Control Moment Gyroscope

Introduction

For this lab, there are three experiments, each with separate sub-parts to perform. For experiment 1, the goal is to determine 3 unknown inertias (the inertia of Body C around the Y axis is denoted J_C, Body A around the Z axis has inertia K_A, and Body C around the X axis has inertia I_C). Experiment 2 consists of finding the hardware gains of the system k_{g1} and k_{g2}. Finally, Experiment 3 illustrates the unique properties of gyroscopes, specifically nutation and precession.

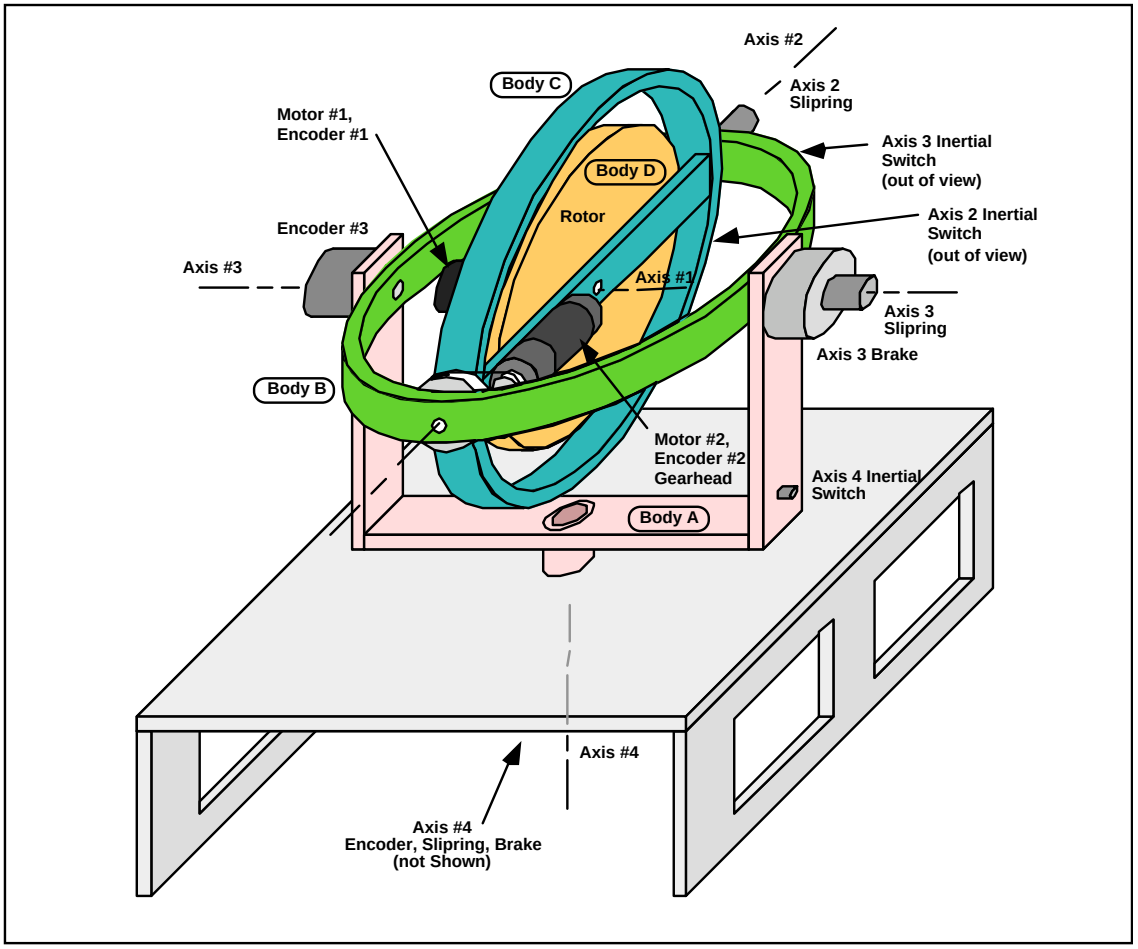

Figure 1: ECP Control Moment Gyro Experiment

Hardware

The Control Moment Gyro Experiment consists of the CMG mechanism and its actuators and sensors. The design features two high torque density (rare earth magnet type) DC servo motors for control effort transmission, high resolution encoders for gimbal (rotating ring) angle feedback, and low friction slip rings for signal and motor power transmission across all gimbals. It also includes inertial switches for high gimbal speed detection and

safety shutdown and electromechanical brakes to facilitate changing dynamic degrees of freedom as well as securing the system during safety shutdown.

The plant, shown in Figure 1, consists of a high inertia brass rotor suspended in an assembly with four angular degrees of freedom. The rotor spin torque is provided by a rare earth magnet type DC motor (motor#1) whose angular position is measured by a 2000 count per revolution optical encoder (encoder #1). The motor drives the rotor through a 3.33:1 reduction ratio that amplifies both the torque and encoder resolution by this factor. In this laboratory, the other degrees of freedom will be locked using the electromagnetic brake switches on the control box. Thus, the system has a single motor/encoder for position and speed control, the most frequently observed applications in practice.

Safety

Read the safety information found in Appendix B on the course webpage

When implementing a controller, use a ruler or other non-sharp object to nudge the various elements of the system to verify that there is no unstable control condition and that the system is safe to manipulate. In these experiments, do not move axes 3 and 4 while their brakes are engaged. This leads to premature wear out of the brakes.

Hardware/Software Equipment Check

Before starting the lab, confirm that the hardware is working by performing the following steps:

Step 1: With power switched off to the Control Box, enter the ECP program by double clicking on its icon. One should see the Background Screen at this point. Gently rotate the inner gimbal ring (the one that encloses the brass rotor). Observe changing readings in the Encoder 2 position and possibly small changes in the Encoder 1 (rotor) position. The Control Loop Status should indicate "OPEN" and the Motor 1 Status, Motor 2 Status, and Servo Time Limit should all indicate "OK".

Step 2: Now press the black "ON" button to turn on the power to the Control Box. Notice that the green power indicator LED is lit, but the motors should remain in a disabled state. Turn off the Axis 3 and 4 Brakes via the toggle switches on the Control Box and *safety check* the controller as per the instructions in Appendix B on the course website. Now move Axes 3 and 4. Observe the corresponding Encoder position values change on the Background Screen. Encoder 1 position will usually change as one moves axis 3.

Experiment 1: System Identification; Inertia Measurement

In the following tests, students will measure three moments of inertia using the principle of *conservation of angular momentum*. The remaining moments of inertia are provided, but could be measured experimentally using this and other fundamental principles.[1]

Read this section before beginning the procedure below. This will clearly explain which axis corresponds to which inertia in the procedures. Follow the procedure to determine the moments of inertia about the specified axes, and thus complete Table 1.2.

Note: This lab will use conservation of angular momentum to determine the moments of inertia:

$$J_1\omega_{1o} + J_2\omega_{2o} = J_1\omega_{1f} + J_2\omega_{2f} \qquad \text{(A)}$$

where J_1 and J_2 are the inertias of two bodies rotating about a common axis, and the subscripts "*o*" and "*f*" denote two time instants, say original and final.

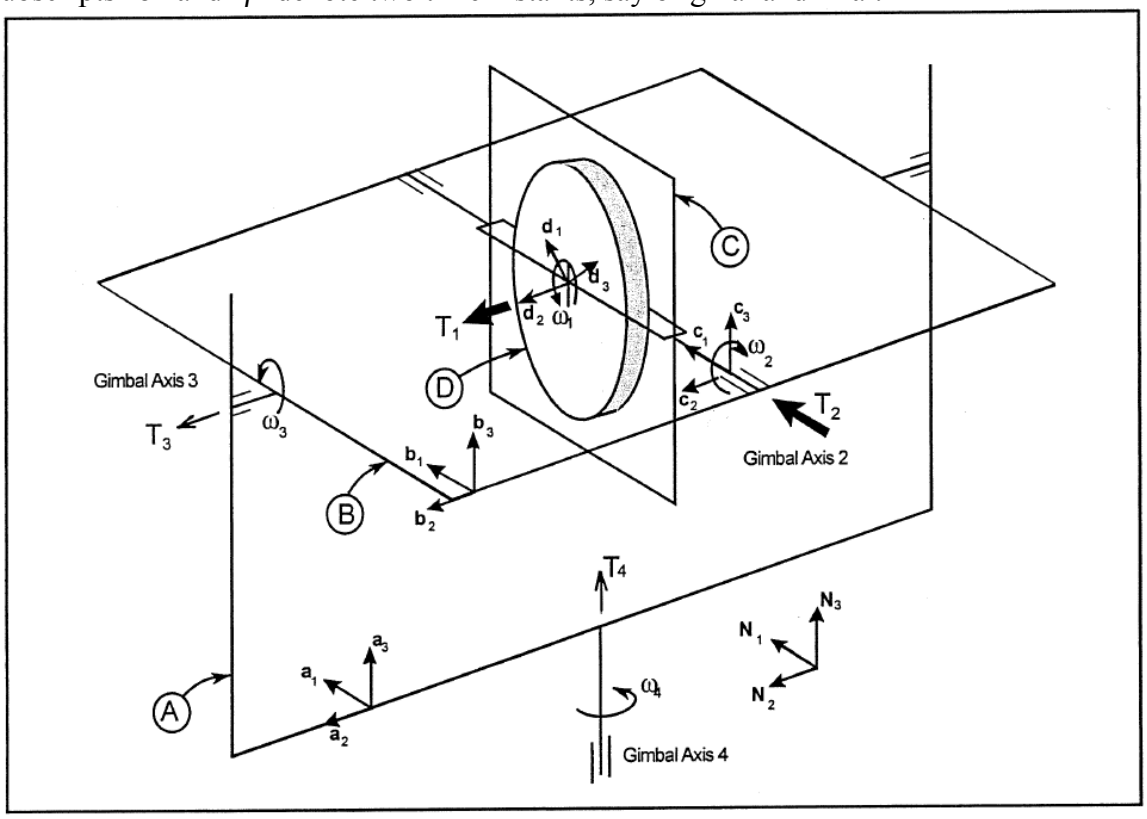

Figure 2: Coordinate System Orientations

[1] The inertia values measured here are those that are less amenable to mass property calculation (e.g. via mechanical geometries, material densities and component weights) and vary the greatest from apparatus to apparatus.

Figure 2 above is the orientation setting with which inertias of various bodies are defined. In the figure, four bodies are labeled, A, B, C, and D. Also, three coordinate directions are labeled, 1, 2 and 3; these coordinates or axes are further named I, J and K and used to define various inertias in Table 1-2 on the next page.

In the first part of the lab, students are expected to determine inertia J_C, which is the inertia of body C about coordinate-direction 2 defined in Fig. 2. In this procedure, one may assume that body D rotates as one piece and Bodies C and B together rotate as one piece. Both rotate about coordinate-direction 2 or axis J. The inertia of Body D about direction 2 is given in Table 1-2 as J_D. The inertia of Body B about the same axis is also given in the table as J_B. Now, if one substitutes J_D as J_1 in Eq. (A), then J_2 in the same equation would be equal to J_B+J_C. Thus, one can determine what J_C would be using the equation along with the measurement data of the angular velocities used in this equation.

In this lab, students should repeat the process above for two similar experiments to determine K_A and I_C and enter the results in Table 1-2 below. There are step-by-step procedures that follow that explain the process in detail.

Note that the measured sensor counts must be converted to radians in the calculations. Please see Table 1-1 below for encoder gains and divide the experimental values by $\mathbf{k}_{ei}$ to obtain motion data in radians. Also, if the calculations give any negative inertia, there is a very strong possibility that one has used an incorrect definition in the inertias of equation (A).

Table 1-1: Encoder Gain Measurements

Axis Number (Encoder Number)	Output / Rev. (counts/rev)	Gain, k_{ei} (counts/rad)
1	**6667**	**1061*32**
2	**24400**	**3883*32**
3	**16000**	**2547*32**
4	**16000**	**2547*32**

Important: Always change sampling time before implementing the control algorithm. *To edit the control algorithm, the current controller must be aborted first.*

Experiment 1a: (J_C)

In this part of Experiment 1, the inertia of Body C will be found experimentally.

1. Turn on the hardware and software. Start the ECP program. Select *Abort Control* on the Background Screen to disable any controller that may be still running on the DSP board from a previous user. Turn on power to the Control Box. Use a ruler or other non-sharp object to nudge the various elements of the system to verify that there is no unstable control condition and that the system is safe to manipulate. In this and all subsequent experiments, do not move axes 3 and 4 while their brakes are engaged. This leads to premature wearout of the

brakes. One should see changes in the encoder counts on the background screen (for the axes that are not locked) as the apparatus is moved.

2. Prepare for the first test. Setup the mechanism as shown in Figure 1-2a. The Axis 3 and 4 brakes are turned on and off via toggle switches on the Control Box. The Axis 2 Virtual Brake is engaged via a button on the Executive Program background screen. (This brake is affected via a simple linear control loop that is closed in firmware resident on the DSP board and uses Motor #2 and Encoder #2 as the sensor and actuator.) Select *Zero Position* (Command menu) to zero the incremental encoder values at these gimbal positions.

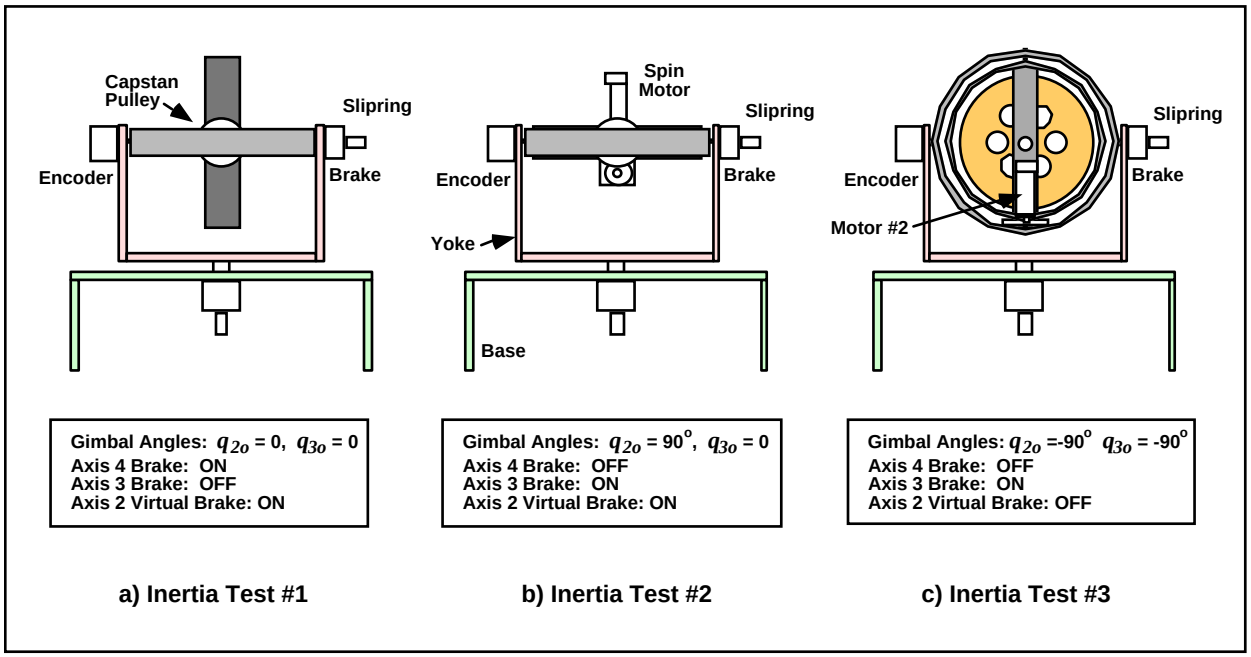

Figure 1-2. Configurations For Moment of Inertia Tests

3. Implement an input. Write a simple real-time algorithm to activate Motor #1 (i.e. put control effort values on the DAC) with a *Control Effort* equal to the (*Commanded Position)*/32.[2] Use the global real-time variables "`control_effort1`" and "`cmd1_pos`" for this purpose. (The *Commanded Position* values are subsequently entered via Setup Trajectory to provide the input to the system.) Review the algorithm

[2] The "/32" factor accounts for a firmware gain that multiplies all commanded position and encoder signals by 32 for increased internal resolution. These control effort counts are converted to a voltage via a digital-to-analog converter (DAC), then to a current via the servo amplifier, to a torque by the motor, and finally to a different torque magnitude via the gear reduction. The scaling of all of these transformations affect the system gain and will be examined in more detail in the section that follows. See also Chapter 4 for a description of the control hardware and software functionality.

with the instructor or laboratory supervisor before proceeding to the next step.

```
begin
control_effort1 = cmd1_pos/32
end
```

4. Implement this algorithm using the following steps:

   a) Enter Setup Control Algorithm via the Setup menu. Set the *Sample Period* to T_S = 0.00442 sec. and select *Edit Algorithm*. This opens up the control algorithm editor. If the editor contains any text select *New* under *File.*

   b) Type in the control algorithm. Select *Save As…* and choose an appropriate name and directory to save this algorithm in. Close the editor by either selecting *Save Changes and Quit* or simply clicking on the upper right hand button.

   c) Stay well clear of the apparatus when initially performing the next step. Select Implement Algorithm to begin immediate execution of the

   algorithm. If all is well, there should be no motion of the system.

   In this and all subsequent experiments immediately after implementing a controller, perform a safety-check of the system using a ruler or other non-sharp, slender object

5. Set up the input and data acquisition system. Go to Trajectory 1 Configuration under the Setup menu and deselect *Unidirectional Moves* (this enables bi-directional trajectories). Enter Impulse[3] and specify a *Amplitude* of 16000 counts, a *Pulse Width* of 1000 ms, a *Dwell Time* of 0 ms, and 2 *repetitions* (this prepares the controller board to input a 16000 count positive-going step followed immediately by a 16000 count negative-going one.) Select *OK,* successively and enter Setup Data Acquisition (Setup menu). Specify ***Encoder 1 Position, Encoder 2 Position, Encoder 3 Position, Encoder 4 Position, Control Effort 1*** and ***Control Effort 2*** as data to be acquired with a *Sample Period* of 2 servo cycles.

[3] Here the "*Impulse*" general form is used to generate a step-like shape. The impulse dialog box is may be used for step-like forms when the pulse-width is specified to be relatively long.

6. Run the test. Go to Execute (Command menu) and verify that the apparatus is in the configuration of Figure 6.1-2a. Select *Normal Data Sampling* and *Execute Trajectory 1 Only* and then *Run.* One should see the rotor spin up then slow down while the inner assembly (bodies B, C, and D) rotates about Axis 3. Is the motion of the rotor and that of the inner assembly in the same direction or opposite directions? Select OK once the data from the maneuver has been uploaded.
7. Plot the data in MATLAB. Be sure to plot **Encoder 1 Velocity** and **Encoder 3 Velocity** on the same plot. (Note, the ECP software only exports position data) and clearly label each with different lines and a legend.

Does the system eventually come to a stop? Why or why not? What does this illustrate about the theoretical conservation of angular momentum in the system over long time periods? Based on these observations, should one use a relatively short data window (seconds) for the above analysis, or a long data window (tens or hundreds of seconds)?

The final report is expected to include:

A diagram identifying the control elements and signals in the Gyro Experiment. Hint, Use Figure 1 or 2 above.

Sensor:	Actuator:
Controller:	Reference Input:
Actuator Output:	System Output:

One (1) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- Plot of Encoder 1 Velocity and Encoder 3 Velocity

One (1) MATLAB script file that converts the position data into velocity data. This will be needed for subsequent experiments. Please place this in an Appendix of your lab report.

For all the questions highlighted, the questions should be copied and pasted into the student's lab report and explicitly answered immediately thereafter.

Experiment 1b: (K_A)

In this part of experiment 1, the value for K_a will be determined experimentally

8. Prepare the hardware for the next test. Setup the mechanism as shown in Figure 1-2b. In order to improve the quality of the final result, the following procedure is recommended in setting the initial orientation of the yoke v. the base (i.e. q_4). Rotate the yoke relative to the base using a very light touch to find a location where there is least friction (some residual friction from the brake may exist in some locations). Although

typically small, any friction may be detected by giving the yoke a very small initial velocity and observing how rapidly it stops. The yoke should be positioned to minimize the travel in any zone of friction for motion in a counterclockwise direction. (E.g. positioned just beyond a zone of friction when moving in a counterclockwise direction).

9. Reconfigure the input given to the system. Go to Trajectory 1 Configuration under the Setup menu, select *Impulse* and change the *Pulse Width* to 2000 ms.. Leave all other parameters the same as in Step 5 above.
10. Repeat Step 6 to execute the input to the system. One should see the rotor spin and the remaining assembly rotate about the base (Encoder #4). Note the nature of the motion.
11. Plot in MATLAB **Encoder #1 and Encoder #4 velocity** data on the same plot. Again, be sure to clearly indicate which data corresponds to which line.

The final report is expected to include:

One (1) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- Plot of Encoder 1 Velocity and Encoder 4 Velocity

Experiment 1c: (I_c)

In the final part of experiment 1, the final parameter will be determined.

12. Prepare the hardware for this test. Setup the mechanism as shown in Figure 1-2c. Set the yoke position as described in Step 8.
13. Edit the previous algorithm to output `cmd1_pos` to `control_effort2`. (i.e. put *Trajectory 1* inputs on DAC2 to drive Motor 2). Select *Save Changes and Quit* to exit the editor. .
14. Stay clear of the apparatus and select Implement Algorithm to begin immediate execution of the algorithm. Safety check the system. If all is well, there should be no motion of the system.
15. Change the input to the system. Go to Trajectory 1 Configuration, select *Impulse* and change the *Pulse Width* to 200 ms. Leave all other parameters the same as in Step 5 above.
16. Repeat Step 6 and run the test. One should see the inner gimbal ring rotate relative to the outer ring and the remaining assembly rotate about the base (Encoder #4). The inner ring will likely contact the limit switch at the end of the maneuver causing the Control Box to power down. This is normal. If it occurs before completing a satisfactory test, simply re-power the Control Box. It is also possible that the limit

switches are contacted during the initial (first 400 ms) portion of the maneuver. If this occurs, reduce the *Pulse Width* duration in Step #15 (to say 150 ms.) and repeat the remainder of the procedure.

17. Plot in MATLAB **Encoder #2 and Encoder #4 velocity data**.

Note, this table is for the student to complete, and is used for later parts of the lab.

Table 1-2. Moment of Inertia Data		
Body	**Inertia Element**	**Value (kg-m^2)**
A	K_A	
B	I_B	*0.0119*
	J_B	*0.0178*
	K_B	*0.0297*
C	I_C	
	J_C	
	K_C	*0.0188*
D	I_D	*0.0148*
	J_D	*0.0273*

The final report is expected to include:

One (1) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- Plot of Encoder 2 Velocity and Encoder 4 Velocity

One (1) Excel table

- Table 1.2, completed

Calculations showing answers for Table 1-2. Hint, you will need to use the values already found in Table 1.1 and 1.2. Be sure to solve this symbolically before plugging in numbers.

Note: this would be best done in MATLAB to prevent algebra answers.

Experiment 2 Control Effort Gain Measurement

In the following tests, the control effort gain for the two plant inputs is measured using Newton's second law (in its rotational form) and the gyroscopic cross product. These relationships have the well-known forms

$$T = J\ddot{q} \qquad \text{(Equation 2-1)}$$

$$T = \omega \times H \qquad \text{(Equation 2-2)}$$

where in the first expression, T is the applied torque, J is the moment of inertia about an axis parallel to T, $\ddot{q}$ is the angular acceleration of J about that axis. In the second expression, H is the momentum of the body, ω is the angular velocity associated with change of direction of the vector H, and T is the applied torque necessary to change the direction of H.

Important:
To edit the control algorithm, the current controller must be aborted first.

Experiment 2a: Control Effort Gain (k_{g1})

In this test, the acceleration of the rotor disk is measured in response to a known control effort signal in the positive and negative directions. Students must measure the resulting accelerations. From the accelerations and rotor inertia value, the applied torque and hence control effort gain may be calculated.

1. Set the hardware for the test Setup the mechanism in the configuration of Figure 1-3a, shown below.
2. Edit the algorithm from Step 13 above to output `cmd1_pos` to `control_effort1`. (i.e. the same as in Inertia tests 1 and 2). Select *Save Changes and Quit* to exit the editor. Select Implement Algorithm and safety check the system.

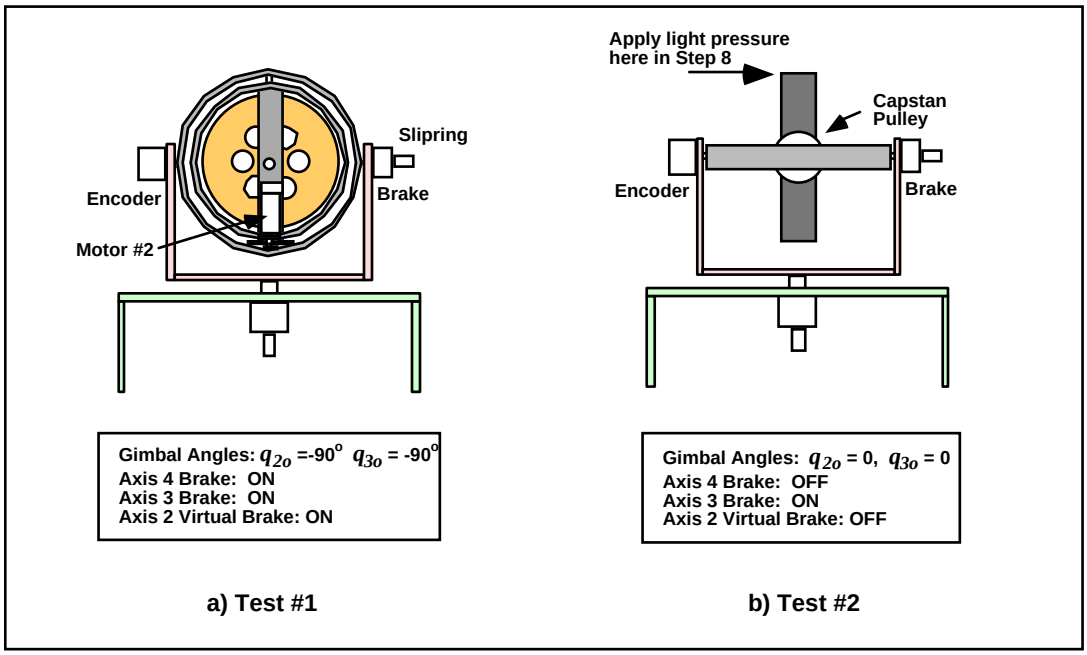

Figure 1-3. Configurations For Control Effort Gain Tests

3. Set up the input. Go to Trajectory 1 Configuration and verify that *Unidirectional Moves* is not selected. Enter Impulse and specify an *Amplitude* of 16000 counts, a *Pulse Width* of 4000 ms, a *Dwell Time* of 0 ms, and 2 *repetitions*.
4. Run the test. Execute this input with *Normal Data Sampling* and *Execute Trajectory 1 Only* selected. One should see the rotor spin up then slow down and possibly reverse motion.
5. In MATLAB, Plot Encoder #1 velocity and control Effort 1 data on the same plot. Use this plot to determine the control gain, being sure to explain the methodology and calculations used to obtained this parameter, including units throughout.

Experiment 2b: Control Effort Gain Test #2 (k_{g_2})

In this test, we measure the torque required to allow the momentum vector, *H*, of the spinning wheel to *precess* at some rate ω according to Equation (2-2).

6. Setup the mechanism in the configuration of Figure 1-3b.
7. Turn on the disk. Stay clear of the mechanism and Initialize Rotor Speed (Command Menu) to 400 RPM. Safety check the system once the rotor has reached approximately 400 RPM as seen on the Background Screen.

8. Read this step carefully before any action. Apply light finger pressure to the upper edge of the inner ring as shown in Figure 1-3b. Note what happens to the motion of the assembly about Axis #4 (the vertical axis at the base of the unit)? This is the fascinating phenomenon of gyroscopic precession. Now apply light pressure in the opposite direction. What happens to the motion of the assembly? For a given amount of finger pressure, does the motion about Axis #4 remain at constant velocity or does it accelerate?

9. Now apply light finger pressure to the side of Encoder#3 to rotate the assembly about Axis #4. What happens to the inner assembly with respect to Axis #2? Do not exceed q_2=± 60 deg. (If a limit switch is contacted due to excessive motion in q_2, repeat steps 6-8.) Return the inner ring to the approximate vertical position (q_2=0).

10. This time, have a fellow student apply light finger pressure to the encoder to cause a small rotation rate about Axis #4 while another student applies a sufficient force on the inner ring (at the location used in Step 8) to keep it approximately vertical (q_2=0). How does the force (and hence torque about Axis #2) required to keep the ring vertical vary with the angular rate about Axis #4?

Please trade places with the fellow students so that each experiences the nature of the applied force. Return the inner ring to the approximate vertical position (q_2=0).

11. Change the data acquisition and control algorithm. In the Setup Control Algorithm dialog box, change sampling period to *Ts*=0.00884 s. Verify that the mechanism is in the configuration of Figure 1-3b. Locate and *Implement* the algorithm "`axis2lock.alg`" as furnished with the system (the code is in the appendix of this lab. You should confirm as students in the past have accidentally edited this!). This simple routine regulates the position of Axis 2 to keep it at the approximate current position. We shall measure the control effort required to do so in the test that follows. Safety check the controller.

12. Rotate the assembly slowly about the vertical axis and note the *Control Effort 2* values displayed. Do not rotate the assembly at a sufficiently high rate to cause *Control Effort 2* to exceed ± 5V. What has changed with regard to resistive torque required to rotate the assembly about the vertical axis (Axis #4) as experienced in Step 9?

13. Change the Impulse amplitude to zero (0) counts and leave all else as specified in Step 3. (This will allow the system to collect data during the "*Execute*" period while maintaining a zero valued reference input)

14. Read this step completely before taking action. Practice the following procedure: Slowly rotate the assembly about the vertical axis at a rate to cause approximately 4-5 V of control effort; release to let the assembly spin freely; wait for a second or two; reverse direction to a

achieve –4 to –5 V control effort, then release again. The goal is to measure the control effort required at motor #2 during the free spinning periods of this maneuver. Once this is consistently achieved, prepare to Execute the *Impulse* maneuver from Step 13. Select *Run*, then perform this procedure.

15. Plot in MATLAB the *Encoder 4 Velocity* and *Control Effort 2* data. Verify that *Control Effort* 2 had at least some values in the range of 4 to 5 V and –4 to –5 V in the portions of the curve when the assembly was spinning freely (the velocity data should be relatively flat during this period). If not, repeat Step 14. Save the final plot.

The final report is expected to include:

Two (2) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- Plot of Encoder #1 velocity and Control Effort 1
- Plot of Encoder #4 velocity and Control Effort 2

For all the questions highlighted, the questions should be copied and pasted into the student's lab report and explicitly answered immediately thereafter.

Experiment 3 Gyroscopic Dynamics: Nutation and Precession

This sequence of experiments illustrates the three dimensional dynamics and control of a gyroscope (these are given by Equation 2-2). Study this equation and the description thereafter.

$$T = \omega \times H \qquad \text{(Equation 2-2)}$$

Explanation of Equation 2-2:
H is the momentum of the body, ω is the angular velocity associated with the change of direction of the vector H and T is the applied torque necessary to change the direction of H.

Important: *Abort the control used in previous step before continuing on.*

Experiment 3a: Nutation: Frequency & Mode Shapes

Follow the procedure to study the nutation motion of the system. Create all the requested plots and include them with this report. Note: When plotting position and velocity on the same figure, place position variables on the left axis and velocity variables on the right axis to avoid scaling issues (hint: "help plotyy").

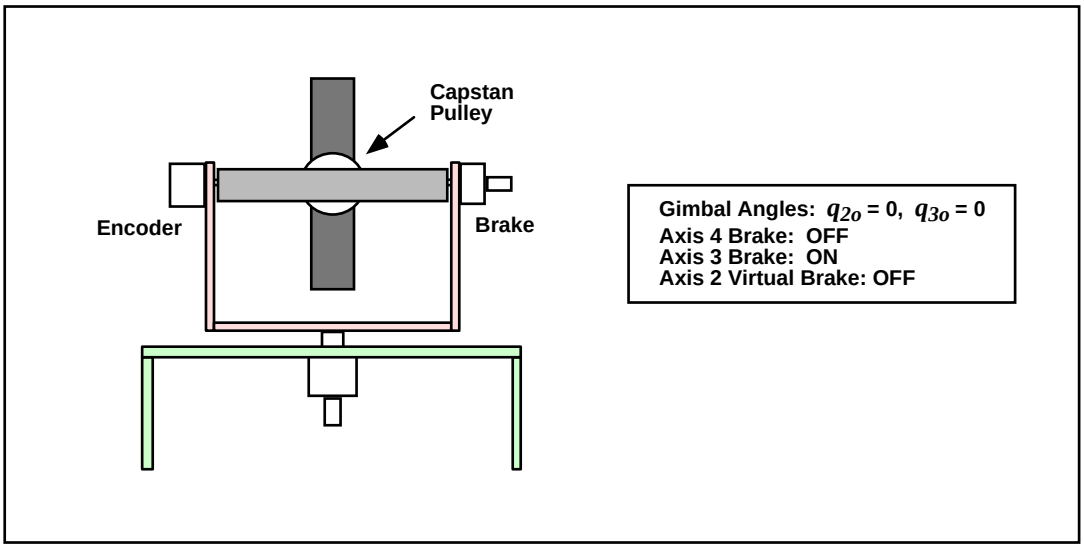

Figure 3-1. Configuration For All Tests In This Section

Procedure

1. Setup the mechanism as shown in Figure 3-1.
2. Write a simple real-time algorithm to activate Motor #2 (i.e. put control effort values on the DAC) with a *Control Effort* equal to the

(*Commanded Position)*/32.[4] Use the global real-time variables "`control_effort2`" and "`cmd1_pos`" for this purpose.

```
begin
control_effort2 = cmd1_pos/32
end
```

3. Set the input to the system. Go to Trajectory 1 Configuration. Enter Impulse and specify an *Amplitude* of 16000 counts, a *Pulse Width* of 50 ms, a *Dwell Time* of 4000 ms, and 1 *repetition* (this prepares the controller board to input a 16000 count positive-going impulse followed immediately by 4 seconds of zero input during which data is collected.
4. Setup Data Acquisition (Setup menu). Specify *Commanded Position 1, Sensor 2 Position, Sensor 4 Position,* and *Control Effort 2* as data to be acquired with a *Sample Period* of 4 servo cycles.
5. Implement the algorithm from Step 2 above with sampling period set to $T_S = 0.00442$ seconds.
6. Initialize Rotor Speed to 200 RPM and zero the encoder positions (Utility menu). Execute the maneuver selecting *Normal Data Sampling* and *Execute Trajectory 1 Only* .
7. Plot the Encoder 2 and Encoder 4 Position data and subsequently the velocity data (which can be obtained from the MATLAB function written in earlier portions of the lab). Note the frequency of the oscillations and the relative amplitude and phase of the Encoder 4 response verses the Encoder 2 response. Save the resulting plots.
8. Turn off the rotor. Disable the rotor speed loop (Command menu). To more rapidly decelerate the rotor, turn off the Control Box. Wait for the rotor to stop (if the Control Box is off, turn it back on at this point). Repeat Steps 6 and 7 for a rotor speed of 400 RPM.
9. Repeat Step 8 for a rotor speed of 800 RPM.

The final report is expected to include:

Three (3) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation. Feel free to use the "subplot" command in MATLAB for this plotting…

- A plot of Encoder #2 and Encoder #4 position at 200 RPM

[4] The "/32" factor accounts for a firmware gain that multiplies all commanded position and encoder signals by 32 for increased internal resolution.

- A plot of Encoder #2 and Encoder #4 position at 400 RPM
- A plot of Encoder #2 and Encoder #4 position at 800 RPM

For all the questions highlighted, the questions should be copied and pasted into the student's lab report and explicitly answered immediately thereafter.

Specifically: comment on the frequency of oscillations and phase between Encoder 2 and Encoder 4 motions. What are the trends and relationships between the three different speed trials?

Experiment 3b: Precession

Follow the procedure below to study the precession motion of the gyroscope. Create all the requested plots and include them in this report.

Procedure

10. Repeat Steps 1 through 6 of Experiment 3a except in Step 3 setup the *Impulse* trajectory for an *Amplitude* of 6000 counts, a *Pulse Width* of 8000 ms, a *Dwell Time* of 0 ms, and 1 *repetition* (this prepares the controller board to input a 6000 count constant input for 8 seconds). The first maneuver should be at 200 RPM and should result in an initial transient series of attenuating nutation oscillations followed by a steady state response. Plot the position data for Encoders 2 and 4 and also their velocity data. Save the plots.
11. Repeat Step 10 for the 400 and 800 RPM cases. Note the change in steady state velocity for Encoder 4 with rotor speed.

The final report is expected to include:

Three (3) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- A plot of Encoder #2 and Encoder #4 position and velocity at 200 RPM
- A plot of Encoder #2 and Encoder #4 position and velocity at 400 RPM
- A plot of Encoder #2 and Encoder #4 position and velocity at 800 RPM

Plot both positions and velocities in the same plot. Put position data on the left axis and velocity data on the right axis. Comment on the change in steady state velocity for Encoder #4 with rotor speed.

For all the questions highlighted, the questions should be copied and pasted into the student's lab report and explicitly answered immediately thereafter.

Experiment 3c: Nutation Damping

First, abort the control that was used in previous experiment. Follow the procedure and implement the following control algorithm (a sample program is saved in the PC at Program Files\ECP Systems\MV\6.2.3.alg):

```
;**************DEFINE USER VARIABLES*********************
#define Ts q1
#define kv q2
#define kdd q3
#define enc2_last q4
;***************INITIALIZE VARIABLES*********************
Ts = 0.00884
kv = 0.005
kdd = kv/Ts
;********************REAL TIME CODE**********************
begin
;CONTROL LAW: output torque = demand – gimbal2 rate feedback
control_effort2 = cmd1_pos/32 – kdd*(enc2_pos – enc2_last)
;UPDATE VARIABLES
enc2_last = enc2_pos
end
```

Important:
To edit the control algorithm, one must abort the current control first.
Experiment with three values of k_v given below and with the angular speeds.

Procedure

12. Augment the algorithm from Step 1 of Experiment 3a to add rate feedback damping at Axis 2. I.e. add a term u_{2damp} of the form

$$u_{2damp} = -k_v\, q_2\, s$$

where k_v is the rate feedback gain, q2 is the position measurement, and s is the derivative operator. Thus, the control effort is just the velocity times the rate feedback gain, k_v. One may use the backwards difference transformation to implement discrete time differentiation according to

$$Velocity = \frac{x(k) - x(k-1)}{T_s}$$

where T_s is the sampling period. Use T_s =0.00884 s. in this experiment.

Have the laboratory supervisor review and approve the algorithms before proceeding.

13. Set the rotor speed to 200 RPM. Beginning with a value of $k_v = 0.005$ Implement the algorithm with T_s =0.00884 seconds.
14. Repeat Step 10 except set the rotor speed at 400 RPM rotor speed. Is there a reduction observed in the nutation mode amplitude?
15. Repeat Steps 13 and 14 for various increasing values of k_v. Do not exceed k_v=0.10, higher values could lead to excessive numerical noise and damage to the system! How are the nutation oscillations affected by increased rate feedback gain? Save the plot for a case where the nutation is well damped.

The final report is expected to include:
Six (6) MATLAB Plot along with titles, labels and legends if necessary that clearly show which plot corresponds to which situation

- Plot of Velocities 2&4 with $k_v = 0.005$ and Rotor Speed of 200 RPM
- Plot of Velocities 2&4 with $k_v = 0.005$ and Rotor Speed of 400 RPM
- Plot of Velocities 2&4 with $k_v = 0.020$ and Rotor Speed of 200 RPM
- Plot of Velocities 2&4 with $k_v = 0.020$ and Rotor Speed of 400 RPM
- Plot of Velocities 2&4 with $k_v = 0.080$ and Rotor Speed of 200 RPM
- Plot of Velocities 2&4 with $k_v = 0.080$ and Rotor Speed of 400 RPM
-

Plot both positions and velocities in the same plot. Put position data on the left axis and velocity data on the right axis. Comment on the change in steady state velocity for Encoder #4 with rotor speed.

For all the questions highlighted, the questions should be copied and pasted into the student's lab report and explicitly answered immediately thereafter.

How are the nutation oscillations affected by increased rate feedback gain?
How does an increase in Rotor Speed affect the amplitude of the oscillations?

Note: because of the number of plots requested in the lab, students will be severely docked points if the plots are not clearly labeled within the report as to which section they pertain to.

Appendix 1: Axis2lock.alg

```
;SET Ts=0.00884 sec
;********** Define User Variables ***************
#define kp q2
#define kd q3
#define enc2_last q4
;************ Initialize Variables ****************
kp=2
kd=8
;********* Real time code which is run every servo period ***
begin
control_effort2=-kp*enc2_pos-kd*(enc2_pos-enc2_last)
enc2_last=enc2_pos
end
```