Assignment 1 in TTK4190
Guidance and Control of Vehicles

Group 47
Alexandra S. Vedeler – 759499
Ella-Lovise H. Rørvik – 759559

Fall 2017

Department of Engineering Cybernetics

Abstract

This is our answers and results for Assignment 1. This is a results and discussion text, and not as official as a report would be in presentation, standards and conventions. This text is more about the fact that you as a reader will be able to understand what we have done, what our results were and what we have been able to conclude from them.

Contents

Problem 1 - Attitude Control of Satellite

The objective of problem 1 is to control attitude of a satellite. The satellites equations of motions are represented in equation (1), [1]. The parameters for this specific satellite are: $\mathbf{I}_{CG} = mr^2\mathbf{I}_{3x3}$, $m = 100kg$, $r = 2.0m$.

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q})\boldsymbol{\omega} \tag{1a}$$

$$\mathbf{I}_{CG}\dot{\boldsymbol{\omega}} - \mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\omega})\boldsymbol{\omega} = \boldsymbol{\tau} \tag{1b}$$

These equations uses the unit quaternions and Euler angles. The unit quaternions are represented as $q = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^\top$ and only the positive η is used, leading to equations (2). The Euler angles are represented as $\boldsymbol{\Theta} = [\phi, \theta, \psi]^\top$ and their representative angular velocities are $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]$. In [1] $\boldsymbol{\omega}$ is often written as $\boldsymbol{\omega} = [p, q, r]$, but because of the parameter $r = 2.0m$, in this rapport $\boldsymbol{\omega}$ is written as $[\omega_1, \omega_2, \omega_3]^\top$.

$$\eta = \sqrt{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} \tag{2}$$

In this lab report the state of the system $\boldsymbol{x} = [\boldsymbol{\epsilon}^\top, \boldsymbol{\omega}^\top]$ and the input to the system $\mathbf{u} = [0, 0, 0, \tau_1, \tau_2, \tau_3]^\top$. The reason for the upper three zeros in the input, is because of $\dot{\boldsymbol{\epsilon}}$ do not have any input, while $\dot{\boldsymbol{\omega}}$ have input $\boldsymbol{\tau}$.

Problem 1.1

Finding the equilibrium point

The equilibrium point is defined as the steady-state solution of a system, meaning $\dot{\mathbf{x}} = 0$ and $\mathbf{u} = \mathbf{0}$.

The equilibrium point $\mathbf{x_0}$ of the closed-loop system $\mathbf{x} = [\boldsymbol{\epsilon}^\top, \boldsymbol{\omega}^\top]^\top$ corresponding to $\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^\top = [1, 0, 0, 0]$ and $\boldsymbol{\tau} = \mathbf{0}$ may be found by setting $\dot{\boldsymbol{\epsilon}} = \mathbf{0}$ and $\dot{\boldsymbol{\omega}} = \mathbf{0}$. The equation for $\boldsymbol{\epsilon}$ and $\boldsymbol{\omega}$ may be found by using equation (2.86) in [1] and (1b). From equation (2.86) in [1] and (1b) the following equations were found:

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}[\eta\mathbf{I}_{3X3} + \mathbf{S}(\boldsymbol{\epsilon})]\boldsymbol{\omega} \tag{3a}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_{CG}^{-1}[\mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\omega})\boldsymbol{\omega} + \boldsymbol{\tau}] \tag{3b}$$

With skew-matrices $\mathbf{S}(\boldsymbol{\epsilon})$ and $\mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\omega})$, as defined in [1] equation (2.10). Equation (3a) may be written as:

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\left\{\begin{bmatrix} \sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} & 0 & 0 \\ 0 & \sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} & 0 \\ 0 & 0 & \sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} \end{bmatrix} + \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix}\right\}\begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3\end{bmatrix} \tag{4}$$

By using equation (4), $\mathbf{q} = [1, 0, 0, 0]^\top$ and $\boldsymbol{\tau} = \mathbf{0}$ the equilibrium point is found to be $\mathbf{x}_0 = [0, 0, 0, 0, 0, 0]^\top$. Since equation (4) have 3 unknown parameters and 3 equations, it was not necessary to include equation (3b). Equation (3b) alone only gives 2 equations and 3 unknown parameters ($\omega_1, \omega_2, \omega_3$), because of the constraints $\mathbf{q} = [1, 0, 0, 0]^\top$ and $\boldsymbol{\tau} = \mathbf{0}$.

Linearization of the equations of motion

A linearization of a function is the first order of its Taylor expansion around the pint of interest, which is the equilibrium point. A general linearized model is $\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u}$. $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$ are the linearized coordinate transformed states of the system and $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$ is the corresponding coordinate transformed input.

A system of the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, may be linearized by using (5). In equation (5) the matrices $\mathbf{A}$ and $\mathbf{B}$ are the matrices of the linearized system, and are calculated by using the Jacobians of $f(\mathbf{x}, \mathbf{u})$, and the equilibrium point.

$$\mathbf{A} = \frac{\partial f(\mathbf{x}, \boldsymbol{u})}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \boldsymbol{u}=\mathbf{0}}} \quad \mathbf{B} = \frac{\partial f(\mathbf{x}, \boldsymbol{u})}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \boldsymbol{u}=\mathbf{0}}} \tag{5}$$

The equation of motion of the satellite , equation (3), is rewritten as a function of the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ in equation (6).

$$\begin{bmatrix} \dot{\boldsymbol{\epsilon}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}[\sqrt{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}\omega_1 - \epsilon_3\omega_2 + \epsilon_2\omega_3] \\ \frac{1}{2}[\epsilon_1\omega_3 + \sqrt{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}\omega_2 - \epsilon_1\omega_3] \\ \frac{1}{2}[-\epsilon_2\omega_1 + \epsilon_1\omega_2 + \sqrt{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}\omega_3] \\ \tau_1\frac{1}{mr^2} \\ \tau_2\frac{1}{mr^2} \\ \tau_3\frac{1}{mr^2} \end{bmatrix} \tag{6}$$

The linearized closed-loop system was calculated using equation (6) , (5) and equilibrium point $\mathbf{x}_0 = [0, 0, 0, 0, 0, 0]$ and $\mathbf{u} = [0, 0, 0, 0, 0, 0]$. The linearized system $\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\mathbf{u}$, with matrices $\mathbf{A}$ and $\mathbf{B}$ is represented in equation (7) :

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \frac{1}{mr^2}\begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{400} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{400} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{400} \end{bmatrix} \tag{7}$$

Problem 1.2

In this problem the following control law was suggested for for controlling the satellite:

$$\boldsymbol{\tau} = -\mathbf{K}_d\boldsymbol{\omega} - k_p\boldsymbol{\epsilon} \tag{8}$$

with $\mathbf{K}_d = k_d\mathbf{I}_{3x3}$, $k_d = 20$ and $k_p = 1$.

Analysis of stability of linearized system

The stability of the linearized system may be analysed looking at the placement of the poles of the system.

The control law in equation (8) may be represented on the form $\mathbf{u} = -\mathbf{K}\mathbf{x}$. Using the control law , from equation (8), the The $\mathbf{K}$-matrix is:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ k_p & 0 & 0 & k_d & 0 & 0 \\ 0 & k_p & 0 & 0 & k_d & 0 \\ 0 & 0 & k_p & 0 & 0 & k_d \end{bmatrix} \tag{9}$$

The linearized system is on the form $\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u}$, that together with the control law in equation (8) may be written as $\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} - \mathbf{B}\mathbf{K}\Delta\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\Delta\mathbf{x}$.

The poles of the system may be found by calculating the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$. The calculation of the eigenvalues was done by using the MATLAB-function `lambda = eig(A)`, with lambda being a vector of the eigenvalues of the closed-loop system with the simple controller. The functionality of finding the eigenvalues of the closed-loop system with control-law given in (8) is implemented in the file stability_system.m . The eigenvalues were found to be $= -0.0250 \pm 0.0250i$, where 3 poles are coinciding. Since the real part of the eigenvalues are negative, the system is stable.

Discussion of significance of real or imaginary poles

When placing poles there are multiple considerations to take into account. Typically the system should be stable, meaning the poles of the system should be placed in the left half of the s-plane. A system having one or more poles lying on the imaginary axis of the s-plane has non-decaying oscillatory components in its homogeneous response. Real poles gives exponentially decaying response components, where the rate increases for more negative poles. With a complex conjugate pole pair in the left-half of the s-plane the combined effect generate response component that is a decaying sinusoidal. The rate of decay for complex poles is specified by the real-part of the pole and the frequency is decided by the imaginary-part of the pole. This means a system with real poles in left-half plane will steadily approach the correct value, while an imaginary pole in left-half plane may reach steady-state value faster than a real-pole on the cost of stability.

When choosing poles, the most important thing is to look into the dynamics of the system. Each pole affects a different part of the system.

Generally since the satellite is in space, it should not have too heavy oscillations because of less interrupting (and stabilizing) forces and being in vacuum. At the same time, adjusting real poles to give as fast response as complex-poles, will imply more power needed. A satellite will have limited thruster capacity, which may lead to saturation on the input to the controller.

Since the satellite should be relatively fast, with limited thrust power, it is suggested to have complex poles, with small imaginary parts /oscillatory parts. This suggestion will lead to a slightly faster positioning response of the satellite, and at the same time save some thrust power but at the expense of somewhat reduced stability.

Problem 1.3

Simulation of system with simple controller

In attitude1.m the system is simulated with a controller based on the the control law from (8). This is done by using a for-loop initialized with Euler-parameters $\phi = 10°$, $\theta = -5°$ and $\psi = 15°$. To initialize the unit quaternion-parameters the `MATLAB`-function `q = euler2q(phi,theta,psi)` from the library `MSS` was used. The function transforms from Euler-angles to unit quaternions. The angular velocity $\boldsymbol{\omega}$ is initialized to $\mathbf{0}$.

The simulated state-response is calculated through a for-loop. The for-loop input is first calculated using the state, with $\mathbf{u} = -\mathbf{K}\mathbf{x}$, with K from (9). Second the Euler angles is calculated based on the state, using the `MATLAB`-function `[phi,theta,psi] = q2euler(q)`. From the equation of motion, (3), $\dot{\mathbf{x}}$ is calculated. The next state is estimated using Euler integration on quaternion-coordinates $\mathbf{q}$ and the angular velocities $\boldsymbol{\omega}$. First the quaternions are normalized, such that it is unit quaternions. The Euler angles, unit quaternions and angular-velocities are saved in a variable table, used for plotting.

The result after the plotting are represented in Figure 1 - 3.

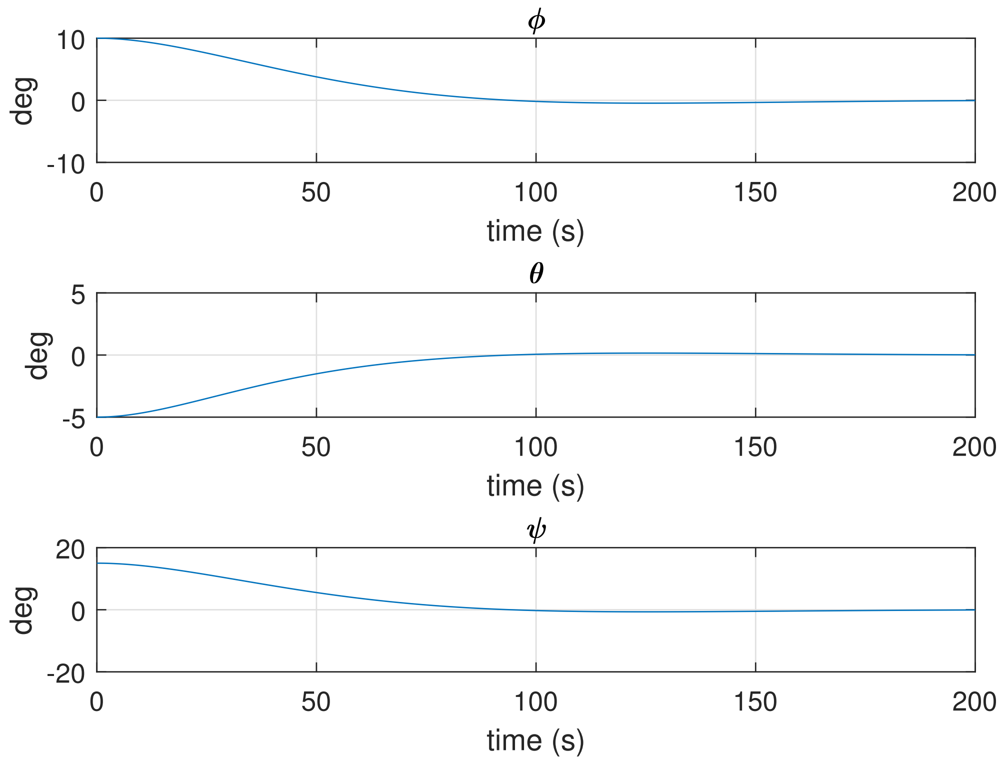

Figure 1: The resulting output Euler angles from the simulation in attitude1.m

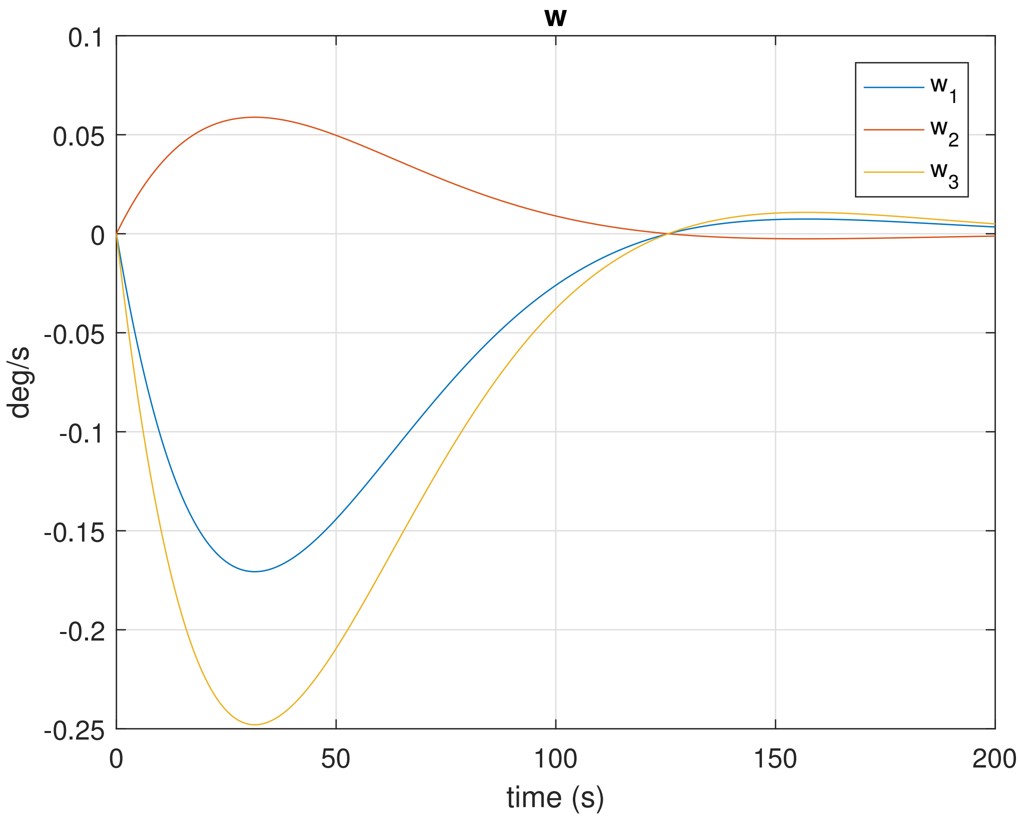

Figure 2: The resulting $\boldsymbol{\omega}$, part of x, from the simulation. of the system and η. $\boldsymbol{\omega}$ is $\mathbf{w}$ in the figure. The figure is from the simulation in attitude1.m

The control law implemented in attitude1.m implements an controller which is proportional to the states, meaning it will try to regulate the states to $\mathbf{0}$. The control law implements a setpoint regulation. The input from the controller impacts $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\epsilon}}$ is impacted by $\boldsymbol{\omega}$, from equation (3). Looking at Figure 1 - 3, the steady-state value is reached after approximately 100 seconds are are zero, meaning the system behaves as expected.

Problem 1.4

A modified attitude control law which controls $\boldsymbol{\epsilon}$ to a desired value $\mathbf{x}_d = [\boldsymbol{\epsilon}_d^\top, \mathbf{0}^\top]^\top$ is:

$$\boldsymbol{\tau} = -\mathbf{K}_d\boldsymbol{\omega} - k_p\tilde{\boldsymbol{\epsilon}} \tag{10}$$

$\tilde{\boldsymbol{\epsilon}}$ is the error in the imaginary part of the quaternion. The error in the quaternions $\tilde{\mathbf{q}}$ is defined as:

$$\tilde{\boldsymbol{q}} := \begin{bmatrix} \tilde{\eta} \\ \tilde{\boldsymbol{\epsilon}} \end{bmatrix} = \bar{\boldsymbol{q}}_d \otimes \boldsymbol{q} \tag{11}$$

with $\bar{\mathbf{q}} = [\eta, -\boldsymbol{\epsilon}^\top]^\top$ and $\mathbf{S}(-\boldsymbol{\epsilon}_d)$ being the skew-symmetric matrix. The quaternion product is defined as :

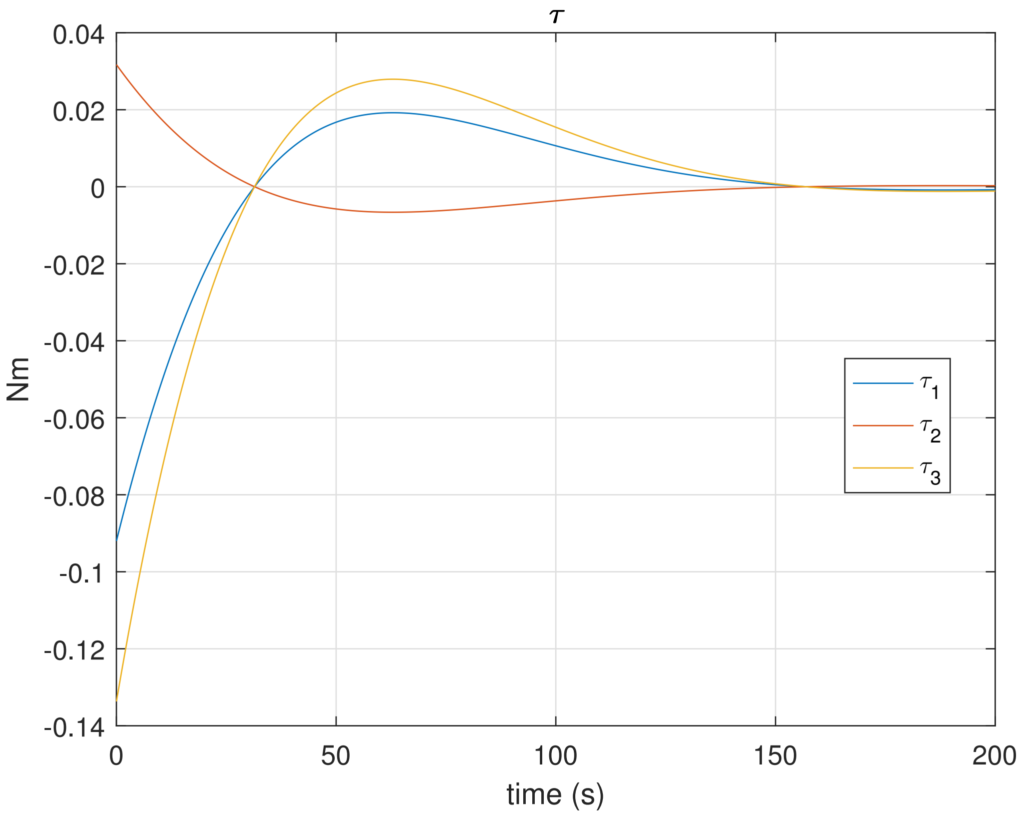

Figure 3: The resulting input $\boldsymbol{\tau}$ from the simulation in attitude1.m

$$\bar{\boldsymbol{q}}_d \otimes \boldsymbol{q} = \begin{bmatrix} \eta_d \eta + \boldsymbol{\epsilon}_d^\top \boldsymbol{\epsilon} \\ \eta_d \boldsymbol{\epsilon} - \eta \boldsymbol{\epsilon}_d + \mathbf{S}(-\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon} \end{bmatrix} \tag{12}$$

Matrix expression for the quaternion error

The matrix expression for the quaternion error expressed on component form is found by using the equation for quaternion error (11) and the definition of the quaternion product (12), and is:

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \eta_d \eta + \epsilon_{d,1}\epsilon_1 + \epsilon_{d,2}\epsilon_2 + \epsilon_{d,3}\epsilon_3 \\ \eta_d \epsilon_1 - \eta \epsilon_{d,1} + \epsilon_{d,3}\epsilon_2 - \epsilon_{d,2}\epsilon_3 \\ \eta_d \epsilon_2 - \eta \epsilon_{d,2} - \epsilon_{d,3}\epsilon_1 + \epsilon_{d,1}\epsilon_3 \\ \eta_d \epsilon_3 - \eta \epsilon_{d,3} + \epsilon_{d,2}\epsilon_1 - \epsilon_{d,1}\epsilon_2 \end{bmatrix} \tag{13}$$

with $\mathbf{q}_d = [\eta_d, \epsilon_{d,1}, \epsilon_{d,2}, \epsilon_{d,3}]^\top$.

Convergence of $\tilde{\mathbf{q}}$

$\tilde{\mathbf{q}}$ converges means $\mathbf{q} \to \mathbf{q}_d$. Finding the $\tilde{\mathbf{q}}$ after convergence means setting $\mathbf{q} = \mathbf{q}_d$ in equation (13) giving:

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \eta\eta + \epsilon_1\epsilon_1 + \epsilon_2\epsilon_2 + \epsilon_3\epsilon_3 \\ \eta\epsilon_1 - \eta\epsilon_1 + \epsilon_3\epsilon_2 - \epsilon_2\epsilon_3 \\ \eta_d\epsilon_2 - \eta\epsilon_2 - \epsilon_3\epsilon_1 + \epsilon_1\epsilon_3 \\ \eta\epsilon_3 - \eta\epsilon_3 + \epsilon_2\epsilon_1 - \epsilon_1\epsilon_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

The result from equation (14) is that the error of the quaternions will converge to zero for $\boldsymbol{\epsilon}$. The reason for η not converging to zero, is the constraint of unit quaternions, which states $\eta\eta + \epsilon_1\epsilon_1 + \epsilon_2\epsilon_2 + \epsilon_3\epsilon_3 = 1$.

Problem 1.5

The simulation of the attitude dynamics of the closed-loop system with control law given by (10) is in the file attitude2.m. For each time iteration, the desired attitude $\mathbf{q}_d(t)$ given by $\phi(t) = 10\sin(0.1t), \theta(t) = 0, \psi(t) = 15\cos(0.05t)$, was converted to radians and then converted to quaternions by the use of the `MATLAB` function `euler2q()`. The desired quaternion value was then conjugated and cross multiplied with the current $\mathbf{q}$ iteration by the use of the `MATLAB` functions `quatconj()` and `quatmultiply()` respectively, as explained in equation (12). By doing so, $\tilde{\mathbf{q}}$ was calculated as per equation (11).

The $\tilde{\boldsymbol{\epsilon}}$ part of $\tilde{\mathbf{q}}$ was extracted and combined with $\boldsymbol{\omega}$ to create the state error and was multiplied with the $\mathbf{K}$, equation (9), to create the control input in the same way as in attitude1.m. The initial values were kept the same as in exercise 1.3 and the k_p and k_d were changed to 10 and 300 respectively.

Simulation results

The simulation of attitude2.m resulted in the plots given in Figure 4 and Figure 5.

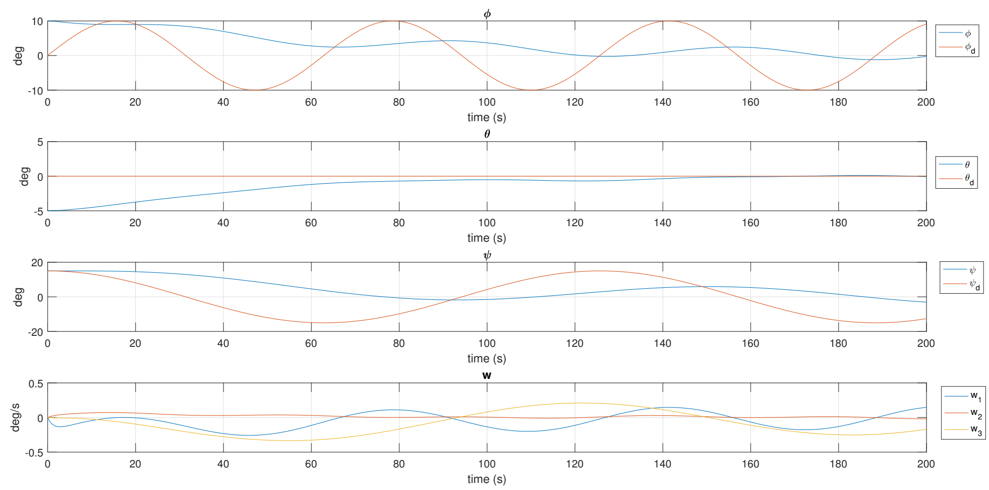

Figure 4: The resulting output euler angles with their corresponding desired values and the resulting output $\boldsymbol{\omega}$ (denoted $\mathbf{w}$ in the plot) from the simulation in attitude2.m.

The Figures show that, with the exception of $\boldsymbol{\theta}$, the angles of the satellite are following a sinusoidal pattern as per their desired values. However, they are not in sync with the

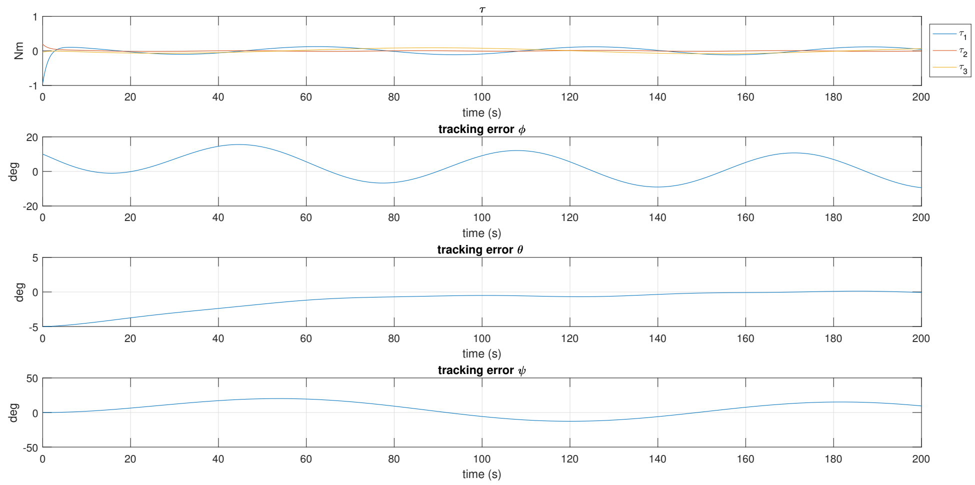

Figure 5: $\boldsymbol{\tau}$, and the tracking errors of the output euler angles from the simulation in attitude2.m

desired signal and the amplitude is far too low compared to their intended forms, resulting in relatively large sinusoidal tracking errors as can be seen in Figure 5. A reason for this misbehaviour is the design of the controller. Our controller as of this problem may be sufficient for static point regulation. However, with frequently changing desired values as we have here, we receive poor results. The two parts of our states $\boldsymbol{\omega}$ and $\boldsymbol{\epsilon}$ are not decoupled, but this fact is lost on the controller as it lacks the nessesary connection between them. With the fact that, even though the angles of the satellite is set to move in a specific way, the desired value of $\boldsymbol{\omega}$ is still set to 0, makes for a contradiction in the controller where neither part is able to do as they are told. The exception is θ and ω_2 who both have desired values of 0 and therefore does not obstruct each other. We would need to correct this in order to achieve satisfying control.

Increasing k_p to 100 resulted in less error in the euler angles. This makes sense as a higher proportional part makes for a faster controller and puts more weight on the error in $\boldsymbol{\epsilon}$.

It is also worth mentioning that the $\mathbf{K}$ matrix used for the controller was found using the linearized version of the system. This may also introduce errors.

Problem 1.6

A modified attitude control law is:

$$\boldsymbol{\tau} = -\mathbf{K}_d\tilde{\boldsymbol{\omega}} - k_p\tilde{\tilde{\boldsymbol{\epsilon}}} \tag{15}$$

with $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega_d}$. Differentiating the desired attitude, Θ_d, from the previous problem and calculating ω_d as

$$\boldsymbol{\omega}_d = \mathbf{T}^{-1}_{\boldsymbol{\Theta}_d}(\boldsymbol{\Theta}_d)\dot{\boldsymbol{\Theta}}_d \tag{16}$$

To simulate the new closed-loop system, small changes was done compared to the file attitude2.m. The new simulation file was named attitude3.m. To find $\mathbf{T}_{\boldsymbol{\Theta_d}}$ the `MATLAB` function `eulerang()` was used and set k_p and k_d to 10 and 300 respectively as before. The $\boldsymbol{\omega}_d$ was calculated from (16) and the new control law, equation (15), was implemented in the new simulation attitude3.m.

Simulation results

We can see from Figure 6 and Figure 7 that the satellite is now able to follow the given reference signal for both angles and angle velocities. From Figure 8 we can see that though there still is an error in the ϕ and ψ they are much smaller than previously. Now that $\boldsymbol{\omega}$ and $\boldsymbol{\epsilon}$ have corresponding desired values, they can both be satisfied by the same actions. The cooperation between the two parts of the state, which are in themselves related, is key to making a good controller. The linearization error discussed in the previous problem may still endure and the controller is not exactly the fastest in the galaxy, but all in all the attitude control is better than before.

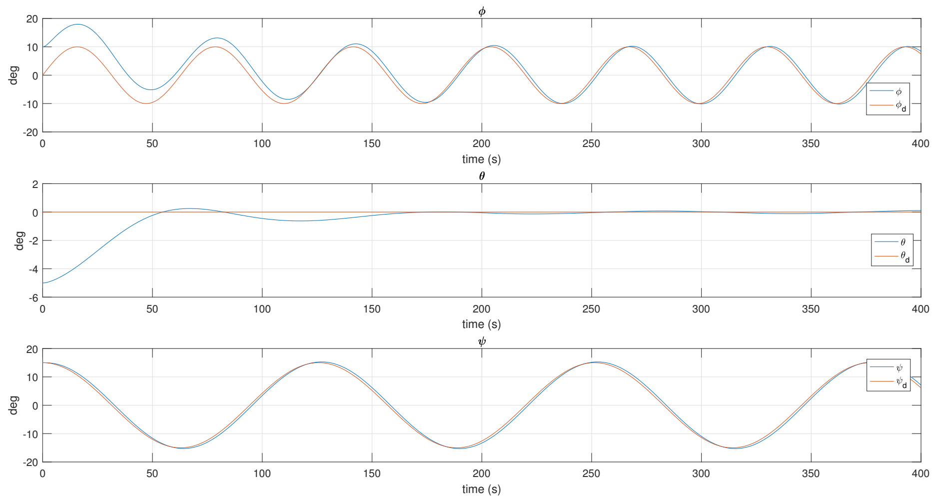

Figure 6: The resulting output euler angles with their corresponding desired values from the simulation in attitude3.

Problem 1.7

A Lyapunov function candidate (Fjellstad and Fossen, 1994) was given:

$$V = \frac{1}{2}\tilde{\boldsymbol{\omega}}^\top \mathbf{I}_{CG}\tilde{\boldsymbol{\omega}} + 2k_p(1-\tilde{\eta}) \tag{17}$$

Assuming $\boldsymbol{\omega_d} = \mathbf{0}$ and $\boldsymbol{\epsilon}_d$ and η_d constants and the control law given by (10), the Lyapunov function in equation (17) is positive and radially unbounded.

The reason for Vs positivity is that $\mathbf{I}_{CG}$ being an identity matrix with a positive number, mr^2, on its diagonal, is positive definite. So the first part of V is positive (when labeling 0 as a positive number). The second part, consisting of $2k_p(1-\tilde{\eta})$, may only be negative if $\tilde{\eta} > 1$. This will never happen however, as $\tilde{\eta}$ is part of the unit quaternion $\tilde{q}$ making $|\tilde{\eta}| \leq 1$. Thus the second part of V will also be positive. The Lyapunov function is usually

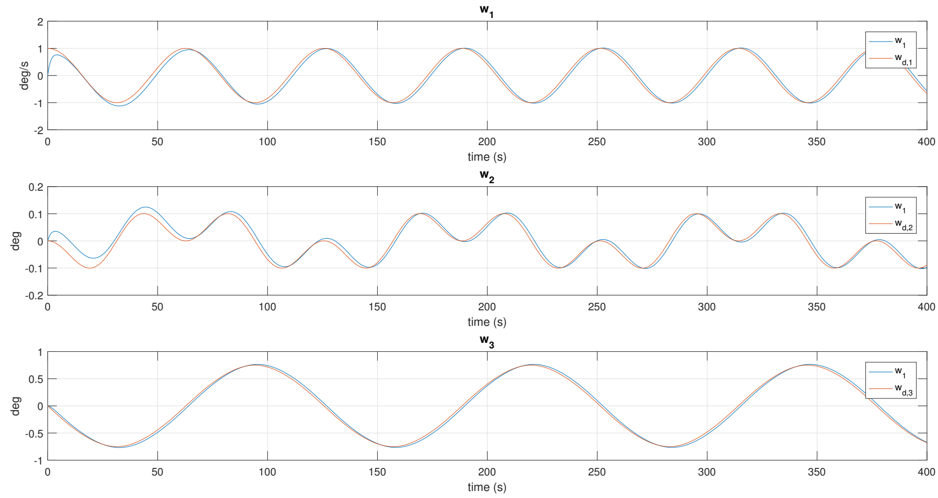

Figure 7: The resulting output $\boldsymbol{\omega}$ (denoted $\mathbf{w}$ in the plots) and the corresponding desired values from the simulation in attitude3.

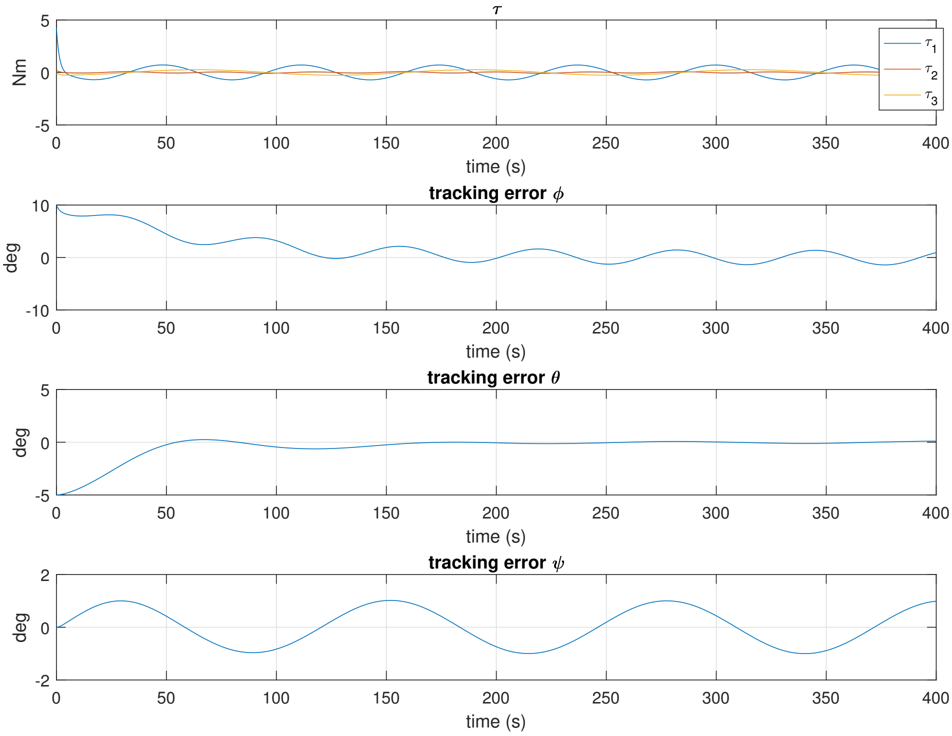

Figure 8: $\boldsymbol{\tau}$, and the tracking errors of the output euler angles from the simulation in attitude3.

thought of as a representation of the energy in the system so the fact that it is never negative makes sense.

The function is radially unbounded because as $\tilde{\boldsymbol{\omega}} \rightarrow \infty$ the first part of V does the same, $\frac{1}{2}\tilde{\boldsymbol{\omega}}^\top \mathbf{I}_{CG}\tilde{\boldsymbol{\omega}} \rightarrow \infty$. We already established that the second part of V, $2k_p(1-\tilde{\eta})$ is positive, meaning that V as a whole is unbounded.

Calculating $\dot{V}$

To calculate $\dot{V}$, $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega}_d = \boldsymbol{\omega}$ was substituted into V before differentiating:

$$\dot{V} = \boldsymbol{\omega}^\top \mathbf{I}_{CG}\dot{\boldsymbol{\omega}} + k_p\tilde{\boldsymbol{\epsilon}}^\top \boldsymbol{\omega} \tag{18}$$

Then, using (1b), $\mathbf{I}_{CG}\dot{\boldsymbol{\omega}}$ was substituted giving this equation:

$$\dot{V} = \boldsymbol{\omega}^\top(\boldsymbol{\tau} - \mathbf{S}(\mathbf{I}_{CG})\boldsymbol{\omega}) + k_p\tilde{\boldsymbol{\epsilon}}^\top \boldsymbol{\omega} \tag{19}$$

Knowing $\boldsymbol{\omega}\mathbf{S}(\mathbf{I}_{CG})\boldsymbol{\omega} = 0$ and substituting $\boldsymbol{\tau}$ from the control law (10) in the equation, we are left with

$$\dot{V} = \boldsymbol{\omega}^\top(k_d\boldsymbol{\omega} - k_p\tilde{\boldsymbol{\epsilon}}) + k_p\tilde{\boldsymbol{\epsilon}}^\top \boldsymbol{\omega} = -k_d\boldsymbol{\omega}^\top \boldsymbol{\omega} \tag{20}$$

Convergence

Barbalat's lemma (A.1)[1], tells us that our $\boldsymbol{\omega}$ will converge to zero if three conditions are satisfied:

- $V \geq 0$
- $\dot{V} \leq 0$
- $\dot{V}$ is uniformly continuous

We already proved the first one when we proved V is positive. For the second condition we can easily see that $\dot{V}$ as a negative quadratic function, due to $\boldsymbol{\omega}^\top\boldsymbol{\omega} \geq 0$ and $k_d \geq 0$ a constant, must be ≤ 0.

To prove $\dot{V}$ is uniformally continuous, we can take a look at $\ddot{V} = -2k_d\boldsymbol{\omega}^\top\dot{\boldsymbol{\omega}}$ and prove it to be bounded. We can assume the $\boldsymbol{\omega}$, and thereby V, does not begin at ∞ and we know $\dot{V}$ is negative, meaning V is always sinking. However V will always be positive, meaning it must stop at 0. Thus $\boldsymbol{\omega}$ will never increase beyond it's initial value. Knowing we are in physical system, it is reasonable to assume $\boldsymbol{\omega}$ is continuous, thus $\dot{\boldsymbol{\omega}}$ cannot go towards ∞. With neither $\boldsymbol{\omega} \rightarrow \pm\infty$ nor $\dot{\boldsymbol{\omega}}\pm \rightarrow \infty$, $\ddot{V} = -2k_d\boldsymbol{\omega}^\top\dot{\boldsymbol{\omega}}$ is bounded. Meaning $\dot{V}$ is uniformly continuous and Barbalat is satisfied. $\boldsymbol{\omega}$ will converge to zero.

According to the curriculum book[1], Barbalat's lemma only guarantees global convergence, meaning our convergence to the equilibrium point is indeed global.

Asymptotic stability

The system is a non-autonomous system so to check stability LaSalle-Yoshizawa's teorem may be used. Three conditions must be satisfied in LaSalle-Yoshizawa's theorem (A.4)[1]

- $V > 0$ and $V(0) = 0$
- $\dot{V} \leq 0$
- V is radially unbounded

We have already proved the last two conditions earlier in this exercise. As for the first one, we have proved that V is always positive, i.e. $V \geq 0$. We also know $\mathbf{I}_{CG} > 0$, or positive definite. This means the first part of V is only equal to zero when $\boldsymbol{\omega} = 0$. As for the second part, we know η to be a function of $\boldsymbol{\epsilon}$, specifically $\eta = \sqrt{1 - \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}}$. The second part is only 0 when $\eta = 1$ which only happens when $\boldsymbol{\epsilon} = 0$, meaning $V = 0$ only when it's input $\mathbf{x}$ is zero, otherwise $V > 0$.

But what if we get stuck? $\dot{V} = -k_d \boldsymbol{\omega}^\top \boldsymbol{\omega} \leq 0$, as already stated, and $V = 0$ only when $\boldsymbol{\omega} = 0$. This does however not mean $\dot{\boldsymbol{\omega}} = 0$. Using (1b) and inserting $\boldsymbol{\omega} = 0$ and $\boldsymbol{\tau}$ into the equation we get

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_{CG}^{-1} k_p \tilde{\boldsymbol{\epsilon}} \neq 0 \tag{21}$$

Since $\dot{\boldsymbol{\omega}} \neq 0$ unless $\tilde{\boldsymbol{\epsilon}} = 0$ we will not get stuck. We will continue until we have reached our desired value.

So LaSalle-Yoshizawa's teorem suggests we have UGAS. However, we are working with unit quaternions which, depending on η may have more than one solution as η may be positive or negative, making the question of global stability more complicated. This assignment have defined η as positive only leaving only one singular solution, but the simulation has not. It is possible to add extra complexity, to the controller so that it will only allow for/choose the positive ηs, thus assuring global stability.

Problem 1.8

In this assignment we have been using unit quaternions which have the advantage of being defined in the whole space. This is unlike euler angles which has singularities at $\pm 90°$ in pitch meaning one will have local stability at best. The singularity of the euler angle is usually not a problem when controlling boats at sea, seeing how they won't reach this angle anyway (and if they do, it is already too late to save them). Considering this assignment is concerned with controlling a satellite in space however, reaching a pitch of $\pm 90°$ is not that unlikely and thus, must be accounted for. Unit quaternions is then a good choice.

Sadly, as all things in life, the unit quaternions also have drawbacks. One of them being the fact that they are 4-dimensional vectors used to describe a 3-dimensional space. This causes them not to be unique as two quaternions may represent the same thing yet have different sings of η as mentioned in the previous problem. This problem is absent from euler angle representation. Another drawback worth mentioning is the fact that, while we are used to describe the world using a 3-dimentional euler angles or similar representations, quaternions are less intuitive for most people.

References

[1] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.