SC 617

End Semester Project
Adaptive Control of Spacecraft with Reaction Wheels

Saketika Chekuri | 190070054
Nishant Mittal | 190070038
Hiya Gada | 19D100007

1 Introduction

The dynamical system we shall be attempting to apply adaptive control is Attitude Control of a Spacecraft via three reaction wheels, each oriented along one of the principle body axis. Each reaction wheel can be controlled using an independent motor, with the motor torque as the control input. The unknown parameters would be the various moments of inertia of the system as well as the various drag torques on the system. The primary references for the dynamics comes from [3].

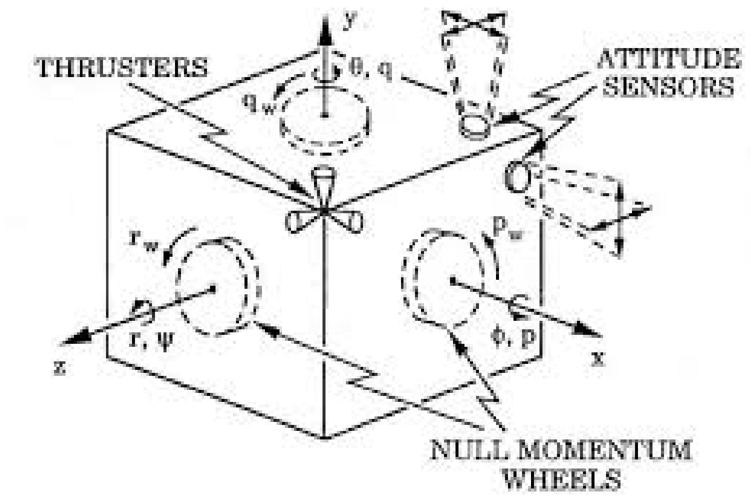


Figure 1: Reaction Wheels Position

As stated on page 38 of [3], the dynamics of the system is given by:

$$[I_{RW}]\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times [I_{RW}]\boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{h_s} - \mathbf{u_s} + \mathbf{L}$$

Where $\boldsymbol{\omega}$ are the rotation rates of the spacecraft around its primary axes as seen from the inertial (ground) frame. $\mathbf{L}$ represents external atmospheric torque from drag, which is a disturbance that we shall be ignore in our analysis. $\mathbf{u_s}$ is our control input torque. The other terms are further discussed below.

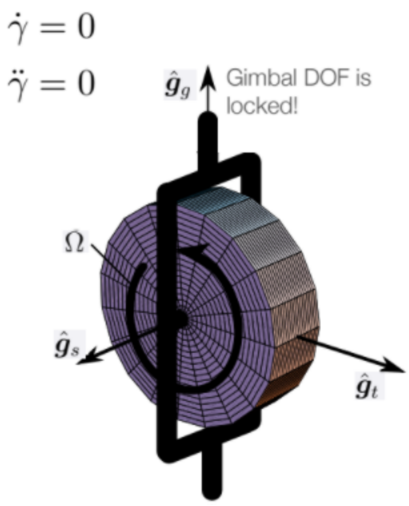


Figure 2: VSCMG with locked axis or Reaction Wheel

The basis vectors $(\hat{g}_s, \hat{g}_t, \hat{g}_g)$ are as shown in the diagram above. We use this notation for each of the 3 reaction wheels, as shown below. Now, as the reactions wheels are aligned with the principle axes of the spacecraft $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$, we have:

$$
\begin{aligned}
&\hat{g}_{s1} = \hat{e}_1 \quad \hat{g}_{t1} = \hat{e}_2 \quad \hat{g}_{g1} = \hat{e}_3 \\
&\hat{g}_{s2} = \hat{e}_2 \quad \hat{g}_{t2} = \hat{e}_3 \quad \hat{g}_{g2} = \hat{e}_1 \\
&\hat{g}_{s3} = \hat{e}_3 \quad \hat{g}_{t3} = \hat{e}_1 \quad \hat{g}_{g3} = \hat{e}_2 \\
&\text{Also,} \\
&\omega_{s1} = \omega_1 \quad \omega_{s2} = \omega_2 \quad \omega_{s3} = \omega_3 \\
&\text{where, } \boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)
\end{aligned}
$$

That gives us:

$$
\begin{aligned}
[I_{RW}] &= [I_s] + \sum_{i=1}^{3} \left(J_{ti} \hat{g}_{ti} \hat{g}_{ti}^\top + J_{gi} \hat{g}_{gi} \hat{g}_{gi}^\top \right) \\
&= [I_s] + \begin{bmatrix} J_{t3} + J_{g2} & 0 & 0 \\ 0 & J_{t1} + J_{g3} & 0 \\ 0 & 0 & J_{t2} + J_{g1} \end{bmatrix}
\end{aligned}
$$

Where $[I_s]$ is the moment of inertia of the spacecraft and J is the moment of inertia with the subscripts representing the t and g axes of the i^{th} reaction wheel as shown in figure 4. Let $\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \Omega_3]^\top$ be the angular velocities of the reaction wheels from the body frame of the spacecraft. h_s is the torque provided by the reaction wheels in the inertial (ground) frame, given as follows,

$$
\mathbf{h_s} = \begin{bmatrix} J_{s1}(\omega_1 + \Omega_1) \\ J_{s2}(\omega_2 + \Omega_2) \\ J_{s3}(\omega_3 + \Omega_3) \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}
$$

Where J_{si} is the moment of inertia of the i^{th} reaction wheel along the s axis as shown in fig 4.
The equation governing $\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \Omega_3]^\top$ (the angular velocities of the reaction wheels) is from pg 30 of [3]:

$$\begin{bmatrix} J_{s1} & 0 & 0 \\ 0 & J_{s2} & 0 \\ 0 & 0 & J_{s3} \end{bmatrix} (\dot{\boldsymbol{\Omega}} + \dot{\boldsymbol{\omega}}) = \mathbf{u_s}$$

Also, to work with the orientation or attitudes of the spacecraft, we shall be using MRP's (properties of which are explained in [2]). Let the angular velocity tracking error be $\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \boldsymbol{\omega_r}$, then the attitude tracking error is given as:

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}[(1-\sigma^2)[I_{3x3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top]\boldsymbol{\delta\omega}$$

Then total dynamics of the system is given by:

$$\boxed{\begin{aligned} [I_{RW}]\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times [I_{RW}]\boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{h_s} - \mathbf{u_s} + \mathbf{L} && (1) \\ [J_s]\dot{\boldsymbol{\Omega}} &= \mathbf{u_s} - [J_s][I_{RW}]^{-1}(-\boldsymbol{\omega} \times [I_{RW}]\boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{h_s} - \mathbf{u_s} + \mathbf{L}) && (2) \\ \dot{\boldsymbol{\sigma}} &= \frac{1}{4}[(1-\sigma^2)[I_{3x3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top]\boldsymbol{\delta\omega} && (3) \end{aligned}}$$

Here, the unknown parameters we are targeting are: $[I_{RW}]$ and $[J_s]$. $\mathbf{u_s}$ is the control input or the commanded motor torque.

2 State-space equations

To simplify our problem, we define state variables $\boldsymbol{a}$ and $\boldsymbol{b}$ as the following,

$$\boldsymbol{a} = [I_{RW}]\boldsymbol{\omega}$$
$$\boldsymbol{b} = [J_s](\boldsymbol{\Omega} + \boldsymbol{\omega})$$

This is a useful definition because the reference $\boldsymbol{\omega}_r$ is 0, which means that the reference $\boldsymbol{a}_r$ don't depend on the parameters and are also equal to 0.
We do not care about the control of $\boldsymbol{\Omega}$, as long as the value is bounded (explained in the next section). As $[J_s]$ is a constant parameter, $\boldsymbol{b}$ must also be bounded. This will be an assumption we take (explained in further sections as well).
Thus we define our final state vector $\boldsymbol{x}$ as the following,

$$\boldsymbol{x} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\sigma})^\top$$

Assume that the parameters are unknown but constant, $[\dot{I_{RW}}] = [\dot{J_s}] = 0$.
We thus get the following dynamics of the system, taking $\boldsymbol{L} = 0$, as we are ignoring disturbance,

$$\dot{\boldsymbol{a}} = -([I_{RW}]^{-1}\boldsymbol{a}) \times \boldsymbol{a} - ([I_{RW}]^{-1}\boldsymbol{a}) \times \mathbf{b} - \mathbf{u_s} \tag{4}$$
$$\dot{\boldsymbol{b}} = \mathbf{u_s} \tag{5}$$
$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}[(1-\sigma^2)[I_{3x3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top]\boldsymbol{\omega} \tag{6}$$

For axes of our choice match the principle axes of our body which results in the inertia matrix being a diagonal matrix, thus $[I_{RW}]$ will also be a diagonal matrix. Let it be,

$$[I_{RW}] = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

The dynamics can be written in the form,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + F(\boldsymbol{x})\boldsymbol{\theta} + g(\boldsymbol{x})\boldsymbol{u}_s$$

where $\boldsymbol{u}_s = (u_1, u_2, u_3)$. Our parameter vector will be,

$$\boldsymbol{\theta} = \left(\frac{1}{I_1}, \frac{1}{I_2}, \frac{1}{I_3}\right)^\top$$

By comparison we also see,

$$f(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$$

$$g(\boldsymbol{x}) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F(\boldsymbol{x}) = \begin{bmatrix} [F_a] \\ [F_b] \\ [F_c] \end{bmatrix}$$

Where,

$$[F_a] = \begin{bmatrix} 0 & -a_2(a_3+b_3) & a_3(a_2+b_2) \\ a_1(a_3+b_3) & 0 & -a_3(a_1+b_1) \\ -a_1(a_2+b_2) & a_2(a_1+b_1) & 0 \end{bmatrix}$$

$$[F_b] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[F_c] = \begin{bmatrix} a_1\left(\frac{{\sigma_1}^2}{4} - \frac{{\sigma_2}^2}{4} - \frac{{\sigma_3}^2}{4} + \frac{1}{4}\right) & -a_2\left(\frac{\sigma_3}{2} - \frac{\sigma_1\,\sigma_2}{2}\right) & a_3\left(\frac{\sigma_2}{2} + \frac{\sigma_1\,\sigma_3}{2}\right) \\ a_1\left(\frac{\sigma_3}{2} + \frac{\sigma_1\,\sigma_2}{2}\right) & -a_2\left(\frac{{\sigma_1}^2}{4} - \frac{{\sigma_2}^2}{4} + \frac{{\sigma_3}^2}{4} - \frac{1}{4}\right) & -a_3\left(\frac{\sigma_1}{2} - \frac{\sigma_2\,\sigma_3}{2}\right) \\ -a_1\left(\frac{\sigma_2}{2} - \frac{\sigma_1\,\sigma_3}{2}\right) & a_2\left(\frac{\sigma_1}{2} + \frac{\sigma_2\,\sigma_3}{2}\right) & -a_3\left(\frac{{\sigma_1}^2}{4} + \frac{{\sigma_2}^2}{4} - \frac{{\sigma_3}^2}{4} - \frac{1}{4}\right) \end{bmatrix}$$

3 Tracking Objective

Our tracking objective is to ensure that $\boldsymbol{\omega} \to 0$ ($\boldsymbol{\omega}_r = 0$) and $\boldsymbol{\sigma} \to 0$ ($\boldsymbol{\sigma}_r = 0$). We do not worry about $\boldsymbol{\Omega}$ of the reaction wheels, as our aim is to ensure that the space craft has attitude control as a whole. We only want $\boldsymbol{\Omega}$ to be bounded (assumption explained ahead).
Thus, for our new defined state variables our tracking objective is,

$$\boldsymbol{a} \to 0 \text{ or } \boldsymbol{a}_r = 0$$
$$\boldsymbol{\sigma} \to 0 \text{ or } \boldsymbol{\sigma}_r = 0$$

where $\boldsymbol{a}_r$ and $\boldsymbol{\sigma}_r$ are the reference trajectories.

4 Non-Adaptive Tracking Control

Introducing some notation,

$$B(\boldsymbol{\sigma}) = (1-\sigma^2)[I_{3x3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top$$

$$[a_I] = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix}$$

$$[\sigma_I] = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

Also, note that the following relations hold:

$$\boldsymbol{\sigma}^T B(\boldsymbol{\sigma}) = (1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\sigma}^\top \tag{7}$$

$$\boldsymbol{\sigma}^\top[a_I] = \boldsymbol{a}^\top[\sigma_I] \tag{8}$$

We simplify our dynamics in 4, 5 and 6 as shown below,

$$\dot{\boldsymbol{a}} = [F_a]\boldsymbol{\theta} - \boldsymbol{u_s} \tag{9}$$

$$\dot{\boldsymbol{b}} = [F_b]\boldsymbol{\theta} + \boldsymbol{u_s} = \boldsymbol{u_s} \tag{10}$$

$$\dot{\boldsymbol{\sigma}} = [F_c]\boldsymbol{\theta} = \frac{1}{4}B(\boldsymbol{\sigma})[a_I]\boldsymbol{\theta} \tag{11}$$

Here, we assume the value of parameter is known. Let us take the following Lyapunov function, which is a function of $\boldsymbol{a}$ and $\boldsymbol{\sigma}$,

$$V = \frac{1}{2}\boldsymbol{a}^\top\boldsymbol{a} + 2K\ln(1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma})$$

where, K is a positive constant. Note that the Lyapunov function that we have chosen is *positive semi-definite*. We differentiate to get the following,

$$\dot{V} = \boldsymbol{a}^\top\dot{\boldsymbol{a}} + \frac{2K(2\boldsymbol{\sigma}^\top\dot{\boldsymbol{\sigma}})}{1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma}}$$

Substituting for $\dot{\boldsymbol{a}}$, $\dot{\boldsymbol{\sigma}}$ from above and using 7, 8 we get:

$$\begin{aligned}\dot{V} &= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + \frac{4K\boldsymbol{\sigma}^\top}{1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma}}\cdot\frac{1}{4}B(\boldsymbol{\sigma})[a_I]\boldsymbol{\theta} \\ &= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + K\boldsymbol{\sigma}^\top[a_I]\boldsymbol{\theta} \\ &= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + K\boldsymbol{a}^\top[\sigma_I]\boldsymbol{\theta} \\ &= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} + K[\sigma_I]\boldsymbol{\theta} - \boldsymbol{u_s}\right)\end{aligned}$$

To make $\dot{V} \leq 0$ (*negative semi-definite*), we need the RHS of this equation to reduce to $-c_1\boldsymbol{a}^\top\boldsymbol{a}$, where c_1 is some positive constant. We choose the control law that satisfies the following relation:

$$\boldsymbol{u_s} = [F_a]\boldsymbol{\theta} + K[\sigma_I]\boldsymbol{\theta} + c_1\boldsymbol{a} \tag{12}$$

The Lyapunov function we have chosen is only *positive semi-definite*, which rules out using Lyapunov theorems to show stability. Therefore, we use signal chasing arguments to show convergence.

1. We have $V \geq 0$ and $\dot{V} \leq 0$, which implies V_∞ exists and is finite. Therefore, V is finite at all times since it is lower bounded and non-increasing $\implies \boldsymbol{a}, \boldsymbol{\sigma} \in \mathcal{L}_\infty$.

2. Since $\dot{V}$ is integrable, $\boldsymbol{a} \in \mathcal{L}_2$.

3. $\boldsymbol{u}_s$ being our control input is bounded, so $\boldsymbol{u}_s \in \mathcal{L}_\infty$.

4. We assume $\boldsymbol{u}_s$ is integrable (a fairly valid assumption for practical scenarios). This implies that $\boldsymbol{b} \in \mathcal{L}_\infty$. Note that this assumption also ensures that $\boldsymbol{\Omega}$ is bounded (because $\boldsymbol{\omega}$ converges as we shall see), satisfying our control objective.

5. From 4, we see that $\dot{\boldsymbol{a}}$ depends only on $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{u}_s$ and $\boldsymbol{\theta}$, which are all bounded, so $\dot{\boldsymbol{a}} \in \mathcal{L}_\infty$

6. We have $\boldsymbol{a} \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and $\dot{\boldsymbol{a}} \in \mathcal{L}_\infty$. From the corollary of Barbalat's lemma, we have $\lim_{t\to\infty} \boldsymbol{a} = 0$.

7. From 11, we see that all terms on RHS are bounded $\implies \dot{\boldsymbol{\sigma}} \in \mathcal{L}_\infty \implies \boldsymbol{\sigma}$ is uniformly continuous.

8. We see that $\boldsymbol{b}$, being $\int \boldsymbol{u}_s$ will be continuous. Since $\dot{\boldsymbol{a}}$ is bounded, $\boldsymbol{a}$ will also be continuous. Therefore, $[F_a]$ will also be continuous.

9. From 12, we see that all of $\boldsymbol{u}_s$, $\boldsymbol{a}$, $[F_a]$ are bounded and continuous, so they will also be component wise integrable, forcing $\boldsymbol{\sigma}$ to be component-wise integrable as well.

10. From Barbalat's lemma, since $\boldsymbol{\sigma}$ is component wise integrable, and is uniformly continuous, $\lim_{t\to\infty} \boldsymbol{\sigma} = 0$

5 Adaptive Tracking Control

Using the certainty equivalence principle, we replace all occurrences of the unknown parameters by their estimates in the control law.

$$\boldsymbol{u_s} = [F_a]\hat{\theta} + K[\sigma_I]\hat{\theta} + c_1\boldsymbol{a} \tag{13}$$

Since here, we only use an estimate of the parameter, we use a new Lyapunov equation, that is a function of $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\sigma}, \tilde{\boldsymbol{\theta}} \left(= \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right)$:

$$\begin{aligned}
V &= \frac{1}{2}\boldsymbol{a}^\top\boldsymbol{a} + 2K\ln(1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma}) + \frac{1}{2\gamma}\tilde{\boldsymbol{\theta}}^\top\tilde{\boldsymbol{\theta}} \\
\dot{V} &= \boldsymbol{a}^\top\dot{\boldsymbol{a}} + \frac{4K\boldsymbol{\sigma}^\top\dot{\boldsymbol{\sigma}}}{1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma}} + \frac{1}{\gamma}\dot{\tilde{\boldsymbol{\theta}}}^\top\tilde{\boldsymbol{\theta}} \\
\dot{V} &= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + \frac{4K\boldsymbol{\sigma}^\top}{1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma}}\cdot\frac{1}{4}B(\boldsymbol{\sigma})[a_I]\boldsymbol{\theta} - \frac{1}{\gamma}\dot{\hat{\boldsymbol{\theta}}}^\top\tilde{\boldsymbol{\theta}} \\
&= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + K\boldsymbol{\sigma}^\top[a_I]\boldsymbol{\theta} - \frac{1}{\gamma}\dot{\hat{\boldsymbol{\theta}}}^\top\tilde{\boldsymbol{\theta}} \\
&= \boldsymbol{a}^\top\left([F_a]\boldsymbol{\theta} - \boldsymbol{u_s}\right) + K\boldsymbol{a}^\top[\sigma_I]\boldsymbol{\theta} - \frac{1}{\gamma}\dot{\hat{\boldsymbol{\theta}}}^\top\tilde{\boldsymbol{\theta}} \\
&= \boldsymbol{a}^\top\left([F_a]\tilde{\theta} + K[\sigma_I]\tilde{\theta} - c_1\boldsymbol{a}\right) - \frac{1}{\gamma}\dot{\hat{\boldsymbol{\theta}}}^\top\tilde{\boldsymbol{\theta}} \\
&= (\boldsymbol{a}^\top[F_a] + \boldsymbol{a}^\top K[\sigma_I] - \frac{1}{\gamma}\dot{\hat{\boldsymbol{\theta}}}^\top)\tilde{\boldsymbol{\theta}} - c_1\boldsymbol{a}^\top\boldsymbol{a}
\end{aligned}$$

Here, $\boldsymbol{u_s}$ is the modified control law that contains the parameter estimates. Take $\dot{\hat{\theta}}$ as follows,

$$\dot{\hat{\boldsymbol{\theta}}} = \gamma\boldsymbol{a}^\top([F_a] + K[\sigma_I]) \tag{14}$$

Thus we get $\dot{V} = -c_1\boldsymbol{a}^\top\boldsymbol{a} \leq 0$ (*negative semi-definite*).
With $V \geq 0$ and $\dot{V} \leq 0$, we carry out signal chasing analysis to show convergence.

1. We have $V \geq 0$ and $\dot{V} \leq 0$, which implies V_∞ exists and is finite. Therefore, V is finite at all times since it is lower bounded and non-increasing $\implies$ $\boldsymbol{a}$, $\boldsymbol{\sigma}$, $\tilde{\boldsymbol{\theta}} \in \mathcal{L}_\infty$.
2. Since $\dot{V}$ is integrable, $\boldsymbol{a} \in \mathcal{L}_2$.
3. $\boldsymbol{u}_s$ being our control input is bounded, so $\boldsymbol{u}_s \in \mathcal{L}_\infty$.
4. We assume $\boldsymbol{u}_s$ is integrable (a fairly valid assumption for practical scenarios). This implies that $\boldsymbol{b} \in \mathcal{L}_\infty$. Note that this assumption also ensures that $\boldsymbol{\Omega}$ is bounded (because $\boldsymbol{\omega}$ converges as we shall see), satisfying our control objective.
5. From 4, we see that $\dot{\boldsymbol{a}}$ depends only on $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{u}_s$ and $\boldsymbol{\theta}$, which are all bounded, so $\dot{\boldsymbol{a}} \in \mathcal{L}_\infty$
6. We have $\boldsymbol{a} \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and $\dot{\boldsymbol{a}} \in \mathcal{L}_\infty$. From the corollary of Barbalat's lemma, we have $\lim_{t\to\infty}\boldsymbol{a} = 0$.

7. From 11, we see that all terms on RHS are bounded $\implies \dot{\boldsymbol{\sigma}} \in \mathcal{L}_\infty \implies \boldsymbol{\sigma}$ is uniformly continuous.

8. We see that $\boldsymbol{b}$, being $\int \boldsymbol{u}_s$ will be continuous. Since $\dot{\boldsymbol{a}}$ is bounded, $\boldsymbol{a}$ will also be continuous. Therefore, $[F_a]$ will also be continuous.

9. $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}$ are bounded, so $\hat{\boldsymbol{\theta}}$ is also bounded. Moreover, it is also continuous, since all terms on RHS are continuous in 14.

10. From 13, we see that all of $\boldsymbol{u}_s$, $\boldsymbol{a}$, $[F_a]$, $\hat{\boldsymbol{\theta}}$ are bounded and continuous, so they will also be component wise integrable, forcing $\boldsymbol{\sigma}$ to be component-wise integrable as well.

11. From Barbalat's lemma, since $\boldsymbol{\sigma}$ is component wise integrable, and is uniformly continuous, $\lim_{t\to\infty} \boldsymbol{\sigma} = 0$,

6 Simulations and Results

The parameters such as inertia and different initial conditions have been derived from [1]. This reference involves a non-adaptive control design. Hence, it was an ideal choice to test our adaptive control design. All the code can be found in the Appendix.
The simulation was performed for 2 minutes, well above the settling time of the states and parameters. A discrete time step of 0.01s was chosen to avoid major integration errors. An RK4 integrator was used to perform the update steps in each loop.
The gain values chosen:

- $c_1 = 1.2$
- $K = 0.1$
- $\gamma = 0.01$

The main objective is to drive the angular velocities to 0. We also need attitude error convergence, however, we would want to avoid oscillations and large fluctuations in the attitude of the spacecraft as this could lead to mechanical stress on the system. Hence, it would be prudent to slow down the spacecraft first, then worry about pointing it in the right direction. Thus, the gain of a was chosen to be an order of magnitude higher than that of σ. Note, large values for the control were not chosen as there is a limited amount of fuel and energy on-board. The trade-off lies between speed of convergence and control magnitude. The parameters were finally tuned to avoid oscillations. Listing the parameters and initial conditions used:

$$I_s = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7.5 \end{bmatrix} kg - m^2$$

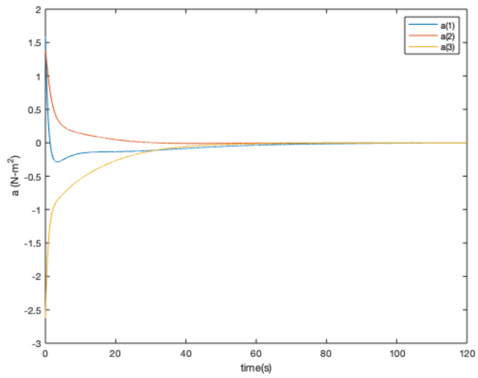
$$J_s = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} kg - m^2$$

$$\omega_i = \begin{bmatrix} 1 \\ 1.75 \\ -2.2 \end{bmatrix} deg/s$$

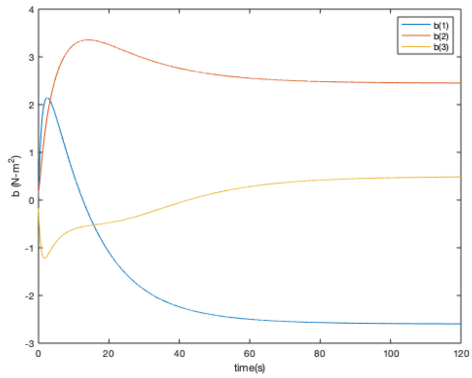
$$\Omega_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\sigma_i = \begin{bmatrix} 0.3 \\ -0.4 \\ 0.5 \end{bmatrix}$$

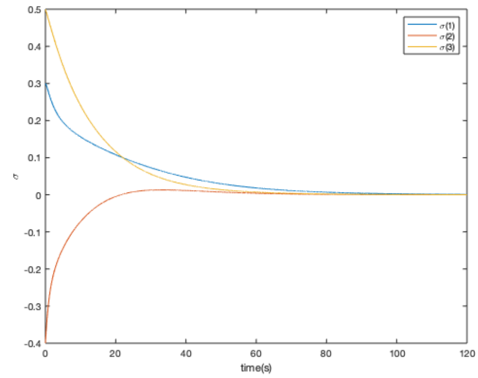
The plots of the simulations are shown below:



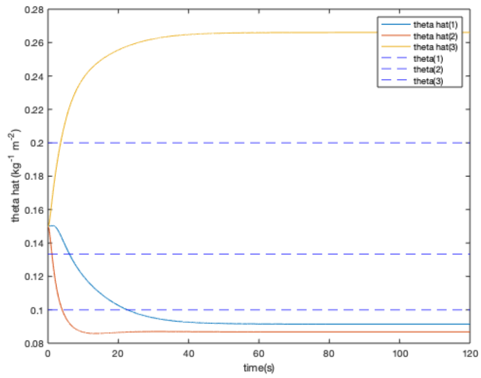
(a) Evolution of a



(b) Evolution of b

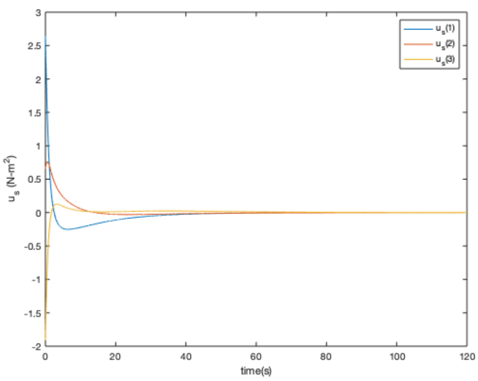


(c) Evolution of σ

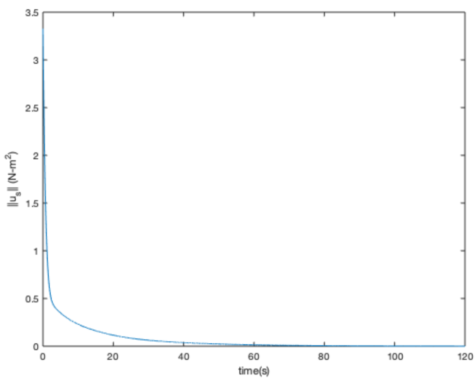


(d) Evolution of parameter estimate

Figure 3: State evolution of modified system

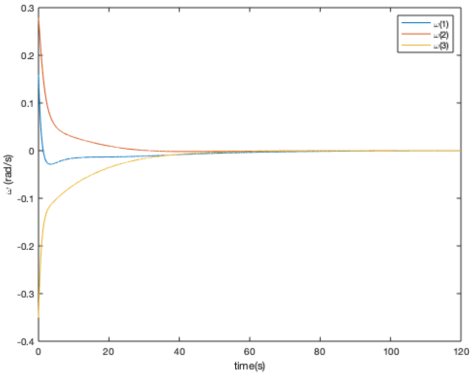


(a) u_s

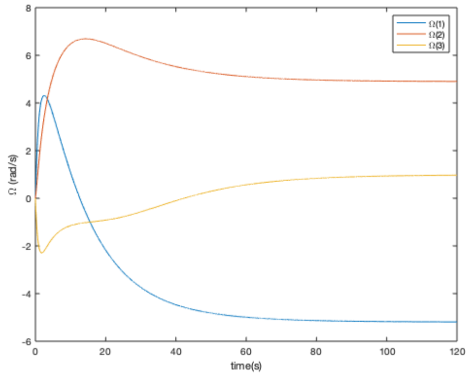


(b) Control Magnitude

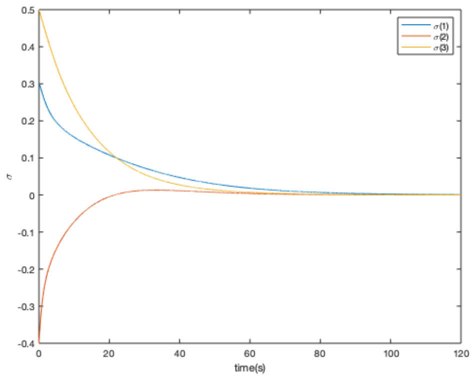
Figure 4: Control Effort



(a) Evolution of ω



(b) Evolution of Ω



(c) Evolution of σ

Figure 5: State evolution of original system

As can be seen from fig 5, the angular rotation rate as well as the attitude error converges to 0 within a minute. As expected, the spacecraft first slows down and then slowly adjusts its attitude.
Note that Ω does not converge to 0, however that is not an issue as the control effort converges to 0 as seen in fig 4. Hence, even if the wheels are spinning, we are not applying any effort(no torque input). As we have reached our objective of stopping the spacecraft from rotating and pointing it in the desired direction, we can claim that the controller works.

Appendices

A Main Files

Listing 1: Main Code

```
clear all
% sim setup
tEnd = 120;
h = 0.01;
t = 0:h:tEnd;
simVec = 1:1:size(t,2)-1;

% Physical - Inital Parameters - define stuff here!
Is = [10, 0, 0; ...
    0, 5, 0; ...
    0, 0, 7.5];

Js = 0.5*eye(3);

omega_inital = [1; 1.75; -2.2]./(2*pi);
Omega_inital = [0; 0; 0];
sigma_inital = [0.3; -0.4; 0.5];

% define values of control gain
gains.c1 = 1.2;
gains.K = 0.1;
gains.gamma = 0.01;

% mapping given parameters to our variable names
theta = [1/Is(1,1);1/Is(2,2);1/Is(3,3)];

% stuff for plotting
x = zeros(9,size(t,2));
control = zeros(3,size(t,2));
theta_hat = zeros(3,size(t,2));

% define intial conditions
x(1:3,1) = Is*omega_inital;
x(4:6,1) = Js*(omega_inital + Omega_inital);
x(7:9,1) = sigma_inital;
theta_hat(:,1) = [0.15; 0.15; 0.15];

for i = simVec
    control(:,i) = us(x(:,i),theta_hat(:,i),gains);
    theta_hat(:,i+1) = thetanext(x(:,i), theta_hat(:,i), gains, h);
    x(:,i+1) = xnext(x(:,i), control(:,i), theta, h);
end
```

```

% plotting stuff

figure;
plot(t,x(1:3,:))
xlabel('time(s)')
ylabel('a (N-m^2)')
legend('a(1)', 'a(2)', 'a(3)')

figure;
plot(t,x(4:6,:))
xlabel('time(s)')
ylabel('b (N-m^2)')
legend('b(1)', 'b(2)', 'b(3)')

figure;
plot(t,x(7:9,:))
xlabel('time(s)')
ylabel('\sigma')
legend('\sigma(1)', '\sigma(2)', '\sigma(3)')

figure;
plot(t,theta_hat)
yline(theta(1),'-- b');
yline(theta(2),'-- b');
yline(theta(3),'-- b');
xlabel('time(s)')
ylabel('theta hat (kg^{-1} m^{-2})')
legend('theta hat(1)', 'theta hat(2)', 'theta hat(3)', 'theta(1)', 'theta
    (2)', 'theta(3)')

figure;
plot(t,control)
xlabel('time(s)')
ylabel('u_s (N-m^2)')
legend('u_s(1)', 'u_s(2)', 'u_s(3)')

figure;
plot(t, vecnorm(control))
xlabel('time(s)')
ylabel('||u_s|| (N-m^2)')

% de-mapping and plotting from a,b, \sigma to \omega, \Omega, \sigma
omega = inv(Is)*x(1:3,:);
Omega = inv(Js)*x(4:6,:) - omega;

figure;
plot(t,omega)
```

```
xlabel('time(s)')
ylabel('\omega (rad/s)')
legend('\omega(1)', '\omega(2)', '\omega(3)')

figure;
plot(t,Omega)
xlabel('time(s)')
ylabel('\Omega (rad/s)')
legend('\Omega(1)', '\Omega(2)', '\Omega(3)')
```

Listing 2: Controller

```
function us = us(x,theta_hat,gains)
%us Controller function goes here

a = x(1:3);
b = x(4:6);
sigma = x(7:9);

sigmaI = [sigma(1), 0, 0; ...
          0, sigma(2), 0; ...
          0, 0, sigma(3)];

us = gains.c1*a + Fa(x)*theta_hat + gains.K*sigmaI*theta_hat;
end
```

Listing 3: Parameter Update

```
function thetaDot = thetaDot(x,gains)
%thetaDot Parameter estimate update Law

a = x(1:3);
b = x(4:6);
sigma = x(7:9);

sigmaI = [sigma(1), 0, 0; ...
          0, sigma(2), 0; ...
          0, 0, sigma(3)];

thetaDot = gains.gamma*(Fa(x) + gains.K*sigmaI)'*a;
end
```

Listing 4: Dynamics

```
function xdot = dynamics(x,us,theta)
%dynamics Gives xdot upon giving the xvector

    xdot(1:3) = Fa(x)*theta - us;
    xdot(4:6) = us;
    xdot(7:9) = Fc(x)*theta;
```

```
    xdot = xdot';
end
```

Listing 5: F_a

```
function Fa = Fa(x)
%Fa Fa for adot

aI = [x(1), 0, 0; ...
     0, x(2), 0; ...
     0, 0, x(3)];
Fa = tilde((x(1:3) + x(4:6)))*aI;
end
```

Listing 6: F_c

```
function Fc = Fc(x)
%Fc Fc for cdot

aI = [x(1), 0, 0; ...
     0, x(2), 0; ...
     0, 0, x(3)];
Fc = (1/4)*BmatMRP(x(7:9))*aI;
end
```

B RK4 Integrators

Listing 7: Dynamics Update

```
function xnext = xnext(x, us, theta,h)
%xnext gives next x based on dynamics and rk4 integrator

k_1 = dynamics(x,us,theta);
k_2 = dynamics((x+0.5*h*k_1),us,theta);
k_3 = dynamics((x+0.5*h*k_2),us,theta);
k_4 = dynamics((x+k_3*h),us,theta);
xnext = x + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;
xnext(7:9) = MRPswitch(xnext(7:9),1);
end
```

Listing 8: Theta Update

```
function thetanext = thetanext(x, theta_hat, gains, h)
%xnext gives next x based on dynamics and rk4 integrator

k_1 = thetaDot(x, gains);
thetanext = theta_hat + k_1*h;
end
```

C Supporting Files

Listing 9: MRP Switching to Shadow Set

```
function s = MRPswitch(q,s2)

% MRPswitch
%
%       S = MRPswitch(Q,s2) checks to see if norm(Q) is larger than s2.
%       If yes, then the MRP vector Q is mapped to its shadow set.
%

q2 = q'*q;
if (q2>s2*s2)
        s = -q/q2;
else
        s = q;
end
```

Listing 10: B(σ)

```
function B = BmatMRP(q)

% BmatMRP(Q)
%
%       B = BmatMRP(Q) returns the 3x3 matrix which relates the
%       body angular velocity vector w to the derivative of
%       MRP vector Q.
%
%               dQ/dt = 1/4 [B(Q)] w
%

s2 = q'*q;
B(1,1) = 1-s2+2*q(1)*q(1);
B(1,2) = 2*(q(1)*q(2)-q(3));
B(1,3) = 2*(q(1)*q(3)+q(2));
B(2,1) = 2*(q(2)*q(1)+q(3));
B(2,2) = 1-s2+2*q(2)*q(2);
B(2,3) = 2*(q(2)*q(3)-q(1));
B(3,1) = 2*(q(3)*q(1)-q(2));
B(3,2) = 2*(q(3)*q(2)+q(1));
B(3,3) = 1-s2+2*q(3)*q(3);
```

Listing 11: Tilde Operator

```
function vecTilde = tilde(vec)
vecTilde = [0, -vec(3), vec(2); ...
            vec(3), 0, -vec(1); ...
            -vec(2), vec(1), 0];
end
```

References

[1] Prof. H. Schaub. *Course Project*. URL: https://drive.google.com/file/d/1sTDNY3fKjBdl1kAcupigkF TzzN/view?usp=sharing.

[2] Prof. H. Schaub. *Modified Rodrigues Parameters*. URL: https://drive.google.com/file/d/1Rt_uLJjodq1O3sMHiJyTHKz2kCc59O4X/view?usp=sharing.

[3] Prof. H. Schaub. *Momentum Exchange Devices*. URL: https://drive.google.com/file/d/1F8s5VM7SRRYPg6wLV53MepebGj2hQUsR/view?usp=sharing.