Adaptive Attitude Control of a Spacecraft SC617 Final Report

Saad Khan
200100085

***Abstract*—The following report presents the control design methods followed by analysis to solve the adaptive attitude control problem for a Spacecraft with either known or unknown inertia matrix.**

I. Spacecraft Dynamics

The rotation dynamics of a 3-D rigid body presented here represents the rotation of the body as a quaternion $\boldsymbol{q} = \begin{bmatrix} q_0 & \boldsymbol{q_v}^T \end{bmatrix}^T$ where $q_0 \in \mathbb{R}$ and $\boldsymbol{q_v} \in \mathbb{R}^3$ such that the quaternion is normalized, *i.e.*,

$$q_0^2 + ||\boldsymbol{q_v}||^2 = 1$$

the angular velocity is represented by $\boldsymbol{\omega} \in \mathbb{R}^3$. The quaternion and the angular velocity dynamics are given by the following equations:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{\boldsymbol{q}}_{\boldsymbol{v}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{q_v}^T \\ q_0 \boldsymbol{I} + S(\boldsymbol{q_v}) \end{bmatrix} \boldsymbol{\omega} := \frac{1}{2} E(\boldsymbol{q}) \boldsymbol{\omega}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{u}$$

Where $S(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$ and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is the inertia matrix and $\mathbf{u} \in \mathbb{R}^3$ is the control input.

A. Error Dynamics

The error quaternion is denoted by $\boldsymbol{s}$ and the angular velocity error is denoted by $\delta\boldsymbol{\omega}$. The angular velocity error with respect to a desired angular velocity $\boldsymbol{\omega}_d$ is given by,

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{R_s}\boldsymbol{\omega}_d$$

Where $\boldsymbol{R_s}$ is the rotation matrix corresponding to the error quaternion $\boldsymbol{s}$ and is defined by,

$$\boldsymbol{R_s} = \boldsymbol{R_q}\boldsymbol{R_{qd}}^T$$

Where $\boldsymbol{R_q}$ and $\boldsymbol{R_{qd}}$ are the rotation matrices corresponding to the current orientation quaternion $\boldsymbol{q}$ and the desired orientation quaternion $\boldsymbol{q_d}$. To obtain the rotation matrix corresponding to a quaternion $\boldsymbol{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$, the following conversion can be used:

$$\boldsymbol{R_q} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

The error quaternion and the angular velocity error dynamics are then given by :

$$\dot{\boldsymbol{s}} = \frac{1}{2} E(\boldsymbol{s}) \delta\boldsymbol{\omega} \tag{1}$$

$$\mathbf{J}\dot{\delta\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{u} - \mathbf{J}(\dot{\boldsymbol{R}}_{\boldsymbol{s}}\boldsymbol{\omega}_d + \boldsymbol{R_s}\dot{\boldsymbol{\omega}}_d) \tag{2}$$

With $\dot{\boldsymbol{R}}_{\boldsymbol{s}}$ given by:

$$\dot{\boldsymbol{R}}_{\boldsymbol{s}} = -S(\delta\boldsymbol{\omega})\boldsymbol{R_s}$$

B. Spacecraft Data

All the simulations performed are with respect to the following initial conditions:

$$\boldsymbol{q_v}(0) = \begin{bmatrix} 0.1826 & 0.1826 & 0.1826 \end{bmatrix}^T$$

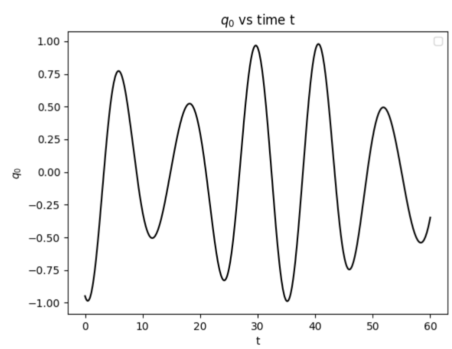
$$q_0(0) = -\sqrt{1 - ||\boldsymbol{q_v}(0)||^2}$$

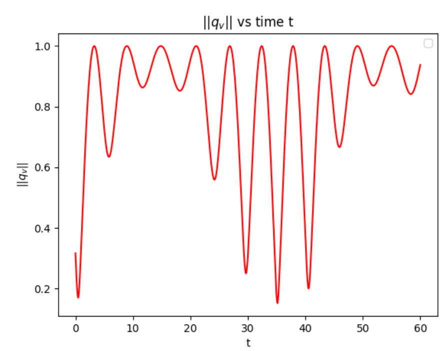
$$\mathbf{J} = \begin{bmatrix} 20 & 1.2 & 0.9 \\ 1.2 & 17 & 1.4 \\ 0.9 & 1.4 & 15 \end{bmatrix}$$

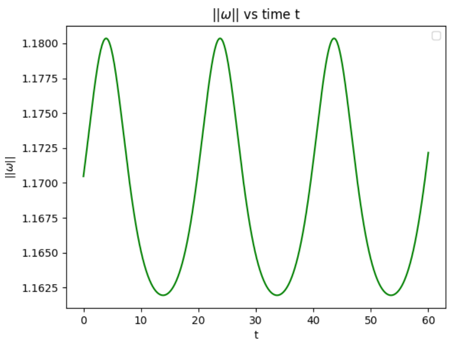
$$\boldsymbol{\omega}(0) = \begin{bmatrix} 0.1 & 0.6 & 1.0 \end{bmatrix}^T$$

C. Uncontrolled Simulation (Task 1)

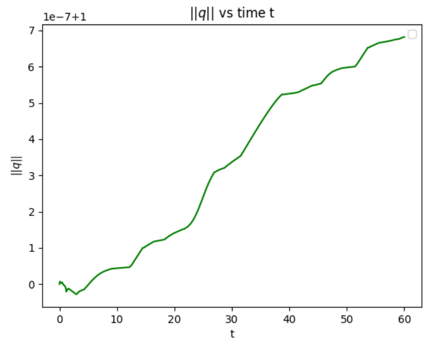
This section presents the results of simulating the spacecraft attitude dynamics in the absence of any control input for 60 seconds.







Clearly the trajectories are **bounded**. To validate the simulation model we plot the norm of the quaternion $\boldsymbol{q}$.



Note that the y-axis is of the order $10^{-7}+1$, validating the quaternion normalization condition.

II. Attitude Stabilization

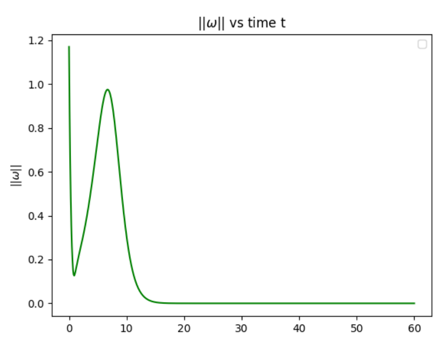
As derived in the problem statement, taking the control input,

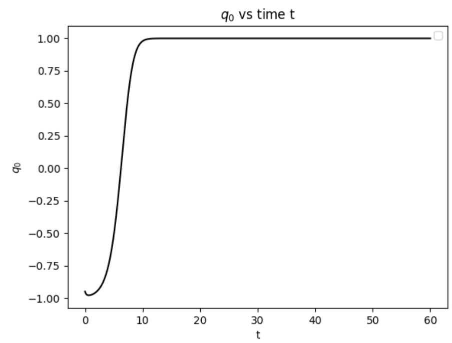
$$\boldsymbol{u} = -k_p \boldsymbol{s_v} - k_2 \delta \boldsymbol{\omega} - W\boldsymbol{\theta*}$$

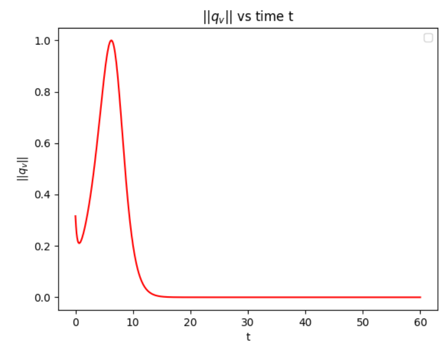
Where $W\boldsymbol{\theta^*} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{J}(\dot{\boldsymbol{R_s}}\boldsymbol{\omega}_d + \boldsymbol{R_s}\dot{\boldsymbol{\omega}}_d)$, for notational conveniences denote $\boldsymbol{\phi} = \dot{\boldsymbol{R_s}}\boldsymbol{\omega}_d + \boldsymbol{R_s}\dot{\boldsymbol{\omega}}_d$.

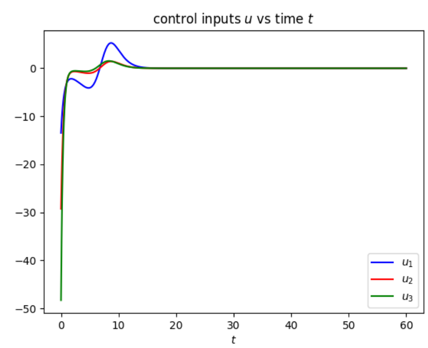
A. Simulation Results (Task 2)

The controller above is simulated for the same initial conditions and the following plots are obtained.









III. Trajectory Tracking

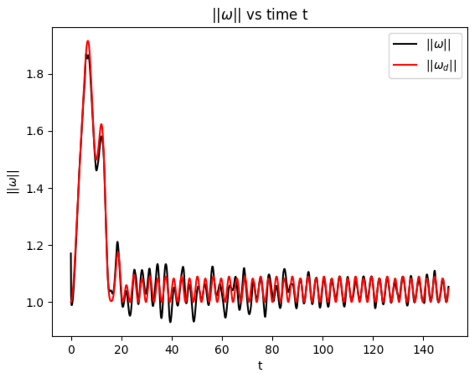
The goal of this section is to track a desired trajectory,

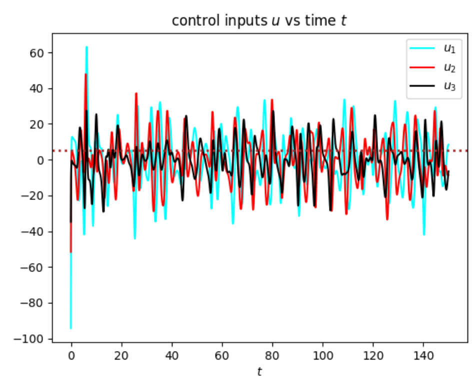
$$\boldsymbol{\omega}_d = \begin{bmatrix} 0.3\cos(t)(1-e^{-0.01t^2}) + (0.08\pi + 0.006\sin(t))te^{-0.01t^2} \\ 0.3\cos(t)(1-e^{-0.01t^2}) + (0.08\pi + 0.006\sin(t))te^{-0.01t^2} \\ 1 \end{bmatrix}$$

With the desired initial orientation $\boldsymbol{q_d}(0) = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$.

A. Simulation Results (Task 3)

Using the same controller as in section 2 we obtain the following plots. As we can see the norm of the angular velocity





tracks the norm of the desired angular velocity sufficiently enough and the scalar part of the error quaternion converges to 1, while the vector components of the error quaternion converge to 0. The control inputs were not bounded by 5N/m in each axis, which may be a result of insufficient tuning.

IV. Integrator Backstepping

This section presents an integrator backstepping based method to desing a control law when the inertia matrix $\mathbf{J}$ is known.

A. Control Design

We now derive the control law and the adaptation law for the error dynamics using integrator backstepping. Consider the $\boldsymbol{s}$ dynamics:

$$\dot{\boldsymbol{s}} = \frac{1}{2}E(\boldsymbol{s})\delta\boldsymbol{\omega}$$

In order to stabilise $\boldsymbol{s}$, we want $\delta\boldsymbol{\omega} = -k_p\boldsymbol{s_v}$. Let,

$$V_1 = \boldsymbol{s_v}^T\boldsymbol{s_v} + (1-s_0)^2$$
$$\implies \dot{V_1} = \boldsymbol{s_v}^T(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v}\times\delta\boldsymbol{\omega}) + (1-s_0)(\boldsymbol{s_v}^T\delta\boldsymbol{\omega})$$
$$\implies \dot{V_1} = \boldsymbol{s_v}^T\delta\boldsymbol{\omega} = -k_p||\boldsymbol{s_v}||^2 < 0$$

Since $\dot{V_1}$ is negative definite $\boldsymbol{s_v}$ asymptotically stabilises to $\mathbf{0}$. Now we want $\delta\boldsymbol{\omega}_d = -k_p\boldsymbol{s_v}$. Define $\boldsymbol{\xi} = \delta\boldsymbol{\omega} - \delta\boldsymbol{\omega}_d = \delta\boldsymbol{\omega} + k_p\boldsymbol{s_v}$.

$$V_2 = \frac{1}{2}\boldsymbol{\xi}^T\mathbf{J}\boldsymbol{\xi}$$
$$\implies \dot{V_2} = \boldsymbol{\xi}^T\mathbf{J}\dot{\boldsymbol{\xi}}$$

Since,

$$\dot{\boldsymbol{\xi}} = \dot{\delta\boldsymbol{\omega}} + k_p\dot{\boldsymbol{s_v}}$$
$$\implies \dot{V_2} = \boldsymbol{\xi}^T(\mathbf{J}\dot{\delta\boldsymbol{\omega}} + k_p\mathbf{J}\dot{\boldsymbol{s_v}})$$
$$\implies \dot{V_2} = \boldsymbol{\xi}^T(-\boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega} + \boldsymbol{u} - \mathbf{J}\boldsymbol{\phi} + \frac{k_p}{2}\mathbf{J}(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v}\times\delta\boldsymbol{\omega}))$$

Choose,

$$\boldsymbol{u} = \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\phi} - \frac{k_p}{2}\mathbf{J}(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v}\times\delta\boldsymbol{\omega}) - k_w\boldsymbol{\xi}$$

This choice of control ensures that,

$$\dot{V_2} = -k_w||\boldsymbol{\xi}||^2$$

Overall candidate Lyapunov Function:

$$V = V_1 + V_2$$
$$\dot{V} = \boldsymbol{s_v}^T\delta\boldsymbol{\omega} - k_w||\boldsymbol{\xi}||^2$$

Note that $\delta\boldsymbol{\omega} = \boldsymbol{\xi} - k_p\boldsymbol{s_v}$,

$$\implies \dot{V} = \boldsymbol{s_v}^T(\boldsymbol{\xi} - k_p\boldsymbol{s_v}) - k_w||\boldsymbol{\xi}||^2$$
$$\implies \dot{V} = -k_p||\boldsymbol{s_v}||^2 + \boldsymbol{s_v}^T\boldsymbol{\xi} - k_w||\boldsymbol{\xi}||^2$$

Using the inequality,

$$\boldsymbol{s_v}^T\boldsymbol{\xi} \leq \frac{||\boldsymbol{s_v}||^2}{2} + \frac{||\boldsymbol{\xi}||^2}{2}$$

We get that,

$$\dot{V} \leq -\left(k_p - \frac{1}{2}\right)||\boldsymbol{s_v}||^2 - \left(k_w - \frac{1}{2}\right)||\boldsymbol{\xi}||^2$$
$$\therefore \dot{V} < 0 \;\forall\; k_p, k_w > \frac{1}{2}$$

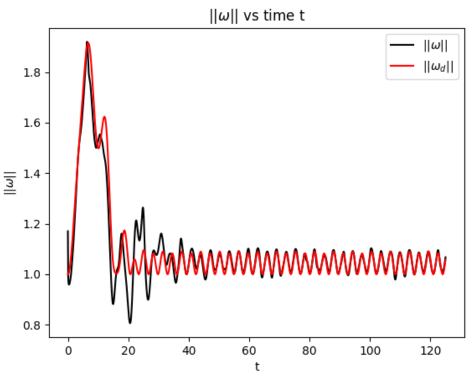
Since the derivative of the Lyapunov function is negative definite, from the Lyapunov stability theorems we conclude that $\boldsymbol{s_v}$, $\boldsymbol{\xi} \to \mathbf{0}$ as $t\to\infty$. And $\delta\boldsymbol{\omega}_d = -k_p\boldsymbol{s_v} \to \mathbf{0}$. Now,

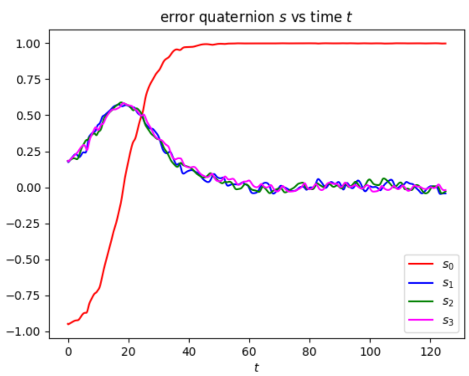
$$\because \boldsymbol{\xi} = \delta\boldsymbol{\omega} + k_p\boldsymbol{s_v} \tag{3}$$

And $\boldsymbol{s_v}$, $\boldsymbol{\xi} \to \mathbf{0}$ as $t\to\infty$, $\implies$ $\delta\boldsymbol{\omega} \to \mathbf{0}$ as $t\to\infty$. Therefore we have achieved the tracking goal by driving the error dynamics to zero. This concludes the control design and the stability analysis.

B. Simulation Results (Task 4)

Using the controller designed using integrator backstepping we get the following plots.





The plots justify that the tracking objective is achieved.

V. Adaptive Integrator Backstepping

This section presents an adaptive integrator backstepping based method to design a control law when the inertia matrix $\mathbf{J}$ is not known.

A. Control Design

We start by re-formulating the control law derived in section 4,

$$\boldsymbol{u} = \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\phi} - \frac{k_p}{2}\mathbf{J}(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v} \times \delta\boldsymbol{\omega}) - k_w\boldsymbol{\xi}$$

$\mathbf{J}$ appears linearly in $\boldsymbol{u}$ and hence we define,

$$B\boldsymbol{\theta^*} = \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\phi} - \frac{k_p}{2}\mathbf{J}(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v} \times \delta\boldsymbol{\omega})$$

Where $B \in \mathbb{R}^{3\times 6}$ is a constant matrix and $\boldsymbol{\theta^*}$ is the vector consisting of the 6 distinct entries of $\mathbf{J}$. Therefore the control law can be re-written as,

$$\boldsymbol{u} = B\boldsymbol{\theta^*} - k_w\boldsymbol{\xi}$$

But since $\boldsymbol{\theta^*}$ is not known, we can replace it by an estimate $\hat{\boldsymbol{\theta}}\boldsymbol{^*}$ as per the Certainty Equivalence principle. Therefore the control law can be written as,

$$\boldsymbol{u} = B\hat{\boldsymbol{\theta}}\boldsymbol{^*} - k_w\boldsymbol{\xi} \tag{4}$$

B. Adaptation Law

Since $\hat{\boldsymbol{\theta}}\boldsymbol{^*}$ is unknown, we use the candidate Lyapunov Function V to be,

$$V = V_1 + V_2 + \frac{1}{2\gamma}(B\tilde{\boldsymbol{\theta}}\boldsymbol{^*})^T(B\tilde{\boldsymbol{\theta}}\boldsymbol{^*})$$

Where,

$$V_1 = \boldsymbol{s_v}^T\boldsymbol{s_v} + (1 - s_0)^2$$

$$V_2 = \frac{1}{2}\boldsymbol{\xi}^T\mathbf{J}\boldsymbol{\xi}$$

$\tilde{\boldsymbol{\theta}}\boldsymbol{^*} = \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}\boldsymbol{^*}$ is the parameter error and $\gamma \in \mathbb{R}$ is the adaptation gain, taking the derivative of V, we get,

$$\dot{V} = \dot{V}_1 + \dot{V}_2 - \frac{1}{\gamma}(B\tilde{\boldsymbol{\theta}}\boldsymbol{^*})^T(B\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{^*})$$

Observe that,

$$\dot{V}_1 = \boldsymbol{s_v}^T\delta\boldsymbol{\omega}$$

$$\dot{V}_2 = \boldsymbol{\xi}^T(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{u} - \mathbf{J}\boldsymbol{\phi} + \frac{k_p}{2}\mathbf{J}(s_0\delta\boldsymbol{\omega} + \boldsymbol{s_v} \times \delta\boldsymbol{\omega}))$$

$$\implies \dot{V}_2 = \boldsymbol{\xi}^T(-B\boldsymbol{\theta^*} + \boldsymbol{u})$$

Substituting $\mathbf{u}$ from (4) we get,

$$\dot{V}_2 = \boldsymbol{\xi}^T(-B\tilde{\boldsymbol{\theta}}\boldsymbol{^*} - k_w\boldsymbol{\xi}) = -k_w||\boldsymbol{\xi}||^2 - \boldsymbol{\xi}^T B\tilde{\boldsymbol{\theta}}\boldsymbol{^*}$$

Substituting the values of $\dot{V}_1$ and $\dot{V}_2$ we get,

$$\dot{V} = \boldsymbol{s_v}^T\delta\boldsymbol{\omega} - k_w||\boldsymbol{\xi}||^2 - \boldsymbol{\xi}^T B\tilde{\boldsymbol{\theta}}\boldsymbol{^*} - \frac{1}{\gamma}(B\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{^*})^T(B\tilde{\boldsymbol{\theta}}\boldsymbol{^*})$$

Substituting $\delta\boldsymbol{\omega}$ from (3)

$$\dot{V} = \boldsymbol{s_v}^T(\boldsymbol{\xi} - k_p\boldsymbol{s_v}) - k_w||\boldsymbol{\xi}||^2 - \left(\boldsymbol{\xi}^T - \frac{1}{\gamma}(B\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{^*})^T\right)B\tilde{\boldsymbol{\theta}}\boldsymbol{^*}$$

Choosing the Adaptation law to be,

$$B\dot{\hat{\boldsymbol{\theta}}}\boldsymbol{^*} = \gamma\boldsymbol{\xi}$$

We get that,

$$\dot{V} = k_p||\boldsymbol{s_v}||^2 + \boldsymbol{s_v}^T\boldsymbol{\xi} - k_w||\boldsymbol{\xi}||^2$$

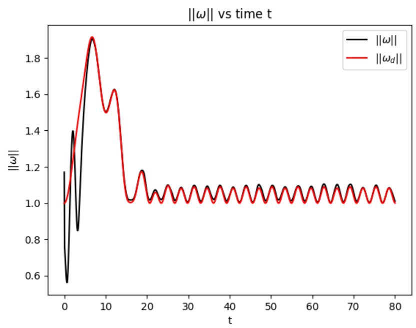
Which translates to,

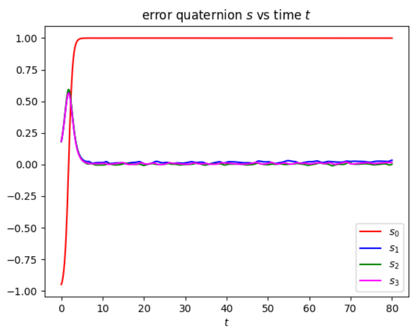
$$\dot{V} \leq -\left(k_p - \frac{1}{2}\right)||\boldsymbol{s_v}||^2 - \left(k_w - \frac{1}{2}\right)||\boldsymbol{\xi}||^2$$

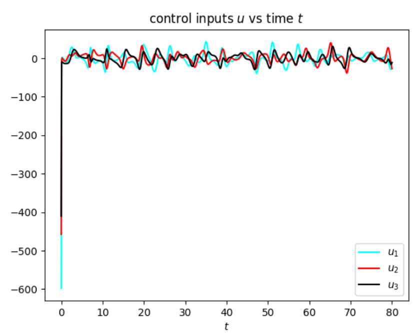
from the inequality of product less than squares.

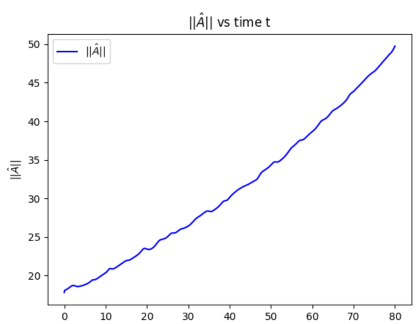
C. *Simulation Results (Task 5)*

The initial estimate of the inertia matrix $\mathbf{J}$ is taken to be $1.3\mathbf{J}$, using the controller (4) designed using adaptive integrator backstepping, we get the following results.









Here $\hat{A} = B\hat{\boldsymbol{\theta}}^*$. Convergence of $\hat{A}$ would guarantee the convergence of $B\hat{\boldsymbol{\theta}}^*$. But from the plots we can see that the parameter error diverges, which means that the parameter does not converge to it's true values.