3 EXPT #1: FAMILIARIZATION WITH EXPERIMENT SYSTEMS

3.1 OBJECTIVE

- To become familiar with the lab equipment (ECP Model 220 system), software and the procedure to obtain typical system responses.
- To use MATLAB in control system application.

3.2 INTRODUCTION

3.2.1 ECP EXECUTIVE CONTROL PROGRAM

Please read Section 2 – "LABORATORY EQUIPMENT OVERVIEW" for a detailed introduction.

3.2.2 Open-loop and Closed-loop Block Diagrams

In this experiment, you will obtain open-loop and closed-loop responses of the position control system. First, we present the block diagrams and transfer functions of the open-loop and closed-loop systems.

Open-loop Block Diagram

The basic 'rigid' plant used in the lab system is equivalent to a rotational mass (total inertia of the turntable/motor system, J (kg.m^2)) supported by bearings (assumed as 'viscous friction' B, N.ms/radian), driven by a servo DC motor.

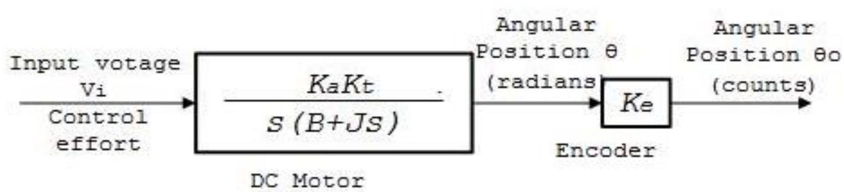


Figure 3.1: DC Motor Open-loop Diagram

$$K_t K_a V_i = J\ddot{\theta} + B\dot{\theta} \tag{3.1}$$

- K_t – torque constant
- K_a – trans-conductance gain
- J – total inertial of turntable system
- B - Viscous friction

From (3.1), the transfer function of the DC motor is:

$$\frac{\theta(s)}{V_i(s)} = \frac{K_a K_t}{s(B+sJ)} = \frac{K_m}{s(B+sJ)} \tag{3.2}$$

The angular position is measured using an encoder with the gain K_e, thus the overall transfer function for open-loop test is

$$G_p(s) = \frac{\theta_o(s)}{V_i(s)} = \frac{K_a K_t K_e}{s(B+sJ)} = \frac{K_m K_e}{s(B+sJ)} = \frac{K_o}{s(1+\tau s)} \qquad (3.3)$$

Where the open-loop gain $K_0 = K_a K_t K_e / B$ (counts/sec per volt) and $\tau = {}^{J}/_{B}$(sec). From Equation (3.3), we obtain the angular velocity (counts/sec) as:

$$\omega(s) = \frac{K_0 V_i(s)}{1+s\tau} \qquad (3.4)$$

Closed-loop Block Diagram

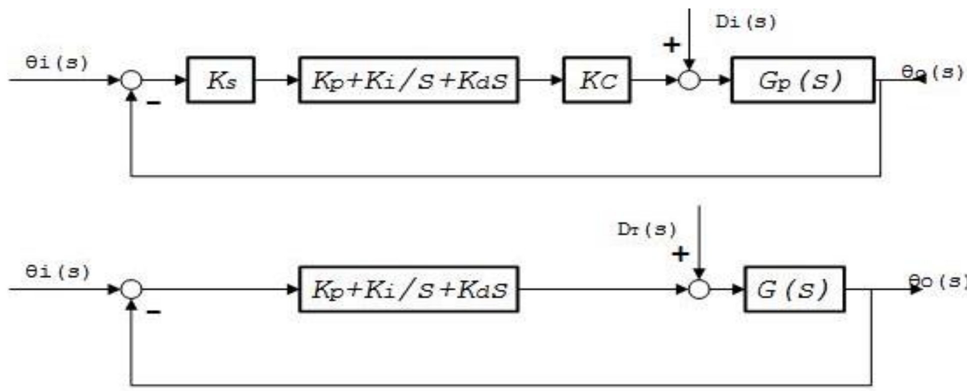


Figure 3.2: ECP 220 System Configuration

The block diagram of the closed-loop system (with a PID controller) is shown in the top diagram in Figure 3.2. In addition to the controller and plant $G_p(s)$, the diagram includes two gains: software gain K_s and D/A converter gain K_c. This block diagram can be redrawn as in the second diagram in Figure 3.2 where

$$G(s) = K_s K_c G_p(s) = \frac{K}{s(B+sJ)}$$

Here the product of the gains K_s and K_c, together with the motor-associated gains K_a and K_t and the encoder gain, are used to define the plant gain: $K = K_s K_c K_e K_a K_t$

The manufacturer-specified nominal values of the constituent factors are as follows:

- Controller Software Gain K_s = 32 Controller input counts/Encoder or reference input counts
- D-to-A Converter (DAC) Gain K_c = 10 volts/32,768 counts = $3.05(10)^{-4}$ volts/count
- Encoder Gain K_e = 16000 pulses(counts) / **2π** radians = 2546.5 counts/radian
- Servo-amplifier Gain $K_a \approx$ 2 Amps / Volt
- Servomotor Torque Constant $K_t \approx$ **0.1** N-m / Amp

The nominal value of K, in the closed-loop configurations, is therefore:

$K = K_s K_c K_e K_a K_t = (32)\,[3.05(10)^{-4}]\,[2546.5]\,(2)\,(0.1) = 4.97$ or $K \approx$ **5** Nm/radian*

The nominal values of B and J will be discussed in Experiment #2.

3.2.3 MATLAB

1. MATLAB

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. You can use MATLAB for the applications of control systems, testing, and measurement. MATLAB is widely used in control system analysis, design, and simulation.

2. CONTROL TOOLBOX

Control System Toolbox™ provides industry-standard algorithms and apps for systematically analyzing, designing, and tuning linear control systems. You can specify your system as a transfer function, state-space, pole-zero-gain, or frequency-response model. Apps and functions, such as *Step* and *Bode*, let you visualize system behavior in time domain and frequency domain. You can tune compensator parameters using automatic PID controller tuning, Bode loop shaping, root locus method, LQR/LQG design, and other interactive and automated techniques. You can also validate your design by verifying rise time, overshoot, settling time, gain and phase margins, and other requirements.

Useful Functions

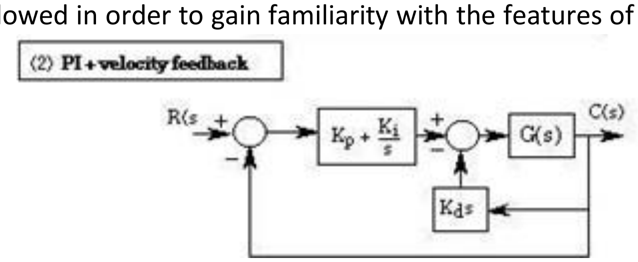
- *tf* - Create transfer function model, convert to transfer function model.
- *pid* - Create PID controller in parallel form, convert to parallel-form PID controller.

References

For more detailed information, please refer to **MATLAB Control System toolbox** website, http://www.mathworks.com/help/control/index.html

3.3 ECP MODEL 220 SYSTEM

In this part, basic operational procedures are followed in order to gain familiarity with the features of the ECP Model 220 system. The default closed-loop "PI+Velocity Feedback" configuration #2 will be used to illustrate various measurements. In section 3.5 we will do experiment with open-loop configuration.



1. Initializing and Selecting Units

From the UTILITY menu, select "Reset Controller". This will set all encoder counts to zero [If "Counts" are not selected, go to USER UNITS under "Setup" and select COUNTS. (16000 Counts = One 360° revolution) Counts will be used in all the experiments. With COUNTS selected, all displayed output magnitudes will be in terms of Counts for position or Counts/sec for velocity.

2. Setup Data Acquisition

From the DATA menu, select the menu "Setup Data Acquisition" and make sure that Commanded Position and Encoder #1 Position are on the list. Select Sample Period = 2 (This implies that data will be collected every second servo cycle, i.e. at intervals of $2T_s$).

3. Setup Control

"Set Up Control Algorithm" window, under which CONTINUOUS TIME, T_s = 0.00442 sec and "PI with Velocity Feedback" have already been selected. Click on SETUP ALGORITHM. This will display the relevant control schematic with the preselected gains K_p =0.2, K_d = 0.01 and K_i = 0. Then click on IMPLEMENT ALGORITHM and OK. This downloads the selected control law to the controller.

Note: 'Implement' will always reset the controller unless inconsistent values are set or the loop is opened. The closed-loop system in configuration #2 is now ready for an input command.

4. Choose the system input command

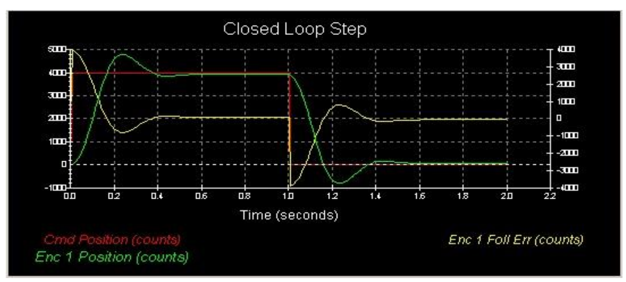
From the COMMAND menu, select "Trajectory" and from the "Trajectory Configuration" window, select STEP (default setting) and then click on "Set Up". Select CLOSED LOOP. This will display the default Step input parameters: STEP SIZE 4000 Counts, DWELL TIME 1000mS, and Number of Repetitions 1.

5. Execute the command

From the COMMAND menu, select "Execute Trajectory" under which check only NORMAL DATA SAMPLING. Then click on RUN. This will initiate the control action. Wait for the data acquisition to be completed, that is, until the ("Upload Successful 100%") message appears. Then click OK.

6. Plot the input and output of the step response

From the PLOTTING menu, select "Setup Plot". Choose for the Left Axis, the variables Commanded Position and Encoder #1 Position. Choose Encoder #1 Following Error for the right axis. Then click on the PLOT DATA button at the lower right hand side of the window. A plot similar to the one shown below should appear.



RESULTS & QUESTIONS

Note how the error varies in the opposite direction of the output and approaches zero in the steady state.

7. Save a copy of the plot for your report

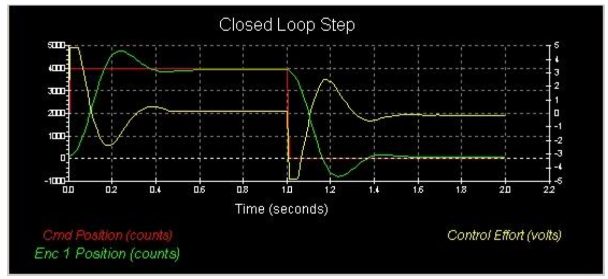
First maximize the display to 'full screen' by clicking the 'maximize' icon at top right. Press the key of PrintSC or Alt+PrintSC, then paste the image to Paint to save the image as a *.jpg file, on the desktop, after giving it a name. (This image file can be manipulated using Windows Paint^{tm} for your report. Among other manipulations, it is most useful to 'invert colors', so that the image is saved and printed using a white background, which will not consume large amounts of printer toner or ink). This procedure for saving plots for your report must be used in all subsequent examples in this experiment as well as in the remaining experiments.

8. Scale the plot for better readability

From the PLOTTING menu -> axis scaling, change the scale of the plots using left/right axis range and horizontal axis.

9. Display the 'Control Effort'

Repeat Step 2 above to add 'Control Effort' (CE) to the list of available variables. Then **repeat** Step 6 above with the 'Control Effort' chosen as the Right Axis variable, instead of the Encoder #1 Error. Then RUN the system with the default step input and then click on the PLOT DATA button.



A plot similar to the one shown below should appear.

RESULTS & QUESTIONS

Note how the CE, which is the input to the servomotor, varies in a manner similar to that of the error, but anticipates the output variation with an observable lead time. The CE is the input whenever the system is operated in the open-loop mode.

3.4 PLOT DATA WITH MATLAB

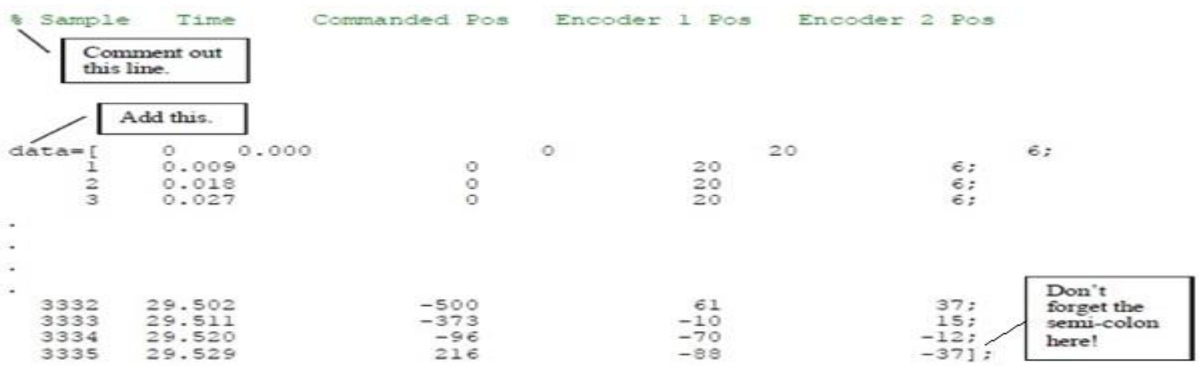
3.4.1 EXPORT RAW DATA

ECP software can export raw data, it allows the currently acquired data to be saved as a text file (*.txt), which can be edited and exported to MATLAB™.

To get raw data, after collecting data:

a. Choose 'Export Raw Data' from the Data menu.
b. Save your data as a text file in the MATLAB bin folder. Give it a name pertaining to your data, like "sinesweep220".
c. Using Windows Explorer, change the extension on your file from '.txt' to '.m'. Moreover, to change from '.txt' to '.m' you may rename the file and change '.txt' to '.m'.
d. Open your m-file in MATLAB for editing.

Change the top and bottom of your file like the example below:



The following is a sample code to plot the data. You can copy the sample code into an m-file.

```
time = data(:,2); %read Time
y=data(:,4);      %set Encode1 Pos as y
u=data(:,6);      % Control Effort as input
plot(time, y);    %Plot data
```

For u=data(:,6), based on the obtained raw data, you have to change the value 6. For example, you could change it to 4 or 5. Compare the plots in Matlab with the screenshot on ECP Executive, try to adjust the parameters of the function 'plot' to make its output close to ECP format.

3.5 OPEN-LOOP TEST AND SIMULATION

In this section we obtain the step response of the open-loop DC motor, first using Matlab simulation then using lab test on the motor. Then we will compare the result.

3.5.1 Matlab Simulation

As an example, set the physical constants as:

```
K = 5.0;             %  Plant gain

B = 0.002;            % Viscous friction (N.m.s/rad)
J = 0.0043;           % kg.m^2

s = tf('s');
dcm_s = K/(J*s+B);    % Speed (counts/s)
dcm_p = dcm_s/s;      % Position (counts)
```

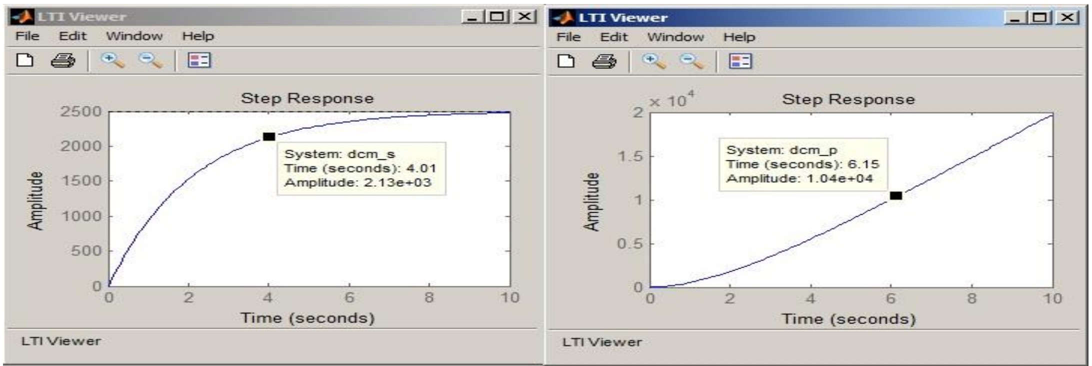
To see the transfer function:

```
>>dcm_s
>>dcm_p
```

To see the step response (speed and position) to step voltage input use:

```
ltiview('step',dcm_s, 0:0.1:10);
ltiview('step',dcm_p, 0:0.1:10);
```

Then you can see a similar display as follows:



Also, with right-click on the plot, click "Plot Types”, you can change the input types.

3.5.2 VERIFICATION ON THE EQUIPMENT

Now we obtain the step response of the open-loop system.

1. Run ECP program. Under ‘Data Acquisition’ menu, make sure that Control Effort (CE) has been added to the list. Set up the 'plot' to display the Encoder #1 Position on the left axis and the CE on the right axis.

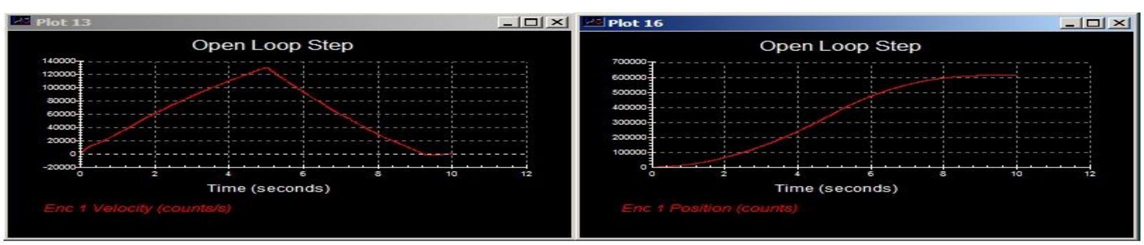
2. Reset the controller from the Utility menu. From the Control Algorithm menu, implement the PID with the following settings:

$$T_s = 0.00442 \text{ sec}, \quad K_p = 1 \quad K_i = 0 \text{ and } K_d = 0.0$$

3. Obtain the **open-loop** step response. From the COMMAND menu, select “Trajectory” and from the “Trajectory Configuration” window, select STEP (default setting) and then click on “Set Up”. Select OPEN LOOP: Step Size of 0.4v, Dwell Time of 5000 ms, and number of Repetitions 1.

4. Go to ‘Execute’ to run the test, and display the plot. Use axis-scaling to display only the section of the response within the dwell period of 5 seconds to make the display appear as a ‘step response’. Note that instead of unit step input 1.0v used in simulation, we used 0.4v input to avoid exceeding the motor speed safety limits.

5. To see the velocity response, set up the ‘plot’ one more time to display the Encoder #1 Velocity on the left axis and the CE on the right axis. Then plot the data.



5. Compare the obtained results with the one from 3.5.1.