AE 353 Design Project 1

Ganesh Ravisankar

I. Introduction

A control moment gyroscope (CMG) is a device consisting of a rotor, a motorized gimbal, and a platform to reorient. The purpose of this device is to influence the angular momentum of a vehicle like a spacecraft to change its equilibrium state and hence its attitude. CMGs are able to influence the net angular momentum through a spinning rotor that results in the exertion of torque on different components of an aerospace system. These gyroscopes are especially useful in outer space systems due to weak gravitational interactions and the absence of 'solid' surfaces to aid in reorientation of the vehicle or system under study.

The objective of this design project is to design a CMG that will come to rest a particular angle that is arbitrarily picked. Tasks like these are often important when considering space systems like satellites and orbital vehicles that may be required to re-orient themselves to avoid collision with objects such as space debris and other space systems. Since these objects are in outer space, they can re-orient themselves with the help of a CMG that will change their attitude to alter their motion in a desired direction.

Solving the system of a CMG analytically often proves difficult because it is a non-linear system. Linearizing the system is a good work-around to this problem. This, however, comes with some trade-offs as linearization is, after all, a linear approximation. Thus, an important consideration is how 'good' this linear approximation is.

II. Problem Statement

The system under consideration consists of a platform that rotates about the vertical axis, a gimbal that rotates about an axis orthogonal to the platform, and a rotor that rotates about the remaining orthogonal axis. The rotor rotates about its center. This property gives rise to the fact that the angular orientation and velocity of the rotor do not pose a significant challenge to the formulation of this problem due to the symmetry of the rotor.

As given in the design project statement, an input torque is applied to the rotor that results in output torques being exerted on the gimbal and platform. These torques result in the rotation of the gimbal and platform about their respective axes and change the orientation of the platform. It is important to consider how much torque is exerted on the platform and gimbal to understand by how much the orientation of the platform changes (in radians) per unit of torque.

The governing equations of the system are given by the matrix:

$$v_1 = -\frac{5 \cdot \left(200\tau_3 \sin\left(q_2\right) \ \ \sin\left(2q_2\right)q_1q_2 \ \ 2\cos\left(q_2\right)q_2q_3\right)}{10\sin^2\left(q_2\right) - 511} \tag{1}$$

$$v_2 = \frac{10 \cdot \left(100\tau_2 - \cos\left(q_2\right)q_1q_3\right)}{11} \tag{2}$$

$$v_3 - \frac{51100\tau_3 \ \ 5\sin\left(2q_2\right)q_2q_3 \ \ 511\cos\left(q_2\right)q_1q_2}{10\sin^2\left(q_2\right) - 511} \tag{3}$$

As discussed, v_3, the acceleration of the rotor, is immaterial to this discussion due to symmetry. The system is then described entirely by (1) and (2). This still is a difficult non-linear differential equation to solve. In order to solve this system, a coherent strategy is required. This is described below.

A. Strategy to solve the problem

This section seeks to provide a brief outline on how to solve the aforementioned problem of reorienting the platform to a particular angle $q_{1,e}$. Listed below are the four key components to solving this system:

1) Linearize the system around the chosen equilibrium state. This will help convert a non-linear problem into a well-defined linear problem which is easier to solve.
2) Verify asymptotic stability. This will confirm, in theory, that the platform angle will converge to a specific value and will remain there, suggesting equilibrium has been reached.
3) Implement a controller. This will provide practical confirmation of a theoretical understanding that the system is asymptotically stable.

III. Methods

In the previous section, three steps were presented to solving this problem:

1) Linearization
2) Verification of asymptotic stability
3) Implementation of controller and resulting graphs.

This section will discuss these three steps in greater detail.

A. Linearization

Linearizing the system is a useful tool to solve it. Linearizing the system means converting equations (1) and (2) into the following form:

$$x = Ax \;\; Bu$$

The expression above is a first-order linear differential equation (LDE) for which an analytical solution is known. To linearize the system, the following steps are necessary:

1) Define state x and input u.
2) Choose an equilibrium state for the system.
3) Find the Jacobians of (1) and (2) with respect to x and u at the equilibrium state.

As discussed previously, the system has three angular positions (given by q_i) and velocities (given by v_i). Due to symmetry, q_i and v_i can be removed from consideration without having an effect on the precision and accuracy of the results of the system. The system also has two input torques (given by τ_i). As such, the state variable of the system is given by:

$$x = \begin{bmatrix} q_1 \\ v_1 \\ q_2 \\ v_2 \end{bmatrix}$$

An important observation that will be important to formulate the matrix A is that $q_i = v_i$.
The input is given by:

$$u = \begin{bmatrix} \tau_2 \\ \tau_3 \end{bmatrix}$$

τ_2 is the magnitude of the torque of the platform on the gimbal and tau_3 is the magnitude of the torque of the gimbal on the rotor. By Newton's third law, equivalently, τ_3 is the magnitude of the torque of the rotor on the gimbal and τ_2 is the magnitude of the torque of the gimbal on the platform.

The following is the equilibrium point that was chosen for the system:

1) $q_{1,e} = \frac{\pi}{20}$
2) $q_{2,e} = \frac{\pi}{4}$
3) $v_{1,e} = 0$
4) $v_{2,e} = 0$
5) $v_{3,e} = \frac{20\pi}{6}$
6) $\tau_{2,e} = 0$
7) $\tau_{3,e} = 0$

In matrix form,

$$x_e = \begin{bmatrix} \frac{\pi}{20} \\ 0 \\ \frac{\pi}{4} \\ 0 \end{bmatrix}$$

$$u_e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

After taking the Jacobians with respect to x and u at the equilibrium state, the following are the matrices A and B:

$$A = \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0.146340017725901 \\ 0 & 0 & 0 & 1.0 \\ 0 & -6.73164081539146 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 1.39744423159397 \\ 0 & 0 \\ 90.9090909090909 & 0 \end{bmatrix}$$

The linearized form of the system is thus:

$$\begin{bmatrix} 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0.146340017725901 \\ 0 & 0 & 0 & 1.0 \\ 0 & -6.73164081539146 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} q_1 - \frac{\pi}{20} \\ v_1 \\ q_2 - \frac{\pi}{4} \\ v_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1.39744423159397 \\ 0 & 0 \\ 90.9090909090909 & 0 \end{bmatrix} \cdot \begin{bmatrix} \tau_2 \\ \tau_3 \end{bmatrix} \tag{4}$$

An important part of this design project is to be able to control the input and find how changing the torque results in a new position. As such, a matrix K is defined such that

$$u = -Kx \tag{5}$$

The values for the matrix K are unknown but they can be found though the Ackermann's method which is implemented in `scipy.signal.place_poles`. Ackermann's method requires three items: the matrix A, the matrix B, and a matrix Σ of arbitrarily chosen eigenvalues. In this case, the following were the eigenvalues:

$$\Sigma = \begin{bmatrix} -1 & -2 & -3 & -4 \end{bmatrix}$$

Negative values were purposefully chosen to ensure asymptotic stability. The matrix K was then found by accessing the `full_state_feedback.gain_matrix`, where `full_state_feedback` is the return of the function `scipy.signal.place_poles`.

B. Verifying theoretical asymptotic stability

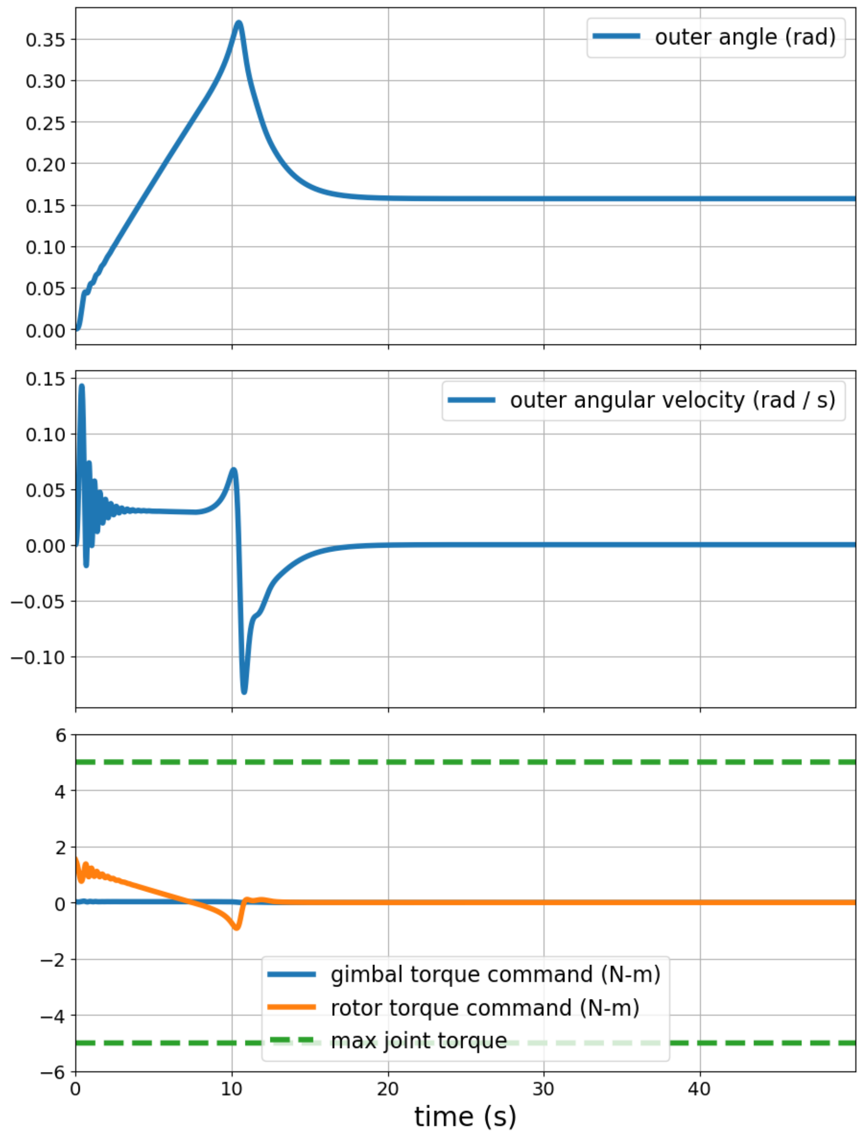
In the previous section, it was mentioned that negative eigenvalues were chosen to ensure asymptotic stability. In order to further confirm that the matrix $A - BK$ did produce negative eigenvalues, the variable `full_state_feedback.computed_poles` was accessed. The computed eigenvalues were all negative and equal to those that were pased into the function, indicating the system wa asymptotically stable in theory.

C. Implementation of controller and graphs

The last step was practically verify asymptotic stability by simulating a controller at the equilibrium state. This was accomplished by creating a class `RobotController` that had the following methods:

1) `__init__(self, k, q1e,q2e,v1e,v2e,v3e,tau2e,tau3e,dt)`: constructor
2) `run(self, q1, v1, q2, v2, q3)`, returns `tau2, tau3`: function that 'ran' the simulation and provided successive values of the torques.

The following image is of the graphs obtained from the simulation:



From the graphs above, it is easy to see that all of angle, velocity, and torque are converging, confirming that the system is asymptotically stable.

IV. Analysis

This section will now present some observations that were made after the main tasks were completed and will attempt to provide a reasonable justification for those observations.

A. Linear trajectory vs. non-linear trajectory

The following are the figures of the linear and non-linear trajectories:

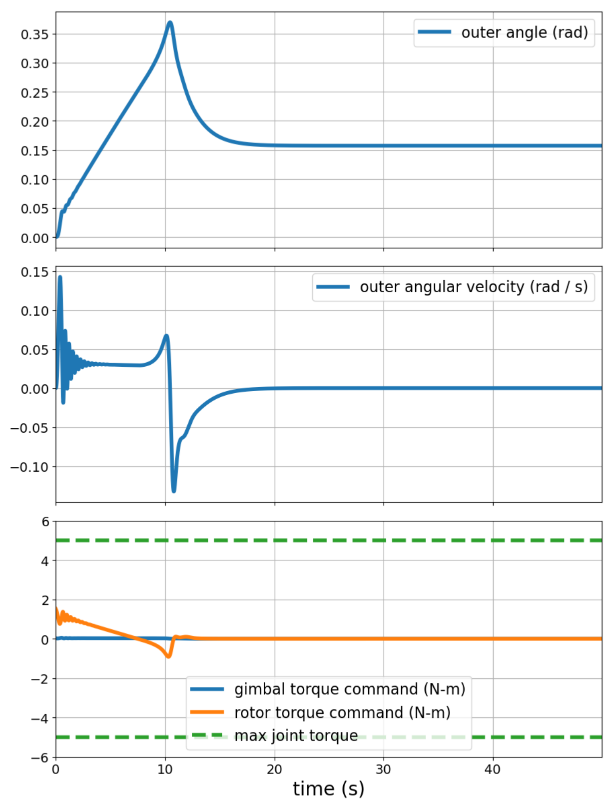


Fig. 1 Linear feedback

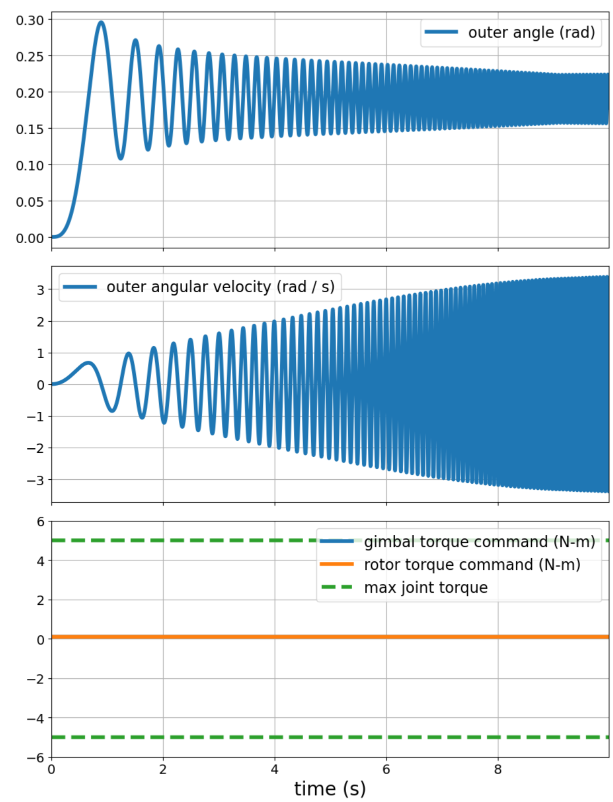


Fig. 2 Non-linear feedback

In the linear feedback, the values all monotonically converge to the final angle. In the non-linear feedback however, the convergence is more sinusoidal with the amplitude steadily decreasing as the values approach the final angle. A possible reason for this is that the solutions to linear feedback are functions of the form $e^{\lambda_i t}$ where λ_i are the eigenvalues for $A - BK$. On the other hand, the non-linear system is described by sinusoidal terms, resulting in the sinusoidal characteristics.

B. Choice of final angle

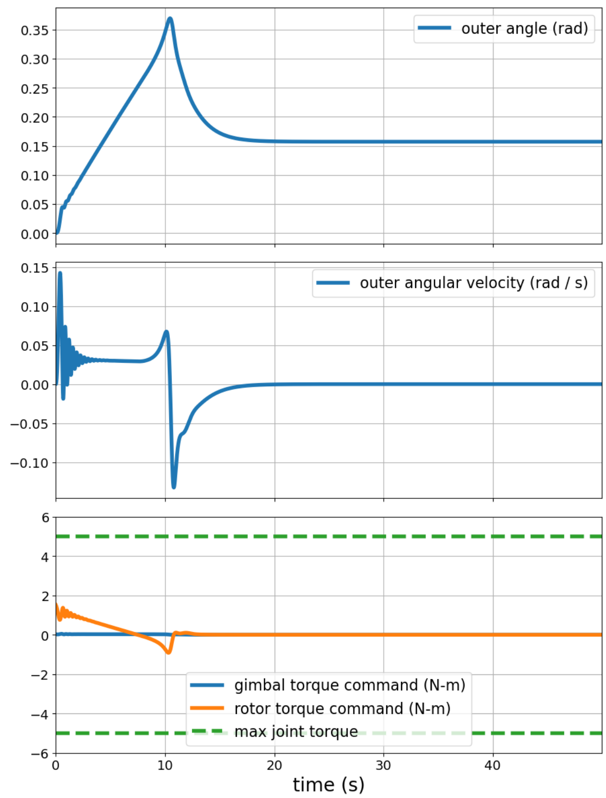


Fig. 3 $q_{1,e} = \frac{\pi}{20}$

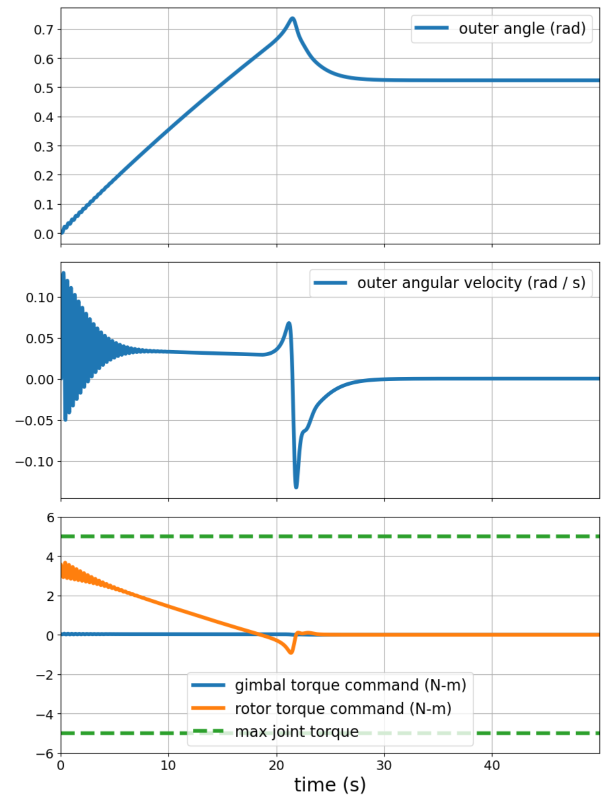


Fig. 4 $q_{1,e} = \frac{\pi}{6}$

The above figure graphs indicate two different final angles, all other things constant. In Fig. 3, for a smaller final angle choice, the equilibrium state is reached fairly quickly. However in Fig. 4, for a larger final angle choice, the equilibrium state is reached at a slower rate. This is logical, as a smaller final angle is closer to the initial position than a larger angle. However, both graphs follow a largely similar trajectory, indicating that the choice of angle does not influence the path followed by the platform to a great extent. This suggests the equilibrium angle does not affect the controllability matrix K.

V. Choice of eigenvalues

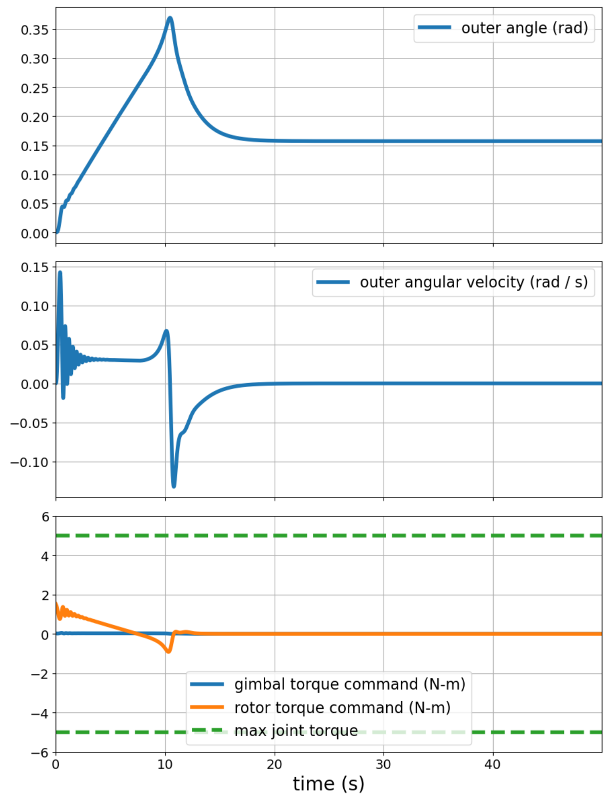


Fig. 5 $\Sigma = \begin{bmatrix} -1 & -2 & -3 & -4 \end{bmatrix}$

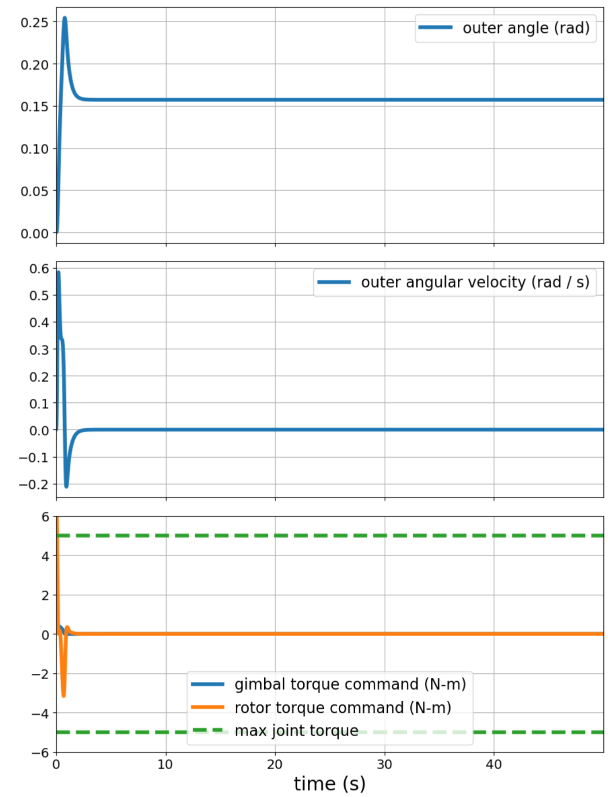


Fig. 6 $\Sigma = \begin{bmatrix} -5 & -6 & -7 & -8 \end{bmatrix}$

Fig. 5 is the graph for eigenvalues of smaller magnitude, and Fig. 6 is the graph for eigenvalues of larger magnitude. Based on the above graphs, it is possible to draw the conclusion that eigenvalues of larger magnitude cause the system to converge to the equilibrium state faster. This could be because larger eigenvalues result in larger values for the entries in the matrix K. This in turn results in the input torque and current internal state of the system changing at a faster rate, resulting in a faster convergence.