XJME2650 Mechatronics and Measurement Systems

Lab Report

DC Motor Speed Control Using PI Control

Author: Yuhui Bi
SID: 201199486
Date: 28 Nov 2019

Catalogue

I. Introduction

In module XJME2650 Mechatronics and Measurement Systems, students are required to understand the control method of PI controller, Laplace transform, system identification, etc. ‘DC Motor Speed Control Using PI Control’ experiment contains two individual experiments, ‘PI Control Gain Tuning’ and ‘Frequency Response’. After connecting the electrical circuit correctly and enabling the motor, different performance of DC motor, such as the change of power consumption and steady state error, could be achieved by tuning frequency, control gains. LabVIEW block diagram and MATLAB programming were used to analyze the method of controlling and obtain other conclusion, such as system identification and Bode diagram. With transfer function for PI controller and transfer function for identified system acquired from MATLAB, transfer function for motor could obtain by mathematical derivation.

II. Aim and Objective

1. Identify the hardware and software in this experiment and connect electrical circuit correctly.
2. Be aware of the signal transmission between hardware and software.
3. Mathematically derive the equation of integration implemented of PI control
4. Obtain experimental results and analysis for 2 individual experiments.
5. Be aware of provided MATLAB codes how to acquire the desired results and how LabVIEW blocks diagrams work for 2 individuals experiments.
6. Use the provided MATLAB codes to acquire system identification and transfer function. Number of zeros and poles should be combined repeatedly to obtain best fitting, i.e. 100% fitting to data, and minimize mean square error as much as possible
7. Understand a Bode diagram and plot it with experimental points and model curve in MATLAB.
8. Draw a block diagram for motor and derive the transfer function mathematically with the help of block diagram, transfer function of PI controller and identified system.

III. Methods

The experiment is composed of 2 individual experiment, ‘PI Control Gain Tuning’ and ‘Frequency Response’.

3.1 Explanation of Hardware and Software (Task1)

3.1.1 Introduction to Experimental Apparatus (Hardware)

Experimental apparatus consists of a high power DC motor with optical encoder, NI myRIO with adapter and bread board, a multimeter and a varying voltage resource shown below.

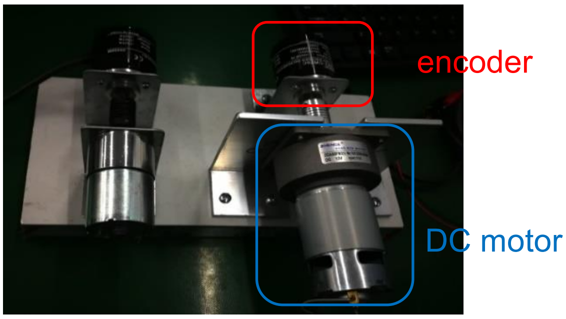


Fig 3.1.1.1: High power DC motor with optical encoder

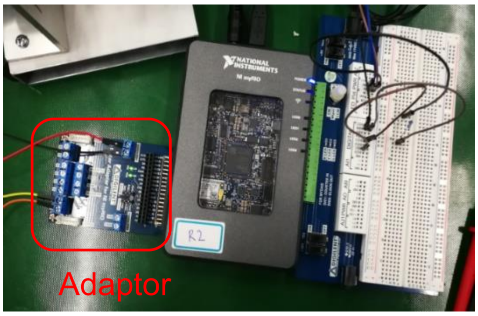


Fig 3.1.1.2: NI myRIO with adaptor and bread board

Fig 3.1.1.3: Multimeter

Fig 3.1.1.4: Varying power source

In this experiment, the type of DC motor is PM55L-048. The optical encoder’s number of pulses per revolution is 1000 Pulses Per Revolution. There’s a gearbox, whose reduction ratio was 31:1, on high power DC motor. The power source maintained 12V. Adaptor is to connect voltage resource and transmit signal. Bread board is to provide a place for connecting partial electrical circuit. Bread board, adapter and myRIO could be treated as a part.

3.1.2 Electrical Circuit

The electrical circuit should be connected as figure fig 3.1.2.1.

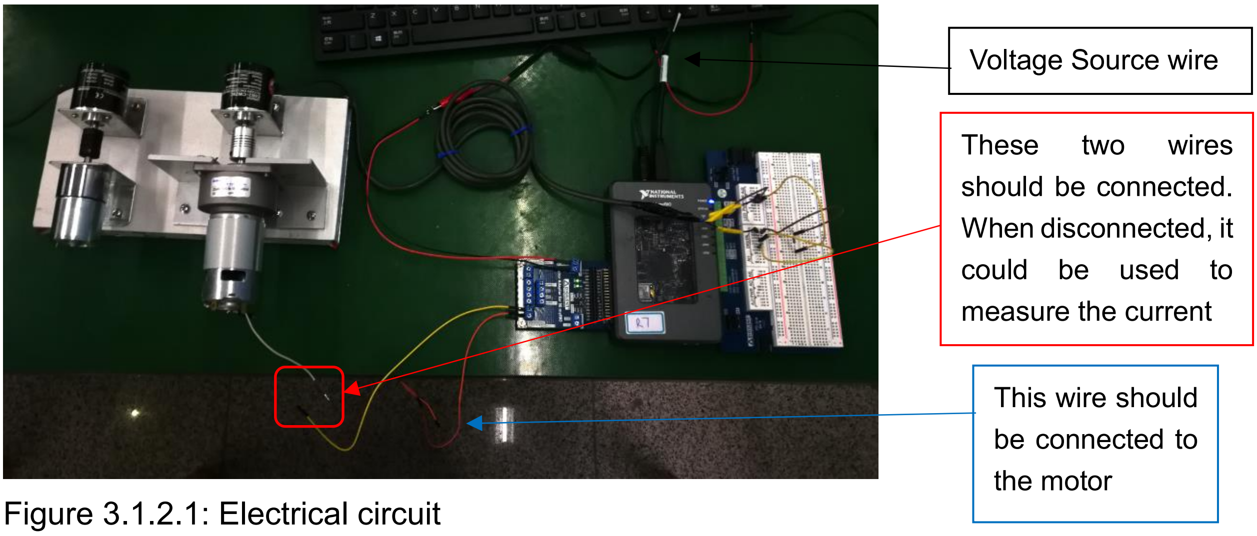


Figure 3.1.2.1: Electrical circuit

3.1.3 Introduction to Experimental Software

The method of controlling for the motor is called Field Programmable Gate Array, which abbreviates as FPGA. FPGA could connect the hardware and software, i.e. between motor with encoder and LabVIEW program.

Fig 3.1.3.1 shows the partial view of front panel in experiment. Reference speed, proportional gain K_p, integral gain K_i and frequency belonged to input signal. The DC motor velocity and motor's power belonged to output signal.

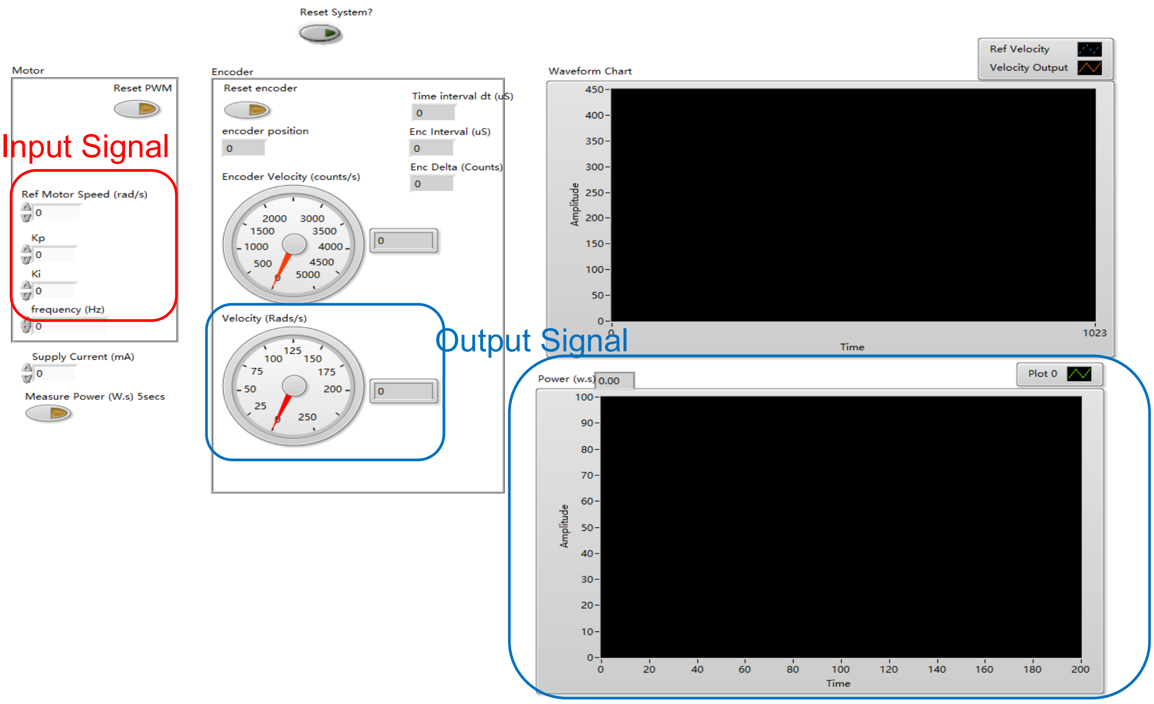
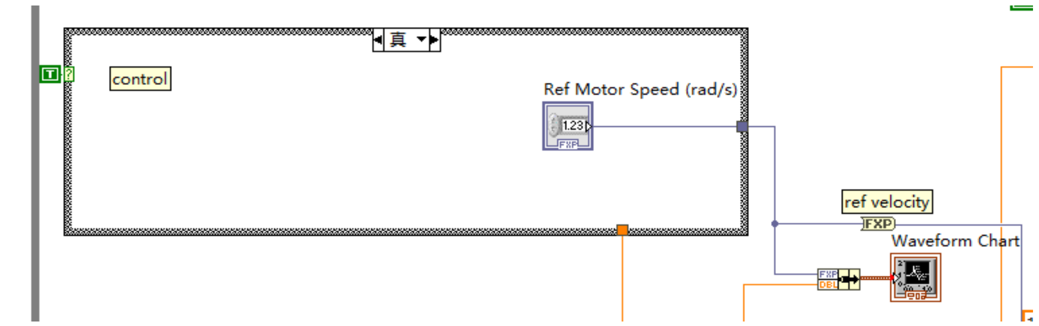


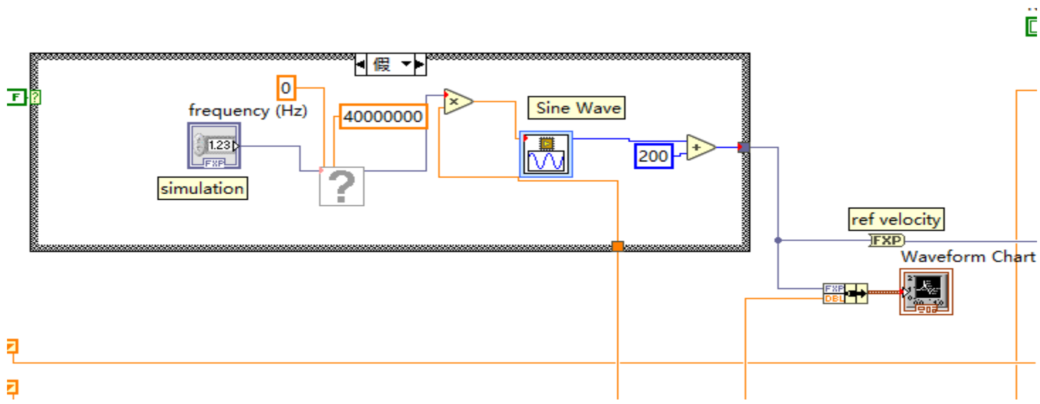
Fig 3.1.3.1: Partial View of Front Panel

Table 3.1.3.2 shows block diagram in LabVIEW with description.

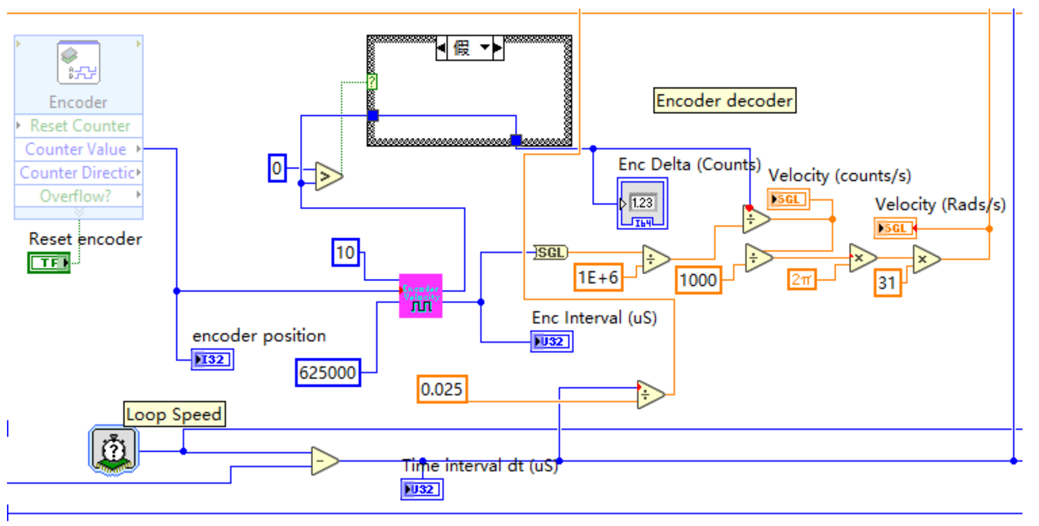
<table>
<tr><th>Block Diagram</th><th>Description</th></tr>
<tr><td>



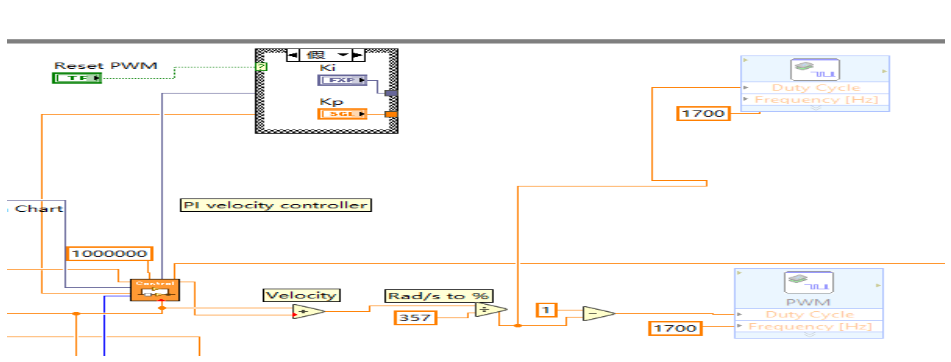
</td><td>This block diagram, used in ‘PI Control Gain Tuning’ experiment, is to input the ‘ref motor speed’ as 200 rad/s and put it to waveform chart as reference compared with velocity output.</td></tr>
<tr><td>



</td><td>This block diagram, used in ‘Frequency Response’ experiment, is to input the ‘ref motor speed’ with some certain value to transform with a sine wave which oscillates around x axis. Adding 200 will make oscillate around 200. Finally put it to waveform chart as reference compared with velocity output.</td></tr>
<tr><td>



</td><td>This block diagram is to count the time and calculate the counts per second of the optical encoder and transfer the speed to motor speed as rad per second. Then put the motor speed on the waveform chart using ‘ref motor speed’ as reference.
Normally, ‘enc delta’>0, false case in the structure case is used and ‘enc delta’ will pass through directly</td></tr>
<tr><td>



</td><td>This block diagram is to input proportional gain K_p and integral gain K_i of PI controller to control the velocity output and consumed power of motor. Detail about PI controller is below.</td></tr>
</table>

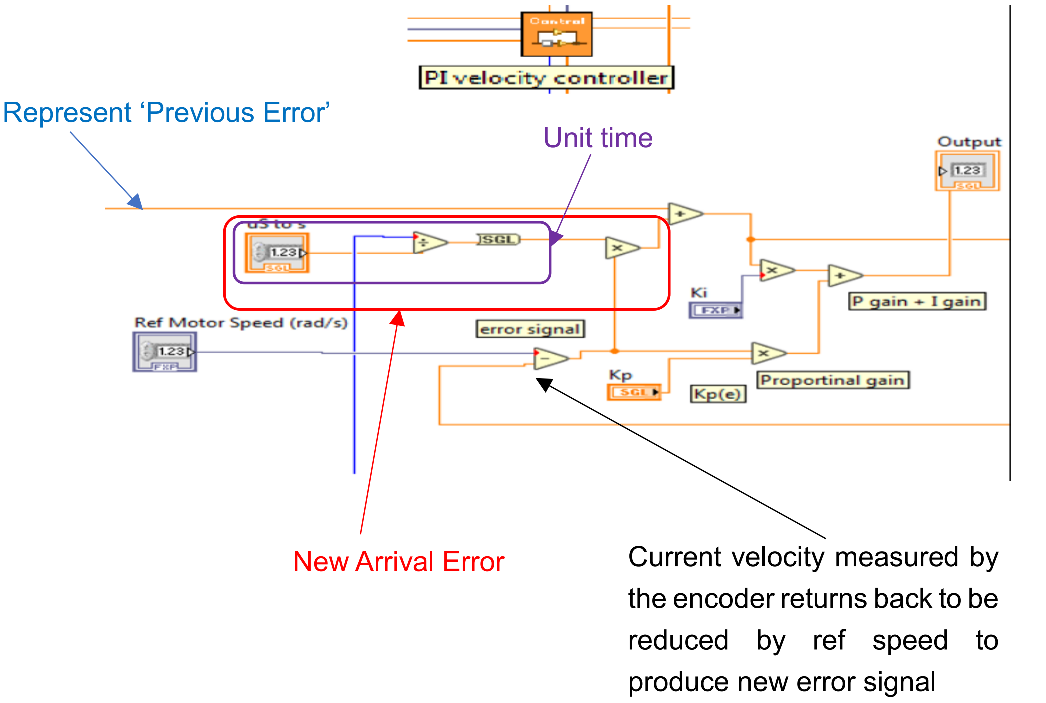
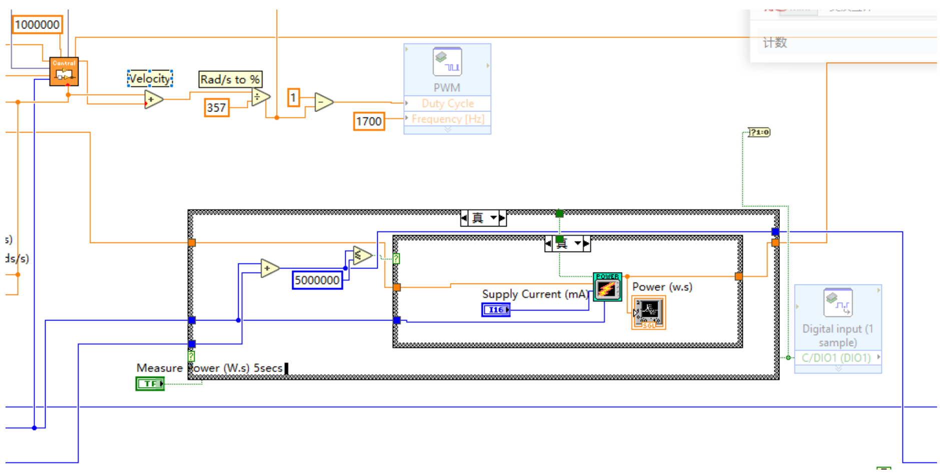
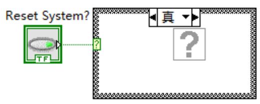
<table>
<tr>
<td>
</td>
<td>Explanation of PI controller:
From the left block diagram, ‘Ref Motor Speed’ reduces current ‘Velocity Output’ to obtain the ‘error signal’. The red frame represents error times the time to obtain the new arrival error. New arrival error plus previous error is the process of accumulation respect to the time, which is regarded as the integration. Proportional gain times error plus integral gain times integration of error equals to velocity, which join in next calculation with current velocity outside PI controller.</td>
</tr>
<tr>
<td>
</td>
<td>This block diagram is to calculate the consumed power in 5 seconds with measured current by multimeter in ‘PI Control Gain Tuning’ experiment.
PWM, abbreviates as Pulse Width Modulation, is to keep the voltage level constant and adjust the time interval of turning the voltage.</td>
</tr>
<tr>
<td>
</td>
<td>This block is used to reset the system.</td>
</tr>
</table>

Table 3.1.3.2: Block Diagram in LabVIEW and its description

3.2 Introduction to ‘PI Control Gain Tuning’

Firstly, open LabVIEW with ‘myRIO DC motor STEP’ program on computer. Enable the motor. Reset all values to 0 and set up parameters of reference speed, proportional gain K_p and integral factor K_i. ‘Ref motor speed’ was 200 rad/s, ‘K_p’ varies 1 to 100 and ‘K_i’ varies 0 to 10. Let frequency be 0 all the time.

In the first trial, K_p could be 1 and K_i could be 0.
There's 3 kind of approaches to change 2 gains.

1) Just increase the value of K_p and keep K_i constant.
2) Just increase the value of K_i and keep K_p constant.
3) Increase 2 values at same time.

Secondly, enable the program. Observe the change of waveform chart of velocity output. When the motor was in steady state, export the velocity data of waveform chart form LabVIEW. Disconnect the wire between motor and myRIO. Use multimeter to connect the circuit to measure the current through the motor.
Thirdly, use current value to enter 'supply current'. Because the current won't change when varying K_p and K_i, don't need to measure the current another time. Click 'Measure Power Secs' button to record the power and observe the waveform chart of power in 5 seconds.
Change the value of K_p and K_i with mentioned approaches, repeat all steps (EXCEPT measurement of current) for several times.

3.3 Introduction to 'Frequency Response'

Firstly, open LabVIEW with 'myRIO DC motor SINE' program on computer. Reset all values to 0. Enter proportional gain K_p with 5 and integral gain K_i with 0.5 and enable the motor.
Secondly, varying frequency from 2 to 10. In the first trial, the frequency could be 2. Run the program and observe the change of waveform of velocity and export it from LabVIEW.

3.4 Block Diagram for Component and Sub-System (Task 2)

LabVIEW software control myRIO with signal directly, which could export signal to LabVIEW. Voltage source was connected to myRIO. Multimeter was used to measure the current through the motor in the first experiment.
As fig 3.4.1 shown, put data to LabVIEW on PC artificially, for example, 'frequency' as 2 Hz, myRIO could control and send the signal to the motor. Receiving the signal, motor will rotate and the rotation of coils of motor could be recorded and counted by encoder. Encoder will collect the data and send signal back to myRIO. For myRIO- motor- encoder cycle, it is single way loop.
Note: Due to the functional integration of bread board, adaptor and myRIO, 3 components are regarded as one.

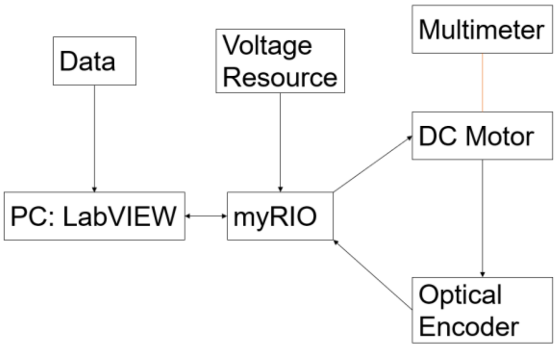


Figure 3.4.1: Block diagram for Components

Figure 3.4.2 shows thw overall block diagram. Enable the motor at first. When input the gain, frequency, ref speed on the LabVIEW and start, LabVIEW will send signal to myRIO to adjust the speed of the motor with signal, motor's speed will be recorded by the optical encoder to deliver the speed output as signal back. The difference (or called error) between reference velocity and velocity output will go through PI controller as signal and prodece new velocity output repeatedly as signal and control myRIO to produce the signal to motor to adjust. Finally, LabVIEW will show the waveform chart including reference velocity and velocity output and finish the calculation of power consumption ,etc.

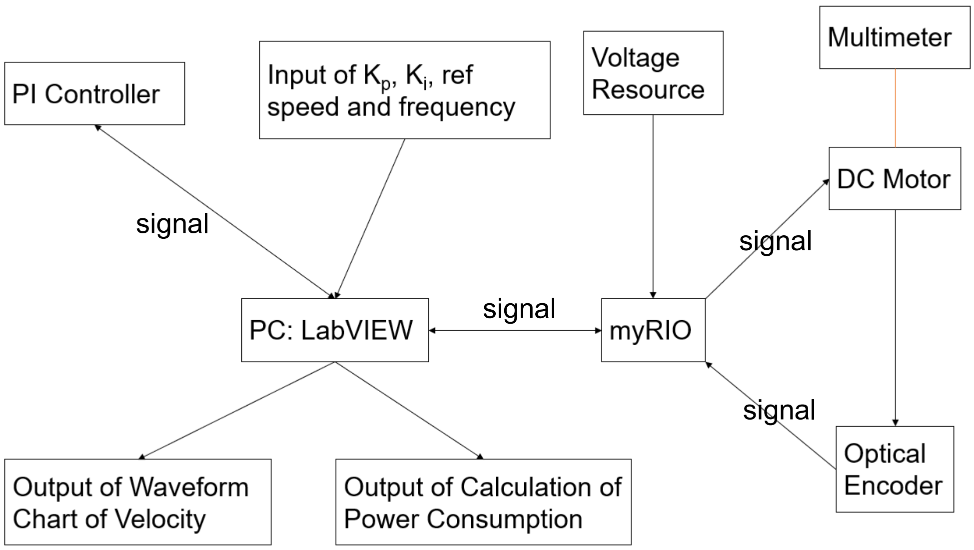


Figure 3.4.2: Block diagram for Components and Sub-Systems

3.5 Formula of the Integration Implemented for the PI Controller (Task 3)

From PI control system, the accumulated or current motor speed u(t) and relationship between accumulated or current motor speed and error signal e(t), equation 3.6.1 and 3.6.2 could be obtained. (Onwubolu, 2005)

$$\mathrm{u(t)} = K_p \times e(t) + K_i \times \int e(\tau)d(\tau) \quad \text{[Equation 3.5.1]}$$

$$\mathrm{R(t) - u(t) = e(t)} \quad \text{[Equation 3.5.2]}$$

Derivative equation 3.6.1 and take Laplace transform:

$$\mathrm{s} \times \mathrm{U(s)} = s \times K_p \times E(s) + K_i \times E(s) \quad \text{[Equation 3.5.3]}$$

R(t)=200 in the experiment, take Laplace transform of equation 3.5.2:

$$\frac{200}{s} - \mathrm{U(s) = E(s)} \quad \text{[Equation 3.5.4]}$$

Take equation 3.6.4 back to equation 3.6.3:

$$\mathrm{s} \times \mathrm{U(s)} = s \times K_p \times [\frac{200}{s} - \mathrm{U(s)}] + K_i \times [\frac{200}{s} - \mathrm{U(s)}]$$

Rearrange it:

$$\mathrm{U(s)} = \frac{200K_p s + 200K_i}{(K_p + 1)s^2 + K_i s} \quad \text{[Equation 3.5.5]}$$

Transfer function for PI controller could be found from equation 3.6.3

$$G_I(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \quad \text{[Equation 3.5.6]}$$

3.6 Implementation of MATLAB Codes (Task 5)

Input the provided codes with calculated frequency, phase angle. Due to the vertical axis of amplitude-frequency diagram in Bode diagram is dB, amplitude ratio (Ar) used in code should be processed as below:

$$\text{Ar used in code} = 20 \times \lg(\text{Ar}) \quad \text{[Equation 3.6.1]}$$

For the term ‘nz’ and ‘np’, number of zero and number of poles, np equals to the number of trials and nz should be keeping plugging in different positive integer to fit the data with 100%and to make MSE (mean square error) enough small in ‘command window’. np=5 and nz=4 is better for fitting.

Figure 3.6.1 shows the MATLAB codes and amplitude ratio used in code.

```
fq=[2,4,6,8,10];
ar=[0.9861,0.6445,0.454,0.331,0.4765];
Ph=[-0.3464,-1.3391,-1.10721,-0.915,-1.508];
Fq=fq*2*pi;
Ar=-20*log10(ar);
deg=Ph*180/pi;
zfr = Ar.*exp(1i*Ph);
data = frd(zfr,Fq);
nz = 4; np = 5;
Gc = tfest(data,np,nz)
bode(Gc);
[mag, pha,hz] = bode(Gc);
```

Negative Sign to make experimental measurement coincide to model identified

Figure 3.6.1: MATLAB codes for Bode Diagram

Figure 3.6.2 shows the command window of MATLAB:

```
Gc =

      -1499 s^4 - 93.64 s^3 - 6.389e06 s^2 + 3.648e06 s - 9.62e08
  ---------------------------------------------------------------------
  s^5 - 112.8 s^4 + 1.154e04 s^3 - 4.698e05 s^2 + 3.224e07 s - 1.265e08

Continuous-time identified transfer function.

Parameterization:
   Number of poles: 5   Number of zeros: 4
   Number of free coefficients: 10
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on frequency response data "data".
Fit to estimation data: 100%
MSE: 5.271e-27
fx >>
```

Figure 3.6.2: MATLAB Command Window

Form command window, Gc could be found.

IV. Result

4.1 Experimental Result from 'PI Control Gain Tuning'

In the first experiment, the voltage is 12 V and the current measured by the multimeter is 280 mA. From equation 3.5.2, steady state error could be calculated.

$$200 - \mathrm{u(t)} = \mathrm{e(t)} \quad \text{[Equation 3.5.2]}$$

Steady state abbreviates as s.s. By varying proportional gain and integral gain, table 4.1.1, which contains 8 trials, could be recorded.

	Kp	Ki	s.s. angular velocity (rad/s)	Power Consumption (Ws)	s.s. error
Trial 1	1	0	187.4824	9.89	12.5176
Trail 2	5	2	201.2205	10.6	1.2205
Trail 3	15	4	198.0807	7.5	1.9193
Trail 4	30	6	195.1410	5.59	4.859
Trail 5	50	2	212.3317	6.13	12.3317
Trail 6	80	2	205.3158	6.75	5.3158
Trail 7	15	6	195.7245	6.92	4.2755
Trail 8	15	8	198.8496	6.87	1.1504

Table 4.1.1: Experimental Data recorded in the 'PI Control Gain Tuning' Experiment

4.1.1 The Relationship between the control gains, steady state error and power consumption (Task 4):

For power consumption:

1) When increasing K_p alone, shown in trial 2,5,6, its power consumption will decrease by 42.17% then increase by 10.11%. There's a valley point for the lowest value of power consumption.
2) When increasing K_i alone, shown in trial 3,7,8, its power consumption will decrease by 7.73% then increased by 0.72%. The value of power consumption keeps decreasing with sharp and slight trend respectively.
3) When increasing K_p and K_i at same time, shown in trial 1,2,3,4, its power consumption increases by 7.18%, decrease by 29.25%, followed by decreasing by 25.47%. There's a peak point for the highest value of power consumption. The last value maybe a valley point for the lowest value because it decreased continuously without increasing trend.

For steady state (ss) error:

4) When increasing K_p alone, shown in trial 2,5,6, its ss error increases by 910.38% then decreased by 56.89%. There's a peak point for the highest value of steady state error.
5) When increasing K_i alone, shown in trial 3,7,8, its ss error increases by 122.76% then decreased by 73.09%. There's a peak point for the highest value of steady state error.
6) When increasing K_p and K_i at same time, shown in trial 1,2,3,4, its ss error decreases by 90.25%, then increased by 57.26%, 153.17%. There is one

valley value for the lowest steady state error.

Detail of data manipulation could be found in part Appendix.

4.2 Experimental Result from ‘Frequency Response’

In the second experiment, the voltage maintains 12 V, the frequency input from 2-10 Hz, table 4.2.1, which contains 5 trials, could be recorded.

Frequency (Hz)	Actual Amplitude	Amplitude Ratio	Period (S)	Time Difference (mS)	Phase Angle (rad)
2	98.605	0.9861	0.537	28.6	-0.3464
4	64.45	0.6445	0.215	45.8	-1.3391
6	45.4	0.454	0.216	36.8	-1.0721
8	33.1	0.331	0.215	31.3	-0.915
10	47.65	0.4765	0.106	25.5	-1.508

Table 4.2.1: Experimental Data Recorded in ‘Frequency Response’ Experiment

For amplitude ratio:
Amplitude ratio equals actual amplitude of velocity output divides amplitude of reference velocity of the motor, i.e. 100.
When increasing the frequency, it keeps decreasing by 34.64%, 29.56%, 27.09% and increased by 43.96% respectively in 5 trials.
For period:
When increasing the frequency, it decreases by 60%, 3.24%, 5.36%, 50.43% respectively in 5 trials. Frequency keeps decreasing with sharp, slight, slight and sharp trend.
For time difference:
Time difference equals to time distance between the ‘reference velocity’ curve and ‘velocity output’ curve of the motor.

Detail for calculation: It equals to peak distance between output curve and reference curve, then plus valley distance between output curve and reference curve. Finally, taking the value to divide by 2 then **transpose the unit to second or milliseconds which will be described in *Appendix***.

When increasing the frequency, it increases by 60.3%, then keep decreasing by 19.7%, 15.1%, 18.3%. Maybe there’s lots of error existing in the first trial. The value of time difference in the first trial should be greater than the second trial’s.
For phase angle:
Assume the function of velocity output below:

$$y = A\sin(\omega t_1 + \varphi) + B$$ [Equation 4.2.1]

For the ideal function of velocity output,

$$y = A\sin(\omega t_2 + \varphi) + B$$ [Equation 4.2.2]

Combining 2 equations together.

$$y = A\sin(\omega t_1 + \varphi) + B = A\sin(\omega t_2 + \varphi) + B$$
$$\sin(\omega t_1 + \varphi) = \sin(\omega t_2 + \varphi)$$

When the curve moves to right, there's always a negative sign. Besides, the delay of signal represents as negative.

$$\text{Phase Angle Difference} = -[\omega\,(t_1 - t_2)] = -\frac{2\pi}{Period} \times Time\ Difference$$ [Equation 4.2.3]

For example, when frequency was 2 Hz, phase angle difference

$$= -\frac{2\pi}{Period} \times Time\ Difference = -\frac{2\pi}{0.537} \times 0.0286 = -0.335\ \text{rad}$$

There's no law about changing of phase angle.

Detail of data manipulation could be found in part Appendix.

4.3 Bode Diagram (Task 6)

After running MATLAB program, fig 4.3.1 shows Bode diagram:

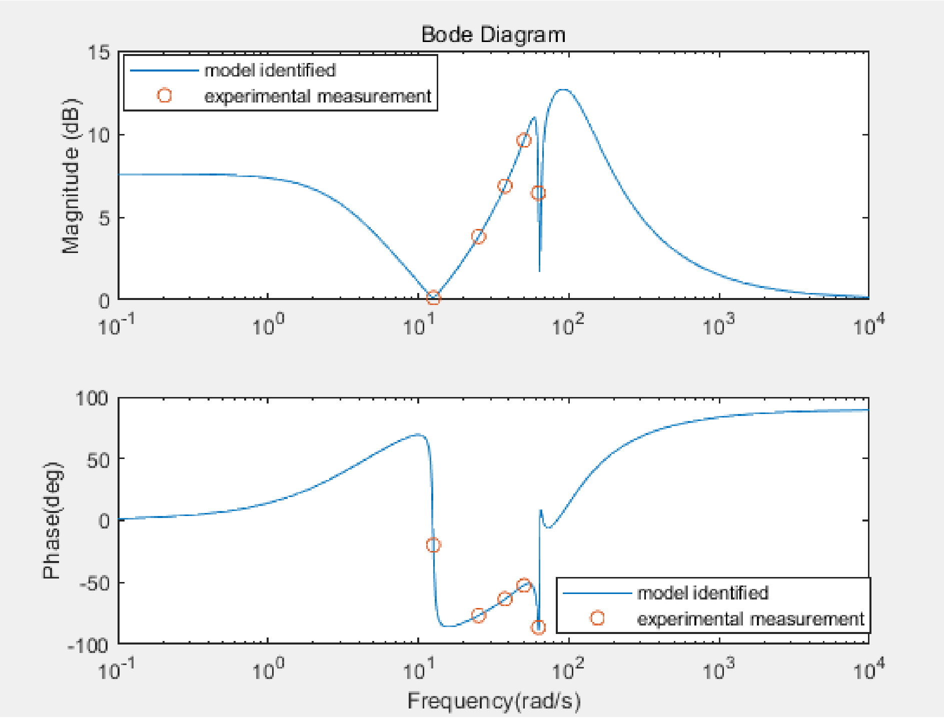


Figure 4.3.1 Bode Diagram

From MATLAB's command window, shown in fig 3.7.2, G_c could be found:

$$G_C = \frac{-1499s^4 - 93.64s^3 - 6.389 \times 10^6 s^2 + 3.648 \times 10^6 s - 9.62 \times 10^8}{s^5 - 112.8s^4 + 1.154 \times 10^4 s^3 - 4.698 \times 10^5 s^2 + 3.224 \times 10^7 s - 1.265 \times 10^8} \quad \text{[Equation 4.3.1]}$$

4.4 Block Diagram and Calculation for G_M (Task 7,8)

Fig 4.4.1 shows the block diagram for G_M. The overall block diagram for the system could be represented as: signal input to the identified system, goes through and times 'transfer function for motor G_M' and 'transfer function for PI controller G_I'. Then signal would go through two paths. The first part times a factor of one then goes back to origin which would be reduced by signal input, whose difference in the process is regarded as error. The second part would export. From input to output, the identified system's transfer function could be represented as G_C.

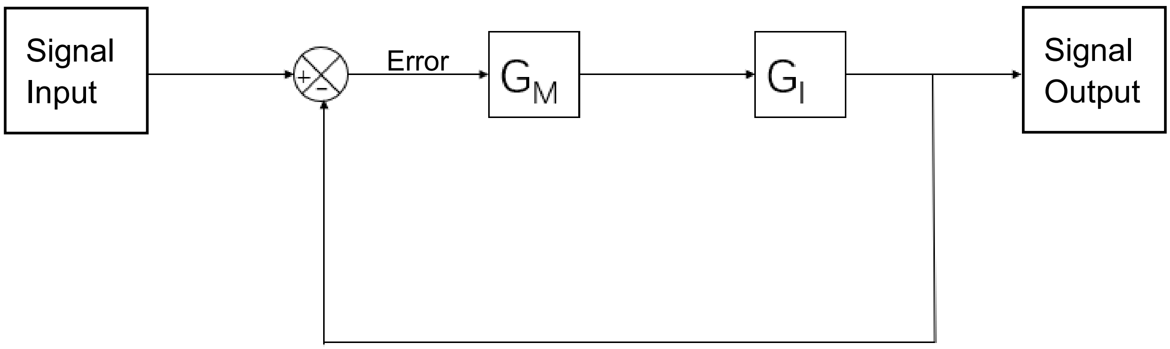


Figure 4.1.1: Block Diagram for G_M

From block diagram above, transfer function of identified system could be represented as:

$$G_C = \frac{G_M G_I}{1 + G_M G_I} \quad \text{[Equation 4.4.1]}$$

From equation derived, G_M could be rearranged:

$$G_M = \frac{G_C}{G_I - G_C G_I} \quad \text{[Equation 4.4.2]}$$

Transfer function G_I has been acquired before:

$$G_I(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \quad \text{[Equation 3.5.6]}$$

$$G_M(s) = \frac{G_C}{G_I - G_C G_I} =$$

$$\frac{\frac{-1499s^4-93.64s^3-6.389\times10^6s^2+3.648\times10^6s-9.62\times10^8}{s^5-112.8s^4+1.154\times10^4s^3-4.698\times10^5s^2+3.224\times10^7s-1.265\times10^8}}{(K_p+\frac{K_i}{s})-(K_p+\frac{K_i}{s})\frac{-1499s^4-93.64s^3-6.389\times10^6s^2+3.648\times10^6s-9.62\times10^8}{s^5-112.8s^4+1.154\times10^4s^3-4.698\times10^5s^2+3.224\times10^7s-1.265\times10^8}} =$$

$$\frac{1499s^5+93.64s^4+6.389\times10^6s^3-3.648\times10^6s^2+9.62\times10^8s}{(K_ps+K_i)(s^5+1.39\times10^3s^4+1.163\times10^4s^3+5.919\times10^6s^2+2.859\times10^7s+8.355\times10^8)}$$

$$G_M(s) = \frac{1499s^5+93.64s^4+6.389\times10^6s^3-3.648\times10^6s^2+9.62\times10^8s}{(K_ps+K_i)(s^5+1.39\times10^3s^4+1.163\times10^4s^3+5.919\times10^6s^2+2.859\times10^7s+8.355\times10^8)}$$ [Equation 4.4.3]

V. Discussion

5.1 Discussion about Steady State Error of PI Controller

Ideally, increasing proportional gain K_p alone will make the steady state error of response decrease. Increasing integral gain K_i alone will eliminate the steady state error of the response. (Aleciatore and Histand, 2012)

But from the result in part 4.1, increasing K_p alone, steady state error increases by 910.38% then decreases, the result from the first trial maybe incorrect. Ideally, the value should be great than second trial's. Increasing K_i alone, systematical steady state error increases followed by decreasing. Ideally, it the effect of elimination should be observed.
Firstly, it is hard to determine the relationship only within 3 trials. Experimental relationship could be determined within a lot of trials.
Secondly, due to the defect of experimental apparatus, such as the lack of filter, a great deal of error existed in the experiment.
Not only the relationship of steady state error between control gains, but also power consumption is hard to determine with limited experimental condition. Improving the performance of experimental apparatus, doing more experiments to record more data can obtain better conclusion.

5.2 Discussion about Power Consumption

In lab instruction, it is to use the multimeter to measure the current through the motor for once. If breaking this law, measuring the current in each trial when tuning proportional gain and integral gain, there's difference between each trial. Here's a couple of value of current when tuning the gains, and the unit is A.
0.272, 0.315, 0.295, 0.295, 0.32, 0.656, 0.58
The first five values are around 0.30 A, from which last two deviates. From that, the change of current cannot be eliminated. It's better to measure the current

in each trial to input in LabVIEW to measure the power consumption.

5.3 Discussion about Phase Angle

Ideally, frequency and period of input signal and output signal should equal with each other. However, there's slightly difference between them. Period of output is used to calculate phase angle, which is affected by the period difference. This effect may involve to and result in the error.

5.4 Discussion about vibration of motor

In both experimental, due to the defect of motor and PI controller, the fitting waveform of output velocity will vibrate such as figure fig 5.4.1.

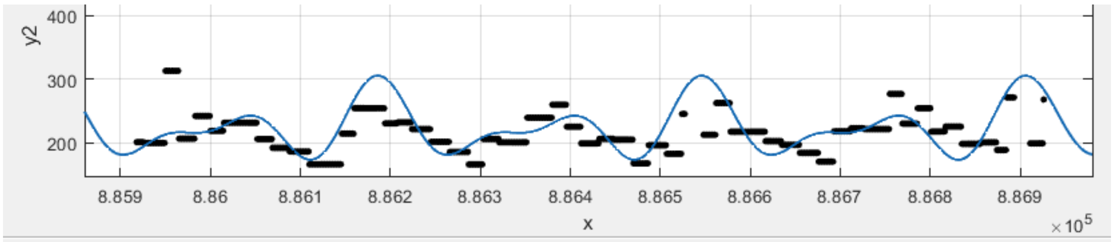


Figure 5.4.1: partial view of curve fitting for output velocity with 10 Hz

When introducing DC motor and PI controller to analyze, their defect is hard to eliminate. Experimental results draw a conclusion that when increasing frequency, fitting waveform of output velocity, the effect of vibration will get fierce. Taking data from valley and peak is not suitable for this case, which should be replaced with better model.

VI. Conclusion

In this experiment, experimental apparatus and block diagrams in LabVIEW software was provided. DC motor with optical encoder was used to obtain difference performance such as motor speed and steady state error of PI controller. Tuning proportional gain, integral gain (in the first experiment) and frequency (in the second) experiment could acquire difference performance of motor. With difference phase angle, amplitude ratio and frequency, system could be identified. Bode diagram, used to analyze frequency response of the system, could be plotted by MATLAB.

There's error and defect existing in this experiment, such as lack of apparatus, i.e. filter, etc, lack of enough data, results in hard to determine relationship between motor performance and control gains. Improving conditions may give better conclusion.

VII. Reference

1. Onwubolu, G. 2005. *Mechatronics Principles and Applications*. Oxford: Elsevier Butterworth-Heinemann.
2. Alciatore, D and Histand, M. 2012. *Introduction to Mechatronics and Measurement Systems.* 4th Edition. New York: McGrew-Hill

VIII. Appendix

8.1 Process of manipulation data in the first experiment:

1) For steady state angular velocity: After exporting data from LabVIEW when the motor was in steady state, use Excel to select all data and find average value for velocity output.
2) For change of steady state error, power consumption, use Excel to write the code: Percentage of change equals the difference between two value dividing the value of former one as fig 8.1.1 shown. To distinguish 3 kind of increasing, green color stands for increasing proportional gain and integral gain at same time, yellow and blue for only increasing proportional and integral gain respectively.

=(L3-L6)/L3

G	H	I	J	K	L	M	N	O	P	Q	R	S
		Kp	Ki	s.s. angular velocity (rad/s)	Power Consumption (Ws)	s.s. error	%power	%error	%power	%error	%power	%error
	Trial 1	1	0	187.482	9.89	12.5176					-0.07179	0.902497
	Trail 2	5	2	201.221	10.6	1.2205	=(L3-L6)/L3	-9.1038099			0.292453	-0.57255
	Trail 3	15	4	198.081	7.5	1.9193			0.07733333	-1.22764	0.254667	-1.53165
	Trail 4	30	6	195.141	5.59	4.859						
	Trail 5	50	2	212.332	6.13	12.3317	-0.1011419	0.5689321				
	Trail 6	80	2	205.316	6.75	5.3158						
	Trail 7	15	6	195.725	6.92	4.2755			0.00722543	0.730932		
	Trail 8	15	8	198.85	6.87	1.1504						

Figure 8.1.1: Data manipulation for change of error, power in the first experiment

8.2 Process of manipulation data in the second experiment:

3) For reference velocity, it is suitable to manipulate data in Excel. For example, f=2Hz, clear data for reference speed could be found in Excel as fig 8.2.1.

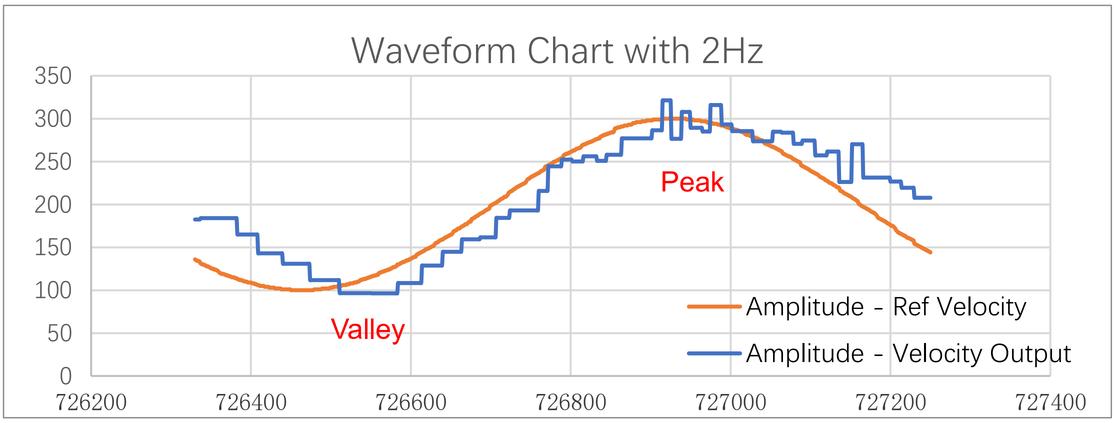


Figure 8.2.1: Waveform chart with 2Hz in the second experiment

4) But as shown in figure above, it's hard to determine the fitting curve for velocity output. Curve Fitting, one of MATLAB's APP, is useful to analyze it. Write code for reading the file as fig 8.2.2:

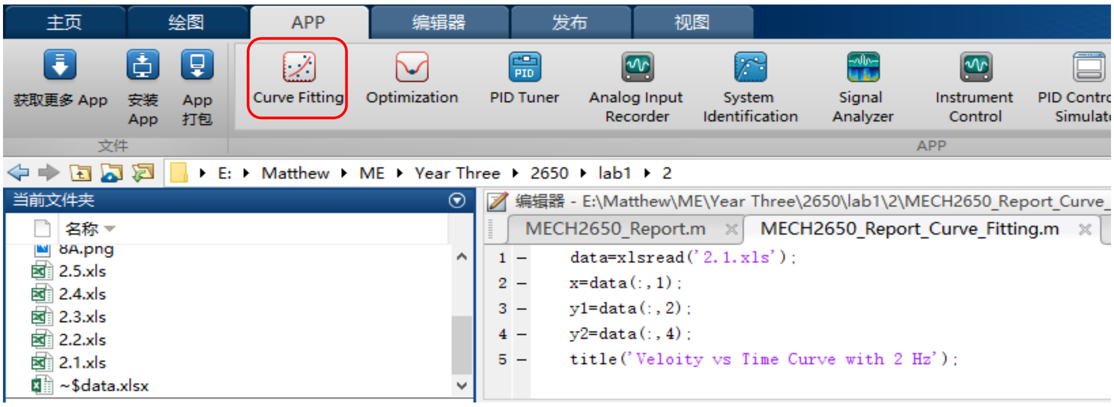


Figure 8.2.2: MATLAB code for reading a file in the second experiment

Then click 'Curve Fitting', set Fourier equation to 4 terms of number generally, suitable fitting curve could be determined as fig 8.2.3. Peak and valley's point data could be captured. It's useful and helpful to capture the output data to calculation amplitude, time difference, etc.

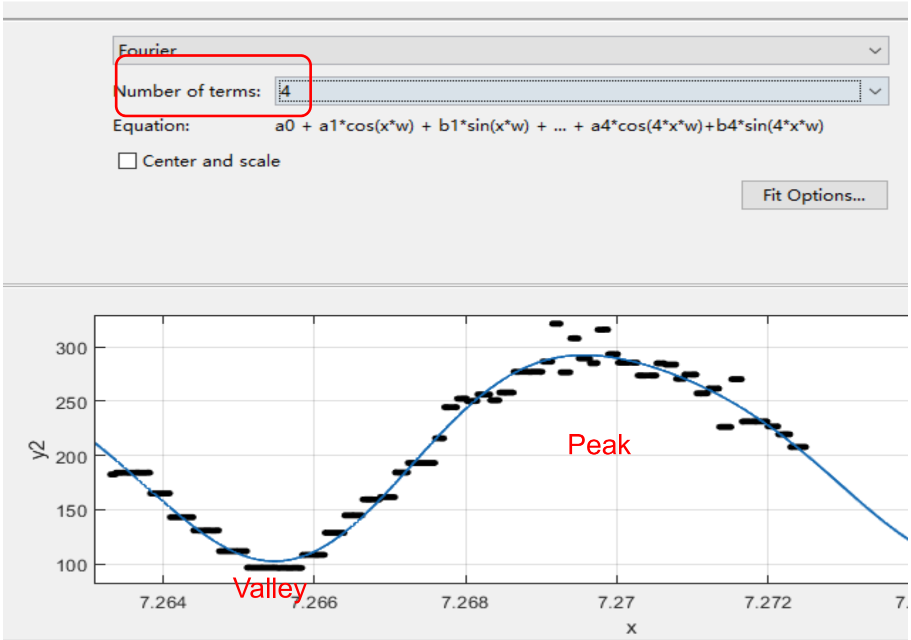


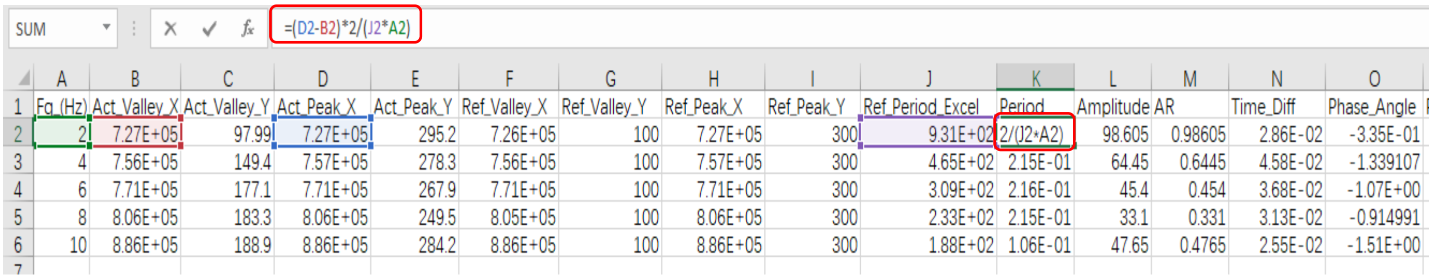
Figure 8.2.3: MATLAB Curve Fitting for determine data in the second experiment

8.3 Important: Transpose from the abscissa's unit of Excel to second

5) When exporting the data to Excel, it is easy to find the unit used in x axis of Excel is not seconds, or milliseconds. It is the sample number of myRIO. For example, unit of time difference and period need to be transposed to seconds or milliseconds. Take 2Hz as example, 2Hz stands for the period is 0.5s. A completed period's sample numbers could be determined, i.e. 2 times the distance between peak to valley. A simple equation below could be used to transpose the unit to second:

$$\frac{Period\ of\ reference\ (s)}{\text{distance between peak to valley of reference}} = \frac{Period\ of\ output\ (s)}{\text{distance between peak to valley of output}} \quad \text{[Equation 8.3.1]}$$

Data in the second experiment and calculation taking 2Hz as example in Excel is shown in fig 8.3.1:



SUM | =(D2-B2)*2/(J2*A2)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Fq (Hz)	Act_Valley_X	Act_Valley_Y	Act_Peak_X	Act_Peak_Y	Ref_Valley_X	Ref_Valley_Y	Ref_Peak_X	Ref_Peak_Y	Ref_Period_Excel	Period	Amplitude	AR	Time_Diff	Phase_Angle
2	2	7.27E+05	97.99	7.27E+05	295.2	7.26E+05	100	7.27E+05	300	9.31E+02	2/(J2*A2)	98.605	0.98605	2.86E-02	-3.35E-01
3	4	7.56E+05	149.4	7.57E+05	278.3	7.56E+05	100	7.57E+05	300	4.65E+02	2.15E-01	64.45	0.6445	4.58E-02	-1.339107
4	6	7.71E+05	177.1	7.71E+05	267.9	7.71E+05	100	7.71E+05	300	3.09E+02	2.16E-01	45.4	0.454	3.68E-02	-1.07E+00
5	8	8.06E+05	183.3	8.06E+05	249.5	8.05E+05	100	8.06E+05	300	2.33E+02	2.15E-01	33.1	0.331	3.13E-02	-0.914991
6	10	8.86E+05	188.9	8.86E+05	284.2	8.86E+05	100	8.86E+05	300	1.88E+02	1.06E-01	47.65	0.4765	2.55E-02	-1.51E+00
7															

Figure 8.3.1: Recorded data and transposing unit to second in the second experiment