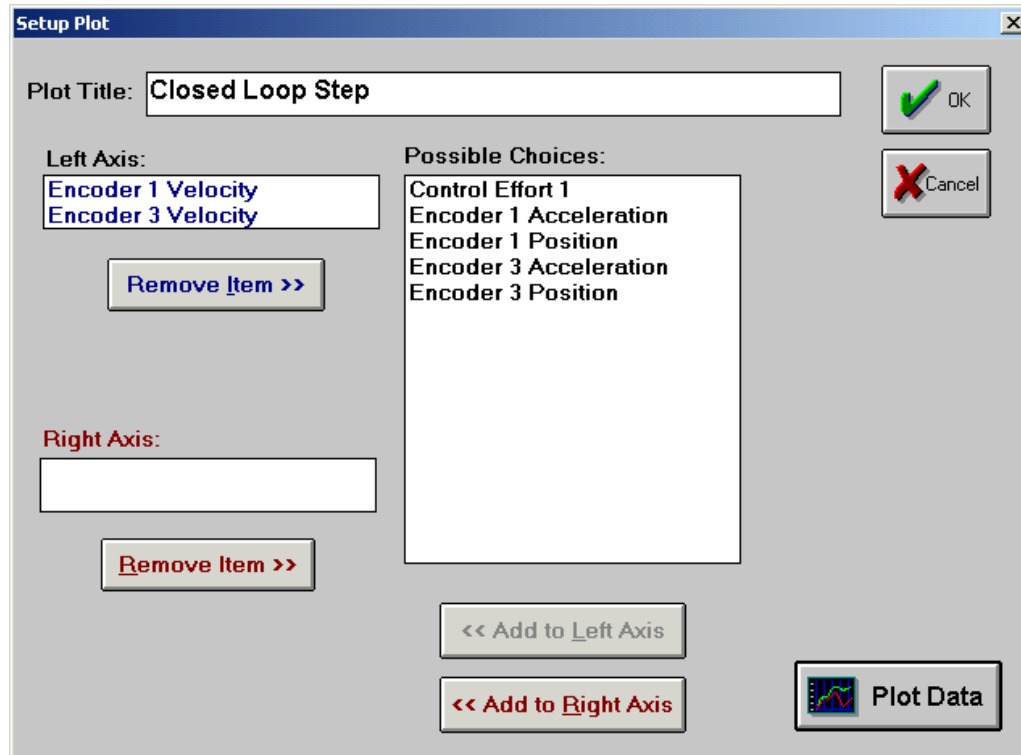


Figure 11: Setup Plot window

2. Select the data you would like to plot by clicking on the names and click on the **Remove Item** or **Add to Left Axis** or **Add to Right Axis** buttons to respectively exclude or include the plot of that variable. You can plot a maximum of 2 variables per axis and depending what you selected under Setup Data Acquisition, the following variables can be plotted.

Encoder 1 Position	= Output Position of Rotor
Encoder 2 Position	= Output Position of Rotor Drum
Encoder 3 Position	= Output Position of Inner Gimbal
Encoder 4 Position	= Output Position of Outer Gimbal
Control Effort 1	= Input Signal to Servo Motor 1
Control Effort 2	= Input Signal to Servo Motor 2
Commanded Position 1	= Reference signal specified under Setup Trajectory 1 (see above)
Commanded Position 2	= Reference signal specified under Setup Trajectory 2 (see above)

Note that **Velocity** and **Acceleration** measurements are found by numerically differentiating the measurements and tend to be noisy! They can, however, provide useful information.

3. For the **open-loop experiment using Servo Motor 1**, it is best to plot the output (Encoder 1 Position or Encoder 3 Position) on the left axis and the input (Control Effort 1) on the right axis, as selected in Figure 11. For the **open-loop experiment using Servo Motor 2**, it is best to plot the output (Encoder 2 Position or Encoder 4 Position) on the left axis and the input (Control Effort 2) on the right axis.
4. Click on **Plot Data** button and a plot of the data will appear in a new window.

Saving Data

Make sure you have performed a successful experiment and were able to plot the data as described in the steps above. Whenever you run a new experiment, the data of the previous experiment will be overwritten in memory. So if you like your measurement and you would like to save the data for your report for plotting purposes, follow these steps:

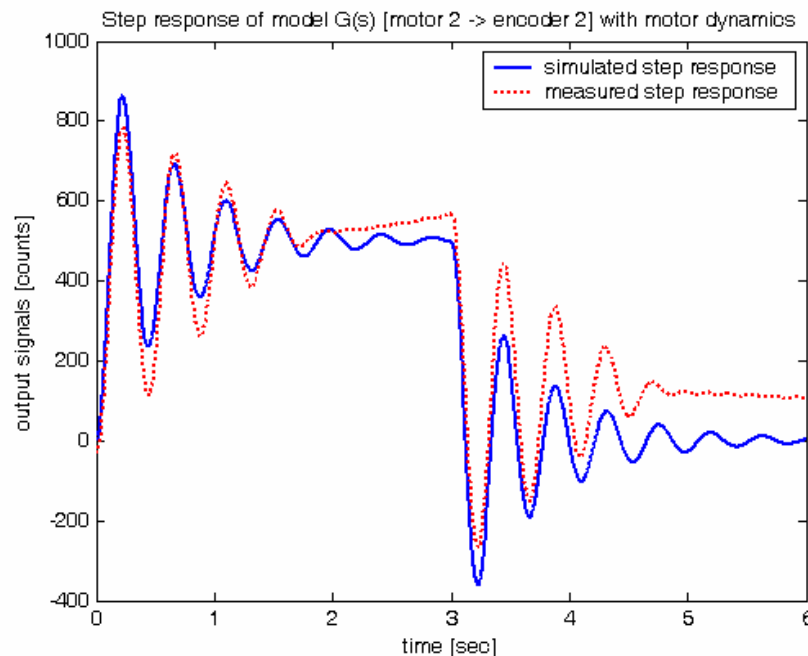
1. Select **Data - Export Raw Data** from the main menu (See Figure 1) to save your data before you start a new experiment.
2. When saving the raw data, make sure you save it in your directory under **C:\labcourse**
3. The saved data will be text file where the data is stored column-wise and can be opened with Notepad and/or Excel. The data file can directly be read by the Matlab program **ecpread** available in your directory. The Matlab program **ecpread** is also used by the Matlab script **maelab** available in your directory to plot your simulation and experimental results for Model Validation and Controller validation (see next page).
4. You can also modify the text file into an m-file so Matlab can read the data directly. Details to convert the text file into an m-file can also be found in your lab handout and requires the following editing steps:
 - a. First line in text file: Comment out the first line with **%**
 - b. Second line in text file: Enter **dummy=** before the opening bracket [**.**
 - c. Last line in text file: put a semicolon **;** behind the closing bracket **]**.
 - d. After last line in text file: define time vector **t**, input vector **u** and output **y** by **selecting the appropriate columns from the dummy variable**. For example, if you have selected to save the control effort (input **u**) and the encoder 1 position (output **y**), this can be done by adding the following lines to the end of the text file:


```
t=dummy(:,2);
y=dummy(:,3);
u=dummy(:,4);
clear dummy;
```
5. Save the raw text file as a file with the extension **.m**. Result is a Matlab script that can be run to read in your measurements.
6. Final note: make sure you use the variables **t**, **u** and **y** to define respectively the time vector, the input vector and the output vector. This allows the data file to also be read by the script file **maelab.m** to validate your models.

VALIDATION OF MODELS

Comparing experimental data with a simulation can validate models of the ECP gyroscope system. Matlab can handle experimental data and simulations and you are provided with a script file called **maelab** to perform all the necessary simulation, validation and control steps. It is imperative to **validate the model before using the model for control design purposes** and to use the **maelab** script file, follow the following steps:

1. Start Matlab
2. In the Matlab command window, type in **pwd** and verify that you are indeed in your working directory under C:\labcourse\
3. Edit the file **models.m** by typing in **edit models** to specify the parameters of the different transfer function you will encounter during the 175a lab:
 - a. G31 (model from motor 1 to encoder 3) in 1st week
 - b. G22 (model from motor 2 to encoder 2) in 2nd week
 - c. G42 (model from motor 2 to encoder 4) in 3rd week
 Make sure to save the **models.m** file before continuing.
4. Run **maelab** script file by typing in **maelab** in the Matlab command window.
5. Specify the Servo Motor you are interested in.
6. Specify the Encoder Output you are interested in.
7. Specify the name of the filename that contains your transfer functions (default **models.m**)
8. Use the menu option to “simulate open loop step response” and enter the values for the step response simulation. Typically the values should be the same as done during the experiment to be able to compare simulations with experiments.
9. NOTE: when asked for a filename, make sure to put the filename between quotes, e.g. **'myfile'**
10. For validation purposes, for each week of your 175b lab, you should have a simulation and an experiment (based on step response) that show close resemblance for model validation purposes, similar to the figure listed below.

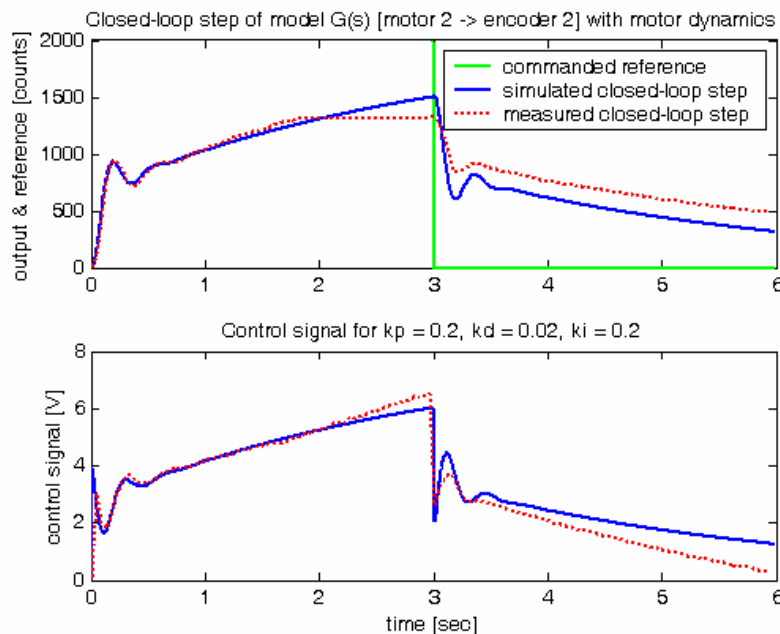


Example of figure for open-loop step-based model validation of week 2 experiment

VALIDATION OF CONTROL

Before implementing a (new) control algorithm, first verify the performance of your proposed P-, PD- or PID-control algorithm with Matlab by running a closed-loop simulation with the Matlab script file called **maelab**. Based on the model that you have validated (as indicated above) It allows you to verify whether your control algorithm will be stable on the actual system and to **verify whether the control signals stay within bounds and are not subjected to excessive oscillations** when making a step on the reference signal. The procedure to **validate the control algorithm before implementation** is as follows:

1. Start Matlab
2. In the Matlab command window, type in **pwd** and verify that you are indeed in your working directory under C:\labcourse\
3. Edit the file **models.m** by typing in **edit models** to specify the parameters of the different transfer function you will encounter during the 175a lab:
 - a. G31 (model from motor 1 to encoder 3) in 1st week
 - b. G22 (model from motor 2 to encoder 2) in 2nd week
 - c. G42 (model from motor 2 to encoder 4) in 3rd week
 Make sure to save the **models.m** file before continuing.
4. Run **maelab** script file by typing in **maelab** in the Matlab command window.
5. Specify the Servo Motor you are interested in.
6. Specify the Encoder Output you are interested in.
7. Specify the name of the filename that contains your transfer functions (default **models.m**)
8. Use the menu option to “Design/evaluate feedback controller” and enter the numerical values for the **kp (proportional gain)**, **kd (derivative gain)** and **ki (integral gain)**. Keep in mind the bounds on the gains to avoid excessive control signals: **$|kp| < 0.5$, $|kd| < 0.05$ and $|ki| < 1$** .
9. Make sure to motivate and argument the choice of your kp, kd and ki values in your lab report based on the figures being created by **maelab** script file.
10. Use the menu option to “Simulate closed loop step response” and enter the values for the step response simulation. The values should be the same as done during an experiment for comparison purposes. Typically a closed-loop step should be in the order of 2000 counts.
11. NOTE: when asked for a filename, make sure to put the filename between quotes, e.g. **'myfile'**. Initially, for control validation purposes, no experimental data is required.
12. For final validation purposes for each week of your 175b lab, you should have a simulation and an experiment (based on a closed-loop step response) that shows close resemblance, similar to the figure below.



Example of figure for validation of control algorithm based on closed-loop step data and simulation

CLOSED-LOOP EXPERIMENTS (for control of the gyroscope)

A closed-loop experiment for the ECP gyroscope system requires the implementation of a so-called *closed-loop control algorithm* and the *specification of an closed-loop trajectory*. To perform a closed-loop experiment, follow these steps:

Setup of Control Algorithm

1. First turn of any closed-loop control algorithm by clicking the large **Abort Control** button in the ECP Main Window (see Figure 1).
2. **ONLY** in case you do a **closed-loop experiment with Servo Motor 2** and the experiment requires you to have the **Rotor (Motor 1) spin at a contact speed** (week 2 & 3 of the lab), spin up the rotor to the RPM specified in the file **info.txt** in your directory.
3. Select **Setup - Control Algorithm** from the main menu bar to implement a *closed-loop control algorithm*.
4. Select **Load from Disk** and go to your directory under C:\labcourse\ to load either one of the following template files:
 - a. Week 1: **template_axis3_control.alg** - template for PD of Motor 1 to control Gyroscope axis 3 via measurement of encoder 3.
 - b. Week 2: **template_axis2_control.alg** - template for PID of Motor 2 to control Gyroscope axis 2 via measurement of encoder 2 and, while rotor motor (motor 1) is spinning at constant speed.
 - c. Week 3: **template_axis4_control.alg** - template for PD of Motor 2 to control Gyroscope axis 4 via measurement of encoder 4, while rotor motor (motor 1) is spinning at constant speed.
4. You should now see the algorithm in your Setup Control Algorithm similar to Figure 2.
5. Make sure to **set the Sampling Period Ts to 0.004420**, as this is the value used in the (default) template files provided to you.
6. Edit the numerical values of the PID or PD control algorithm by clicking on the Edit button (see Figure 2). Make sure to select **File - Save Changes** and **Quit** to save your edited/modified control algorithm.
7. Make sure to **LIMIT THE VALUE OF kd (the derivative gain) to 0.05** to avoid amplification of high frequency sensor noise, resulting in a very noisy control signal. In case of a high frequency control signal you will hear/feel the **Gyroscope buzzing** and **should be avoided**.
8. Make sure to **LIMIT THE VALUE OF kp (the proportional gain) to 0.5** to avoid large control signals.
9. Make sure to **LIMIT THE VALUE OF ki (the integral gain) to 1** to avoid large control signals.
10. By a click on **Implement Algorithm** button to actually compile and implement the algorithm on the ECP control box. You will see a small Download window with a progress bar, indicating the compilation and download progress. **Keep in mind the bounds on kp, kd and to slowly increase the value of ki as indicated above before implementing the control algorithm!**

NOTE: it is good practice to have someone keep his/her your finger close to the RED OFF button on the front of the ECP control box. In case the algorithm or the trajectory results in a violently moving Gyroscope system, push the RED OFF button on the front of the ECP control box to avoid damage to the ECP gyroscope system.

11. Close the Setup Control Algorithm Window by clicking the **OK** button and your main ECP window should now indicate that the (closed-loop) control algorithm is implemented via the **Control Loop Status: CLOSED**.

Control Loop Status: CLOSED
