Spacecraft attitude control: Application of fine trajectory linearization control

Course Project

Name: Reva Dhillon

Roll Number: AE21B108

Course: AS5545

Instructor: Professor Luoyi Tao

30/11/2024

Contents

Chapter 1

Introduction

The aim of this project is to explore a method for spacecraft attitude control based on fine trajectory linearization control. The motivation and the fine linearization method has been described, the controller design has been discussed and the simulation results have been presented. The difficulties faced in implementing the controller have also been highlighted.

The spacecraft attitude control system plays a very important role in space missions. It is vital in attitude pointing and tracking maneuvers. The onboard sensors must be aligned in the direction mandated by the mission requirements. This motivates the need for a higher accuracy control system.

The equations governing spacecraft attitude dynamics are highly nonlinear. One of the most popular methods to design a model-based controller for a nonlinear dynamic system involves applying linear control theories to the linearized model of the nonlinear system. Traditionally, a first-order Taylor series expansion is used. However, this introduces an error in the linearized model, especially in highly nonlinear systems.

In this project, an error-less linearization technique, the fine linearization technique, is described and applied to improve the trajectory linearization control (TLC) method. The traditional TLC algorithm applies the Jacobian linearization based on first-order Taylor series expansion to linearize a nonlinear model along the desired trajectory point by point. However, a linearization error is introduced by this approach. The fine TLC method does not face this error.

Both the traditional TLC and fine TLC methods are applied to construct a spacecraft attitude controller and their performances are compared.

Chapter 2

Fine Linearization

2.1 The fine linearization method

The fine linearization method employs the fundamental theorem of calculus:

$$g(c,s) - g(c',s) = \left(\int_0^1 \left. \frac{\partial g}{\partial c} \right|_{c' + \lambda(c-c')} d\lambda \right) (c - c') \tag{2.1}$$

where,

- The line segment $c' + \lambda(c - c')$ remains in the domain defined.
- $0 \leq \lambda \leq 1$

Equation 2.1 is a generalized form of Jacobian linearization. For simplicity, this is what is referred to as fine linearization. There is no linearization error in this method.

2.2 Application to a general system

To demonstrate how fine linearization may be applied to a nonlinear system, consider the following system with state vector x and input vector u.

$$\dot{x} = f(x,u,t) \tag{2.2}$$

Let x_a be an arbitrary state vector and u_a be an input vector relevant to x_a. By fine linearization of $f(x,u,t)$ about $[x_a, u_a]$ we obtain,

$$f(x,u,t) - f(x_a,u_a,t) = A(x,x_a,u,u_a,t)(x - x_a) + B(x,x_a,u,u_a,t)(u - u_a) \tag{2.3}$$

where,

$$A(x,x_a,u,u_a,t) = \int_0^1 \left. \frac{\partial f}{\partial x} \right|_{\begin{bmatrix} x_a \\ u_a \end{bmatrix} + \lambda \begin{bmatrix} x - x_a \\ u - u_a \end{bmatrix}} d\lambda \tag{2.4}$$

$$B(x,x_a,u,u_a,t) = \int_0^1 \left. \frac{\partial f}{\partial u} \right|_{\begin{bmatrix} x_a \\ u_a \end{bmatrix} + \lambda \begin{bmatrix} x - x_a \\ u - u_a \end{bmatrix}} d\lambda \tag{2.5}$$

The error-less approach presented above generates a quasi-linear parameter varying (Quasi-LPV) model of system 2.2.

Chapter 3

Fine Trajectory Linearization Control Method

The TLC is a nonlinear control design method, which combines a nonlinear dynamic inversion and a linear time-varying feedback stabilization. This chapter describes the method and modifies it by implementing fine linearization.

3.1 System considered

- Consider the following system:

$$\dot{x} = f(x, u, t) \tag{3.1}$$

- Let $\bar{x}$ be the nominal trajectory we want to track.
- The objective of the controller is to design the input u so that the system may accurately track the state trajectory $\bar{x}$.
- Let $\bar{u}$ be the nominal input satisfying,

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}, t) \tag{3.2}$$

- Assume that the exact inverse dynamics of the system is achievable. Then,

$$\bar{u} = F(\bar{x}, \dot{\bar{x}}, t) \tag{3.3}$$

- If the exact inverse dynamics of the system is achievable, TLC can be employed. The next section describes the components of the TLC algorithm.

3.2 TLC Algorithm

Figure 3.1 graphically illustrates the conceptual structure of TLC. It consists of 3 parts:

- Inverse Dynamics
- Differentiator
- Regulator

3.2.1 Inverse Dynamics

- The inverse dynamics of the plant model is used as a feedforward, open-loop controller.
- Equation 3.3 gives the inverse dynamics of the system under consideration.
- If it is not possible to implement the exact inverse, a pseudo-inversion may be used as an approximation.

3.2.2 Differentiator

- The differentiator calculates the time derivative of x.
- Practically, when $\bar{x}$ is a noisy signal, the time derivative of $\bar{x}$ is computed using realizable pseudo-differentiators. This is somewhat inaccurate but computationally feasible.
- First-order and second-order differentiators are represented by transfer functions as:

$$D_1(s) = \frac{\omega_d s}{s + \omega_d} \tag{3.4}$$

$$D_2(s) = \frac{\omega_d^2 s}{s^2 + 2\omega_d \xi_d s + \omega_d^2} \tag{3.5}$$

where,

  - ω_d is the bandwidth of lowpass filters that smooths the differentiators input to avoid output saturation by high-frequency noises.
  - $\xi_d = 0.707$ is fixed as the optimal damping ratio.

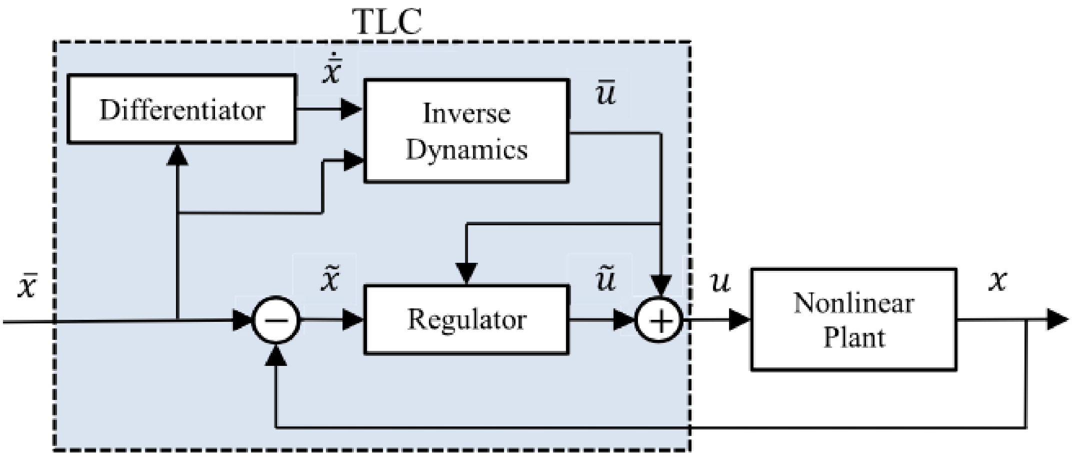


Figure 3.1: The conceptual structure of TLC.

3.2.3 Regulator

- The regulator generates the input tracking error $\tilde{u} = u - \bar{u}$.
- Let the state tracking error be $\tilde{x} = x - \bar{x}$. From the tracking error dynamics,

$$\dot{\tilde{x}} = f(x, u, t) - f(\bar{x}, \bar{u}, t) \tag{3.6}$$

- The tracking error is a nonlinear time-varying system. In TLC, the feedback regulator design is based on the linearized tracking error dynamics along the nominal trajectory.
- We use fine linearization to compute the linearized tracking error dynamics as given below.

$$\dot{\tilde{x}} = A(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t)\tilde{x} + B(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t)\tilde{u} \tag{3.7}$$

  where,

$$A(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t) = \int_0^1 \left.\frac{\partial f}{\partial x}\right|_{\begin{bmatrix}\bar{x}\\ \bar{u}\end{bmatrix}+\lambda\begin{bmatrix}\tilde{x}\\ \tilde{u}\end{bmatrix}} d\lambda \tag{3.8}$$

$$B(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t) = \int_0^1 \left.\frac{\partial f}{\partial u}\right|_{\begin{bmatrix}\bar{x}\\ \bar{u}\end{bmatrix}+\lambda\begin{bmatrix}\tilde{x}\\ \tilde{u}\end{bmatrix}} d\lambda \tag{3.9}$$

- The regulator uses a control law of the form,

$$\tilde{u} = K(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t)\tilde{x} \tag{3.10}$$

  where $K(\tilde{x}, \bar{x}, \tilde{u}, \bar{u}, t)$ is the linear parameter-varying control gain.

- Finally, $u = \bar{u} + \tilde{u}$.

In this manner, the TLC algorithm computes the input u and the states x. Since fine linearization has been employed, this version of TLC is called fine TLC. In the next chapter, this algorithm is applied to the spacecraft attitude control system.

Chapter 4

Application of Fine TLC to Spacecraft Attitude Control

In this chapter, the spacecraft attitude is first modeled, then a control law for this system, based on fine TLC, is presented.

4.1 Spacecraft Attitude Model

- We utilize the quaternions (q) to represent the orientation of the spacecraft.
- The quaternion kinematics differential equation is:

$$\dot{q} = \frac{1}{2}Q(\omega)q \tag{4.1}$$

  where,

  - $q = [q_1\ q_2\ q_3\ q_4]^T$ is the quaternion vector.
  - $\omega = [\omega_1\ \omega_2\ \omega_3]^T$ is the angular velocity vector of the spacecraft.
  - $Q : \mathbb{R}^3 \to \mathbb{R}^{4\times 4}$ such that

$$Q(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{4.2}$$

- The attitude dynamics of the rigid spacecraft is given by Euler's equation:

$$\dot{\omega} = J_s^{-1}(T - S(\omega)h) \tag{4.3}$$

  where,

  - J_s is the spacecraft inertia matrix.
  - T is the total input torque vector.
  - h is the total angular momentum vector, $h = J_s\omega$.
  - $S : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ such that

$$S(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{4.4}$$

4.2 Spacecraft Attitude Control Algorithm

The spacecraft attitude model in this approach is viewed as a cascade connection of two components:

1. Attitude kinematics
2. Attitude dynamics

4.2.1 Steps summarized

- To start, we consider just the attitude kinematics.
- ω is viewed as the input and we design a feedback controller for ω that is capable of tracking a nominal trajectory $\bar{q}$.
- We then look at the attitude dynamics.
- Designed ω from the first controller is viewed as the nominal trajectory $\bar{\omega}^*$ for the next component.
- We design the second controller to calculate the input torque T.
- We use the fine TLC algorithm for the trajectory tracking described above.

4.2.2 Controller 1 - Attitude Kinematics

- **Inverse Dynamics:** The nominal input $\bar{\omega}$ in terms of the nominal trajectory $\bar{q}$ and its time derivative $\dot{\bar{q}}$ is given by:

$$\bar{\omega} = 2G(\bar{q})\dot{\bar{q}} \tag{4.5}$$

where $G : \mathbb{R}^4 \to \mathbb{R}^{3\times 4}$ such that

$$G(\bar{q}) = \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \tag{4.6}$$

- **Differentiator:** This is used to calculate the derivative of $\bar{q}$.
- **Regulator:** The regulator generates the input tracking error.
  - The input tracking error $\tilde{\omega} = \omega - \bar{\omega}$ and the state trajectory error $\tilde{q} = q - \bar{q}$.
  - Applying fine linearization,

$$\dot{\tilde{q}} = A(\tilde{\omega}, \bar{\omega})\tilde{q} + B(\tilde{q}, \bar{q})\tilde{\omega} \tag{4.7}$$

where,

$$A(\tilde{\omega}, \bar{\omega}) = \int_0^1 \frac{\partial \dot{q}}{\partial q}\bigg|_{\begin{bmatrix}\bar{q}\\ \bar{\omega}\end{bmatrix} + \lambda \begin{bmatrix}\tilde{q}\\ \tilde{\omega}\end{bmatrix}} d\lambda = \frac{1}{2}Q(\bar{\omega}) + \frac{1}{4}Q(\tilde{\omega}) \tag{4.8}$$

$$B(\tilde{q}, \bar{q}) = \int_0^1 \frac{\partial \dot{q}}{\partial \omega}\bigg|_{\begin{bmatrix}\bar{q}\\ \bar{\omega}\end{bmatrix} + \lambda \begin{bmatrix}\tilde{q}\\ \tilde{\omega}\end{bmatrix}} d\lambda = \frac{1}{2}G^T(\bar{q}) + \frac{1}{4}G^T(\tilde{q}) \tag{4.9}$$

- This is further simplified to yield,

$$\dot{\tilde{q}} = \frac{1}{2}Q(\bar{\omega})\tilde{q} + \frac{1}{2}G^T(q)\tilde{\omega} \tag{4.10}$$

- For traditional TLC,

$$\dot{\tilde{q}} \simeq \left.\frac{\partial \dot{q}}{\partial q}\right|_{\begin{bmatrix}\bar{q}\\ \bar{\omega}\end{bmatrix}} \tilde{q} + \left.\frac{\partial \dot{q}}{\partial \omega}\right|_{\begin{bmatrix}\bar{q}\\ \bar{\omega}\end{bmatrix}} \tilde{\omega} = \frac{1}{2}Q(\bar{\omega})\tilde{q} + \frac{1}{2}G^T(\bar{q})\tilde{\omega} \tag{4.11}$$

- Now that the system has been linearized, we want to design the controller. We define the augmented error vector $q_{agm} = \left[\int \tilde{q}^T dt\ \tilde{q}^T\right]^T$ which yields,

$$\dot{q}_{agm} = \begin{bmatrix} 0_{4\times4} & I_{4\times4} \\ 0_{4\times4} & \frac{1}{2}Q(\bar{\omega}) \end{bmatrix} q_{agm} + \begin{bmatrix} 0_{4\times3} \\ \frac{1}{2}G^T(q) \end{bmatrix} \tilde{\omega} \tag{4.12}$$

- As per the TLC theory, we design a controller $\tilde{\omega} = K(t)q_{agm}$ where $K = [K_I\ K_P]$ such that

$$\dot{q}_{agm} = \begin{bmatrix} 0_{4\times4} & I_{4\times4} \\ -\omega_{kin}^2 I_{4\times4} & -2\omega_{kin} I_{4\times4} \end{bmatrix} q_{agm} \tag{4.13}$$

  where,

  1. Scalar ω_{kin} is the first tunable control parameter.
  2. $K_I = -2\omega_{kin}^2 G(q)$
  3. $K_P = -G(q)(4\omega_{kin} I_{4\times4} + Q(\bar{\omega}))$

- Therefore, the control input for attitude kinematics is given by $\omega = \tilde{\omega} + \bar{\omega}$. This completes controller 1 for the attitude kinematics.

4.2.3 Controller 2 - Attitude Dynamics

- The nominal trajectory $\bar{\omega}^*$ is equal to ω from the first controller.

- **Inverse Dynamics:** The nominal input torque $\bar{T}$ in terms of the nominal trajectory $\bar{\omega}^*$ and its time derivative $\dot{\bar{\omega}}^*$ is given by:

$$\bar{T} = J_s\dot{\bar{\omega}}^* + S(\bar{\omega}^*)h^* \tag{4.14}$$

  where $h^* = J_s\bar{\omega}^*$.

- **Differentiator:** This is used to calculate the derivative of $\bar{\omega}^*$.

- **Regulator:** The regulator generates the input tracking error.

  - The input tracking error $\tilde{T} = T - \bar{T}$ and the state trajectory error $\tilde{\omega}^* = \omega - \bar{\omega}^*$.
  - Applying fine linearization,

$$\dot{\tilde{\omega}}^* = A'(\tilde{\omega}^*, \bar{\omega}^*)\tilde{\omega}^* + B'\tilde{T} \tag{4.15}$$

where,

$$\begin{aligned} A'(\tilde{\omega}^*, \bar{\omega}^*) &= \int_0^1 \left.\frac{\partial \dot{\omega}}{\partial \omega}\right|_{\begin{bmatrix}\bar{\omega}^* \\ \bar{T}\end{bmatrix} + \lambda \begin{bmatrix}\tilde{\omega}^* \\ \tilde{T}\end{bmatrix}} d\lambda \\ &= -J_s^{-1}\left[S\left(\frac{1}{2}(\omega + \bar{\omega}^*)\right)J_s + S\left(\frac{1}{2}J_s(\omega + \bar{\omega}^*)\right)\right] \end{aligned} \tag{4.16}$$

and

$$B' = \int_0^1 \left.\frac{\partial \dot{\omega}}{\partial T}\right|_{\begin{bmatrix}\bar{\omega}^* \\ \bar{T}\end{bmatrix} + \lambda \begin{bmatrix}\tilde{\omega}^* \\ \tilde{T}\end{bmatrix}} d\lambda = J_s^{-1} \tag{4.17}$$

- Therefore,

$$\dot{\tilde{\omega}}^* = -J_s^{-1}\left[S\left(\frac{1}{2}(\omega + \bar{\omega}^*)\right)J_s + S\left(\frac{1}{2}J_s(\omega + \bar{\omega}^*)\right)\right]\tilde{\omega}^* + J_s^{-1}\tilde{T} \tag{4.18}$$

- For traditional TLC,

$$\begin{aligned} \dot{\tilde{\omega}}^* &\simeq \left.\frac{\partial \dot{\omega}}{\partial \omega}\right|_{\begin{bmatrix}\bar{\omega}^* \\ \bar{T}\end{bmatrix}} \tilde{\omega}^* + \left.\frac{\partial \dot{\omega}}{\partial T}\right|_{\begin{bmatrix}\bar{\omega}^* \\ \bar{T}\end{bmatrix}} \tilde{T} \\ &= -J_s^{-1}\left[S\left(\bar{\omega}^*\right)J_s + S\left(J_s(\bar{\omega}^*)\right)\right]\tilde{\omega}^* + J_s^{-1}\tilde{T} \end{aligned} \tag{4.19}$$

- Now that the system has been linearized, we want to design the controller. We define the augmented error vector $\omega_{agm} = \left[\int (\tilde{\omega}^*)^T dt \ (\tilde{\omega}^*)^T\right]^T$ which yields,

$$\dot{\omega}_{agm} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & A'(\tilde{\omega}^*, \bar{\omega}^*) \end{bmatrix} \omega_{agm} + \begin{bmatrix} 0_{3\times3} \\ B' \end{bmatrix} \tilde{T} \tag{4.20}$$

- As per the TLC theory, we design a controller $\tilde{T} = K'\omega_{agm}$ where $K' = [K_I' \ K_P']$ such that

$$\dot{\omega}_{agm} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ -\omega_{dyn}^2 M^2 & -2\omega_{dyn} M \end{bmatrix} \omega_{agm} \tag{4.21}$$

where,

1. Scalar ω_{dyn} is the second tunable control parameter.
2. M is chosen to be:

$$M = \begin{bmatrix} \sqrt{J_{11}} & 0 & 0 \\ 0 & \sqrt{J_{22}} & 0 \\ 0 & 0 & \sqrt{J_{33}} \end{bmatrix}^{-1} \tag{4.22}$$

   where, J_{ii} for $i = 1, 2, 3$ are the main diagonal elements of J_s.
3. $K_I' = -\omega_{dyn}^2 J_s M^2$
4. $K_P' = S\left(\frac{1}{2}(\omega + \bar{\omega}^*)\right)J_s + S\left(\frac{1}{2}J_s(\omega + \bar{\omega}^*)\right) - 2\omega_{dyn}J_s M$

- Therefore, the control input for attitude dynamics is given by $T = \tilde{T} + \bar{T}$. This completes controller 2 for the attitude dynamics.

4.3 Control parameters and attitude command trajectory generation

- The parameters ω_{kin} and ω_{dyn} must be tuned as per disturbances and actuator capabilities to guarantee system stability.

- The nominal trajectory $\bar{q}$ must be such that it does not exceed the control system capacity. Hence, a feasible trajectory $\bar{q}$ must be generated.

- A smooth trajectory of the following form is generated and utilized to avoid a sudden jump from initial q_0 at initial time t_0 to desired q_d:

$$\bar{q} = \frac{\bar{Q}}{||\bar{Q}||_2} \tag{4.23}$$

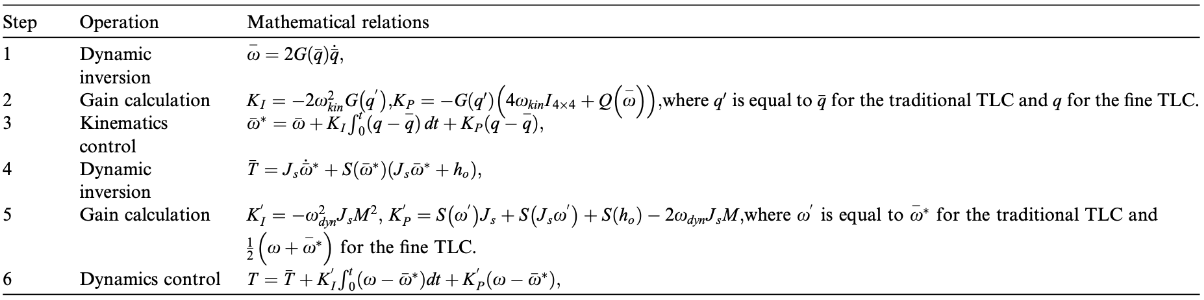
  where,

$$\bar{Q} = q_d + exp\left(-\frac{t - t_0}{\tau}\right)(q_0 - q_d) \tag{4.24}$$

  Here,

  - τ is a time constant and the third tunable control parameter. If τ is small, the variation is steeper and if τ is large, the variation is more gradual.
  - $||\bar{Q}||_2 = \sqrt{\bar{Q}^T\bar{Q}}$ included in the equation so that the norm of the quaternion is 1.

- This completes the discussion of the fine TLC method applied to the spacecraft attitude control problem. The table given in figure 4.1 summarizes the algorithm discussed. The next chapter analyses the simulation results.



Step	Operation	Mathematical relations
1	Dynamic inversion	$\bar{\omega} = 2G(\bar{q})\dot{\bar{q}}$,
2	Gain calculation	$K_I = -2\omega_{kin}^2 G\left(q'\right), K_P = -G(q')\left(4\omega_{kin}I_{4\times4} + Q\left(\bar{\omega}\right)\right)$,where q' is equal to $\bar{q}$ for the traditional TLC and q for the fine TLC.
3	Kinematics control	$\bar{\omega}^* = \bar{\omega} + K_I\int_0^t(q - \bar{q})\,dt + K_P(q - \bar{q})$,
4	Dynamic inversion	$\bar{T} = J_s\dot{\bar{\omega}}^* + S(\bar{\omega}^*)(J_s\bar{\omega}^* + h_o)$,
5	Gain calculation	$K_I^{'} = -\omega_{dyn}^2 J_s M^2,\ K_P^{'} = S\left(\omega^{'}\right)J_s + S\left(J_s\omega^{'}\right) + S(h_o) - 2\omega_{dyn}J_s M$,where $\omega^{'}$ is equal to $\bar{\omega}^*$ for the traditional TLC and $\frac{1}{2}\left(\omega + \bar{\omega}^*\right)$ for the fine TLC.
6	Dynamics control	$T = \bar{T} + K_I^{'}\int_0^t(\omega - \bar{\omega}^*)dt + K_P^{'}(\omega - \bar{\omega}^*)$,

Figure 4.1: Spacecraft attitude control algorithm - Summary

Chapter 5

Numerical Simulation Results

This chapter discusses the numerical simulations carried out and presents an analysis of them. The results reported by the reference have been replicated and discussed.

5.1 Model parameters and variables used

- Simulation time = 1800 s
- Desired Euler angle trajectory, $Euler_d = [Roll_d \; Pitch_d \; Yaw_d]^T$ such that,

$$Roll_d = \begin{cases} -90\cos(0.025t) \; deg & 0 \leq t < 1100 \; s \\ 0 \; deg & t \geq 1100 \; s \end{cases} \tag{5.1}$$

$$Pitch_d = \begin{cases} -30\sin(0.005t) \; deg & 0 \leq t < 1100 \; s \\ 0 \; deg & t \geq 1100 \; s \end{cases} \tag{5.2}$$

$$Yaw_d = \begin{cases} 75 \; deg & 0 \leq t < 1100 \; s \\ 0 \; deg & t \geq 1100 \; s \end{cases} \tag{5.3}$$

- $\omega_{kin} = 0.01$
- $\omega_{dyn} = 15$
- The nominal inertia matrix J_s is,

$$J_s = \begin{bmatrix} 147 & 6.5 & 6 \\ 6.5 & 158 & 5.5 \\ 6 & 5.5 & 137 \end{bmatrix} kgm^2 \tag{5.4}$$

  The reference also accounts for an uncertainty in the spacecraft inertia matrix such that the true inertia matrix is $(1 + c)J_s$. The influence of c has been illustrated.

- The external disturbances T_d,

$$T_d = \begin{bmatrix} 1 + 2\sin(0.005t) \\ -1 - 5\cos(0.005t) \\ 2 - 4\cos(0.005t) \end{bmatrix} Nm \tag{5.5}$$

- Initial conditions:

  - $\omega_0 = [1 \; -1 \; 1]^T \; deg/s$
  - $T_0 = [0 \; 0 \; 0]^T \; Nm$
  - $Roll(0) = 42°$, $Pitch(0) = -19°$, $Yaw(0) = -7°$

- The maximum control torque $T_{\max} = 0.2 \; Nm$

- Performance index J_x:

$$J_x = \frac{1}{2}\int_0^{\infty}\left[\frac{1}{180^2}(Euler - Euler_d)^T(Euler - Euler_d)\right]dt \tag{5.6}$$

- Performance index J_u:

$$J_u = \frac{1}{2}\int_0^{\infty}\left[\frac{1}{T_{\max}^2}T^T T\right]dt \tag{5.7}$$

5.2 Results and Analysis

- The reference solves the model differential equations using a fourth-order Runge-Kutta method with a step size of $0.05s$. It assumes that all of the measurements are obtained every 1 s, and the control torque is fed into the spacecraft model every 1 s in the simulations.

- In my simulations, I use the *ode*45 function in MATLAB to solve the differential equations with an adaptive time step.

- In the reference, in order to approximate the time derivative of $\bar{\omega}^*$, the second-order pseudo differentiator 3.5 with $\omega_d = 0.25$ is used. The following issues have been observed:

  - The paper gives no details on exactly how it was implemented. The values of $\bar{\omega}^*$ are only available at discrete locations and we do not have a continuous function to which we can apply the differentiator.
  - I first attempted to convert the differentiator to the time domain and use the convolution integral. However, these computations were very time-intensive.
  - I next attempted to fit the curve using a Fourier series expansion and took the Laplace transform of that fitted curve. The differentiator could then be applied. While the results looked reasonable, the time taken was still significant.
  - I then used the backward Euler method and found that the results were comparable and the time taken was much lesser.

- The paper also hasn't explicitly mentioned how they incorporated the torque saturation effects. I tried to include those, however, with saturation effects, the function becomes non-smooth leading to more errors getting accumulated.

- The paper referred also gives a result for the performance for different amounts of noise for the control system. No implementation details have been shared by them for this. Hence, this graph has not been reproduced.

- Another place where I believe the paper could have detailed their method of implementation more was in the final computation of the quaternions. Once we have the final angular velocities we would like to recompute the quaternions and Euler angles obtained corresponding to these angular velocities. However, from standard methods, the accuracy obtained is not as high as desired.
  - I first tried to implement equation 4.1. However, high errors were observed.
  - I then tried to analytically integrate the equation to find the solution. The results from this had very high oscillations.
  - I finally used a rearranged form of equation 4.1 which gave the best result so far, though it does suffer from inaccuracies.

The following subsections display and discuss the results.

5.2.1 Case 1 - $\tau = 1$ $c = 0$

The results for when $\tau = 1$, $c = 0$ have been given below.

From the reference paper:

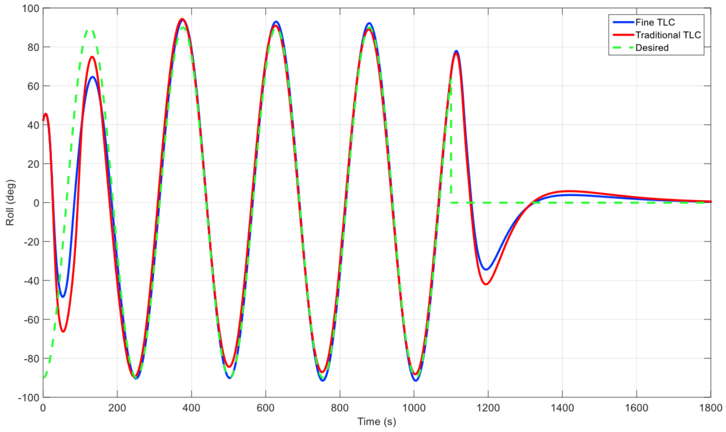


Figure 5.1: Reference: Roll angle variation

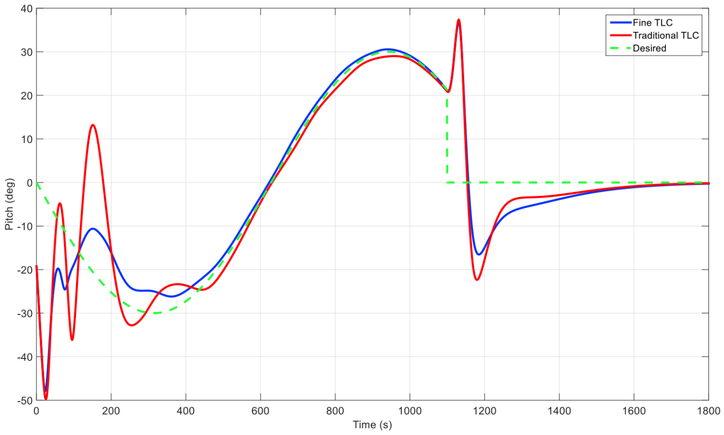


Figure 5.2: Reference: Pitch angle variation

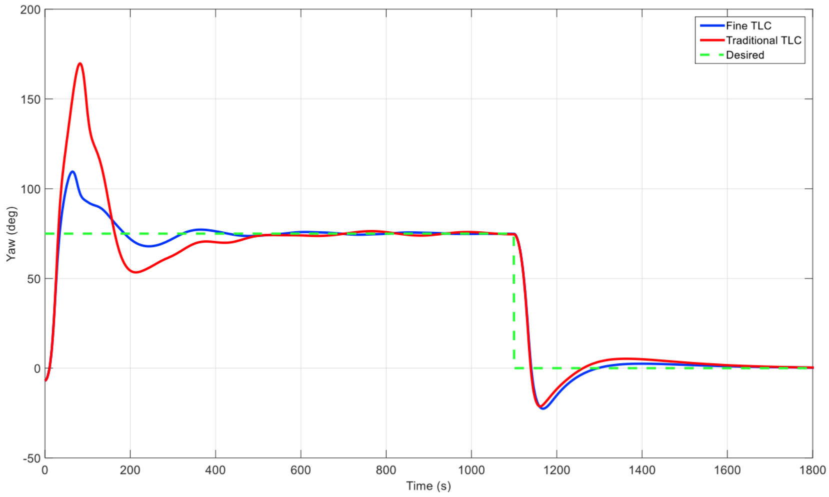


Figure 5.3: Reference: Yaw angle variation

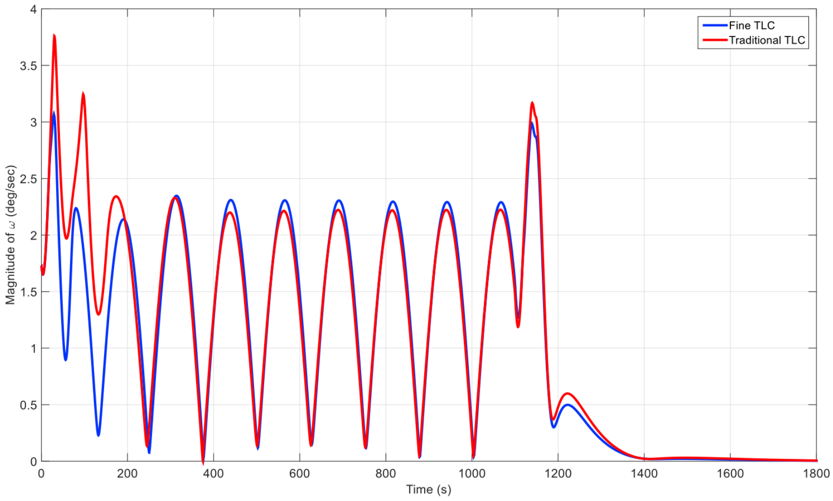


Figure 5.4: Reference: Angular velocity magnitude variation

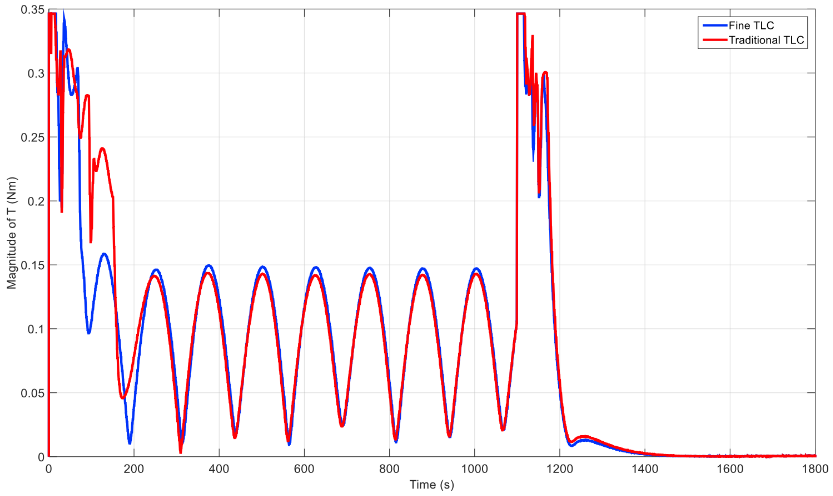


Figure 5.5: Reference: Control torque variation (saturated in $\sqrt{3}T_{\max}$)

Reproduced results:

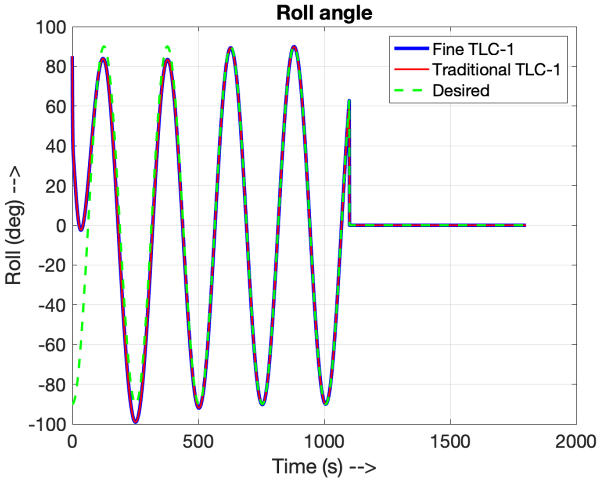


Figure 5.6: Reproduced: Roll angle variation from the first controller

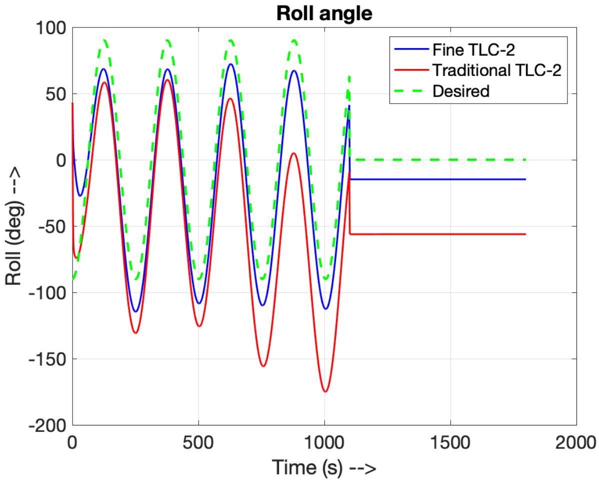


Figure 5.7: Reproduced: Roll angle variation from the second controller

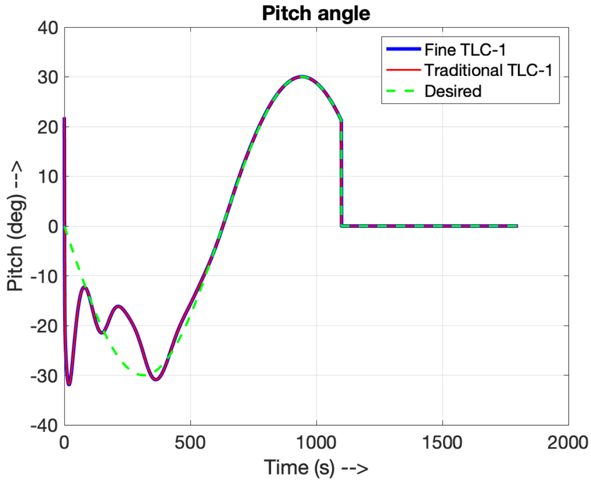


Figure 5.8: Reproduced: Pitch angle variation from the first controller

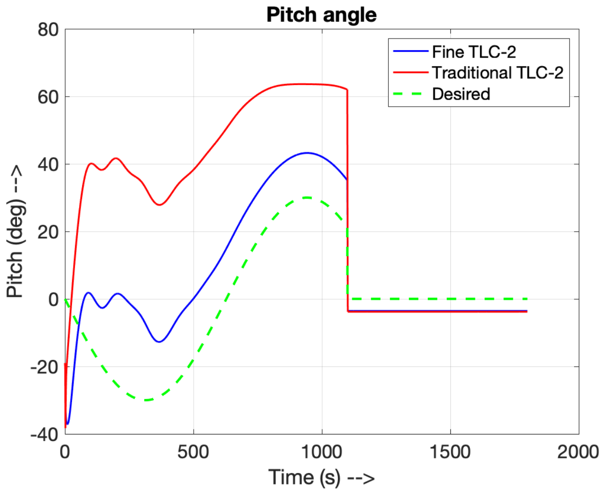


Figure 5.9: Reproduced: Pitch angle variation from the second controller

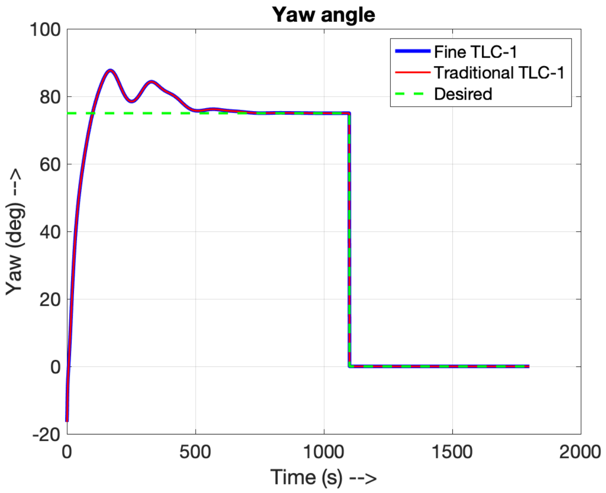


Figure 5.10: Reproduced: Yaw angle variation from the first controller

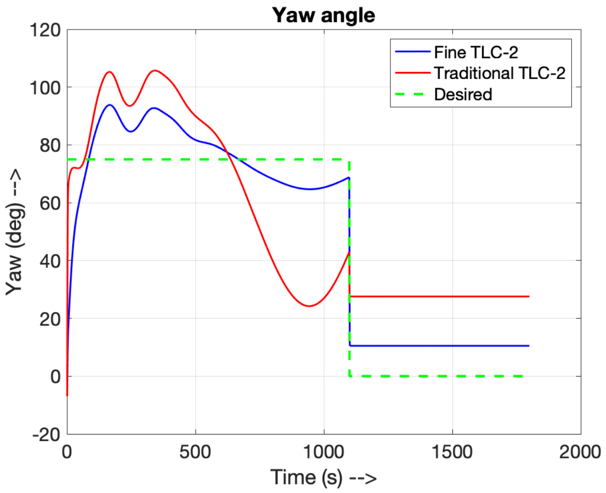


Figure 5.11: Reproduced: Yaw angle variation from the second controller

The plot below shows the variation of the final angular velocity magnitude. There is a spike at 1100 s that was not in the results reported by the paper. This may be attributed to the maximum allowable value of T. The results I have produced have not imposed the upper limit on T as this leads to high numerical errors. This is why the plots for the control torque also include higher values. Even with the deviations, the trends are observed to match.

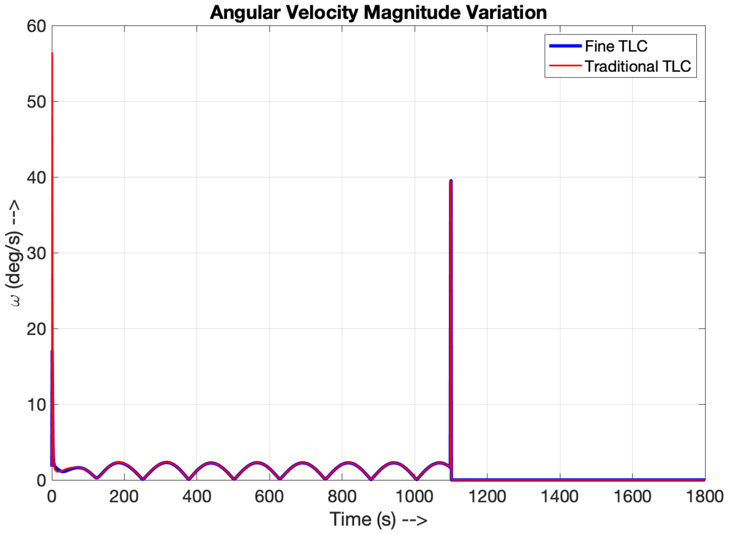


Figure 5.12: Variation of the final angular velocity magnitude

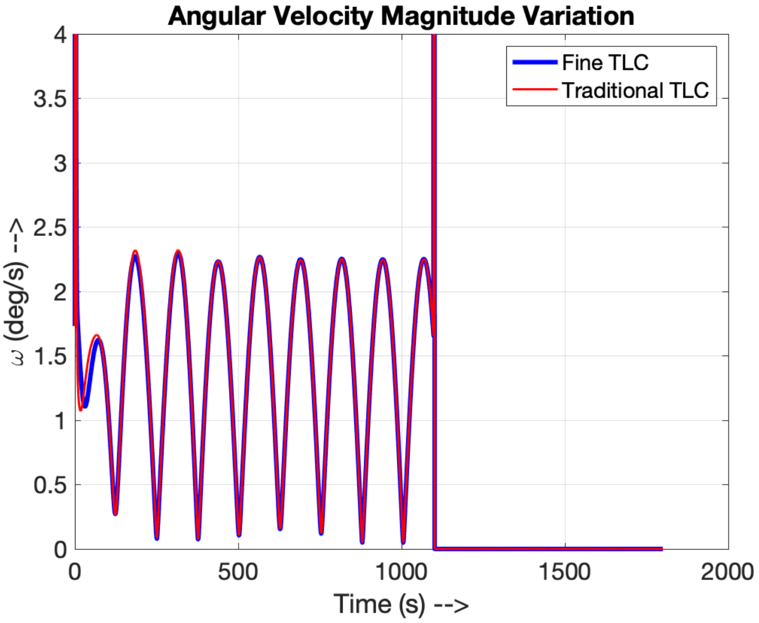


Figure 5.13: Variation of the final angular velocity magnitude - Zoomed in

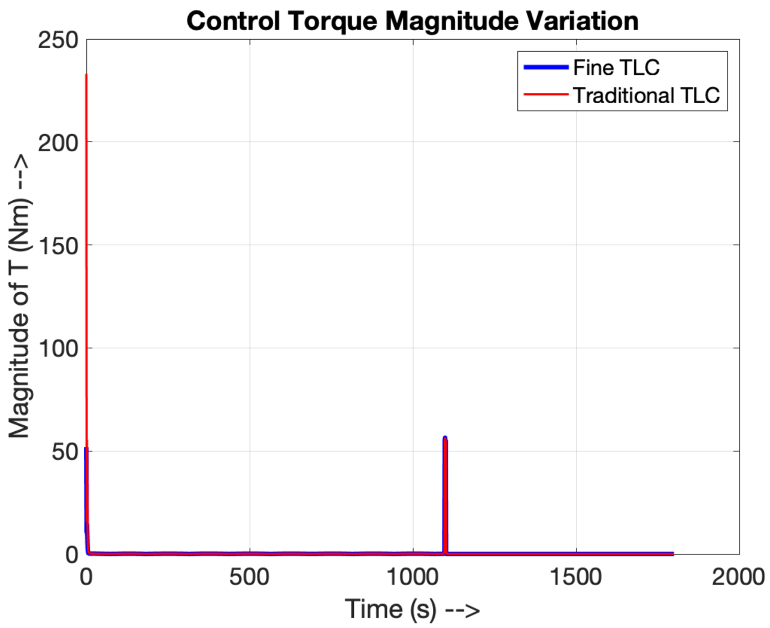


Figure 5.14: Variation of the control torque magnitude

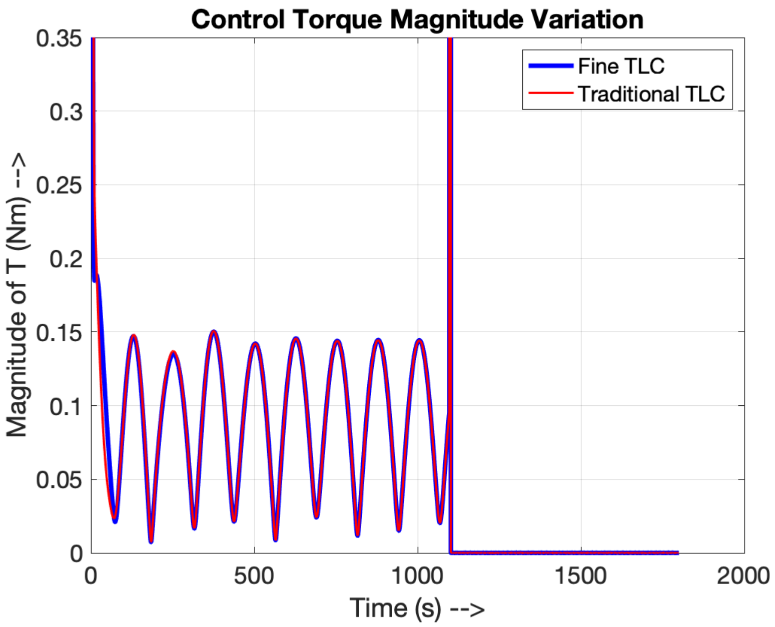


Figure 5.15: Variation of the control torque magnitude - Zoomed in

5.2.2 Case 2 - Performance Indices

For the plots for the performance indices, the deviations observed may again be attributed to the torque upper bound. However, the trends have been captured.

From the reference paper:

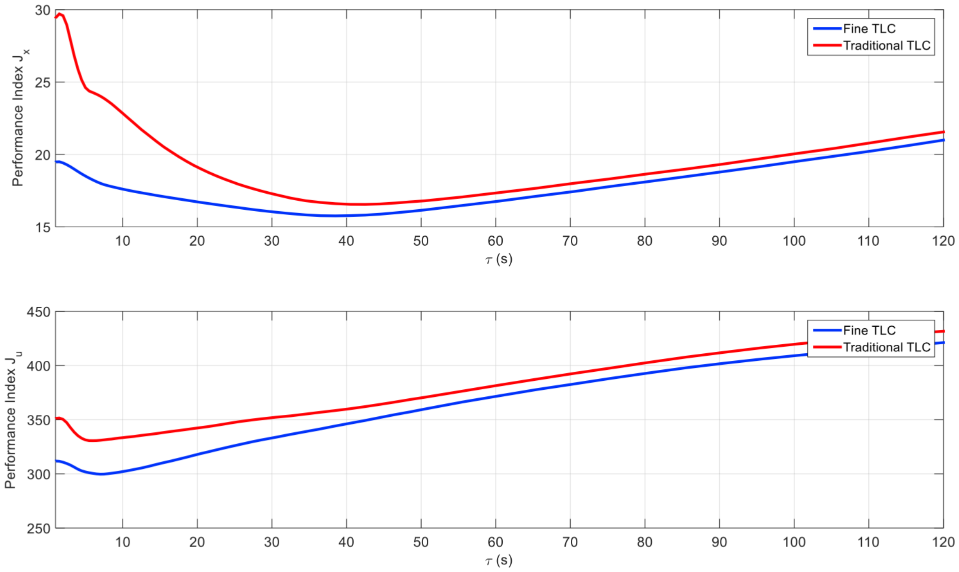


Figure 5.16: Reference: Performance indices - Variation with τ

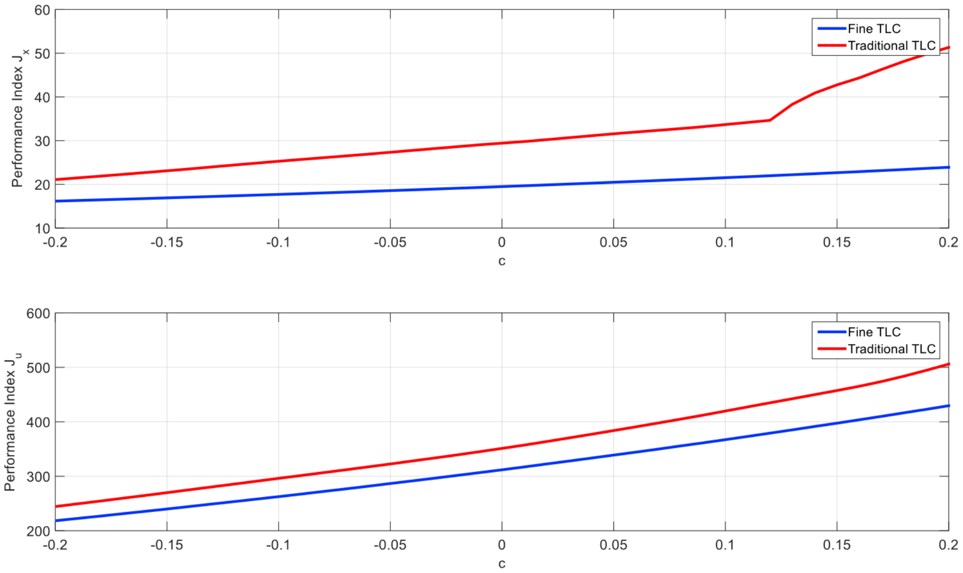


Figure 5.17: Reference: Performance indices - Variation with c

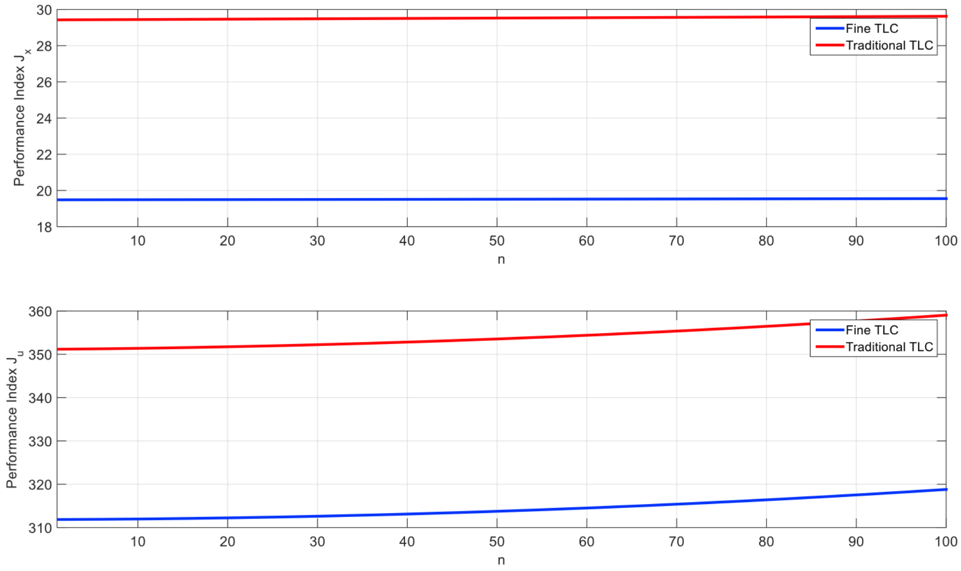


Figure 5.18: Reference: Performance indices - Variation with noise

Reproduced results:

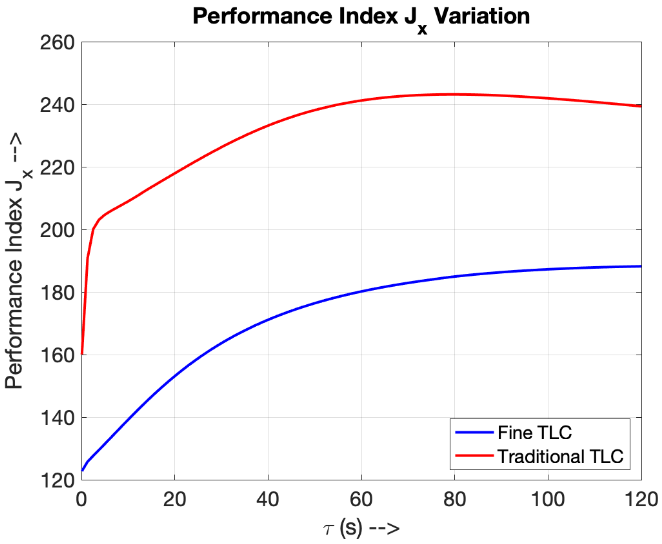


Figure 5.19: Performance index J_x - Variation with τ

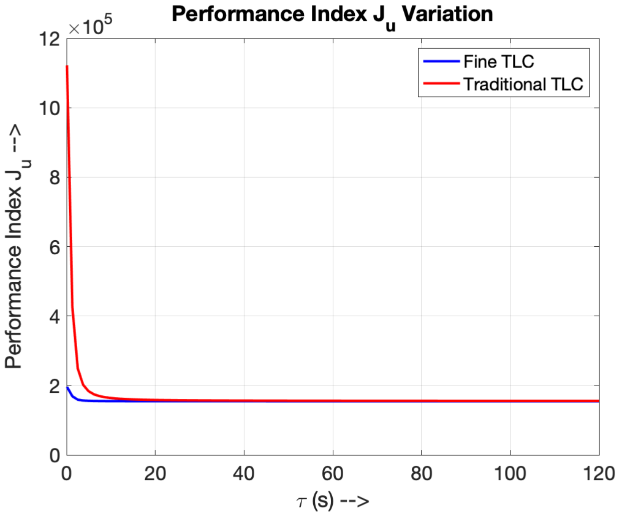


Figure 5.20: Performance index J_u - Variation with τ

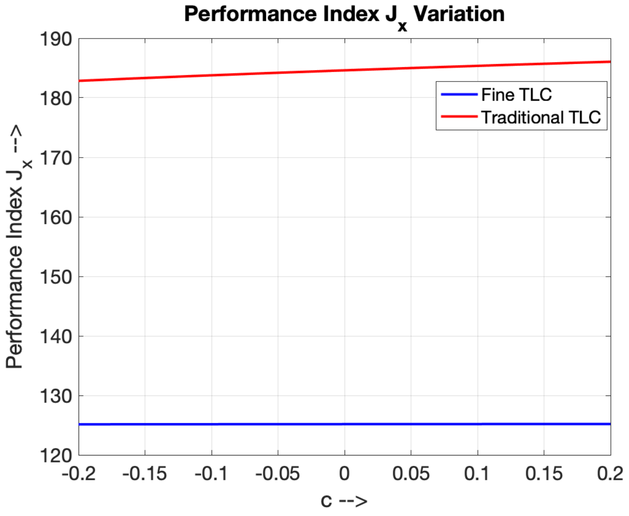


Figure 5.21: Performance index J_x - Variation with c

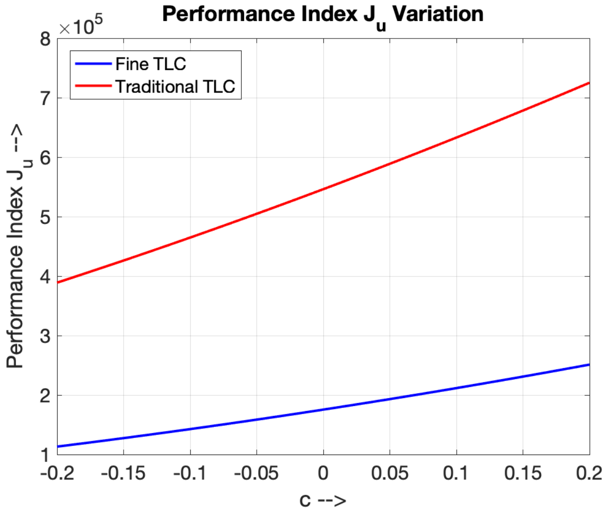


Figure 5.22: Performance index J_u - Variation with c

Chapter 6

Concluding Remarks

- This project analyses the fine TLC method applied to the spacecraft attitude control system.
- The reference used is: Lasemi, N & Shaker, HR 2021, 'Spacecraft Attitude Control: Application of Fine Trajectory Linearization Control', Advances in Space Research, vol. 68, no. 9, pp. 3663-3676. https://doi.org/10.1016/j.asr.2021.08.018
- The results in the paper have been analyzed and it can be concluded that the fine TLC approach offers an improvement in accuracy and superior performance as can be seen from the numerical simulations.
- The method is useful for the spacecraft attitude system amongst other nonlinear dynamical systems.
- All the simulations have been produced using MATLAB.