NTNU

TTK4190

Assignment 1

Author:
Shiv Jeet Rai

MTTK

September 10, 2015

1: 6-DOF Kinematics and Kinetics

Task 1.1

$$\dot{\mathbf{p}} = \mathbf{U}(\mathbf{Q})\mathbf{v}$$
$$\dot{\mathbf{Q}} = \mathbf{T}(\mathbf{Q}\omega)$$

Euler angles (zyx convention):

- $\mathbf{Q} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$
- $\mathbf{U}(\mathbf{Q}) = \mathbf{R}_b^n(\mathbf{Q}) = \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\phi} = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$
- $\mathbf{T}(\mathbf{Q}\omega) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \omega_{b/n}^b$

Unit Quaternions:

- $\mathbf{Q} = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T$
- $\mathbf{U}(\mathbf{Q}) = \mathbf{R}_b^n(\mathbf{Q}) = \begin{bmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1\epsilon_2 - \epsilon_3\eta) & 2(\epsilon_1\epsilon_3 + \epsilon_2\eta) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\eta) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2\epsilon_3 - \epsilon_1\eta) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\eta) & 2(\epsilon_2\epsilon_3 + \epsilon_1\eta) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{bmatrix}$
- $\mathbf{T}(\mathbf{Q}\omega) = \frac{1}{2} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \omega_{b/n}^b$

Rotation matrix:

- $\mathbf{Q} = \mathbf{R}_b^n = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$
- $\mathbf{U}(\mathbf{Q}) = \mathbf{R}_b^n = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$
- $\mathbf{T}(\mathbf{Q}\omega) = \mathbf{R}_b^n \mathbf{S}\omega_{b/n}^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$

Task 1.2

Assumptions:

- North-East-Down (NED) is inertial frame
- Only rotational motion about CG. This means that the transational velocity is contstant or 0.
- Vessel is rigid

We derive the kinetic equations of a satellite from Euler's Second Axiom: $\vec{h}_g = I_g \vec{\omega}_{b/i}$

$$\vec{\tau} = \frac{{}^i d}{dt}(I_{CG}\vec{\omega}_{b/i}) = \frac{{}^i d}{dt}(I_{CG}\vec{\omega}_{b/n}) = \frac{{}^b d}{dt}(I_{CG}\vec{\omega}_{b/n}) + (\vec{\omega}_{b/n} I_{CG}) \times \vec{\omega}_{b/n}$$

$$\vec{\tau} = \underline{\underline{I_{CG}\dot{\vec{\omega}}_{b/n} - (I_{CG}\vec{\omega}_{b/n}) \times \vec{\omega}_{b/n}}} \qquad Q.E.D$$

Task 1.3

- The main advantage of describing rotations using Euler angles is that it gives a good visually interpretation. Is more intuitive than unit quaternions. The disadvantage is that when $\theta = \pm 90$ degrees the representation is singular ($\mathbf{T}^-1$ is not possible). In addition computing trignometric functions (Euler representation) takes more time than non-trognometric functions.
- The benefit of unit quaternions is that we don't have to think about the singularity problem above. The representation is non-singular. In addition we don't have to check $R^T R = I$ or $det(R) = 1$, instead we check if $|q| = 1$ is satisfied. Thus the computation of unit quaternions is easier for the computer. The disadvantage is that it isn't easy to get a visual representation by using quaternions, it's not simply intuitive.
- Rotational Matrix: The benefit of having the rotations desribed by the rotation matrix is that we dont need to compute the rotation matrix, since we already got it. By looking at Task 1.1 we clearly see that the representation in form of the rotation matrix clearly is the less computational heavy. That is a big advantage. The disadvantage however is, like the unit quaternions, that it is less intuitive than Euler Angles. Here we have 9 parameters which describes the dynamics.

2: Attitude Control using Euler Angles

Task 2.1

- **PD-Controller:** I chosed $\vec{\tau} = -\mathbf{K}_d\vec{\omega} - \mathbf{K}_p\mathbf{T}_Q^T(\mathbf{Q})\tilde{Q}$ as my controller. I arrived this controller by considering the given controller in **Task 3.1**.

- **Chosing controller gains:** Since we haven't learned how to do it by Lyapnov yet, I used try and error. The criterias I used were:

  - System should be stable with my controller gains.
  - The respective angles should reach the reference in about 10-20 simulation-seconds (e.g from 0 to π) due to a satelitte is energy conservative and thus doesn't uses a lot of input power.

$$K_p = \begin{bmatrix} 354.9 & 0 & 0 \\ 0 & 364.0 & 0 \\ 0 & 0 & 154.7 \end{bmatrix} \qquad K_d = \begin{bmatrix} 1080 & 0 & 0 \\ 0 & 1080 & 0 \\ 0 & 0 & 480 \end{bmatrix}$$

- **Initial values:** Could be anything except $\theta = \pm 90n$degrees, where $n = \pm 1, 2, 3.....$
  I chose $Q_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \qquad \omega_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$

- **Desired values:** Due to my Q_0 my system was stable if $\theta \neq \pm\pi/2$.
  Thus $Q_{d\_works} = \begin{bmatrix} \pi & \pi/4 & \pi \end{bmatrix}^T \qquad Q_{d\_don't\_works} = \begin{bmatrix} \pi & \pi & \pi \end{bmatrix}^T$

  Note: The oscillations on $Q_{d\_don't\_works}$ are not desired (destroys the thrusters) and θ don't even reach desired value.

  Higher K_p and K_d gives a faster respons (and gets unstable for $theta = \pm 90$degrees), but as mentioned earlier this satelitte is chosen with lower gains since I believe it to being conservative. It can be that due to solar panels it can and should have higher gains, but I have chosen those I have. Hope it's okay.

Task 2.2

- **Kind of disturbances:** To name a few:

  - solar wind affecting $\dot{\vec{\omega}}$
  - solar wind causing disturbances in sensors on $\dot{\mathbf{Q}}$.
    In this task I will use solar wind affecting $\dot{\vec{\omega}}$. This disturbance will generated by a random and multiplied with a constant parameter, called noise_amp, at each iteration.

New system:

$$\begin{bmatrix} \dot{\vec{Q}} \\ \dot{\vec{\omega}}_{b/n} \end{bmatrix} = \begin{bmatrix} \vec{T}(\vec{Q}\omega) \\ = I_{CG}^{-1}(\vec{\tau} + I_{CG}\vec{\omega}_{b/n}) \times \vec{\omega}_{b/n}) + noise \end{bmatrix}$$

- **Noise effect on system**: Noise at each iteration is given as

$$noise = noise\_amp(rand(1,3) - rand(1,3) = 0.5 * \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where x,y,z are between -1 and 1.

As it can be seen on the plots, it is not neccessary to implement a PID since the perbturbations are small.

- **PID controller:** $= \vec{\tau} = -\mathbf{K}_d\vec{\omega} - \mathbf{K}_p\mathbf{T}_Q^T(\mathbf{Q})\tilde{Q} - \mathbf{K}_i\mathbf{T}_Q^T \int \tilde{Q}$
I used the previous controller as inspiration, and just added an integral part.

- **Chosing controller gains:** Used try and error.

As mentioned I didn't use an integral term due to low perturbatations, as an option told in the HINTS to this assignment. What we are supposed to see with integral term is that the integral term removes any stationary difference we may have due to noise. In this can already small.

$$K_p = \begin{bmatrix} 354.9 & 0 & 0 \\ 0 & 364.0 & 0 \\ 0 & 0 & 154.7 \end{bmatrix} \qquad K_d = \begin{bmatrix} 1080 & 0 & 0 \\ 0 & 1080 & 0 \\ 0 & 0 & 480 \end{bmatrix}$$

$noise\_amp = 0.5$ in my case.
This means I have noise from -0.5 rad/s to 0.5 rads/s, which is really high. However it works

- **filter?** I have not implemented a filter in this task.

- **Initial values:**
I chose $Q_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \qquad \omega_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$

- **Desired values:** $Q_d = \begin{bmatrix} \pi & \pi/4 & \pi \end{bmatrix}^T$

- Yes, due to the perturbations being small I would say my controller counteract in good manner.

Note: Since my gains are small, I would have expected the noise to act with higher perbturbations, but I can't see that I have implemented it in the PD-controller wrong.

3: Attitude Control using Unit Quaternions

Task 3.1

- **Computed Quaternion:** $q_d = \begin{bmatrix} 0.9760 & 0.1006 & 0.1648 & -0.1006 \end{bmatrix}$
- **Euler approach VS Quaternion approach:** I tried to change both K_p and K_d to see if this had something to say how the quaternion-approach and euler-angle-approach reacted in the simulation, with the given initial- and desired values (ω_0 was set equal to $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$). The result was that the quaternion-approach reached the reference faster on ϕ and ψ, on θ the euler-approach reached faster.

  Figure 2 and 3 show for theta over 90 degrees. Figure 3 is quaternion, which doesnt crash.