SAPIENZA
UNIVERSITÀ DI ROMA

Dynamic Feedback Linearizability of Ingenuity

A.A. 2025 - 2026

Luca Franzin
Andrea Gravili
Giuseppe D'Addario
Federico Tranzocchi
Sapienza University of Rome

Contents

Chapter 1

Introduction

The exploration of Mars has been significantly advanced with the success of the NASA Perseverance rover mission, whose primary objective is to search for signs of past microbial life and collect samples of Martian rock and soil. A key component of this mission is the *Ingenuity* helicopter (Figure 1.1), which demonstrates the potential of aerial vehicles for planetary exploration. Ingenuity's flights provide valuable data on the Martian atmosphere and, in general, environmental conditions that can be leveraged for future missions.

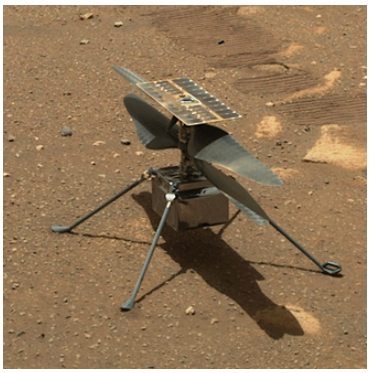

Figure 1.1: Ingenuity at Wright Brothers Field on 6 April 2021, its third day of deployment on Mars. Image source: Wikipedia.

1.1 The Ingenuity helicopter

Unlike the more common quadrotors, Ingenuity is a *co-axial* helicopter with two counter-rotating rotors stacked on a single mast. This design choice allows for a more compact structure that does not need a tail rotor to counteract torque. Instead of controlling motion through varying the speed of the two rotors, Ingenuity uses a mechanism called *swashplate* [1] (see Figure 1.2) to change the pitch of the rotor blades cyclically and collectively.

- *Cyclic pitch* changes the angle of the blades as they rotate, which tilts the thrust vector allowing to generate the forces and moments needed for horizontal motion (roll and pitch).
- *Collective pitch* adjusts the angle of all blades of a rotor simultaneously, controlling the total magnitude of the thrust for vertical motion.

This actuation method yields flight dynamics that are fundamentally different from those of multi-rotor drones.

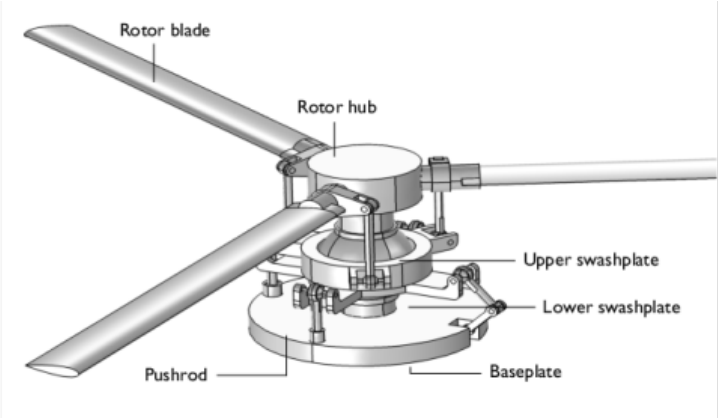

Figure 1.2: A helicopter swashplate mechanism. It translates non-rotating control inputs from the servos into the rotating frame of the rotor blades, controlling both collective and cyclic pitch. Image source: COMSOL Application Gallery.

1.1.1 Control

Controlling Ingenuity is a challenging task due to two main reasons. First, the Martian atmosphere is much thinner than Earth's, with a density of about 1% that of sea level on Earth. This results in reduced aerodynamic forces and moments, making it difficult to generate sufficient lift and, in general, it alters the helicopter's flight dynamics. This requires the two rotors to spin at very high speeds (over 2500 RPM) to generate enough lift for takeoff and maneuvering, leading to complex aerodynamic effects [2].

Second, the helicopter's dynamics are nonlinear and coupled. In fact, the forces generated by the rotors depend on the vehicle's orintation, and the translational and rotational motions are linked. While linear controllers have been succesfully used for Ingenuity's flight control [1], their performance can be limited, especially during aggressive maneuvers or in the presence of disturbances such as wind gusts. In general, standard linear control techniques tend to struggle in handling these nonlinearities.

Dynamic feedback linearization

We adopt an approach from nonlinear control theory known as *dynamic feedback linearization* (DFL) [3] that aims to cancel out the coupled nonlinearities in the system dynamics through feedback, resulting in a decoupled linearized system that can be controlled using linear control methods. The main idea is to find a suitable change of coordinates and a nonlinear control law that cancels out the unwanted nonlinear terms in the dynamics.

The "dynamic" aspect of DFL refers to an extension of the method that is necessary when the control inputs do not immediately affect the outputs, requiring the controller to have its own internal dynamics. As we will show, this is precisely the case for controlling the position of a co-axial helicopter like Ingenuity using the swashplate mechanism.

1.2 Objective

The main goal of this project is to formally analyze the dynamic feedback linearizability of the Ingenuity helicopter, design a state feedback controller for *TODO: add task*, and validate its performance through simulations. Specifically, we aim to:

1. Develop a dynamic model of Ingenuity based on Newton-Euler equations, capturing the key dynamics and couplings introduced by the co-axial rotor configuration;
2. analyze such dynamics to determine the conditions under which the system can be linearized using dynamic feedback;
3. design the nonlinear control law that achieves linearization and decoupling;
4. *TODO: implement a controller to carry out some task*;
5. evaluate the performance of our controller in simulation under various scenarios and for different maneuvers.

1.3 Structure

The report is organized as follows. In Chapter 2 we present the detailed dynamic model of the Ingenuity helicopter deriving the equations of motion. In Chapter 3 we provide a mathematical formulation of the dynamic feedback linearization technique and derive the specific control law for our system. In Chapter 4 we describe the simulation environment, the implementation details of the controller, and the test scenarios. In Chapter 5 we present the results of our experiments and evaluate the controller's performance. Finally, in Chapter 6 we summarize our findings and discuss potential future work.

Chapter 2

Ingenuity dynamic model

In this chapter, we develop the mathematical model of the Ingenuity helicopter using Newton-Euler equations. We start by defining the reference frames and the notation used throughout the report, the state and input vectors of the system, and derive the translational and rotational equations of motion governing the vehicle's flight. The dynamic model derived here is based on the formulation presented in [4] and will serve as the basis for the control system design analyzed in Chapter 3.

2.1 Coordinate frames

To describe the motion of Ingenuity, we define two main right-handed coordinate frames, as illustrated in Figure **??**:

- The *inertial frame* $\{I\}$, which is a fixed non-accelerating frame used as a global reference. Its origin is located at the takeoff point on the Martian surface, and its axes are denoted by (x_i, y_i, z_i). We adopt a z-up convention, with the z_i axis pointing vertically upwards in the opposite direction of gravity.
- The *body frame* $\{B\}$, which is a frame attached to the vehicle with its origin at the center of mass. Its axes (x_b, y_b, z_b) are aligned with the principal axes of Ingenuity, with x_b pointing forward, y_b pointing to the right, and z_b pointing up along the rotor mast.

The orientation of the body frame with respect to the inertial frame is described by a rotation matrix $\boldsymbol{R} \in SO(3)$ which transforms vectors from the body frame to the inertial frame. Such a matrix is paramterized using the ZYX Euler angles convention, defined by the yaw (ψ), pitch (θ), and roll (ϕ) angles:

$$\boldsymbol{R} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{2.1}$$

where $c_\alpha = \cos(\alpha)$ and $s_\alpha = \sin(\alpha)$, for any angle α.

2.2 State and input

The *state* of the system captures its complete dynamic condition at any given time. The *input* vector represents the control commands used to control the system.

State vector The state of the Ingenuity helicopter is described by a 12-dimensional vector $\boldsymbol{\xi} \in \mathbb{R}^{12}$, which includes its position, linear velocity, orientation, and angular velocity:

$$\boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{V}^b \\ \boldsymbol{\Theta} \\ \boldsymbol{\omega} \end{pmatrix} \in \mathbb{R}^{12} \tag{2.2}$$

where:

- $\boldsymbol{P} = (x, y, z)^T \in \mathbb{R}^3$ is the position of the vehicle's center of mass in the inertial frame $\{I\}$;
- $\boldsymbol{V}^b = (V_x^b, V_y^b, V_z^b)^T = (u, v, w)^T \in \mathbb{R}^3$ is the linear velocity of the center of mass expressed in the body frame $\{B\}$;
- $\boldsymbol{\Theta} = (\phi, \theta, \psi)^T \in \mathbb{R}^3$ are the Euler angles representing the orientation of the body frame $\{B\}$ with respect to the inertial frame $\{I\}$;
- $\boldsymbol{\omega} = (p, q, r)^T \in \mathbb{R}^3$ is the angular velocity of the vehicle expressed in the body frame $\{B\}$.

Input vector Ingenuity is controlled via the net thrust produced by its two rotors and the net torques applied to the body. We define a 4-dimensional input vector $\boldsymbol{u} \in \mathbb{R}^4$ that abstracts the complex swashplate mechanism and rotor dynamics into the following components:

$$\boldsymbol{u} = \begin{pmatrix} F_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \in \mathbb{R}^4 \tag{2.3}$$

where:

- F_T is the magnitude of the total thrust force acting along the body z_b axis;
- τ_ϕ, τ_θ, and τ_ψ are the net control torques applied around the body axes x_b, y_b, z_b, respectively.

2.3 Equations of motion

The dynamics are separated into *kinematic* and *dynamic* equations: kinematics describe how the position and orientation evolve based on the velocities, while dynamics describe how the velocities change based on the applied forces and torques.

2.3.1 Kinematics

The kinematic equations describe the geometry of motion without considering the forces that cause it.

The rate of change of the inertial position is the linear velocity in the inertial frame, which is obtained by transforming the body-frame velocity using the rotation matrix in Eq. (2.1):

$$\dot{\boldsymbol{P}} = \boldsymbol{R}\boldsymbol{V}^b \tag{2.4}$$

The rate of change of the Euler angles is related to the body-frame angular velocity through the following transformation:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \dot{\boldsymbol{\Theta}} = \boldsymbol{W}(\boldsymbol{\Theta})\boldsymbol{\omega} = \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.5}$$

where $t_\alpha = \tan(\alpha)$, for any angle α.

2.3.2 Translational dynamics

The translational dynamics are derived from *Newton's second law*, which we express in the body frame $\{B\}$:

$$m\dot{\boldsymbol{V}}^b = \boldsymbol{F}^b_{\text{tot}} - m\boldsymbol{\omega} \times \boldsymbol{V}^b \tag{2.6}$$

where m is the mass of the vehicle, and the term $m(\boldsymbol{\omega} \times \boldsymbol{V}^b)$ is the Coriolis force that arises from differentiating the velocity in a rotating frame. The total force $\boldsymbol{F}^b_{\text{tot}}$ in the body frame is th sum of the thrust force, gravity, and aerodynamic drag:

$$\boldsymbol{F}^b_{\text{tot}} = \boldsymbol{F}^b_{\text{thrust}} + \boldsymbol{F}^b_{\text{gravity}} + \boldsymbol{F}^b_{\text{drag}} \tag{2.7}$$

In detail:

- The *thrust force* is the force generated by the two rotors acting along the positive z_b axis of the body frame $\{B\}$:

$$\boldsymbol{F}^b_{\text{thrust}} = \begin{pmatrix} 0 \\ 0 \\ F_T \end{pmatrix} \tag{2.8}$$

- The *gravitational force* is the weight of the vehicle acting downwards in the inertial frame $\{I\}$ rotated into the body frame $\{B\}$ using the transpose of the rotation matrix in Eq. (2.1):

$$\boldsymbol{F}^b_{\text{gravity}} = \boldsymbol{R}^T \boldsymbol{F}^i_{\text{gravity}} = \boldsymbol{R}^T \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{2.9}$$

- The *aerodynamic drag force* is modeled as a simple linear damping proportional to the body-frame velocity:

$$\boldsymbol{F}^b_{\text{drag}} = -\boldsymbol{A}_{\text{trans}} \boldsymbol{V}^b \tag{2.10}$$

  where $\boldsymbol{A}_{\text{trans}}$ is a diagonal matrix containing the translational drag coefficients along each body axis (see Chapter 4).

2.3.3 Rotational dynamics

The rotational dynamics are derived from *Euler's equation* for rigid body rotation, expressed in the body frame $\{B\}$:

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}^b_{\text{tot}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \tag{2.11}$$

where $\boldsymbol{I}$ is the inertia matrix, and the term $\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}$ represents the gyroscopic torques due to the rotating frame. The total torque $\boldsymbol{\tau}^b_{\text{tot}}$ in the body frame is the sum of the control torques and aerodynamic damping torques:

$$\boldsymbol{\tau}^b_{\text{tot}} = \boldsymbol{\tau}^b_{\text{control}} + \boldsymbol{\tau}^b_{\text{drag}} \tag{2.12}$$

In detail:

- The *control torques* are the torques generated by the swashplate mechanism around each body axis, and are given by the control inputs in Eq. (2.3):

$$\boldsymbol{\tau}^b_{\text{control}} = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \tag{2.13}$$

- The *aerodynamic damping torques* are modeled as linear damping opposing the angular velocity:

$$\boldsymbol{\tau}^b_{\text{drag}} = -\boldsymbol{A}_{\text{rot}}\boldsymbol{\omega} \tag{2.14}$$

  where $\boldsymbol{A}_{\text{rot}}$ is a diagonal matrix containing the rotational drag coefficients around each body axis (see Chapter 4).

2.4 Complete state-space model

Combining the kinematic and dynamic equations, we obtain the complete *nonlinear* state-space model of the Ingenuity helicopter in the form $\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$:

$$\dot{\boldsymbol{\xi}} = \begin{pmatrix} \dot{\boldsymbol{P}} \\ \dot{\boldsymbol{V}}^b \\ \dot{\boldsymbol{\Theta}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}\boldsymbol{V}^b \\ \frac{1}{m}\left(\boldsymbol{F}^b_{\text{thrust}} + \boldsymbol{F}^b_{\text{gravity}} + \boldsymbol{F}^b_{\text{drag}}\right) - \boldsymbol{\omega} \times \boldsymbol{V}^b \\ \boldsymbol{W}(\boldsymbol{\Theta})\boldsymbol{\omega} \\ \boldsymbol{I}^{-1}\left(\boldsymbol{\tau}^b_{\text{control}} + \boldsymbol{\tau}^b_{\text{drag}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \end{pmatrix} \tag{2.15}$$

We will be using this set of 12 coupled nonlinear differential equations as the basis for our analysis of dynamic feedback linearization in Chapter 3.

Chapter 3

Control design

This chapter details the design of a nonlinear controller for the Ingenuity helicopter model developed in Chapter 2. Due to the underactuatued and coupled nature of its dynamics, standard linear control techniques are not suitable for achieving trajectory tracking. We therefore employ the technique of *dynamic feedback linearization* (DFL) to cancel the system's inherent nonlinearities and achieve a linear input-output mapping.

The objective is to design a control law for the input vector $\boldsymbol{u}$ defined in Eq. (2.3) that forces the position $\boldsymbol{P}(t)$ and yaw angle $\psi(t)$ of Ingenuity to track a desired smooth trajectory $(\boldsymbol{P}_d(t), \psi_d(t))$. To achieve this, we will first analyze the system's input-output relationship to determine the *relative degree* of each output with respect to the inputs. Then, we will design a *dynamic compensator* with an internal state $\boldsymbol{\zeta}$ to augment the system and achieve full relative degree, so as to get a feedback linearizing control law. Finally, we will design an *outer-loop* linear tracking controller to stabilize the tracking error for the linearized system.

3.1 Dynamic feedback linearization

For a general nonlinear system of the form $\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{G}(\boldsymbol{\xi})\boldsymbol{u}$ with output $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{\xi})$, the goal of feedback linearization is to find a *coordinate transformation* and a control law that renders the input-output map linear. This is achieved by differentiating each output y_i until at least one input u_j appears. The number of differentiation required is called the *relative degree* r_i of the output y_i.

If the total relative degree $r = \sum r_i$ is less than the dimension of the state n, the system has residual internal dynamics (*zero dynamics*) that may be unstable. On the other hand, if $r = n$ the system can be fully linearized.

When the control inputs do not appear in a way that allows for direct cancellation of nonlinearities, that is when the *decoupling matrix* is singular, we introduce a *dynamic compensator* with its own internal state $\boldsymbol{\zeta}$ and dynamics $\dot{\boldsymbol{\zeta}}$. This involves augmenting the system with integrators on the input channels, which leads to an increased dimension of the state space to match the total relative degree.

3.1.1 Ingenuity DFL

We apply the DFL framework to the dynamics of Ingenuity. The outputs we want to control are the position $\boldsymbol{P} = (x, y, z)^T$ and yaw angle ψ, so we define the output vector as:

$$\boldsymbol{y} = \begin{pmatrix} x \\ y \\ z \\ \psi \end{pmatrix} \tag{3.1}$$

We can now differentiate each component of the output vector $\boldsymbol{y}$ until a component of the input vector $\boldsymbol{u}$ appears, and determine the corresponding relative degree.

Yaw channel

The yaw dynamics are governed by the rotational kinematics and the rotational dynamics. Differentiating the yaw angle ψ once gives:

$$\dot{y}_4 = \dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r$$

This expression depends only on the states ϕ, θ, q, r and no inputs appear, so we differentiate again:

$$\ddot{\psi} = \underbrace{(s_\phi \sec\theta)\dot{q} + (c_\phi \sec\theta)\dot{r}}_{\text{input-dependent}} + \underbrace{(\dot{\phi} c_\phi \sec\theta + s_\phi \dot{\theta} \sec\theta \tan\theta) q + (-\dot{\phi} s_\phi \sec\theta + c_\phi \dot{\theta} \sec\theta \tan\theta) r}_{\text{state-dependent} = \boldsymbol{f}_\psi(\boldsymbol{\xi})}$$

where $\boldsymbol{f}(\boldsymbol{\xi})$ is the *drift* term, and the input-dependent part can be expressed as:

$$\begin{pmatrix} 0 & s_\phi \sec\theta & c_\phi \sec\theta \end{pmatrix} \dot{\boldsymbol{\omega}}$$

where $\dot{\boldsymbol{\omega}}$ is given by the rotational dynamics in Eq. (2.11). Substituting this expression, we obtain:

$$\ddot{\psi} = \boldsymbol{f}_\psi(\boldsymbol{\xi}) + \begin{pmatrix} 0 & s_\phi \sec\theta & c_\phi \sec\theta \end{pmatrix} \boldsymbol{I}^{-1} \left(\boldsymbol{\tau}^b_{\text{control}} + \boldsymbol{\tau}^b_{\text{drag}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \right) \tag{3.2}$$

We see that the control torques $\boldsymbol{\tau}^b_{\text{control}}$ appear linearly (in particular, the two inputs $u_3 = \tau_\theta$ and $u_4 = \tau_\psi$), so the relative degree of the yaw channel is $r_\psi = 2$.

Position channel

The position dynamics reveal that the system is *underactuated*. From the translational kinematics in Eq. 2.4, we have:

$$\dot{\boldsymbol{P}} = \boldsymbol{R}(\boldsymbol{\Theta}) \boldsymbol{V}^b$$

which is independent of the inputs. We therefore differentiate twice:

$$\begin{aligned} \ddot{\boldsymbol{P}} &= \dot{\boldsymbol{R}} \boldsymbol{V}^b + \boldsymbol{R} \dot{\boldsymbol{V}}^b \\ &= \cancel{\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{V}^b} + \boldsymbol{R} \left[\frac{1}{m} \left(\boldsymbol{F}^b_{\text{thrust}} + \boldsymbol{F}^b_{\text{gravity}} + \boldsymbol{F}^b_{\text{drag}} \right) - \cancel{\boldsymbol{\omega} \times \boldsymbol{V}^b} \right] \\ &= \frac{1}{m} \boldsymbol{R} \begin{pmatrix} 0 \\ 0 \\ \textcircled{F_T} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} - \frac{1}{m} \boldsymbol{R} \boldsymbol{A}_{\text{trans}} \boldsymbol{V}^b \end{aligned}$$

The input F_T appears, but the control inputs τ_ϕ, τ_θ do not. We must continue differentiating:

$$P^{(3)} = \frac{d}{dt}\left[\frac{F_T}{m}\boldsymbol{R}\boldsymbol{e}_3 - g\boldsymbol{e}_3 + \frac{1}{m}\boldsymbol{R}\boldsymbol{F}^b_{\text{drag}}\right]$$

The time derivatve of the second term is zero. The third term depends on $\boldsymbol{V}^b$ and $\boldsymbol{\omega}$, but not on its derivative $\dot{\boldsymbol{\omega}}$, and thus will not introduce any new input. As for the first term, we have:

$$\frac{d}{dt}\left[\frac{F_T}{m}\boldsymbol{R}\boldsymbol{e}_3\right] = \frac{\dot{F}_T}{m}\boldsymbol{R}\boldsymbol{e}_3 + \frac{F_T}{m}\dot{\boldsymbol{R}}\boldsymbol{e}_3$$

The term $\dot{F}_T$ appears, which is not a control input. We therefore differentiate one last time:

$$P^{(4)} = \frac{d}{dt}\left[\frac{\dot{F}_T}{m}\boldsymbol{R}\boldsymbol{e}_3 + \frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{e}_3 + \dots\right]$$

The first term gives:

$$\frac{d}{dt}\left[\frac{\dot{F}_T}{m}\boldsymbol{R}\boldsymbol{e}_3\right] = \frac{\ddot{F}_T}{m}\boldsymbol{R}\boldsymbol{e}_3 + \frac{\dot{F}_T}{m}\dot{\boldsymbol{R}}\boldsymbol{e}_3$$

Again, we have a non-input term $\ddot{F}_T$. The second term gives:

$$\frac{d}{dt}\left[\frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{e}_3\right] = \dots + \frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}(\dot{\boldsymbol{\omega}})\boldsymbol{e}_3$$

where $\dot{\boldsymbol{\omega}}$ appears, which is a direct linear function of the input torques $\boldsymbol{\tau}^b_{\text{control}}$. Therefore, the relative degree of each position channel is $r_x = r_y = r_z = 4$.

Dynamic compensator

The appearance of $\ddot{F}_T$ in $\boldsymbol{P}^{(4)}$ indicates that the thrust input F_T does not directly influence the outputs in a way that allows for cancellation of nonlinearities. We need to introduce a dynamic compensator for the thrust channel, augmenting the state of the system with two additional controller states:

$$\boldsymbol{\zeta} = \begin{pmatrix}\zeta_1\\ \zeta_2\end{pmatrix} = \begin{pmatrix}F_T\\ \dot{F}_T\end{pmatrix} \tag{3.3}$$

The dynamics of the compensator are given by a simple integrator chain driven by a new *virtual input* v_T:

$$\dot{\boldsymbol{\zeta}} = \begin{pmatrix}\dot{\zeta}_1\\ \dot{\zeta}_2\end{pmatrix} = \begin{pmatrix}\zeta_2\\ v_T\end{pmatrix} = \begin{pmatrix}\dot{F}_T\\ \ddot{F}_T\end{pmatrix} \tag{3.4}$$

The thrust F_T applied to Ingenuity is now treated as a state variable ζ_1 that evolves according to the compensator dynamics. The extended state of the system is $(\boldsymbol{\xi}, \boldsymbol{\zeta}) \in \mathbb{R}^{14}$.

Linearizing control law

The total relative degree of the augmented system is now:

$$r = r_x + r_y + r_z + r_\psi = 4 + 4 + 4 + 2 = 14$$

which equals the dimension of the extended state space ($n_{\text{ext}} = 12 + 2 = 14$). Therefore, the system is fully state linearizable with no zero dynamics.

We define the extended input vector as:

$$\tilde{\boldsymbol{u}} = \begin{pmatrix} \ddot{F}_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \tag{3.5}$$

The dynamics of the output can now be expressed in the standard affine form:

$$\begin{pmatrix} P^{(4)} \\ \ddot{\psi} \end{pmatrix} = \boldsymbol{l}(\boldsymbol{\xi}, \boldsymbol{\zeta}) + \boldsymbol{J}(\boldsymbol{\xi}, \boldsymbol{\zeta})\tilde{\boldsymbol{u}} \tag{3.6}$$

where $\boldsymbol{l}(\boldsymbol{\xi}, \boldsymbol{\zeta})$ is the drift vector containing all state-dependent terms, and $\boldsymbol{J}(\boldsymbol{\xi}, \boldsymbol{\zeta})$ is the 4×4 decoupling matrix.

Decoupling matrix $\boldsymbol{J}$ The decoupling matrix $\boldsymbol{J}$ contains the coefficients of the extended inputs $\tilde{\boldsymbol{u}}$ in the output derivative equations. By carrying out the differentiations in the previous section, we can identify its components:

$$\boldsymbol{J}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = \left[\begin{array}{c|c} \frac{1}{m}\boldsymbol{R}\boldsymbol{e}_3 & -\frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}(\boldsymbol{e}_3)\boldsymbol{I}^{-1} \\ \hline \boldsymbol{0}_{1\times 1} & \begin{pmatrix} 0 & s_\phi \sec\theta & c_\phi \sec\theta \end{pmatrix} \boldsymbol{I}^{-1} \end{array}\right] \tag{3.7}$$

The block $\boldsymbol{J}_{11}$ is a 3×1 vector representing the effect of $\ddot{F}_T$ on $\boldsymbol{P}^{(4)}$. The block $\boldsymbol{J}_{12}$ is a 3×3 matrix representing the effect of the control torques on $\boldsymbol{P}^{(4)}$. The block $\boldsymbol{J}_{21}$ is zero, since the yaw acceleration is independent of the thrust dynamics $\ddot{F}_T$. The block $\boldsymbol{J}_{22}$ is a 1×3 vector representing the effect of the control torques on $\ddot{\psi}$.

This decoupling matrix is dependent on F_T and becomes singular if $F_T = 0$. In this condition, the torques have no effect on the position, and the system is uncontrollable. In practice, we implement a check to disable the controller if the thrust magnitude approaches zero.

Drift vector $\boldsymbol{l}$ The drift vector $\boldsymbol{l}$ contains all the state-dependent terms from the differentiation that do not multiply the extended inputs. It can be partitioned as:

$$\boldsymbol{l} = \begin{pmatrix} \boldsymbol{l}_{pos} \\ l_\psi \end{pmatrix}$$

The yaw drift l_ψ is derived from Eq. (3.2):

$$l_\psi(\boldsymbol{\xi}) = (\dot{\phi}c_\phi \sec\theta + s_\phi\dot{\theta}\sec\theta\tan\theta)q + (-\dot{\phi}s_\phi \sec\theta + c_\phi\dot{\theta}\sec\theta\tan\theta)r \tag{3.8}$$

The position drift $\boldsymbol{l}_{\text{pos}}$ contains all terms from the fourth derivative of position that are not part of the decoupling matrix. Its full expression is complex, so we highlight only its main components:

$$\boldsymbol{l}_{pos} = \frac{2\dot{F}_T}{m}\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{e}_3 + \frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}^2(\boldsymbol{\omega})\boldsymbol{e}_3 + \frac{F_T}{m}\boldsymbol{R}\boldsymbol{S}(\dot{\boldsymbol{\omega}}_{\text{drift}})\boldsymbol{e}_3 + \dots \tag{3.9}$$

where $\dot{\boldsymbol{\omega}}_{\text{drift}} = \boldsymbol{I}^{-1}(\boldsymbol{\tau}^b_{\text{drag}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega})$ is the part of the angular acceleration not caused by control inputs. The ellipsis represents higher-order drag terms which are often simplified in implementation.

Control law The linearizing control law is chosen to cancel these complex terms:

$$\tilde{\boldsymbol{u}} = \boldsymbol{J}^{-1}(\boldsymbol{\xi}, \boldsymbol{\zeta}) \left[\boldsymbol{v} - \boldsymbol{l}(\boldsymbol{\xi}, \boldsymbol{\zeta})\right] \tag{3.10}$$

where $\boldsymbol{v} = (v_1, v_2, v_3, v_4)^T$ is the new simplified input vector. Applying this control law transforms the input-output dynamics into a set of decoupled linear integrator chains:

$$\begin{cases} x^{(4)} = v_1 \\ y^{(4)} = v_2 \\ z^{(4)} = v_3 \\ \ddot{\psi} = v_4 \end{cases} \tag{3.11}$$

Outer-loop tracking controller

The final step is to design the input $\boldsymbol{v}$ to ensure the tracking errors converge to zero. Let the position error be $\boldsymbol{e}_P = \boldsymbol{P}_d - \boldsymbol{P}$ and the yaw error be $e_\psi = \psi_d - \psi$. For the fourth-order position dynamics, we design a controller with feedforward and state feedback on the error dynamics:

$$v_{1,2,3} = \boldsymbol{P}_d^{(4)} + \boldsymbol{K}_3 \boldsymbol{e}_P^{(3)} + \boldsymbol{K}_2 \ddot{\boldsymbol{e}}_P + \boldsymbol{K}_1 \dot{\boldsymbol{e}}_P + \boldsymbol{K}_0 \boldsymbol{e}_P \tag{3.12}$$

where $v_{1,2,3} = (v_1, v_2, v_3)^T$ and $\boldsymbol{K}_i = \text{diag}(k_{i,x}, k_{i,y}, k_{i,z})$ are diagonal gain matrices.

For the second-order yaw dynamics, we use a PD controller with feedforward:

$$v_4 = \ddot{\psi}_d + k_{d,\psi} \dot{e}_\psi + k_{p,\psi} e_\psi \tag{3.13}$$

The gains in Eqs. (3.12) and (3.13) are chosen to place the poles of the respective error characteristic equations in the left-half of the complex plane, ensuring exponential convergence of the tracking errors to zero.

3.2 Control architecture summary

The complete control architecture is shown in Figure 3.1. The outer loop computes the virtual input $\boldsymbol{v}$ based on the tracking error. The inner loop computes the state-dependent terms $\boldsymbol{l}$ and $\boldsymbol{J}$ and uses the DFL law (3.10) to calculate the extended inputs $\tilde{\boldsymbol{u}}$. The torque commands are sent to Ingenuity, while the thrust rate command v_T is integrated through the dynamic compensator to produce the thrust input F_T.

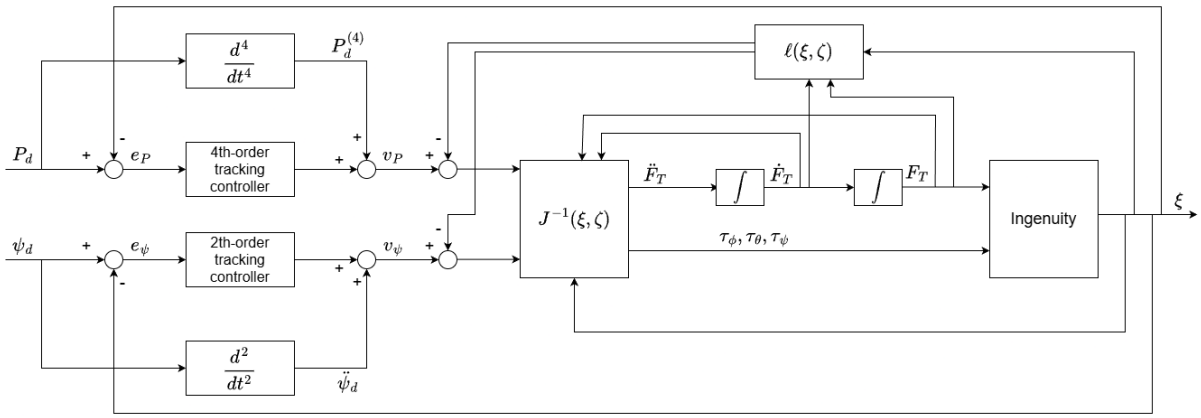

Figure 3.1: Block scheme of the complete DFL control system.

Chapter 4

Simulation setup

In this chapter, we describe the simulation environment used to validate the control law computed in Chapter 3. We define the physical parameters of the Ingenuity helicopter, the controller tuning, the reference trajectories used for testing, the external disturbances applied to check robustness, and the performance metrics used to evaluate the results.

4.1 Simulation environment

The simulation is implemented in MATLAB. Since the system dynamics derived in Chapter 2 are nonlinear and continuous in time, we use a standard numerical solver for ordinary differential equations (`ode15s`) to integrate the system dynamics.

All the simulations are run with a variable time step to ensure numerical stability, but the results are analyzed and plotted over a fixed time span of $T = 30$ seconds. The initial condition for all simulations corresponds to Ingenuity being on the ground (zero altitude) and the rotors spinning to generate a thrust equal to the vehicle's weight (hovering condition), to simulate a realistic takeoff scenario.

4.2 Model parameters

The physical parameters used in the simulation represent the Ingenuity helicopter operating in the Martian environment. The mass and gravity acceleration are taken from the mission data [1]. The inertia tensor is approximated as a diagonal matrix, and the aerodynamic and drag coefficients are estimated to provide a realistic damping effect. Table 4.1 summarizes these parameters.

Table 4.1: Physical parameters of the Ingenuity model.

Parameter	**Symbol**	**Value**
Mass	m	1.8 kg
Mars Gravity	g	3.69 m/s^2
Inertia (x-axis)	I_{xx}	0.02 kg·m^2
Inertia (y-axis)	I_{yy}	0.02 kg·m^2
Inertia (z-axis)	I_{zz}	0.03 kg·m^2
Translational drag	$\boldsymbol{A}_{\text{trans}}$	$\text{diag}(0.05, 0.05, 0.1)$ N·s/m
Rotational drag	$\boldsymbol{A}_{\text{rot}}$	$\text{diag}(0.01, 0.01, 0.05)$ N·m·s

4.3 Controller implementation

4.3.1 Gain tuning

The dynamic feedback linearization controller transforms the nonlinear system into decoupled chains of linear integrators. We stabilize these chains by assigning the poles of the error dynamics.

For the position subsystem, which has a relative degree of $r = 4$, the error dynamics are described by the linear differential equation:

$$e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e = 0$$

To ensure a fast response with no overshoot, we placed all four poles at $\lambda = -2$ on the real axis. By expanding the characteristic polynomial $(s + 2)^4$, we obtained the following gains:

$$\boldsymbol{k}_p = [k_0, k_1, k_2, k_3] = [16, 32, 24, 8]$$

These scalars define the diagonal gain matrices used in Eq. (3.12) as $\boldsymbol{K}_i = k_i \boldsymbol{I}_3$, ensuring identical behavior for the x, y, z axes.

For the yaw subsystem ($r = 2$), we selected proportional and derivative gains to ensure stable tracking:

$$\boldsymbol{k}_\psi = [k_{p,\psi}, k_{d,\psi}] = [4, 4]$$

4.3.2 Actuator saturation

Real actuators cannot produce infinite forces and torques. To make the simulation realistic, we implemented saturation limits on the control inputs based on the known capabilities of Ingenuity's rotors [1].

Thrust limits The thrust produced by the rotors is limited by the rotational speed and the atmospheric density. The thrust-to-weight ratio (TWR) of Ingenuity is between 1.3 and 1.6; we selected a nominal maximum limit of 145%:

$$F_{\text{max}} = 1.45 \cdot mg \approx 9.63 \text{ N}$$

We also enforced a minimum thrust limit of $F_{\text{min}} = 0.3mg$ to prevent the total thrust from reaching zero, which would cause a singularity in the decoupling matrix $\boldsymbol{J}$ (see Chapter 3).

Torque limits The control torques $\tau_\phi, \tau_\theta, \tau_\psi$ are saturated at ± 0.05 Nm to represent the physical limits of the cyclic and collective pitch mechanism.

4.4 Reference trajectories

We define a set of reference trajectories to test the performance of the controller under different conditions. They are chosen so as to evaluate the controller's ability to track both simple and smooth paths, as well as to more aggressive maneuvers.

Box trajectory This trajectory simulates a patrol mission: the helicopter takes off to an altitude of 5 meters and then follows a square path with a side length of 10 meters. The motion along each segment is generated using quintic polynomials to ensure smooth transitions in position, velocity, acceleration, jerk, and snap. The yaw angle is kept constant at zero throughout the trajectory

Helix trajectory This trajectory requires the helicopter to ascend while performing a circular motion. It is defined analyitically as:

$$\begin{cases} x_d(t) = R\cos(\omega_{\text{ref}} t) \\ y_d(t) = R\sin(\omega_{\text{ref}} t) \\ z_d(t) = \min(v_z t, z_{\max}) \end{cases}$$

where the radius is $R = 2$ m, the angular rate is $\omega_{\text{ref}} = 0.5$ rad/s, and the vertical velocity is $v_z = 0.2$ m/s.

4.5 Disturbances

To verify the robustness of the controller, we simulate a wind gust disturbance acting on the helicopter during flight. Since the DFL technique relies on exact cancellation of nonlinearities, model uncertainties or external forces can degrade performance. A sudden random constant force vector is added to the translational dynamics (Eq. (2.6)) for a specific duration:

$$\boldsymbol{F}_{\text{gust}} \in [-3, 3]\ \text{N} \quad \text{applied for } t \in [10, 20]\ \text{s} \tag{4.1}$$

This force acts as an unmodeled disturbance that the DFL law cannot cancel out; the outer linear feedback loop must compensate for it to maintain the trajectory.

4.6 Performance metrics

To quantitatively evaluate the controller, we compute the following metrics based on the discrete simulation data sampled at time steps t_k, for $k = 1, \ldots, N$.

Tracking error (RMSE) The root mean square error (RMSE) measures the average deviation of the helicopter from the desired path. In the discrete domain, it is defined as:

$$E_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} ||\boldsymbol{P}_d(t_k) - \boldsymbol{P}(t_k)||^2} \tag{4.2}$$

A lower RMSE indicates better tracking performance.

Average thrust usage To evaluate the energy demand and the safety margin of the maneuver, we compute the average percentage of the maximum available thrust used during the flight:

$$J_{\%} = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{F_T(t_k)}{F_{\max}}\right) \times 100 \tag{4.3}$$

A value close to 100% indicates that the system is operating near its physical limits, leaving little room for error or disturbance rejection, while a lower value implies a safer and more efficient flight.

Settling time *TODO*

Chapter 5

Results

5.1 TODO: task

5.2 Analysis

Chapter 6

Conclusion

6.1 Summary

6.2 Future work

Bibliography

[1] H. F. Grip et al., "Flight control system for nasa's mars helicopter," in *AIAA Scitech 2019 Forum*, 2019. DOI: 10.2514/6.2019-1289

[2] B. Balaram et al., "Mars helicopter technology demonstrator," in *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018. DOI: 10.2514/6.2018-0023

[3] M. Vendittelli, *Quadrotor control via dynamic feedback linearization*, Course Slides, Elective in Robotics, Sapienza University of Rome, 2024.

[4] M. Vendittelli, *Ingenuity dynamic model (co-axial rotor helicopter)*, Course Slides, Elective in Robotics, Sapienza University of Rome, 2024.