DP1: Control Moment Gyroscope Control of a Spacecraft

Patrick D. Thornton and Varsha Krishnakumar
University of Illinois at Urbana-Champaign

The objective of this exploration is to implement and verify a feedback controller that uses a single-gimbal control moment gyroscope (CMG) to reorient and control the direction of a spacecraft platform in a gravitational field. The controller and simulation are implemented, designed, and tested in a Python Jupyter Notebook environment.

I. Nomenclature

CMG	=	Control Moment Gyroscope
ODE	=	Ordinary Differential Equation
f	=	Equations of motion as a function
q_1	=	Angle (rad) of platform
$\dot{q}_1$	=	Angular Velocity (rad/s) of platform
$\ddot{q}_1$	=	Angular Acceleration (rad/s) of platform
q_2	=	Angle (rad) of gimbal
$\dot{q}_2$	=	Angular velocity (rad/s) of gimbal
$\ddot{q}_2$	=	Angular velocity (rad/s) of gimbal
τ	=	Torque (N $\cdot$ m) applied by the platform on the gimbal
u	=	Inputs for the system as a vector
x	=	State Vector

II. Introduction

A Control Moment Gyroscope (CMG) is comprised of a spinning rotor and motorized Gimbals that can control the angular momentum of the rotor [*]. A CMG is most commonly used as an attitude control device for a spacecraft's attitude control system. While a CMG can have more than one motorized gimbals, the focus of this project is on the *single-gimbal* CMG. A single-gimbal CMG influences the rotor's angular momentum in a way that a large torque can be applied. Furthermore, since there is only one motorized gimbal, the angle of the platform can only be changed in one plane. A single-gimbal CMG is able to conserve on power and electricity, since it requires little input on both.

In this project, the motion of the platform will be analyzed based on the theoretical system's equilibrium conditions. The platform test angle for the intended design of the Single-Gimbal is noted as $q_{1_{des}} = \pi$. To design such a controller, a non-linear state space model of the system is initialized. The relevant testing and configuration of this model is performed in a python environment.[†]

III. Theory

A. Equations of Motion for CMG System

The CMG system is modeled based on a model in state-space form[‡]. The motion of this particular CMG system is governed by the set of Ordinary Differential Equations below

$$\ddot{q}_1 = \frac{0.01\sin(2q_2)\dot{q}_1\dot{q}_2 + 20\cos(q_2)\dot{q}_2 - 39.24\sin(q_1)}{9.002 + 0.02\cos^2(q_2)} \tag{1a}$$

$$\ddot{q}_2 = -909.09\cos(q_2)\dot{q}_1 + 90.909\tau \tag{1b}$$

[*]https://en.wikipedia.org/wiki/Control_moment_gyroscope
[†]https://github.com/varshakrishnakumar/AE-353-Design-Project-1
[‡]https://tbretl.github.io/ae353-sp22/projectsthe-system

In order to design a proper controller, the system must be modeled in state-space form. The equations of motion must first be linearized in order to approximate the system with the desired state-space model. To do so, the system was first rewritten as a set of first-order ordinary differential equations.

$$\dot{x} = f = \begin{bmatrix} v_1 \\ \frac{0.01\sin(2q_2)v_1v_2 + 20\cos(q_2)v_2 - 39.24\sin(q_1)}{9.002 + 0.02\cos^2(q_2)} \\ v_2 \\ -909.09\cos(q_2)v_1 + 90.909\tau \end{bmatrix} \tag{2}$$

$vTheequilibriumpointforwhichtolinearizethesystemaboutisthesetofvariablesfor\mathrm{f}thatcause = 0, andthus\mathrm{x}remainsconsta$

1) $v_{1e} = v_{2e} = 0$ as is apparent from the first and third elements of $\dot{x}$.
2) As a result, $\tau_e = 0$ then becomes a requirement as well.
3) From the second element, it is important to note that the choice of q_{1e} (i.e., the desired angle at which to hold the platform) must be an integer multiple of π. In plain terms, this means that the spacecraft can only maintain stability in a completely vertical orientation. $q_{1e} = \pi$ was chosen.
4) q_{2e} may have any value. $q_{2e} = 0$ was made as a reasonable choice.

The state and input may now be defined based on this choice of equilibrium point:

$$x = \begin{bmatrix} q_1 - q_{1e} \\ v_1 - v_{1e} \\ q_2 - q_{2e} \\ v_2 - v_{2e} \end{bmatrix} = \begin{bmatrix} q_1 - \pi \\ v_1 - 0 \\ q_2 - 0 \\ v_2 - 0 \end{bmatrix} \tag{3a}$$

$$u = \begin{bmatrix} \tau - \tau_e \end{bmatrix} = \begin{bmatrix} \tau - 0 \end{bmatrix} \tag{3b}$$

The matrices A and B may then be found by calculating the Jacobian of f with respect to the state vector x and the input vector u. These were evaluated at the chosen equilibrium points q_{1e}, q_{2e}, v_{1e}, v_2e.

$$A = \left.\frac{\partial f}{\partial x}\right|_{(q_{1e}, v_{1e}, q_{2e}, v_{2e})} = \begin{bmatrix} 0. & 1. & 0. & 0. \\ 4.34936821 & 0. & 0. & 2.21680337 \\ 0. & 0. & 0. & 1. \\ 0. & -909.09090909 & 0. & 0. \end{bmatrix} \tag{4a}$$

$$B = \left.\frac{\partial f}{\partial u}\right|_{(q_{1e}, v_{1e}, q_{2e}, v_{2e})} = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 90.90909091 \end{bmatrix} \tag{4b}$$

B. Controller Analysis

To implement a closed-loop system, the input matrix u was chosen as follows, where K is some constant matrix. This form implies that the input depends on the state.

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

With this definition of u, the system may be rewritten as:

$$\dot{x} = (A - BK)x \tag{6}$$

In order to achieve a reliable controller, the resulting matrix $F = (A - BK)$ must have have eigenvalues with only negative real parts. A higher magnitude for the real part indicates that the system achieves equilibrium in a shorter time.

Further, zero imaginary part was desired as a lower magnitude for the imaginary part indicates that the system has less oscillation before equilibrium.

Through python implementation, a working K matrix below was determined using the A and B matrices, as well as the desired eigenvalues in the vector below:

$$p = \begin{bmatrix} -10. & -8. & -4. & -2. \end{bmatrix} \tag{7}$$

The resulting K from this process is shown below:

$$K = \begin{bmatrix} 3.6143 & -8.27568 & -1.61862 & 0.264 \end{bmatrix} \tag{8}$$

It is apparent that not all valid K matrices yield the desired CMG behavior, so this process will be constrained further. The next appropriate step is to efficiently constrain the time frame for the system achieving a desirable platform angle. With this, the controller will be featured as computationally and time efficient. To achieve this, an analysis on its response, $x(t)$ must be performed. Here, the matrix F, as found previously, must be diagonalized so that evaluating the matrix exponential, e^{Ft}, is simplified.

The relation used for this process if as follows:

$$x(t) = Ve^{St}V^{-1}x(0) \tag{9}$$

Here, V is a matrix comprised of the eigenvectors of F, and S is a diagonal matrix with F's eigenvalues. The corresponding results of the analysis on the controller's response is shown in *Figure 1*.

IV. Experimental Methods

Jupyter Notebook was utilized as code interface for simulating and testing the CMG model. The instructions for compilation of the project code is detailed in the GitHub repository associated with this report *.

A matrix of controller gains, **K**, in theory results in a controller that produces an asymptotically stable CMG system if the matrix $F = (A - BK)$ has eigenvalues with all negative real parts.

To test the stability of the controller in achieving and maintaining the desired platform angle, the following parameters were set:

1) q_{1e} was set to the desired platform angle of π rad. This is a completely vertical position where the platform's load mass is held above the platform.
2) All other equilibrium variables were set to a value of 0 as discussed prior.
3) The initial platform angle q_{1i} at time $t = 0$ was set to a value of $\pi - 0.2$ rad.
4) The initial platform velocity v_{1i} was set to a value of $-0.2\frac{rad}{s}$.
5) The other initial variables q_{2i} and v_{2i} were set to a value of 0, so that the rotor wheel starts flush with the platform.
6) The gains matrix was set to $K = \begin{bmatrix} 3.6143 & -8.27568 & -1.61862 & 0.264 \end{bmatrix}$

Under these conditions, the controller code was executed. The plots formulated from this initial test case may be seen in *Figure 1*.

To test the limits of initial conditions for which the controller can achieve stability, python was used to run and collect data on 1000 simulations each for varying q_{1i} and v_{1i}. Simulations for q_{1i} sampled from $q_{1i} \in \begin{bmatrix} q_{1e} - 0.6 & q_{1e} + 0.6 \end{bmatrix}$, with all other initial conditions set to zero. Simulations for v_{1i} sampled from $v_{1i} \in \begin{bmatrix} v_{1e} - 1.2 & v_{1e} + 1.2 \end{bmatrix}$, with $q_{1i} = 0$ and all other initial conditions set to zero.

*https://github.com/varshakrishnakumar/AE-353-Design-Project-1

V. Results

The results of the single-CMG simulation are shown in *Figure 1*.

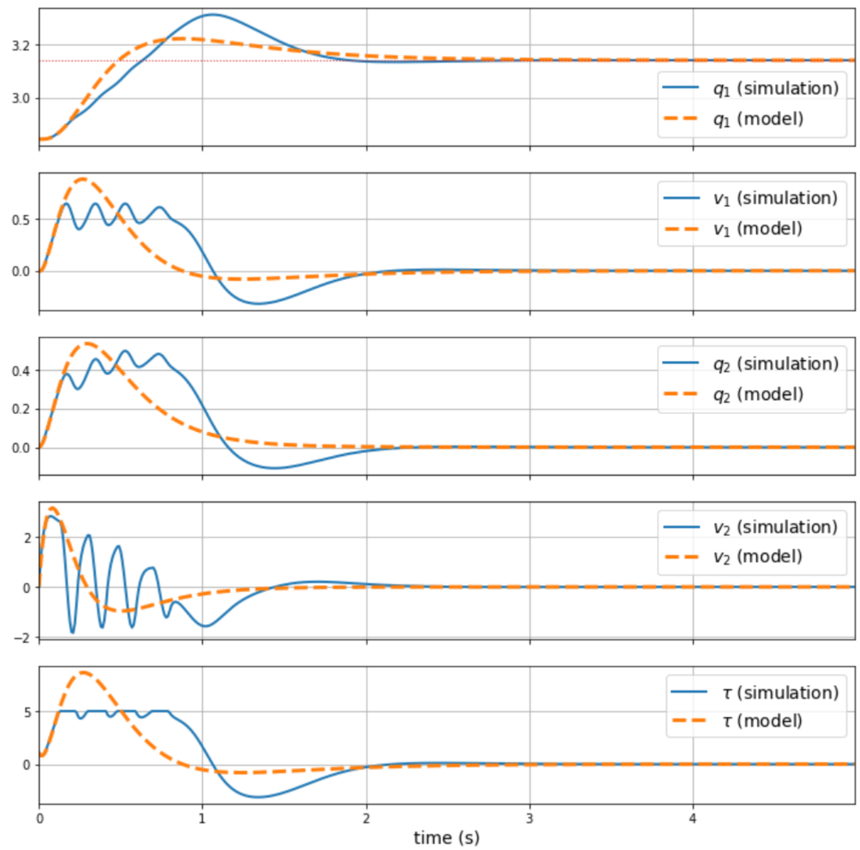

Fig. 1 Results of Simulation

The most critical outcomes from this simulation and corresponding testing are:

1) For the initial condition of $q_{1_i} = \pi - 0.2$ and an equilibrium condition of $q_{1_e} = \pi$, the platform angle converges to the desired angle of π in approximately 2 seconds. This indicates the stability of the controller, as well as the efficiency of the model for desired angles close to the equilibrium condition.
2) Similarly, the desired angular velocity of the platform, $v_1 = 0$, is achieved within approximately 2 seconds.
3) For the initial condition of $q_{2_i} = 0$ and an equilibrium condition of $q_{2_e} = 0$, the gimbal angle converges to the desired angle of 0 within 2 seconds.
4) Similarly, the desired angular velocity of the gimbal, $v_2 = 0$, is achieved within 2 seconds.

The results for the other five conducted trials of varying initial platform and angular velocity are detailed in the table below.

As seen from the results, the controller is efficient in stabilizing for initial conditions that are close to the equilibrium or desired conditions. A key feature of the system is that there is little to no oscillation. This is attributed to the fact that the eigenvalues selected for the system have no imaginary parts, and only contain negative real parts. Nevertheless, the rotor is not able to provide enough torque to enable the desired motion of the platform. Therefore, initial conditions that are further away from the selected equilibrium conditions are not practically feasible.

The ranges of the initial conditions, q_{1_i} and v_{1_i}, can be observed in *Figure 2* and *Figure 3*. The controller achieves stability for the range $q_{1_i} = [\pi - 0.38378, \pi + 0.38378]$ for the platform angle and for the range

$v_{1_i} = [-0.93093, 0.93093]$ for the platform velocity. For initial condition values outside this range, it can be seen that the stability of the system falters.

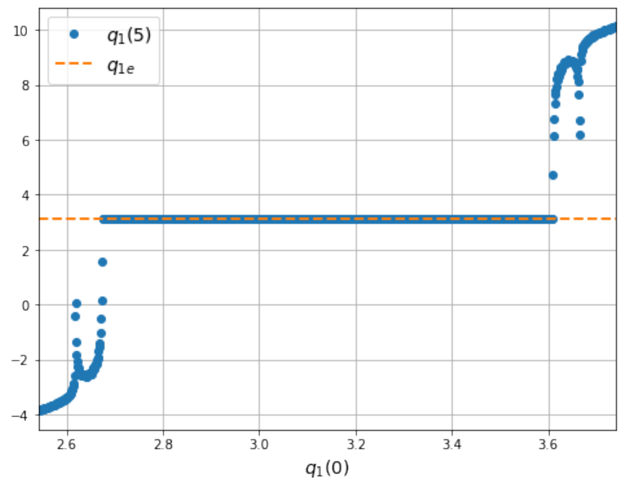

Fig. 2 Range of q_{1_i}

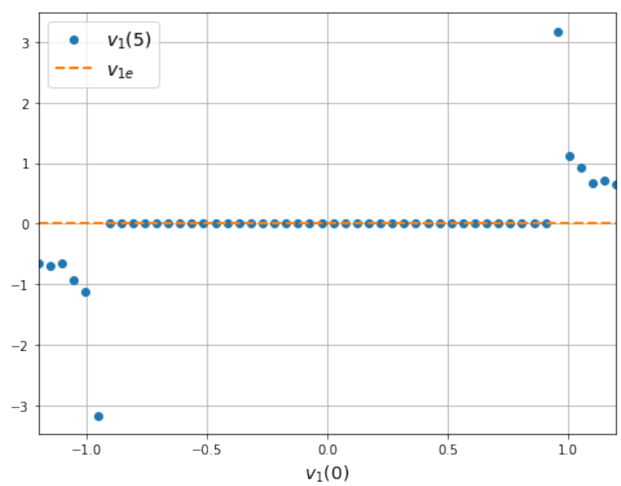

Fig. 3 Range of v_{1_i}

VI. Conclusion

The designed single-CMG model is efficient in achieving the desired equilibrium conditions in a small time frame, and is additionally able to attain stability appropriately. The utilization of the *Signal* package from the *SciPy* Python library was useful in the generation of the K gains matrix which was crucial in the efficiency of the system. Although the system loses the ability to attain stability as the initial conditions stray further away from the equilibrium conditions, further testing and generation of K matrices can be useful in achieving this objective.

Acknowledgments

Majority of the driver code for the implementation of the CMG model was provided in lecture in AE 353 at the University of Illinois conducted by Professor Timothy Bretl and the Teaching Assistant, Mr. Jacob Kraft. The guided learning on the theory of control systems and coding techniques assisted in the modelling and implementation of this project. Furthermore, the utilization of *Campuswire* as a tool for query clarification further improved theoretical understanding.

References

[1]: "Control moment gyroscope," Wikipedia Available: https://en.wikipedia.org/wiki/Control_moment_gyroscope. [2]: Krishnakumar, V., and Thornton, P., "Varshakrishnakumar/ae-353-design-project-1: Designing and implementing a control moment gyroscope (CMG)," GitHub Available https://github.com/varshakrishnakumar/AE-353-Design-Project-1. [3]: "Projects," AE 353: Aerospace Control Systems Available: https://tbretl.github.io/ae353-sp22/projectsthe-system 1.

Appendix

Day	Task	Person or People
01/28/2022	Initiation and set-up of report document	Patrick Thornton
02/01/2022	Basic structure for Abstract and Introduction sections	Varsha Krishnakumar
02/02/2022	Creation of GitHub repository with basic outline of model generating code in python environment	Varsha Krishnakumar
02/03/2022	Description of Equation of Motion for the CMG system	Varsha Krishnakumar
02/03/2022	Description of method for finding equilibrium points	Patrick Thornton
02/04/2022	Exposition of theory and model with python code for the controller: worked together on analyzing faults in the system's motion and possible reasons for inefficient working	Patrick Thornton & Varsha Krishnakumar
02/09/2022	Analyzed, debugged, formatted, and annotated existing code.	Patrick Thornton
02/09/2022	Completed working code, implemented plots to visualize state of the system to determine if stability is achieved.	Patrick Thornton
02/09/2022	Adapted the in-class interactive eigenvalue plot notebook for use with the CMG system, devised an improved system for determining the K matrix, and conducted testing for various K matrices.	Patrick Thornton
02/10/2022	Conducted testing of controller for various initial conditions	Patrick Thornton & Varsha Krishnakumar
02/11/2022	Finalized *Experimental Methods* and *Results* sections.	Patrick Thornton & Varsha Krishnakumar
02/11/2022	Improved report formatting and made adjustments based on comments	Patrick Thornton & Varsha Krishnakumar
02/17/2022	Made adjustments on Experimental Methods and Results sections based on comments	Varsha Krishnakumar
02/18/2022	Devised new plan for generating K matrix and revamped efficiency of the system	Patrick Thornton
02/18/2022	Edited and put together short video comprising of the jist of the project	Patrick Thornton
02/18/2022	Revamped writing of the report by including more analysis on linear state feedback, gain matrix, eigenvalues and results	Varsha Krishnakumar
02/18/2022	Over-viewed and corrected sentence structure and grammar throughout the report for finalization	Patrick Thornton & Varsha Krishnakumar