Imperial College London Control UROP

Athanasios Kantas, Ali Orkun Ozkan, Henry Hausamann

July-September 2022

1 Abstract

The 2DoF quadrotors used in the second year Control Systems ELEC50004 laboratory are controlled using a multi-loop PID controller, designed by the frequency-domain approach. This paper covers a new, modern and faster way of achieving attitude and position control by means of using reduced-order observer-based feedback control, designed by the state-space approach.

2 Introduction

The aim of this project is to create a state-feedback position control scheme for a quadrotor drone with two degrees of freedom, namely pitch and unilateral motion. This was done by controlling position, velocity, pitch angle, pitch rate and angular moment of the drone while rejecting disturbances to the velocity and pitch rate dynamics, which where modelled as uncontrollable states. In addition, a reduced order observer had to be designed to create an estimation of the unobservable states within the system. This allows for a "full-state" feedback control law design by the separation principle, excluding the disturbances which are uncontrollable.

3 Contributions

The initial research and creation of the 3D observer and control scheme was performed jointly by all 3 group members. Orkun and Athanasios then expanded the observer to contain a fourth state while Henry adjusted the control scheme to work in 4D. The group then split the remaining duties into two parts, with Orkun and Henry implementing the attitude control schemes on the ANT-X platform, while Athanasios mathematically proved the 7D non-linear state-space model for the position dynamics from first principles and linearising said model. The 7D control scheme was then created and verified in Simulink by Henry, with Orkun leading implementation and debugging on the ANT-X platform. The proof for the 7D and 4D controller coordinate transformation and the 7D controller were provided by Athanasios and Orkun, respectively. Athanasios directed the documentation and writing of the report and Orkun assisting primarily under the peer review process.

4 Actuator and Pitch Dynamics

4.1 State Space Representation

First step is to design a reduced-order observer in order to estimate the unmeasured states: the actual pitch moment M_a on the quad-rotor and the angular rate disturbance d, given that it cannot be measured directly from the system. These states will be later used for a state-feedback controller.

The pitch angle dynamics of the quad-rotor are defined in the following system:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

The state x is defined as the following vector:

$$x = \begin{bmatrix} \theta & q & M_a & d_q \end{bmatrix}^T \tag{1}$$

Where $x_1 = \begin{bmatrix} \theta \\ q \end{bmatrix}$ and $x_2 = \begin{bmatrix} M_a \\ d_q \end{bmatrix}$ correspond to the measured and unmeasured parts of vector x in equation (1) respectively. The vector x_2 is unmeasured but observable such that $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Also, θ is the pitch angle of the actuators, q is the pitch rate, M_a is the pitch moment, and d is a constant torque disturbance.

Therefore, the actuator and pitch dynamics are described by:

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{M_a} \\ \dot{d_q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -{\omega_n}^2 & -2\xi\omega_n & \mu & 1 \\ 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ q \\ M_a \\ d_q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/\tau \\ 0 \end{bmatrix} M_c^d, \tag{2}$$

where: $\mu = 358.5$ is the input gain, $\omega = 3.44$ is the natural frequency, $\xi = 0.084$ is the damping ratio, and $\tau = 0.035$ is a constant.

4.2 Reduced-Order Observer Design

In the 4D system (2) described in the previous section, only θ and q can be measured from the output $y = Cx = \theta + q$. As s result, a reduced-observer feedback system is to be designed to estimate M_a and d_q as accurately as possible. Below is a block diagram showing the structure of the reduced order observer:

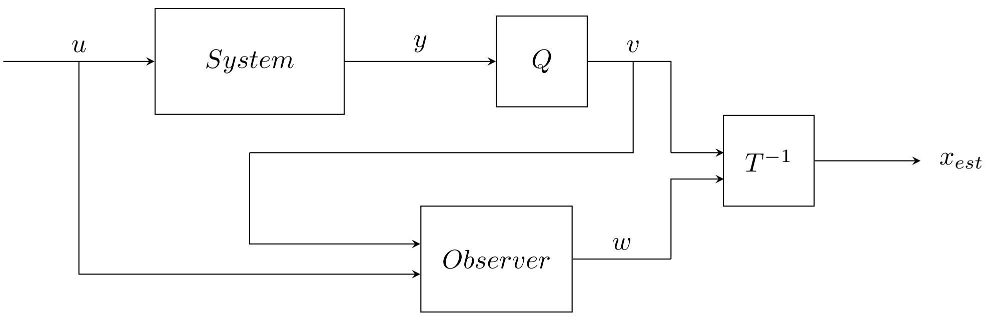

Figure (I): Reduced Order Observer Diagram

1. Determine Q

Matrix Q is used to extract the information about x_2 from the output y of our system, as seen in figure (I). The first step to compute it is to identify the C_2 matrix, derived from the C matrix elements that correspond to x_2, as seen in (1).

It is easy to see that since y contains no information about x_2 then:

$$C_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{thus from the following identity: } QC = \begin{bmatrix} I & C_2 \end{bmatrix}$$

$$\text{It can be concluded that: } Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \tag{3}$$

2. Determine L

Before matrix L is computed, the system equation of the observer is first be defined:

$$\dot{\xi} = F\xi + Hv + Gu,$$
$$w = \xi + Lv, \tag{4}$$

where ξ is the internal variable of the observer and w is the output, equal to the estimate of $x_2 = \begin{bmatrix} M_a \\ d_q \end{bmatrix}$, and v is the modified output $v = Qy = x_1$ as seen in figure (I). The system should follow the following two properties:

- Asymptotic stability is required, so matrix F should have eigenvalues with negative real part.
- The system must react fast, so the eigenvalues must have a great magnitude.
- The on-board controller works in discrete time, so a continuous-time to discrete-time transform occurs in the flight-control implementation. If the magnitude of the eigenvalue is too large, then the discrete model could become unstable.

It is known from theory [1] that matrix F is given by:

$$F = A_{22} - LA_{12}$$

So it must be ensured that the properties apply on this expression. To check that, the MATLAB "place" function was used, by providing the values of: A_{12} , A_{22} , and the poles at $-30, -25$.

The end result was:

$$L = \begin{bmatrix} 0 & 0.0005 \\ 0 & 6.2500 \end{bmatrix}$$

Matrix L was chosen so that the settling time of the estimation error of x_2 is less than 0.1s.

3. Computation of Observer

The formulas to find the rest of the matrices defining the observer system are known so:

$$H = FL - LA_{11} + A_{21} = \begin{bmatrix} 0.0059 & -0.0271 \\ 310.6320 & -678.5796 \end{bmatrix}$$

$$G = -LB_1 + B_2 = \begin{bmatrix} 28.5714 \\ 0 \end{bmatrix}$$

$$F = \begin{bmatrix} -28.7 & 0.000 \\ -9.4106 \times 10^3 & -26.3 \end{bmatrix}$$

The observer system is now complete.

4.3 Deriving the Equilibrium Point

The input u to the observer is the control moment M_c^d, thus a formula must be derived for the input in terms of the equilibrium state that the drone aims to reach. This is done by setting the equilibrium conditions: $\dot{x^e} = Ax^e + Bu^e = 0$:

$$\dot{\theta^e} = 0 \quad \Rightarrow \quad q^e = 0$$

$$\dot{d_v^e} = \dot{d_q^e} = 0 \quad \Rightarrow \quad d_v^e = d_v \ , \ d_q^e = d_q$$

$$\dot{q^e} = 0 \quad \Rightarrow \quad -\omega_n^2\theta_e - 2\xi\omega_n q_e + \mu M_a^e + d_q^e = 0 \quad \Rightarrow \quad -\omega_n^2\theta^e + \mu M_a^e + d_q = 0 \quad \Rightarrow \quad M_a^e = \frac{\omega_n^2}{\mu}\theta^e - \frac{d_q}{\mu}$$

$$\dot{M}_a = 0 \quad \Rightarrow \quad \frac{1}{\tau}[M_a^e - M_c^{d\,e}] = 0 \quad \Rightarrow \quad M_a^e = M_c^{d\,e} \quad \Rightarrow \quad \boxed{M_c^{d\,e} = \frac{\omega_n^2}{\mu}\theta^e - \frac{d_q}{\mu}} \tag{5}$$

To test the observer, the equilibrium angle value θ^e is given. The input $u = M_c^d$ is then calculated from that. All of the states can thus be observed by the observer during time period taken for the equilibrium point to be reached.

4.4 Designing the Controller

The system is now fully defined and all the states can be observed. As a result, a controller can be set in place so that the drone states reach the desired equilibrium state described in (5).

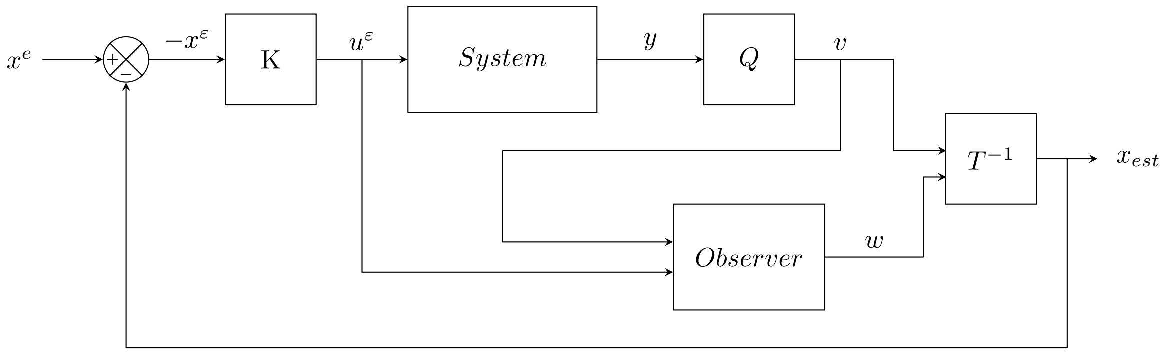

Figure (2) : Closed Loop System Diagram

The input u^ε to the closed loop system is thus defined as: $u^\varepsilon = -Kx^\varepsilon = -Kx + Kx^e$, where x^e is the equilibrium state that the system reaches over time. This is because x^ε asymptotically converges to 0.

Change of Coordinates

Below is the proof of how when the system is described in terms of the error coordinates the dynamics of the system do not change.

Our original system:

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{M}_a \\ \dot{d}_q \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -{\omega_n}^2 & -2\xi\omega_n & \mu & 1 \\ 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ q \\ M_a \\ d_q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/\tau \\ 0 \end{bmatrix} M_c^d$$

The equilibrium state values are summarised below from the previous section:

$$q^e = 0, \qquad M_a^e = -\frac{\omega_n^2}{\mu}\theta^e + \frac{d_q^e}{\mu} = M_c^{d\,e}, \qquad d^e\ ,\ \theta^e \text{ are constant.}$$

Hence:

$$\begin{aligned} \dot{\theta}^\varepsilon &= \dot{\theta} - \dot{\theta}^e \qquad \text{where:} \qquad \dot{\theta}^e = q^e = 0 \\ &= q = q^\varepsilon + q^e \\ &= q^\varepsilon \end{aligned}$$

$$
\begin{aligned}
\dot{q^\varepsilon} &= \dot{q} - \dot{q^e} \qquad \text{where:} \qquad \dot{q^e} = 0 \\
&= q = -\omega_n^2\theta - 2\omega_n q + \mu M_a \\
&= -\omega_n^2(\theta^\varepsilon + \theta^e) - 2\omega_n(q^\varepsilon + q^e) + \mu(M_a^\varepsilon + M_a^e) + d^\varepsilon + d^e \qquad \text{where:} \qquad M_a^e = -\frac{\omega_n^2}{\mu}\theta^e + \frac{d_q^e}{\mu} \\
&= -\omega_n^2\theta^\varepsilon - 2\omega_n q^\varepsilon + \mu M_a^\varepsilon + d^\varepsilon
\end{aligned}
$$

$$
\begin{aligned}
\dot{M_a^\varepsilon} &= \dot{M_a} - \dot{M_a^e} \qquad \text{where:} \qquad \dot{M_a^e} = 0 \\
&= \dot{M_a} = -\frac{1}{\tau}[M_a - M_c^d] \\
&= \dot{M_a} = -\frac{1}{\tau}[M_a^\varepsilon + M_a^e - M_c^{d\,\varepsilon} - M_c^{d\,e}] \qquad \text{where:} \qquad M_a^e = M_c^{d\,e} \\
&= -\frac{1}{\tau}[M_a^\varepsilon - M_c^{d\,\varepsilon}]
\end{aligned}
$$

It has thus been proven that for the 4D linear system the matrices A^ε and B^ε from the system $\dot{x^\varepsilon} = A^\varepsilon x^\varepsilon + B^\varepsilon x^\varepsilon$ are equal to matrices A and B from the original system.

Because of the closed loop, the input u^ε can now be described in terms of the states and their desired equilibrium value:

We can thus prove that:

$$
\begin{aligned}
u^\varepsilon = M_c^d = -Kx^\varepsilon = -K\begin{bmatrix}\theta^\varepsilon \\ q^\varepsilon \\ M_a^\varepsilon \\ d_q^\varepsilon\end{bmatrix} &= -K\begin{bmatrix}\theta \\ q \\ M_a \\ d_q\end{bmatrix} + K\begin{bmatrix}\theta^e \\ q^e \\ M_a^e \\ d_q^e\end{bmatrix} \\
\Rightarrow u^\varepsilon &= -K\begin{bmatrix}\theta \\ q \\ M_a \\ d_q\end{bmatrix} + K\begin{bmatrix}\theta^e \\ 0 \\ -\frac{\omega_n^2}{\mu}\theta^e + \frac{d_q^e}{\mu} \\ d_q^e\end{bmatrix}
\end{aligned} \tag{6}
$$

Thus describing the transformed system in terms of the current and the desired states:

$$
\begin{aligned}
\dot{x^\varepsilon} &= A^\varepsilon x^\varepsilon + B^\varepsilon x^\varepsilon = Ax^\varepsilon + Bu^\varepsilon \qquad \text{where:} \qquad u^\varepsilon = -Kx^\varepsilon && (7) \\
&= Ax^\varepsilon - BKx^\varepsilon = (A - BK)x^\varepsilon && (8) \\
&= (A - BK)\begin{bmatrix}\theta^\varepsilon \\ q^\varepsilon \\ M_a^\varepsilon \\ d_q^\varepsilon\end{bmatrix} \qquad \Rightarrow \qquad \dot{x^\varepsilon} = (A - BK)\begin{bmatrix}\theta - \theta^e \\ q \\ M_a + \frac{\omega_n^2}{\mu}\theta^e - \frac{d_q^e}{\mu} \\ 0\end{bmatrix} && (9)
\end{aligned}
$$

It should be noted that M_a and d_q are both generated by the observer, thus the reason why it is correct to use them for feedback is due to the Separation Principle. The disturbance d_q is not a state that can be controlled, so: $d_q^e = d_q \Rightarrow d_q^\varepsilon = 0$. The controller gain matrix can thus be only implemented for the 3 controllable states. To ensure stability of the closed loop system the matrix $A - BK$ must be Hurwitz stable. The 3 poles where chosen at: [-10 -50 -60].
The control gain matrix is thus given by:

$$K = \begin{bmatrix}2.7909 & 0.3924 & 3.1798\end{bmatrix}$$

4.5 Simulating and Testing the Closed Loop System

4.5.1 Simulation

The closed loop control scheme was easy to simulate using a Simulink model, with d_q being simulated as a non-zero initial condition in the system state integrator. The simulation model served two main purposes

- Verifying the asymptotic stability of the control scheme
- Regulating the feedback control gain matrix K

The gain K was adjusted such that the maximum range of M_c^d would fall between 1 and -1.

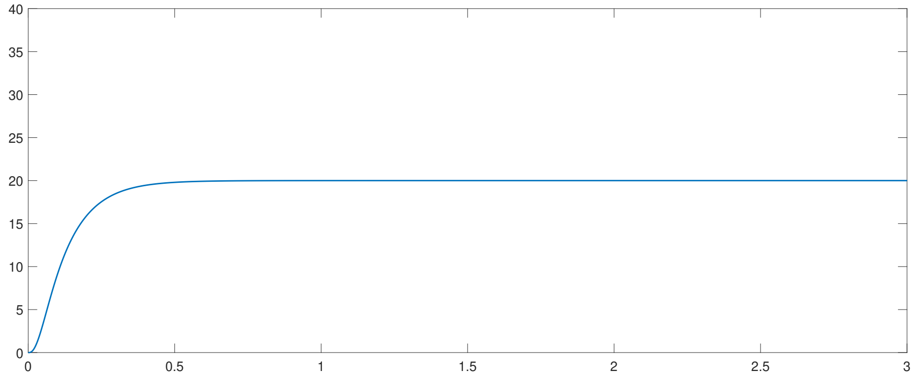

Figure (3): Time history of θ of undisturbed system in simulation [3D]

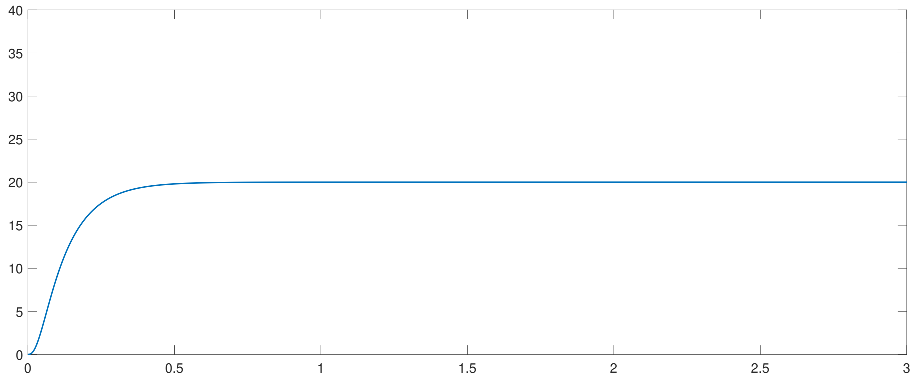

Figure (4): Time history of θ of disturbed system in simulation [4D]

4.5.2 Application on Physical Drone

Applying our 4D control scheme to the ANT-X drone was faced with one major issue: the control torque M_c^d would almost always oscillate in an exponentially increasing pattern during the arming stage of test runs, causing the drone to crash. This was solved by only applying thrust to the rotors when a setpoint command was given to the drone.

After much testing, it was discovered that the observer would only estimate the states M_a and d_q accurately within a small range of poles: {-25, -30} being identified as suitable. An additional source of inaccuracy was eliminated by resetting the observer integrator prior to each new setpoint command.

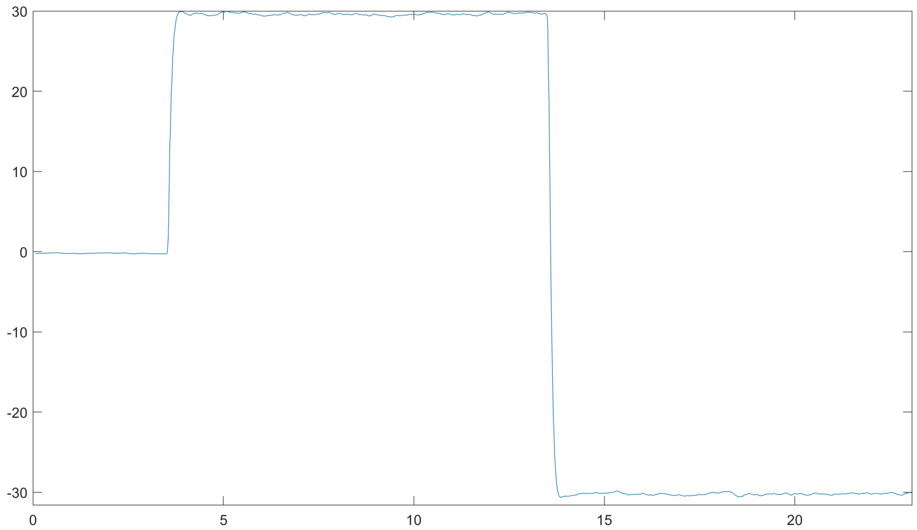

Figure (5): Time history of θ of undisturbed system [3D]

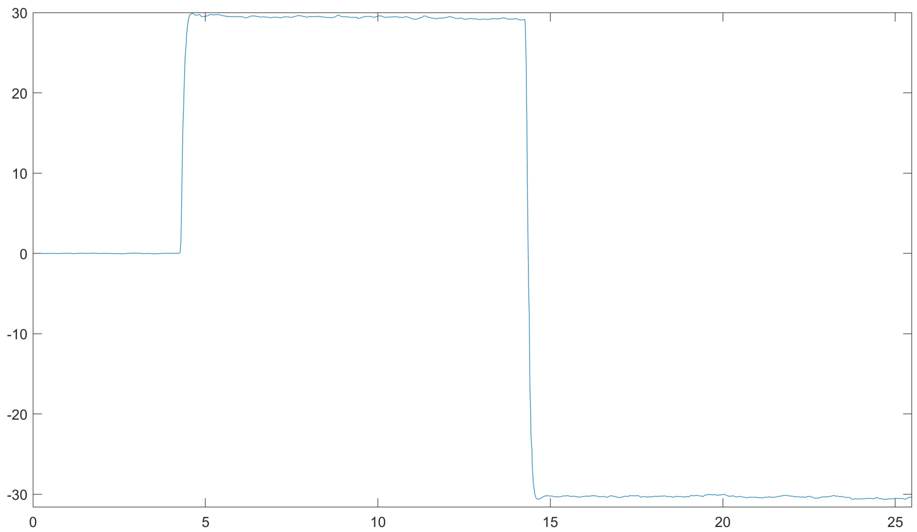

Figure (6): Time history of θ of disturbed system [4D]

Both Fig. 5 and Fig. 6 are logged data from the ANT-X platform where the drone is commanded to hold an angle to a square wave of magnitude 30 degrees, without and with disturbance rejection, respectively.

Conclusions gathered from Fig. 5 are that the controller is operating as desired, since the angle is being reached and is robust enough to successfully execute rapid changes of angles, with an expected amount of noise. The settle time is within reason and the overshoot is tolerable.

Fig. 6 confirms that scaling the 3D control scheme to mitigate for any delay operates desirably, as seen by the noisiness of the angle being much more dampened.

5 Position and Velocity Dynamics

The drone used in this experiment, has 2 degrees of freedom, pitch rotation and axial translation. Section 3 covered rotation, so this section is the natural continuation of the previous one, which covers translation.

5.1 Complete State Space Representation

To have complete control of the drone, the equations of motion must first be found, which will later be linearized so that the state space model can be created.

5.1.1 Obtaining the non linear model

To begin, some points must be made in order to model the drone system correctly:

- The frame of reference used to describe the equations of motion is non inertial, it is thus rotating and moving with the drone itself. Thus, fictitious forces must be added to the equations of motion using the D'Alambert principle
- The centre of mass of the drone, and the axis of rotation are not the same point, but at distance L apart. From the Parallel axis theorem, a new Moment of Inertia is calculated: $J' = J_\theta + mL^2$
- The mass of the cart holding the drone m_d must be taken into account, so the combined mass is: $m' = m_d + m_c$

As a result, the centrifugal force should be added to the force equation since the centre of mass is preforming circular motion around the axis of rotation. Moreover, the inertial torque produces a linear force and the inertial force produces a torque.

Below are the two equations of motion obtained from mechanical analysis:

$$\Sigma F = mq^2 L \sin\theta - \eta \bar{T} \sin\theta - cv_x - m\dot{q}L\mathrm{cos}\theta - \mathrm{d_v} = 0 \tag{10}$$

$$\Sigma \tau = \mu M_a - mgL\sin\theta - kq - mL\dot{v}cos\theta - d_q = 0 \tag{11}$$

Both equations of motion are a result of Newton's second law for linear and rotational motion, thus is known that: $\Sigma F = m'\dot{v}$ & $\Sigma\tau = J'\dot{q}$. However, the two equations of motion (10) & (11) must be solved simultaneously to give expressions for $\dot{v}$ and $\dot{q}$ separately.

After separation, the non-linear equations are given by:

$$\dot{q} = \frac{M_a m'\mu - d_q m' - km'q + d_v\lambda\cos\theta - \frac{\lambda^2 q}{2}\sin 2\theta + c\lambda v\cos\theta - gm'\lambda\sin\theta + \frac{\bar{T}\eta\lambda}{2}\sin 2\theta}{J'm' - \lambda^2\cos^2\theta} \tag{12}$$

$$\dot{v} = \frac{d_q\lambda\cos\theta - J'd_v - J'cv - J'\bar{T}\eta\sin\theta + \frac{g\lambda^2}{2}\sin 2\theta + J'q^2\lambda\sin\theta - M_a\mu\lambda\cos\theta + k\lambda q\cos\theta}{J'm' - \lambda^2\cos^2\theta} \tag{13}$$

Where: $= mL$

5.1.2 Linearizing the System

The last step is to linearize the equations using the Jacobian linearization matrices:

$$J_x = \begin{bmatrix} \frac{\partial f1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}, J_u = \begin{bmatrix} \frac{\partial f1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial u_1} & & \frac{\partial f_m}{\partial u_n} \end{bmatrix}$$

Computed at $x = x^e$ and $u = u^e$, where:

$$\text{The state vector: } x = \begin{bmatrix} p & v & \theta & q & M_a & d_v & d_q \end{bmatrix}^T \tag{14}$$

$$\text{The equilibrium state: } x^e = \begin{bmatrix} p & 0 & 0 & 0 & M_a^e & 0 & 0 \end{bmatrix}^T \tag{15}$$

$$\text{The input: } u = u^e = M_c^d \tag{16}$$

So that if $\dot{x} = Ax + Bu = f(x, u)$, then:

$$\delta\dot{x} = J_x \delta x + J_u \delta u \Rightarrow$$

$$J_x = A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{J'c}{\sigma_1} & -\frac{J'\bar{T}\eta - gL^2m^2}{\sigma_1} & \frac{Lkm}{\sigma_1} & -\frac{Lm\mu}{\sigma_1} & -\frac{J'}{\sigma_1} & \frac{Lm}{\sigma_1} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{Lcm}{\sigma_1} & -\frac{Lgmm' - L\bar{T}\eta m}{\sigma_1} & -\frac{km'}{\sigma_1} & \frac{'}{\sigma_1} & \frac{Lm}{\sigma_1} & -\frac{m'}{\sigma_1} \\ 0 & 0 & 0 & 0 & -1/\tau & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$$J_u = B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/\tau & 0 & 0 \end{bmatrix}^T \tag{18}$$

Where $\sigma_1 = J'm' - L^2m^2$

This is the complete linear state-space symbolic representation of the drone. It should be noted that this linearization was done at $\theta^e = 0$, however that was not necessary due to the addition of the disturbances, that allow the system to be linearised for any angle $|\theta < 35°|$. Note, that the formula for M_a^e used in (15) is proven later in the section 5.2.2.

5.1.3 Numerical Representation of the Linear Model

It is clear from the model that most of the constants are known. These include:

- Mass of the drone: $m_d = 0.374\ kg$, Mass of the cart: $m_c = 0.160\ kg$, Combined Mass: $m' = 0.534\ kg$
- Distance between centre of mass and axis of rotation: $L = 1.00 \times 10^{-3}$ m
- Gravitational constant: $g = 9.81\ \text{m} \cdot \text{s}^{-2}$
- The normalized thrust: $\bar{T} = 0.400$, Thrust scaling constant: $\eta = 6.34\,\text{kg} \cdot \text{m} \cdot \text{s}^{-2}$
- Friction coefficient: $c = 0.624\ \text{kg} \cdot \text{s}^{-1}$
- Actual angular momentum coefficient: $\mu = 358.5\,\text{s}^{-1}$

- $\tau = 0.035\,\mathrm{s}$

However, the following constants are unknown:

- Resistive torque coefficient caused by the aerodynamic force: k
- New Moment of Inertia: J'

To find the values of k and J', the relationship between the second order system in actuator dynamics and their actual coefficient values must be utilised. More specifically, the Aerodynamic resistance coefficient is defined as: $k = 2J'\xi\omega_n$, and the weight torque: $mgL = J'\omega_n^2$. By solving together the two expressions:

$$J' = \frac{mgL}{\omega_n^2} = 3.1 \times 10^{-3}\ \mathrm{kg \cdot m^2} \quad \text{and} \quad \mathrm{k = 2J'\omega_n\xi} = 1.79 \times 10^{-5}\ \mathrm{kg \cdot m^2 \cdot rad \cdot s^{-1}}$$

Finally, the 7-dimensional matrix A is given numerically (using 5 significant figures) by:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.1695 & -4.7448 & 4.0510 \times 10^{-4} & -810.52 & -1.8742 & 2.2609 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1.4108 & -6.1101 & -0.57841 & 1.1573 \times 10^{6} & 2.2609 & -3.2281 \times 10^{3} \\ 0 & 0 & 0 & 0 & -28.571 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5.2 Observer-Controller Integration and Simulation

5.2.1 Design

A reduced order obsever must be designed to allow for the hidden states: M_a, d_v, d_q to be observed. The poles used to find L were placed at [-10, -15, -20]. With those values, matrix L was generated and upon which matrices F, G H were constucted. Their values are given below:

$$L = \begin{bmatrix} -8.3499 \times 10^{-22} & 2.4591 \times 10^{-5} & 0 & -2.0385 \times 10^{-5} \\ 1.087 \times 10^{-15} & -32.04 & 0 & 0.0224 \\ -1.3334 \times 10^{-18} & 0.0393 & 0 & -0.0326 \end{bmatrix}$$

$$F = \begin{bmatrix} -5 & -9.5105 \times 10^{-17} & -0.0658 \\ 3.638 \times 10^{-12} & -60 & 0 \\ 3.7643 \times 10^{4} & -1.579 \times 10^{-13} & -105 \end{bmatrix}$$

$$H = \begin{bmatrix} 9.1848 \times 10^{-20} & -0.0027 & -2.4123 \times 10{-4} & 0.0022 \\ -6.5238 \times 10^{-14} & 1.885 \times 103 & 152.16 & -1.3464 \\ 1.0858 \times 10^{-16} & -3.1977 & -0.3852 & 2.6321 \end{bmatrix} \quad G = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

5.2.2 Simulation

Just like in 4D, a way of knowing what input value of M_c^d is needed for a given desired state x^e must be found so that the accuracy observer can be tested. This is done by finding the equilibrium of the system $u^e = M_c^{d\,e}$, then finding an expression for the input in terms of the state variable to be controlled.

Thus $\dot{x} = Ax^e + Bu^e = 0$:p

$$\dot{p} = 0 \quad \Rightarrow \quad v^e = 0$$

$$\dot{\theta} = 0 \quad \Rightarrow \quad q^e = 0$$
$$\dot{d_v} = 0 \quad \Rightarrow \quad d_v^e = d_v$$
$$\dot{d_q} = 0 \quad \Rightarrow \quad d_q^e = d_q$$
$$\dot{v} = 0 \quad \Rightarrow \quad 0 - J' c\, v^e - (L^2 g m^2 - J' \bar{T}\eta)\, \theta^e - Lkm\, q^e + Lm\mu\, M_a^e - J'\, d_v^e - Lm\, d_q^e = 0 \tag{19}$$

$$\dot{q} = 0 \quad \Rightarrow \quad 0 - Lcm\, v^e + (L\bar{T}\eta m - Lgmm')\, \theta^e - km'\, q^e + \mu m'\, M_a^e - Lm\, d_v^e - m'\, d_q^e = 0 \tag{20}$$

By solving together (19) and (20) :

$$M_a^e = \frac{\bar{T}\eta\, d_q^e + gmL\, d_v^e}{\bar{T}\eta\mu} \qquad , \qquad \theta^e = \frac{d_v^e}{\bar{T}\eta} \tag{21}$$

Finally:

$$\dot{M_a} = 0 \quad \Rightarrow \quad -\frac{1}{\tau}(M_a^e + M_c^{d\,e}) = 0 \quad \Rightarrow \quad M_c^{d\,e} = M_a^e \quad \Rightarrow \quad M_c^{d\,e} = \frac{\bar{T}\eta\, d_q^e + gmL\, d_v^e}{\bar{T}\eta\mu}$$

$$\text{where:} \quad d_v^e = \bar{T}\eta\, \theta^e, \quad \text{so by rewriting the } M_c^{d\,e} \text{ expression:} \quad M_c^{d\,e} = \frac{d_q^e + gmL\, \theta^e}{\mu} \tag{22}$$

This way, all the states x_{est} can now be observed overtime, as the observer tracks the actual states x to reach the desired equilibrium angle θ^e.

5.3 Design of Controller

Similarly to 4D, below is the proof of the new system in terms of the error coordinates and the input in terms of the current state and the desired one:

$$\begin{aligned}
\dot{\theta^\varepsilon} &= \dot{\theta} - \dot{\theta^e} \qquad \text{where:} \qquad \dot{\theta^e} = q^e = 0 \\
&= q = q^\varepsilon + q^e \\
&= q^\varepsilon
\end{aligned}$$

$$\begin{aligned}
\dot{p^\varepsilon} &= \dot{p} - \dot{p^e} \qquad \text{where:} \qquad \dot{p^e} = v^e = 0 \\
&= v = v^\varepsilon + v^e \\
&= v^\varepsilon
\end{aligned}$$

$$\begin{aligned}
\dot{v^\varepsilon} &= \dot{v} - \dot{v^e} \qquad \text{where:} \qquad \dot{v^e} = 0 \\
&= \dot{v} = a_v v + a_\theta \theta + a_q q + a_{M_a} M_a + a_{d_v} d_v + a_{d_q} d_q = A_v \bar{x} \\
&= a_v(v^\varepsilon + v^e) + a_\theta(\theta^\varepsilon + \theta^e) + a_q(q^\varepsilon + q^e) + a_{M_a}(M_a^\varepsilon + M_a^e) + a_{d_v}(d_v^\varepsilon + d_v^e) + a_{d_q}(d_q^\varepsilon + d_q^e) \\
&= A_v \bar{x}^\varepsilon + a_\theta \frac{d_v^e}{\bar{T}\eta} + a_{M_a} \frac{\bar{T}\eta d_q^e + gmLd_v^d}{\bar{T}\eta\mu} + a_{d_v} d_v^e + a_{d_q} d_q^e \\
&= A_v \bar{x}^\varepsilon + \left(-\frac{L^2 g m^2 - J'\bar{T}\eta}{\bar{T}\eta} + \frac{L^2 m^2 \mu g}{\bar{T}\eta\mu} - J' \right) d_v^e + \left(\frac{Lm\mu\bar{T}\eta}{\bar{T}\eta\mu} - Lm \right) d_q^e \\
&= A_v \bar{x}^\varepsilon
\end{aligned}$$

$$
\begin{aligned}
\dot{q^\varepsilon} &= \dot{q} - \dot{q^e} \qquad \text{where:} \qquad \dot{q^e} = 0 \\
&= \dot{q} = b_v v + b_\theta \theta + b_q q + b_{M_a} M_a + b_{d_v} d_v + b_{d_q} d_q = A_q \bar{x} \\
&= b_v(v^\varepsilon + v^e) + b_\theta(\theta^\varepsilon + \theta^e) + b_q(q^\varepsilon + q^e) + b_{M_a}(M_a^\varepsilon + M_a^e) + b_{d_v}(d_v^\varepsilon + d_v^e) + b_{d_q}(d_q^\varepsilon + d_q^e) \\
&= A_q \bar{x}^\varepsilon + b_\theta \frac{d_v^e}{\bar{T}\eta} + b_{M_a} \frac{\bar{T}\eta d_q^e + gmL d_v^d}{\bar{T}\eta\mu} + b_{d_v} d_v^e + b_{d_q} d_q^e \\
&= A_q \bar{x}^\varepsilon + \left(\frac{\mu m' gmL}{\mu} + \frac{L\bar{T}\eta m}{\bar{T}\eta} - \frac{Lgmm'}{\bar{T}\eta} - Lm \right) d_v^e + \left(\frac{m'\mu}{\mu} - m' \right) d_q^e \\
&= A_q \bar{x}^\varepsilon
\end{aligned}
$$

$$
\begin{aligned}
\dot{M_a^\varepsilon} &= \dot{M}_a - \dot{M_a^e} \qquad \text{where:} \qquad \dot{M_a^e} = 0 \\
&= \dot{M}_a = -\frac{1}{\tau}\left(M_a - M_c^d\right) \\
&= \dot{M}_a = -\frac{1}{\tau}(M_a^\varepsilon + M_a^e - M_c^{d\,\varepsilon} - M_c^{d\,e}) \qquad \text{where:} \qquad M_a^e = M_c^{d\,e} \\
&= -\frac{1}{\tau}(M_{a\varepsilon} - M_{c\varepsilon}^d)
\end{aligned}
$$

Thus proving that for the 7D linear system, the coordinate transformation does not alter the dynamics of the system, ie. $A^\varepsilon = A \quad \text{and} \quad B^\varepsilon = B$.

Below is the proof for the formula of the closed loop system input u^ε in terms of the current and desired equilibrium states x and x^e:

$$
u^\varepsilon = -Kx^\varepsilon = -Kx + Kx^e = -K \begin{bmatrix} p \\ v \\ \theta \\ q \\ M_a \\ d_v \\ d_q \end{bmatrix} + K \begin{bmatrix} p^e \\ v^e \\ \theta^e \\ q^e \\ M_a^e \\ d_v^e \\ d_q^e \end{bmatrix} = -K \begin{bmatrix} p \\ v \\ \theta \\ q \\ M_a \\ d_v \\ d_q \end{bmatrix} + K \begin{bmatrix} p^e \\ 0 \\ d_v^e/\bar{T}\eta \\ 0 \\ (\bar{T}\eta\, d_q^e + gmL\, d_v^e)/\bar{T}\eta\mu \\ d_v^e \\ d_q^e \end{bmatrix} \tag{23}
$$

The drone should thus converge to an equilibrium for a given desired value of p.

The new error-coordinate system can now thus be described in the following equation:

$$
\dot{x^\varepsilon} = A^\varepsilon x^\varepsilon + B^\varepsilon x^\varepsilon = Ax^\varepsilon + Bu^\varepsilon \qquad \text{where:} \qquad u^\varepsilon = -Kx^\varepsilon \tag{24}
$$

$$
= Ax^\varepsilon - BKx^\varepsilon = (A - BK)x^\varepsilon \tag{25}
$$

$$
= (A - BK) \begin{bmatrix} p^\varepsilon \\ v^\varepsilon \\ \theta^\varepsilon \\ q^\varepsilon \\ M_a^\varepsilon \\ d_v^\varepsilon \\ d_q^\varepsilon \end{bmatrix} \quad \Rightarrow \quad \dot{x^\varepsilon} = (A - BK) \begin{bmatrix} p - p^e \\ v \\ \theta - d_v^e/\bar{T}\eta \\ q \\ M_a - (\bar{T}\eta\, d_q^e + gmL\, d_v^e)/\bar{T}\eta\mu \\ 0 \\ 0 \end{bmatrix} \tag{26}
$$

Similarly to 4D, M_a, d_v and d_q are generated by the observer. The reason why it is possible to use their estimated values for feedback is due to the Separation Principle. Moreover, the two disturbance states are uncontrollable thus: $d_v^e = d_v \Rightarrow d_v^\varepsilon = 0$, $d_q^e = d_q \Rightarrow d_q^\varepsilon = 0$. The controller gain matrix can thus be only implemented for the 5 controllable states. To calculate

the control gain matrix, the MATLAB "place" function was used. The 5 poles where chosen at: $[-4-4i-4+4i-7-10-15]$.
The control gain matrix is thus given by:

$$K = \begin{bmatrix} 0.0061 & 0.0025 & 0.0033 & 0.0005 & 9.6806 \end{bmatrix}$$

5.4 Simulating and Testing the Complete Closed Loop System

5.4.1 Simulation

A negative feedback mechanism with a controller is utilised to control the states position, velocity, angle, angular velocity, and torque.

The simulation was utilised to tune the controllers poles such that the following conditions are met:

- Verifying the asymptotic stability of the derived control scheme
- Regulating the feedback control gain matrix K.
- Tuning controller such that the settle time for position control is at a maximum of 2 seconds

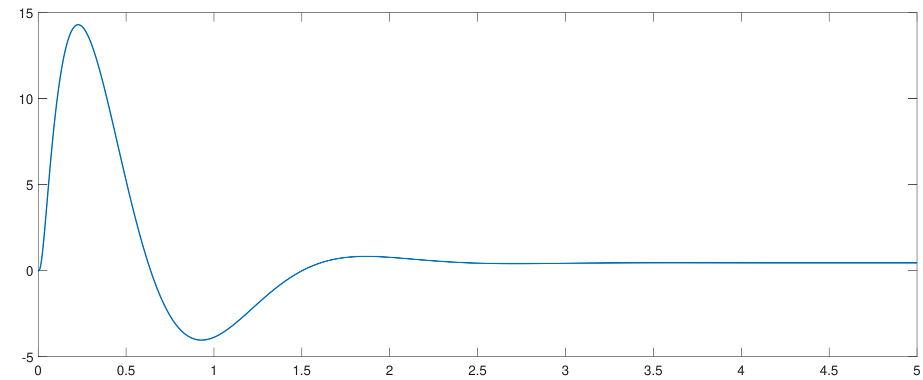

Figure (7): Time history of θ in simulation

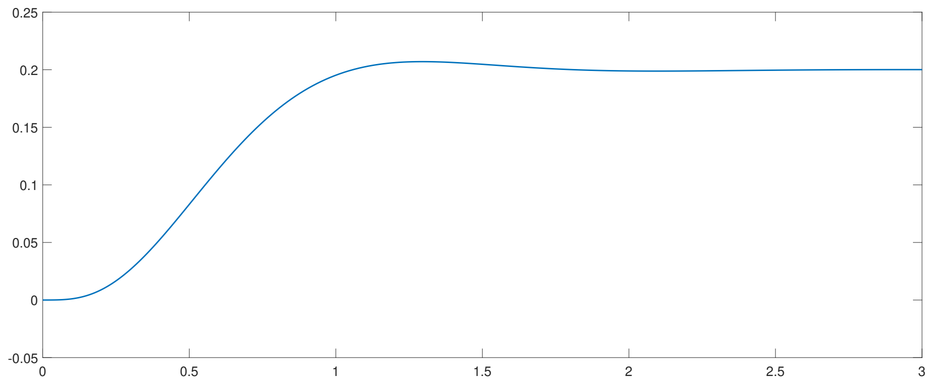

Figure (8): Time history of x in simulation

5.4.2 Application on Physical Drone

Applying the 7D control scheme to the ANT-X drone faced implementation issues which resulted in position control not being completed. The major issue was the lack of torque control measurement, despite the theoretical model being correct. Position control on the ANT-X platform was not achieved as a result.

6 Conclusion

Angular control was achieved with accuracy. Unfortunately, there were technical difficulties faced with the application observer of the 7D system, which didn't allow for M_a to be correctly observed, even though the numerical simulation showed that the design is reasonable.

7 References

[1] A.Astolfi, "Systems and Control Theory, An Introduction", Lecture Notes, Imperial College London, 2021

[2] Simos Evangelou, Thomas Parisini, "Control Systems Laboratory - 2DOF Drone", Lab Manual, Imperial College London, 2021