DP3: Control of a Spacecraft with a Star Tracker and Reaction Wheels

Kameron Jackson and Patrick Thornton
University of Illinois at Urbana-Champaign

I. Nomenclature

ODE	=	Ordinary Differential Equation
ψ	=	yaw angle (rad)
ϕ	=	roll angle (rad)
θ	=	pitch angle (rad)
ω_x	=	angular velocity (rad/s) about the body-fixed x-axis
ω_y	=	angular velocity (rad/s) about the body-fixed y-axis
ω_z	=	angular velocity (rad/s) about the body-fixed z-axis
τ_1	=	torque (N * m) applied to wheel 1
τ_2	=	torque (N * m) applied to wheel 2
τ_3	=	torque (N * m) applied to wheel 3
τ_4	=	torque (N * m) applied to wheel 4
x	=	state matrix
u	=	input matrix

II. Introduction

Consider a spacecraft controlled via reaction wheels and star tracker. Suppose it is possible to hold the yaw, pitch, and roll close to zero despite the movement of stars. Our goal is to design a controller and observer that is stable while keeping the system close to the equilibrium condition of 0 for all parameters. This is a challenge that can be seen in real implications of star trackers and control systems. The theory section derives a sensor state-space model of our system by linearizing about an equilibrium point and then applies eigenvalue placement to design linear state feedback. Upon linearization a controller and observer are created. The methods section describes how we implemented and tested our controller and observer in simulation, and in particular describes the simulations we conducted to quantify the extent of the data set and functionality of the system. Section results presents the results of these experiments. Section conclusion concludes with opportunities for future work.

III. Requirements and Verification

A. Requirements

For the default arrangement of seven stars, initial spacecraft orientation sampled at random, initial angular velocity set to zero on all axes, and sensor noise set to the default of 0.1, the spacecraft shall maintain an orientation of approximately $\psi = \phi = \theta = 0$. Because it is understood that the controller and sensor model system will not consistently achieve perfect results, the root-mean-square error (RMSE) for each orientation and for a 60-second simulation must remain less than $\frac{\pi}{6}$ for at least 80% of the simulations run. RMSE over N steps is defined below where x corresponds to ψ, ϕ, and θ.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - 0)^2}{N}} \tag{1}$$

B. Verification

PyBullet will be used to simulate the spacecraft and star tracker system. 500 simulations will be run with a maximum time of 60 seconds, and the data generated by these simulations will be imported into a Jupyter Notebook for analysis with Python. For each simulation, the RMSE for ψ, ϕ, and θ with respect to their target values of zero will be recorded and plotted in a histogram. For each orientation parameter, the 80th percentile RMSE will be determined. If the 80th percentile RMSE for each orientation parameter is less than $\frac{\pi}{6}$, this indicates that 80% of the simulations run yielded an RMSE less than $\frac{\pi}{6}$, and thus the requirement is met.

IV. Theory

In the process of designing a proper controller and observer, the equations describing the dynamic system are to be linearized so that the system can be placed into sensor state-space form. Given a set of ODEs, the system first has to be rewritten in a way such that any second-order ODEs are replaced by a set of two first order ODEs. To do so, equilibrium points were arbitrarily chosen to be 0 for all parameters then verified. The following shows the process of linearization and determining the state space model

A. Equations of Motion for the Spacecraft

If we define the nonlinear inputs to the spacecraft and reaction wheel system as follows,

$$m = \begin{bmatrix} \psi & \theta & \phi & w_x & w_y & w_z \end{bmatrix}^T \qquad n = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}^T \tag{2}$$

Then we can write the equations of motion for the system in the form

$$\dot{m} = f(m,n) = \begin{bmatrix} \frac{w_y \sin(\phi) + w_z \cos(\phi)}{\cos(\theta)} \\ w_y \cos(\phi) - w_z \sin(\phi) \\ w_x + w_y \sin(\phi)\tan(\theta) + w_z \cos(\phi)\tan(\theta) \\ -\frac{55\sqrt{2}\tau_1}{1484} + \frac{55\sqrt{2}\tau_2}{1484} - \frac{150 w_y w_z}{371} \\ -\frac{55\sqrt{2}\tau_3}{1484} + \frac{55\sqrt{2}\tau_4}{1484} + \frac{150 w_x w_z}{371} \\ -\frac{55\sqrt{2}(\tau_1+\tau_2+\tau_3+\tau_4)}{2084} \end{bmatrix} \tag{3}$$

We determine the equilibrium point of (2) by setting the equation equal to 0 and solving for m_e and n_e. When we also consider the goal of keeping the spacecraft close to zero yaw, pitch, and roll, we find that

$$m_e = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \qquad n_e = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

We then define the linear state and input as

$$x = m - m_e \qquad u = n - n_e \tag{5}$$

and find the matrices A and B as the Jacobian of f with respect to m and n respectively, evaluated at the equilibrium point:

$$A = \left.\frac{\partial f}{\partial m}\right|_{(m_e,n_e)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \left.\frac{\partial f}{\partial n}\right|_{(m_e,n_e)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{55\sqrt{2}}{1484} & \frac{55\sqrt{2}}{1484} & 0 & 0 \\ 0 & 0 & -\frac{55\sqrt{2}}{1484} & \frac{55\sqrt{2}}{1484} \\ -\frac{55\sqrt{2}}{2084} & -\frac{55\sqrt{2}}{2084} & -\frac{55\sqrt{2}}{2084} & -\frac{55\sqrt{2}}{2084} \end{bmatrix} \tag{6}$$

Combining (4) and (5), we now have a linear approximation to the dynamic system in state-space form:

$$\dot{x} = Ax + Bu \tag{7}$$

B. Verifying Controllability of the Spacecraft System

In order to determine if the spacecraft system is controllable, we build the controllability matrix W_C below and verify that it is full rank

$$W_C = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{8}$$

where n is the number of states, i.e., the number of rows in A. Through python implementation, we find that the rank of W is 6, and the largest possible rank for a matrix of equivalent dimensions is also 6. W_C is full rank, and thus the system is controllable.

C. Star Tracker Sensor Model

Just as in Subsection A, we start by writing the equations of the sensor model below.

$$o = \begin{bmatrix} y_0 \\ z_0 \\ \vdots \\ y_6 \\ z_6 \end{bmatrix} = g(\psi, \theta, \phi, \alpha_0, \delta_0, \cdots, \alpha_6, \delta_6) \tag{9}$$

We define the output as

$$y = o - g(m_e, n_e) \tag{10}$$

and substitute the equilibrium values and respective α_i and δ_i into (8).

We then compute C by

$$C = \left.\frac{\partial g}{\partial m}\right|_{(m_e, n_e)} = \begin{bmatrix} -2.65142599685905 & 0.0400056306980763 & -0.398721896095952 & 0 & 0 & 0 \\ -0.0400056306980764 & 2.68556348587671 & 0.263378514224308 & 0 & 0 & 0 \\ -2.625 & 0 & -0.396729947403024 & 0 & 0 & 0 \\ 0 & 2.68495986711101 & 0 & 0 & 0 & 0 \\ -2.65142599685905 & -0.0400056306980763 & -0.398721896095952 & 0 & 0 & 0 \\ 0.0400056306980764 & 2.68556348587671 & -0.263378514224308 & 0 & 0 & 0 \\ -2.625 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.625 & 0 & 0 & 0 & 0 \\ -2.65142599685905 & -0.0400056306980763 & 0.398721896095952 & 0 & 0 & 0 \\ 0.0400056306980764 & 2.68556348587671 & 0.263378514224308 & 0 & 0 & 0 \\ -2.625 & 0 & 0.396729947403024 & 0 & 0 & 0 \\ 0 & 2.68495986711101 & 0 & 0 & 0 & 0 \\ -2.65142599685905 & 0.0400056306980763 & 0.398721896095952 & 0 & 0 & 0 \\ -0.0400056306980764 & 2.68556348587671 & -0.263378514224308 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

From (10), we may now write the linear approximation of the sensor model in state-space form:

$$y = Cx \tag{12}$$

D. Verifying Observability of the Sensor Model

In the same manner as controllability, we must first determine if the observability matrix W_O is full rank. To do so, we build W_O as follows:

$$W_O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{13}$$

The rank of W_O is found to be 6. Because W_O is full rank, the system is thus observable.

E. Linear Quadratic Regulator Gain Matrices

In order to design a controller and observer for our system, the gain matrices for both must be calculated. In order to calculate these matrices, we used the linear quadratic regulator approach. This approach calculates the optimal cost of the system via:

$$\int_{t_0}^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \tag{14}$$

The equation to find the K matrix is as follows:

$$K = R^{-}1B^T p \tag{15}$$

The equation to find the L matrix is as follows:

$$L = P_0 C^T Q_0 \tag{16}$$

F. Controller and Observer Design

To design the controller and Observer we found the gain matrix for each, K and L respectively. For the controller design the gain matrix, K is as follows:

$$K = \begin{bmatrix} -0.5 & 0 & -0.7 & -3.7 & 0 & -2.6 \\ -0.5 & 0 & 0.7 & 3.7 & 0 & -2.6 \\ -0.5 & -0.7 & 0 & 0 & -3.7 & -2.6 \\ -0.5 & 0.7 & 0 & 0 & 3.7 & -2.6 \end{bmatrix} \tag{17}$$

Now that the controller gain matrix has been found and the controller created, we moved on to the creation of the observer by finding its gain matrix, L. The resulting matrix, L is as follows:

$$L = \begin{bmatrix} -3.58799999999999 & 1.18117153662188 \cdot 10^{-14} & -5.07419826179465 & -26.8414144160575 & 4.552862466 \\ -3.588 & -1.22334169523271 \cdot 10^{-14} & 5.07419826179465 & 26.8414144160575 & -4.84622643 \\ -3.588 & -5.07419826179468 & 7.88705663505369 \cdot 10^{-15} & 5.41942980695403 \cdot 10^{-14} & -26.8414 \\ -3.58799999999999 & 5.07419826179468 & -6.39155025223964 \cdot 10^{-15} & -3.97965909194736 \cdot 10^{-14} & 26.84141 \end{bmatrix} \tag{18}$$

G. Stability

To verify that the controller is stable, we utilized the creation of a closed loop system, which we called F. The equation for F is as follows:

$$F = A - BK \tag{19}$$

Where the A and B matrices were found using the jacobian. The K matrix was found using LQR, Linear Quadratic Regulator. To verify the stability of the controller the eigenvalues of F were found to have all negative real parts, indicating stability. The same methodology was implemented for the G matrix, this time with the following equation.

$$G = L - AC \tag{20}$$

Confirming that G produce all negative eigenvalues, we also confirmed its stability.

V. Experimental Methods

A. Controller and Observer Implementation

After finding the gain matrices, we implemented the controller and observer through state space estimation, the formula for which can be seen below.

$$\hat{x} = A\hat{x} + Bu - L(C\hat{x} - y) \tag{21}$$

The observer allows for state estimation for the given sensor data and the model. This step is needed since the sensor data does not provide the full statespace model. As previously discussed in the theory section the controller and observer implementation relied on the finding the gain matrices for both. In order to do that, LQR was used to find the gain matrix K, for the controller and G, for the observer.

B. Experiments to Determine Run Time Until Failure for Series of Trials

By running the experiments we were able to verify the functionality of the system as a whole.

C. Experiments to Determine RMSE of Yaw, Pitch, and Roll for Series of Trials

The RMSE was calculated by running the simulation and recording the data from each trial and using equation 1.

VI. Results and Discussion

A. Run Time Until Failure for Series of Trials

The histogram in Fig. 1 displays the results for run time until failure of the 500 simulations run under the specifications described above. The most critical observations from these trials are as follows:

1) The significant majority of trials met the failure criteria in under 10 seconds of run time. This indicates that the controller and observer system may be tuned further, and this will be revisited in the future.
2) There is a smaller peak in trials found at the 120 s time step. This indicates that a significant number of trials (36) successfully ran the full 120 s of run time without any star going out of view of the scope or any reaction wheel having an angular velocity exceeding a magnitude of 50 rad/s. After further tuning of the controller and observer, the maximum run time may be adjusted.

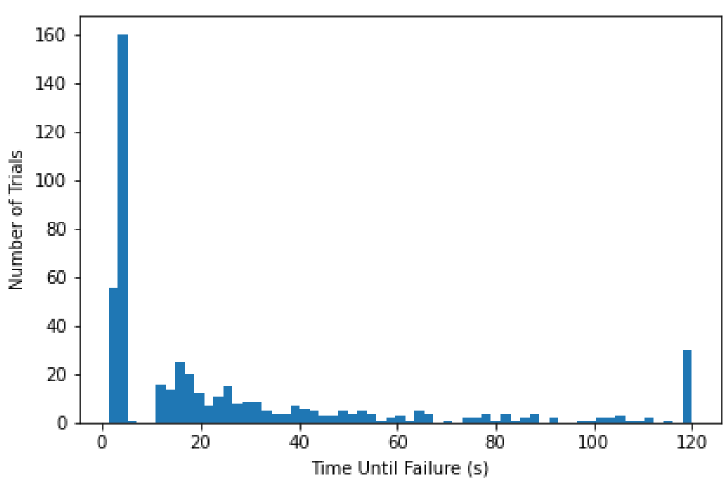

Fig. 1 Time Until Failure Across 500 Simulations

B. RMSE of Yaw, Pitch, and Roll for Series of Trials

The set of histograms in Fig. 2 displays the results for calculated RMSE values for yaw, pitch, and roll for the same 500 simulations run under the specifications described above. The most critical observations from these trials are as follows:

1) The histogram for RMSE results for spacecraft yaw have a right-skewed trend, meaning the majority of results lean towards lower values.
2) The histogram for RMSE results for spacecraft pitch roughly follow a more normal trend, meaning the majority of results are near the middle, while more extreme values are rarer.
3) The histogram for RMSE results for spacecraft roll have a slightly right-skewed trend, while appearing more uniform than the RMSE of Yaw.
4) The RMSE results for roll are significantly greater than those of the yaw and pitch.

From Table 1, we see the 80th percentile values for RMSE for yaw, pitch, and roll. Because all values were found to be below the value of $\frac{\pi}{6}$ established by our requirement, our system successfully passes verification.

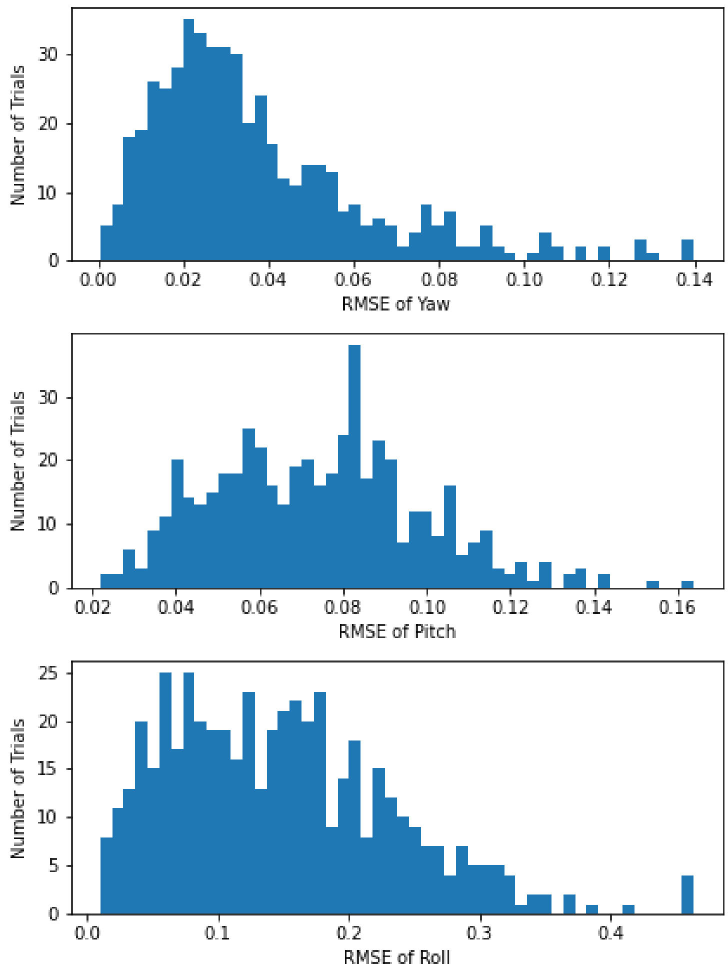

Fig. 2 RMSE Data of Orientation Parameters

80th percentile of RMSE	Value less than $\frac{\pi}{6}$ (0.523598)?
0.0533993 (yaw)	Yes
0.092652 (pitch)	Yes
0.220654 (roll)	Yes

Table 1 Summary of Verification

VII. Conclusion

Acknowledgments

Appendix

Day	Task	Person or People
03/22/2022	Creation and basic organization of Overleaf report document and Github repository	Patrick Thornton
03/23/2022	Completed calculating A and B matrices and documented process in report	Patrick Thornton
03/24/2022	Wrote code for finding C matrix and state space model for sensor. Wrote code for creation of observer.	Kameron Jackson
03/25/2022	Wrote introduction section and added gain matrices and C matrix in theory section.	Kameron Jackson
03/25/2022	Debugged code to ensure proper functionality of matrices and verify that code is both controllable and observable.	Both
03/25/2022	Completed rewriting sensor model for each star and computed values of C matrix Wrote out process for determining C matrix and added the full matrix in report	Patrick Thornton
03/25/2022	Completed verification of controllability and observability in code and documented the process in report	Patrick Thornton
03/25/2022	Finished theory section by finding both gain matrices, K and L, added findings into theory section	Kameron Jackson
03/31/2022	Continued working on finding proper L matrix, encountered few bugs within code (array sizing). Viewed feedback from dp3 draft one submission and started to fix report based on feedback	Kameron Jackson
04/01/2022	Elaborated on finding gain matrices and went into more detail in theory section, providing a more concise view into our methodology	Kameron Jackson
04/01/2020	Wrote requirement and verification of the system, tuned gains matrices for controller and observer, wrote code to extract simulation data, and created histograms for results from multiple simulations. Then wrote results and discussion section.	Patrick Thornton
04/01/2022	Wrote experimental methods section	Kameron Jackson