Department of Systems & Control
Department of Biomechanical Design

Trajectory control and dynamic modeling of a tailless flapping-wing robot

Master Thesis

J. Kelbling

24-11-2020

Delft University of Technology

Challenge the future

Acknowledgement

I would like to express my sincere gratitude to my research supervisors, dr. ir. E. Steur and dr. ir. A. Seth, for their continuous support, feedback and guidance throughout this project. Their patience and vision have inspired me. They have taught me the methodology to carry out the research and how to present the research works as clearly as possible.

Besides my supervisors, I would like to thank dr. D. A. Olejnik, dr. ir. M. Karásek and dr. G.C.H.E. de Croon for providing data and meeting me from time to time to tackle the parameter estimation problem.

I wish to acknowledge the support and great love of my family; I am indebted to your love and acceptance. Last but not the least, I wish to express my deepest gratitude to my caring, loving and wonderful fiancée, Yasmin. They kept me going on.

Jelle Kelbling

Trajectory control and dynamic modeling of a tailless flapping-wing robot

J. Kelbling

24-11-2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology

Abstract

The DelFly Nimble is a type of tailless flapping-wing micro air vehicles (FWMAVs) that has received an increasing amount of attention. FWMAVs show efficient and agile flight possibilities at small scale. The aerodynamics and dynamics of these flapping vehicles are challenging and not fully understood. In this work, a strategy to implement a 3D dynamic model and a trajectory control algorithm is proposed for the DelFly Nimble by the use of a quasi-steady state framework. The design of a suitable tracking control algorithm is the essence of this task. A globally defined smooth nonlinear geometric framework of the flapping-wing vehicle's rigid body dynamics is introduced as a basis for the analysis. This grants an unambiguous coordinate-free dynamic model in which problem of singularities are avoided. The Nimble has four inputs used to control the six translational and rotational degrees of freedom. A nonlinear tracking controller is chosen on the special Euclidean group $SE(3)$ for the underactuated aerial vehicle, where position and yaw trajectory tracking are achieved. The full system is classified into the coupled attitude and position subsystems. Using the Lyapunov Stability theorem, the nonlinear controller is shown to achieve almost global asymptotic tracking of the attitude error dynamics of the Nimble and almost global asymptotic tracking of the position error dynamics of the center of mass of the Nimble, enabling sufficient tracking of aggressive maneuvers. Finally, the dynamic model and the controller are examined with numeric simulations. From the results can be concluded that the nonlinear control design allows for aggressive aerobatic maneuvers while maintaining stability of the closed-loop system, provided that the control inputs and damping forces remain moderate.

***Keywords**:* Autonomous vehicles, Nonlinear control systems, Lie algebra, Flapping flight, Flapping wings, Micro air vehicles, Robotics, Trajectory tracking, Geometric Control

Delft University of Technology
Department of
Delft Center for Systems & Control
BioMechanical Engineering

Dated: 24-11-2020

Supervisors: dr. ir. E. Steur

dr. ir. A. Seth

Readers: dr. ir. A.J.J. van den Boom

dr. ir. J. Moore

Acronyms

Acronym	Meaning
AHRS	Attitude Heading Reference System
CFD	Computational Fluid Dynamics
CoM	Center of Mass
CoP	Center of Pressure
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
EOM	Equations of Motion
FBD	Free-Body Diagram
FBL	Feedback linearization
FCF	Flapping Counter Force
FCT	Flapping Counter Torque
FWMAV	Flapping-wing Micro Aerial Vehicle
GLS	Generalized Least Squares
I/O	Input and Output
IMU	Inertial Measurement Unit
KF	Kalman Filter
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MAV	Micro Aerial Vehicle
MIMO	Multiple-Input Multiple-Output
MLE	Maximum Likelihood Estimation
NLS	Nonlinear Least Squares
OLS	Ordinary Least Squares
OMCS	OptiTrack-Motion Capture System
PD	Proportional Derivative
PID	Proportional Integral Derivative
QP	Quadratic Program
RK4	Runge-Kutta 4
RMSE	Root-Mean-Square Error
SMC	Sliding mode control
UAV	Unmanned Aerial Vehicle
WLS	Weighted Least Squares

List of Symbols

Notation	Meaning
γ	Dihedral angle
β	Wing root angle
α	Angle of attack
Φ	Wing-stroke amplitude
π	Ratio of a circle its circumference to its diameter
τ	Torque vector in inertial frame
τ_x, τ_y, τ_z	Torques around the x-, y- and z-axes in inertial frame
τ_b	Torque vector in body frame
τ_b'	The unrotated torque vector in body frame
$\tau_{b,v}$	Virtual torque input
$\tau_{b,x},\ \tau_{b,y},\ \tau_{b,z}$	Torques around the x-, y- and z-axes in body frame
f_b	Force vector in body frame
f_b'	The unrotated force vector in body frame
$f_{b,v}$	Virtual thrust input
$f_{b,x}, f_{b,y}, f_{b,z}$	Forces in the x-, y- and z-directions in body frame
ω	Angular velocity vector
ϕ, θ, ψ	Euler angles describing the orientation of the body with respect to the inertial frame (pitch, roll and yaw).
Θ	Vector containing the Euler angels
q	Quaternion
ω_b	Angular velocities in body frame
$\omega_{b,d}$	Desired angular velocities in body frame
ω_c	Control angular velocity
p, q, r	Roll, pitch and yaw angular velocities
d_x, d_y, d_z	Linear aerodynamic damping force coefficients along the x-, y- and z-axes in body frame
$\bar{d}$	Wingbeat-cycle-averaged drag force
d	Constant damping force coefficient
b	Constant damping force coefficient
c_1	Constant thrust force coefficient
c_2	Constant thrust force coefficient
k_1, k_2, k_3, k_4	Runge-Kutta 4 steps
Δt	Step size
X_n	State at time step n

$d_{b,x}, d_{b,y}, d_{b,z}$	Constant force coefficient for the body along the x-, y- and z-axes in body frame
$d_{l,x}, d_{l,y}, d_{l,z}$	Constant force coefficient for the left wing along the x-, y- and z-axes in body frame
$d_{l,x}, d_{l,y}, d_{l,z}$	Constant force coefficient for the left wing along the x-, y- and z-axes in body frame
$D_{l,x}, D_{l,y}, D_{l,z}$	Damping force for the left wing along the x-, y- and z-axes in body frame
$D_{r,x}, D_{r,y}, D_{r,z}$	Damping force for the right wing along the x-, y- and z-axes in body frame
$d_{r,x}, d_{r,y}, d_{r,z}$	Constant force coefficient for the left wing along the x-, y- and z-axes in body frame
l_l	Constant length vector for the left wing
$l_{l,x}, l_{l,y}, l_{l,z}$	Constant length parameter for the left wing along the x-, y- and z-axes in body frame from the Com to the CoP
l_r	Constant length vector for the left wing
$l_{l,x}, l_{l,y}, l_{l,z}$	Constant length parameter for the left wing along the x-, y- and z-axes in body frame from the Com to the CoP
l_γ	Constant length parameter from the CoM to the CoP
l_β	Constant parameter representing length of the wing root
h	Constant length parameter representing the height from the wing root to the CoP
ς_L	Rotation parameter for the left wing
ς_{L_s}	Fixed rotation offset parameter for the left wing
ϱ_L	Rotation parameter for the left wing
ς_R	Rotation parameter for the right wing
ς_R	Fixed rotation offset parameter for the right wing
ϱ_R	Rotation parameter for the right wing
$\mathrm{R}_{\mathrm{L}_s}$	Fixed rotation matrix describing the rotation of the CoP of the left wing
R'_L	Rotation matrix describing the rotation of the CoP of the left wing
R_L	Resultant rotation matrix of the CoP of the left wing
$\mathrm{R}_{\mathrm{R}_s}$	Fixed rotation matrix describing the rotation of the CoP of the right wing
R'_R	Rotation matrix describing the rotation of the CoP of the right wing
R_R	Resultant rotation matrix of the CoP of the right wing
$\mathcal{F}$	Aerodynamic mapping

$\mathcal{J}$	Jacobian of the generalized forces with respect to the control inputs
U	Speed of the wing with respect to the body
C_D	Drag coefficient
S	Wing area
ρ	Density of the air
f_L	Flapping frequency of the left wing
f_R	Flapping frequency of the right wing
F_L	Force of the center of pressure of the left wing
F_R	Force of the center of pressure of the right wing
g	Gravitational acceleration
m	Body mass
I	Identity matrix
J	Mass moment of inertia in body frame
I_{xx}, I_{yy}, I_{zz}	Mass moment of inertia around the x-, y- and z-axes in body frame
X	State of the system
$\dot{X}$	State derivative of the system
p	Position of the body frame with respect to the inertial frame
p_d	Desired position of p
v	Velocity of the body frame with respect to the inertial frame
v_d	Desired velocity of v
$\dot{v}$	Acceleration of the body frame with respect to the inertial frame
$x^{\mathcal{J}}, y^{\mathcal{J}}, z^{\mathcal{J}}$	Inertial frame coordinates
$x^{\mathcal{B}}, y^{\mathcal{B}}, z^{\mathcal{B}}$	Body frame coordinates in the longitudinal, lateral and vertical directions
v_b	Body velocity vector
u, v, w	Body velocities along the x^b-, y^b- and z^b-axes
$\alpha_x, \alpha_y, \alpha_z$	Body accelerations along the x^b-, y^b- and z^b-axes
$V_{l,CoP}$	Air velocity vector of the left wing its CoP
$V_{r,CoP}$	Air velocity vector of the right wing its CoP
$u_{l,CoP}, v_{l,CoP}, w_{l,CoP}$	Air velocities of the left wing its CoP projected onto the x^b-, y^b- and z^b-axes
$u_{r,CoP}, v_{r,CoP}, w_{r,CoP}$	Air velocities of the right wing its CoP projected onto the x^b-, y^b- and z^b-axes
R	Unless specified differently this rotation matrix describes $R_{\mathcal{B}}^{\mathcal{J}}$, which is the rotation from body frame $\{\mathcal{B}\}$ to inertial frame $\{\mathcal{J}\}$
R_L	The rotation matrix for the left wing
R_R	The rotation matrix for the right wing

R_d	Desired attitude
R_c	Control attitude
W_b	Wrench in body frame
Ad_H	Adjoint mapping
$\mathcal{B}$	Body fixed coordinate frame of the Nimble
$\mathcal{I}$	Inertial fixed coordinate frame (earth)
Φ	Vector containing the model parameters
$\hat{\Phi}$	Vector containing the estimated model parameters
Φ_0	Vector containing the initial model parameters
$J(\Phi)$	Cost function that aims to optimize the model parameters
$SE(3)$	The set of positions and orientations of the rigid body moving in a three-dimensional space is called the Special Euclidean group
$SO(n)$	The set of n-dimensional Special Orthogonal group
$\mathfrak{so}(3)$	The space of 3×3 skew-symmetric matrices and the Lie algebra of $SO(3)$
Re	Numbers defined on the real axis
$\mathbb{R}^n$	The n-dimensional Euclidean space
$\mathbb{S}^n$	The n-dimensional unit sphere
$\lVert\cdot\rVert$	The Euclidean norm of $(\cdot)$
$\mathbb{N}$	The set of integers
e_p	Tracking error vector of the position in inertial frame
e_v	Tracking error vector of the velocity in inertial frame
e_{ω_b}	Tracking error vector of the angular velocity in body frame
Ψ	Real valued error function on $SO(3) \times SO(3)$
k_p	Control parameter for e_p
k_v	Control parameter for e_v
k_{ω_b}	Control parameter for e_{ω_b}
$\lambda_{min}, \lambda_{max}$	Smallest and highest eigenvalues of the inertia tensor matrix
$\mathcal{V}_R$	Lyapunov function for the attitude loop
$\mathcal{V}_p$	Lyapunov function for the position loop
ϵ_R, ϵ_p	Constant used in the $\mathcal{V}_R$ and $\mathcal{V}_p$
$\mathcal{W}_R$	Upper bound of $\dot{\mathcal{V}}_R$
$\mathcal{M}_{R1}$, $\mathcal{M}_{R2}$	Lower bound and upper bound of $\mathcal{V}_R$
$\mathcal{M}_p$	Upper bound of $\mathcal{V}_p$
$\mathcal{W}_p, \mathcal{W}_{pR}$	Upper bounds of $\dot{\mathcal{V}}_p$
k_R, k_{ω_b}	Control gain parameters
$\mathcal{A}_R, \mathcal{A}_{\omega_b}$	Region of attraction of the attitude and angular velocity
B	Upper bound of linear acceleration

ϑ	Axis-angle
$\mathcal{U}$	Control inputs
$\mathcal{V}$	Virtual control input
σ	Trajectory
c_n	Trajectory coefficients
σ^*	Optimal trajectory
$\mathcal{L}$	Cost function

Contents

List of Figures

List of Tables

1. Introduction

1.1. Background and motivation

Recently, bio-inspired flapping-wing micro-air vehicles (FWMAVs) have been introduced [1]–[6]. FWMAVs are a subcategory of unmanned aerial vehicles (UAVs). The creation of these small-scale robots is motivated by various applications, such as: reconnaissance, assistance in agriculture, search and rescue missions and aiding animal biologists to research insect flight, for example that of hummingbirds, bats and fruit flies. Traditional fixed-wing robots differ greatly from flying animals and are unsuitable for small-scale vehicles like FWMAVs, since they require high angular speeds and relatively more energy in order to produce enough lift at low Reynolds numbers [7]. FWMAVs can be used to gain insight in known animal processes and subsequently test existing hypotheses on animal flight control.

FMWAVs usually are small in size, light in weight and have the potential to improve their flight performance as technology enhances. Due to the harsh restrictions in size and weight of FWMAVs, technological challenges arise. Most existing vehicles lack agility and sufficient power and therefore cannot take off properly or fly for a duration longer than one minute. Thus, they cannot match the flight performance of their biological equivalents. FWMAVs can be divided into two types, namely: tailed FWMAVs and tailless FWMAVs. Tailless designs are more agile than tailed designs. However, tailless FWMAVs are inherently unstable and thus have the need for active control systems.

There are not many FWMAVs without tails on the market yet. An example of a tailless FWMAV is the DelFly Nimble [8]. This particular robot has two wings on each side, flapping in opposite direction on either side. The design of the DelFly Nimble is inspired by animals, such as hummingbirds and fruit flies. It has two main goals, namely: to mimic flying animals and to autonomously perform tasks in unknown environments. Like a quadrotor and several insects such as fruit flies, the DelFly Nimble is capable of vertical flight and can achieve sideways flight by pitching or rolling its body.

1.2. Problem statement

The orientation of the DelFly Nimble is presently stabilized by a PD control design. A joystick is used in order to generate set-points for the actuators and let it fly in the desired directions [8]. The DelFly Nimble can also perform subtle movements during hover or travel large distances and is inherently safe due to a lightweight design without propellers. Also, hypotheses, can be tested on dynamically scaled robots [9] or theoretical modeling. One example is the hypothesis that fruit flies use the translation-induced coupled yaw torque to rotate their body in the direction of the banked turn [10]. So far, the development of this particular robot has mainly been improved by trial-and-error and a physical model is lacking [8].

Previous studies, conducted between 1999 and 2019, show that controlling FWMAVs, and thus controlling the DelFly Nimble, is a challenge due to the unsteady nonlinear aerodynamics, extraordinary force generation mechanisms, flapping oscillations, nonlinear coupled flight dynamics and disturbances [9]–[15]. Furthermore, it should be noted that at small-scale, sensors become less accurate and power reduces considerably. The PD controller limits the use of the development of the DelFly Nimble as it is flies around one trimmed condition. to account for these nonlinear system dynamics. Besides attitude control, a trajectory control design is not yet available for the DelFly Nimble. This is desirable for efficiently realizing measurements, demonstrations and recreation. The implementation of a trajectory control design would bring autonomous flight a step closer.

In order to design a trajectory controller, a dynamic model is needed. Regardless, there is still lack of understanding the aerodynamic force mechanisms and a full dynamic model. It is desirable to develop a physics-based dynamical system in a virtual environment. Learning-based control methods would be impractical due to the vulnerability of the DelFly Nimble and it would be costly to repair it after every experiment. Furthermore, these control methods will work after trial and error, but cannot guarantee stability. A physics-based model would allow for dynamic simulations and model-based control. Such a model would give knowledge about the interaction between aerodynamics, flight kinematics and impact of design parameters. For newer versions of the system, the dynamic model and control design can be updated and still be used. For example, it is interesting to find ways to improve flight performance and efficiency. Physical or control parameters could be tuned and optimized in a virtual environment. Also, a dynamic model is of interest by itself, for example to understand the DelFly Nimble. Although it seems attractive to use detailed fluid dynamic simulations, this is not practicable. These simulations cannot be computed fast enough to control the DelFly Nimble as they require high computational effort [8]. Often, quasi-steady models are used to estimate the flapping-wing aerodynamics for a prescribed kinematic trajectory [11], [16]–[19]. In order to apply control strategies, a physics-based model is not only convenient, but might even be necessary employing a trajectory controller.

1.3. Aims

The first aim of this study is to develop a nonlinear global dynamic model of the DelFly Nimble.

- The first subaim is to design a dynamic model of the DelFly Nimble's body.
- The second subaim is to create an aerodynamic model for the DelFly Nimble.
- The third subaim is to make the dynamic model well-suited for:
  - control design and stability analysis;
  - simulation and parameter optimization.

The second aim of this study is to propose a nonlinear trajectory tracking controller for the DelFly Nimble.

- The first subaim is to create two output tracking controllers, namely:
  - An attitude tracking controller that asymptotically tracks the commanded attitude of the DelFly Nimble;
  - A position tracking controller that asymptotically tracks the commanded position of the DelFly Nimble.
- The second subaim is to let the DelFly Nimble perform three aerobatic maneuvers that require aggressive complex flight responses with large initial tracking errors in which position and attitude aggressive maneuvers are tracked, namely:
  - Recovering from an upside-down orientation;
  - A $360°$ flip;
  - Following an elliptic helix trajectory.

1.4. Organization

This work is organized as follows: the preliminary notions and the mechanisms of the DelFly Nimble are defined in Chapter 2. Chapter 3 describes the development of the system dynamics and flapping-wing aerodynamics of the DelFly Nimble. The system configuration space is represented by a differentiable manifold employing the tools of differential geometry. Furthermore, in Chapter 4, results and validations are presented for the dynamic model. Thereafter, in Chapter 5, the control design is described. The controller allows for tracking different degrees of freedom for both the position and attitude closed loops. Intermediate stability properties are given and subsequently used for the stability analysis of the total interconnected closed-loop system. In Chapter 6, several numerical results are defined that demonstrate the closed-loop behavior and complex aerobatic maneuvers for the DelFly Nimble. Finally, the conclusions of this work and recommendations for future research are presented in Chapter 7.

2. DelFly Nimble

The DelFly Nimble project started in 2005 at the Delft University of Technology. Since then, subjects as design, aerodynamics and artificial intelligence have been studied. The tailless design of the Nimble is inspired by animals like the hummingbird and fruit fly [8]. The DelFly Nimble is a programmable flying insect-inspired robot. The Nimble is controlled through insect-inspired motion actuation with four flapping wings, flapping around 16.5 Hz and weighing 29.4 grams. The vehicle does not have a tail, which makes it greatly agile and less vulnerable to collisions. The two bio-inspired wing pairs on the left and on the right sides flap in counter-phase and clap and peel with each other to alleviate thrust force. The wings fully determine the motion of the Nimble and enable vertical and sideways flight by pitching or rolling the Nimble’s body. It is an underactuated system that has four DoF and is controlled by motions of the wings, inspired by the fruit fly. It can fly for about five minutes and has a battery capacity of 180 mAh at 3.7V. Moreover, it is capable of agile and subtle motions when hovering and in fast forward flight. Inherent safety is guaranteed due to a lightweight design without propellers. The DelFly Nimble, shown in Figure 2-1, is able to autonomously hover and perform rapid escape maneuvers just as flying insects. The agile robot is able to make 360° flips with angular accelerations up to $5000^\circ/s^2$ in open loop and has a maximally attainable speed of $7\,ms^{-1}$. Regardless of the variety in morphology and system aspects, FWMAVs all encounter passive aerodynamic damping, termed flapping-counter-force (FCF) and flapping-counter-torque (FCT) with similar characteristics [10].

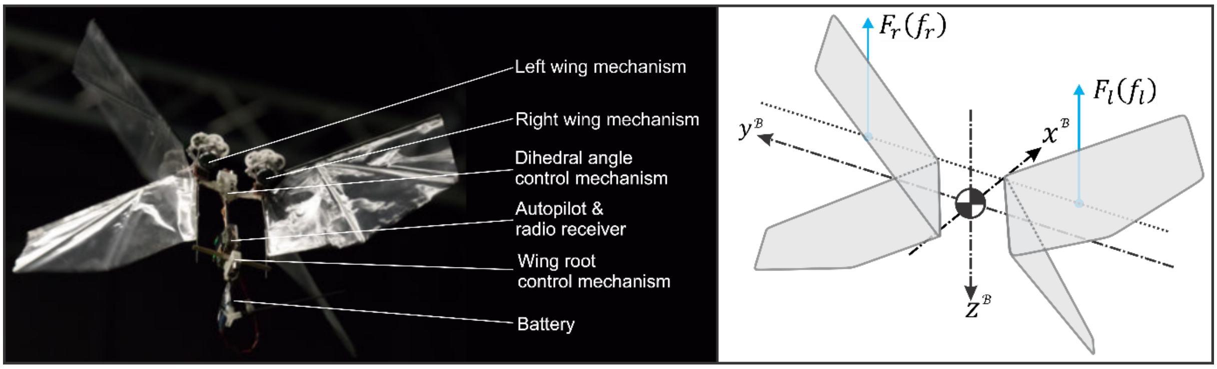

Figure 2-1: The DelFly Nimble [1]*. On the right the model that will be used is portrayed with the same orientation as the DelFly Nimble on the left.*

The free-body diagram with the concept of the control inputs is depicted in Figure 2-2. In the body diagram of the DelFly Nimble, the x-axis represents the longitudinal direction and points forwards, the y-axis represents the lateral direction and points to the right and the z-axis represents the vertical direction and points downwards. The two rotary actuator servos adjust the dihedral angle γ and the tips of the wing-pair roots. By actuating γ, the centerline of the flapping stroke changes, together with the thrust vector with respect to the CoM (Figure 2-2A). This creates a pitch torque and regulates the forward/backward movement of the Nimble. The roll angle is adjusted by changing the difference of the thrust magnitude of the wings, by applying different flapping frequencies (Figure 2-2B). This enables the Nimble to move sideways. The yaw angle is regulated by changing the wing root angle β, resulting in twisted thrust vectors generated by the wings

(Figure 2-2C). The total thrust is regulated by the synchronous flapping frequency of the wings (Figure 2-2D). The combination of the inputs enables the Nimble to hover and move in any direction with rapid, yet smooth, and consistent transitions.

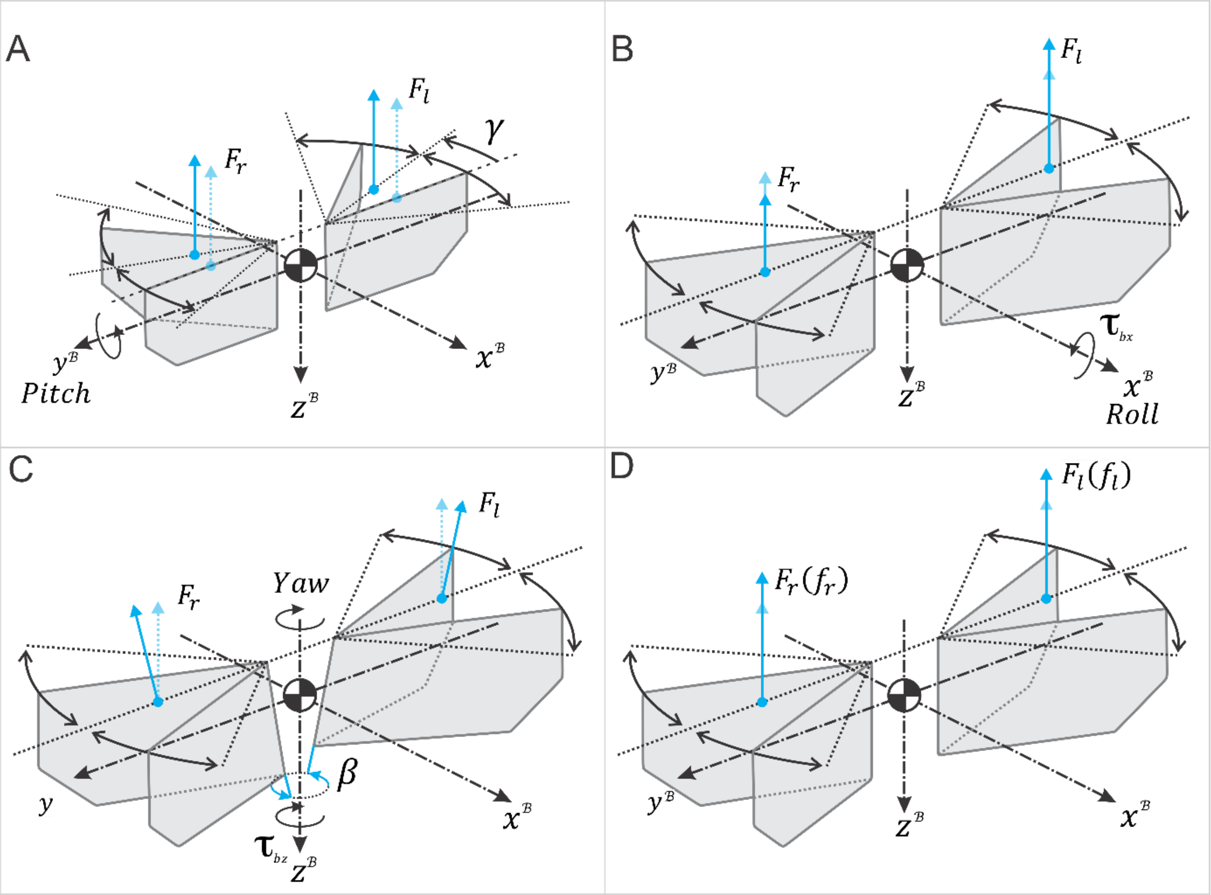

Figure 2-2: Free-body diagram in $\mathbb{R}^6$ of the DelFly Nimble. (A) Pitch moment (B) Roll moment (C) Yaw moment (D) Thrust. The force position is depending on the input γ $[rad]$. The force its orientation is determined by both the inputs γ and β and its magnitude by the frequencies of the wing f_l and f_r.

Figure 2-3 shows the inputs and outputs of the DelFly Nimble. There are four inputs: (1) the flapping frequency for the left wing f_l, (2) the flapping frequency for the right wing f_r, (3) the wing root angle β and (4) the dihedral angle γ. By regulating the inputs of the Nimble, forces and torques are generated that update the body dynamics. Data can be obtained in a testing facility, named the Cyberzoo. There are several sensors to measure the states of the Nimble [20]. 12Prime 17W OptiTrack cameras are measuring the x-, y- and z-position, attitude quaternions and the dihedral angle with a rate up to 360 Hz. The onboard IMU AHRS sensor obtains the angular velocities (p, q and r) and the linear accelerations (a_x, a_y and a_z), as is shown in Figure 2-3. Furthermore, the flapping frequencies (f_l and f_r), setpoints, dihedral angle γ commands and servo feedback are also recorded [10][8].

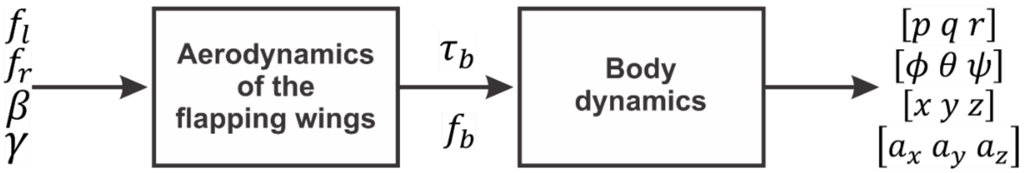

Figure 2-3: Inputs and outputs of the DelFly Nimble.

3. System dynamics

In this section the dynamic model for the DelFly Nimble in $\mathbb{R}^6$ is derived. Little knowledge has been available for the dynamics of the DelFly Nimble. Therefore, previous works have focused on black-box, grey-box and white-box models [5],[7],[49]–[51]. The design of the system dynamics and aerodynamics is crucial for the understanding of the physics of the Nimble and the design of an appropriate control algorithm. The dynamic models of FWMAVs are often too complex to be used on board by a controller or too simple to fully explore the potential of the DelFly Nimble.

In contrast to black-box models, white-box models allow for employing knowledge of the dynamics of the Nimble, preserving the link to the physics of the system. In order to obtain a single model that is valid for the entire flight envelope, it is desirable to implement a white-box model in $\mathbb{R}^6$, which can be used for control design and simulation. Simulation allows for changing physical and control parameters and gives insight of the closed-loop system to gain understanding of the complex behavior of these systems, such as when actuators saturate and what trajectories would be feasible. Furthermore, this allows for changing physical parameters such as length dimensions and show how this changes the behavior of the Nimble.

A white box approach was used to describe the longitudinal dynamics [7], [8]. The rigid body dynamics are actuated by forces and torques that depend on the mean aerodynamic forces, which are applied on the centers of pressure of the wings. This particular model described in this section is based on conventional aircraft dynamics and aerodynamics. In this work, the Nimble is modelled as a rigid body in $\mathbb{R}^6$, regulating its thrust along its vertical body-fixed axis and regulating torques around all three body-fixed axes. The rigid body dynamics captures the underactuated system, its intrinsic instability and the coupling between the translational and rotational nonlinear dynamics. Since this model has a white-box structure, it is useful for insight in the physics and dynamics of the DelFly Nimble. Subsequently, it allows for virtual simulation and parameter changes instead of trial-and-error in real life. Insight can also be obtained by simulating the closed-loop system to gain understanding when input saturation occurs and what trajectories would be feasible. Ultimately, this model can also be applied to other FWMAVs regardless the shape, kinematics and configuration of the wings.

Section 3.1 introduces the configuration space, appropriate reference frames and a suitable set of degrees of freedom in $\mathbb{R}^6$. Section 3.2 discusses the dynamics of a rigid body. The aerodynamics of the flapping wings are derived in Section 3.3. The inverse aerodynamics is discussed in Section 3.4. And lastly, in Section 3.5 a summary of the dynamics and concluding remarks are presented.

3.1. Rigid body configuration description

To describe angles in 3D Euclidean space, at least three parameters are required. The configuration space of the Nimble with respect to $\{\mathcal{I}\}$ is considered as an element of $SE(3)$, denoted as

$$SE(3) \coloneqq \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, \quad R \in SO(3), \ p \in \mathbb{R}^3 \right\},$$

with $p \in \mathbb{R}^3$ the position and $R \in SO(3)$ the rotation matrix from body frame to inertial frame $\left(R_{\mathcal{B}}^{\mathcal{I}}\right)$. The Euclidean space is a continuous set of points in conjunction with a structure describing orthogonality measuring length. The crucial aspect is that topological and algebraic properties concur to real physical properties of particles their motion. Powerful mathematical tools are at hand which allow to describe motion of rigid bodies in a geometrical and global way [23]. The geometric motion of the rigid body is written as elements of

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} \omega_b & v \\ 0 & 0 \end{pmatrix}, \quad \omega_b \in \mathfrak{so}(3), \quad v \in \mathbb{R}^3 \right\}.$$

In this work, the right-handed coordinate frame is used. The attitude of the body of the Nimble, a vector in $\mathbb{R}^3$, can be described by using a linear 3×3 transformation matrix between the body reference frame and the inertial frame (earth). The set of every rotation matrix is called the special orthogonal group in $\mathbb{R}^3$, indicated by $SO(3)$. The algebraic structure is a smooth manifold and is length-preserving. Every rotation matrix $R \in SO(3)$ has the properties of associativity, identity and inverse as

$$SO(3) \coloneqq \{R \in \mathbb{R}^{3\times3}, \qquad R^T R = I = R R^T, \qquad \det(R) = 1\}.$$

This is a Lie group and is called the special orthonormal group. Note that also improper matrices exist and will not be used. The inertial frame $\mathcal{I}$ with basis vectors $\{x^{\mathcal{I}}\, y^{\mathcal{I}}\, z^{\mathcal{I}}\}$ is fixed with respect to the earth. The body frame $\mathcal{B}$ with basis vectors $\{x^{\mathcal{B}}\, y^{\mathcal{B}}\, z^{\mathcal{B}}\}$ is fixed to the CoM of the rigid body. $R_{\mathcal{B}}^{\mathcal{I}}$ represents the rotation matrix from the body frame to the inertial frame. The three orthogonal unit basis vectors of frame $\mathcal{B}$ resolved in frame $\mathcal{I}$ are represented by the three column vectors of R. Idem, the three orthogonal unit basis vectors of frame $\mathcal{I}$ resolved in frame $\mathcal{B}$ are represented by the three row vectors of R. The body-fixed frame rotates around the inertial frame, by rotating around the z-axis (ψ), y-axis (θ) and the x-axis (ϕ) $\in \mathbb{R}^3$, according to the aerospace notation. The final rotation matrix is given by

$$R_{\mathcal{B}}^{\mathcal{I}}\,(\phi, \theta, \psi) = R_z(\psi) R_y(\theta) R_x(\phi) = R_{\mathcal{I}}^{\mathcal{B}^T} SO(3), \tag{3-1}$$

where $c_i = \cos(i)$, $s_i = \sin(i)$ and $t_i = \tan(i)$. The first and second axes of the body-fixed frame, $x^{\mathcal{B}}$ and $y^{\mathcal{B}}$ lie in the plane of the flapping motion as depicted in Figure 3-1. The third body-fixed axis $z^{\mathcal{B}}$ is normal to this plane. The configuration of the Nimble is defined by the location of the CoM and the attitude with respect to the inertial frame.

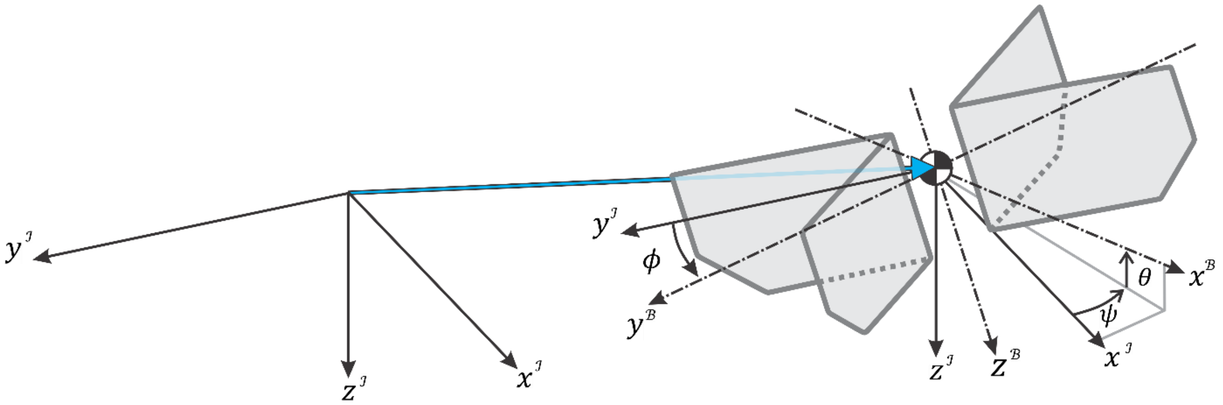

Figure 3-1: Schematic configuration of the Nimble. The inertial fixed reference system ($\mathcal{I}$) and the body reference frame system ($\mathcal{B}$), with the Euler angles ϕ, θ and ψ. The body reference frame can be converted to the fixed reference frame by means of the rotation matrix $R_{\mathcal{B}}^{\mathcal{I}}$.

The rotation matrices are able to represent all body attitudes globally and uniquely. The tangent space at the identity element is called the Lie algebra. $R(t)$ is a differentiable function of time. By differentiating $R(t)^T R(t) = I$ to time, $\dot{R}R^T$ and $R^T\dot{R}$ are skew-symmetric matrices belonging to $\mathfrak{so}(3)$ with the properties

$$\mathfrak{so}(3) \coloneqq \left\{\widetilde{\omega}_b \in \mathbb{R}^{3\times 3}, \qquad -\widetilde{\omega}_b = \widetilde{\omega}_b^T\right\}.$$

The Lie group makes it possible to know the angular velocity of a rigid body ω_b, without knowing the actual pose. The Lie algebra portrays the local properties of its parent Lie group, while the geometric properties are preserved. The angular velocity is described by the notation

$$\widetilde{\omega}_{\mathcal{I}}^{\mathcal{I},\mathcal{B}} \coloneqq R_{\mathcal{B}}^{\mathcal{I}} \dot{R}_{\mathcal{I}}^{\mathcal{B}},$$

in which the in the angular velocity of frame $\mathcal{I}$ with respect to frame $\mathcal{B}$ is expressed in frame $\mathcal{B}$.

3.2. Rigid body dynamics

Using solely orthonormal matrices gives both a globally and unique formulation for the attitude of the Nimble. Moreover, this allows the employment of the geometric control approach, which will be elaborated in Chapter 5. In order to describe the rigid body, the following parameters are defined in Table 3-1.

Table 3-1: Parameters of the rigid body dynamics.

Symbol	Unit	Description
$m \in \mathbb{R}^+$	$[kg]$	Body mass
$g \in \mathbb{R}$	$\left[\frac{Nm}{s^2}\right]$	Gravitational acceleration
$J \in \mathbb{R}^{3\times3}$	$[kgm^2]$	Inertia matrix in frame $\mathcal{B}$
$R \in SO(3)$	$[-]$	Rotation matrix from frame $\mathcal{B}$ to frame $\mathcal{I}$
$p = [x\ y\ z] \in \mathbb{R}^3$	$[m]$	Position of the center of mass in frame $\mathcal{I}$
$v = [\dot{x}\ \dot{y}\ \dot{z}]^T \in \mathbb{R}^3$	$\left[\frac{m}{s}\right]$	Velocity of the center of mass in frame $\mathcal{I}$
$v_b = [u\ \ v\ \ w]^T \in \mathbb{R}^3$	$\left[\frac{m}{s}\right]$	Velocity of the center of mass in frame $\mathcal{B}$
$\omega = \left[\dot{\phi}\ \dot{\theta}\ \dot{\psi}\right]^T \in \mathbb{R}^3$	$\left[\frac{rad}{s}\right]$	Angular velocity in frame $\mathcal{I}$
$\omega_b\ = [\,p\ q\ r]^T \in \mathbb{R}^3$	$\left[\frac{rad}{s}\right]$	Angular velocity in frame $\mathcal{B}$
$f_b\ = \left[f_{b,x}\ f_{b,y}\ f_{b,z}\right]^T \in \mathbb{R}^3$	$[N]$	Applied forces in frame $\mathcal{B}$
$f_d\ = \left[f_{d,x}\ f_{d,y}\ f_{d,z}\right]^T \in \mathbb{R}^3$	$[N]$	Aerodynamic damping force of body in frame $\mathcal{B}$
$\tau_b\ = \left[\ \tau_{b,x}\ \tau_{b,y}\ \tau_{b,z}\right]^T \in \mathbb{R}^3$	$[Nm]$	Applied torques in frame $\mathcal{B}$

Rigid body systems are commonly expressed in body frame. This is convenient, because it makes use of assumptions of the rigid body and since they are often equipped with inertial measurement units (IMU) measuring the state variables. The flight dynamics are described in Equation (3-2). $I_{(\cdot\cdot)}$ are the principle components of the mass moment of inertia along the three axes. $f_{b_{(\cdot)}}$ describes the longitudinal, lateral and vertical forces in body frame. $\tau_{b_{(\cdot)}}$ describes the longitudinal, lateral and vertical moments in body frame. Finally, f_d represents the damping force of the body. Despite the fact that the Nimble does not resemble a conventional aircraft, the reviewed literature and previous work for the Nimble show that the flight dynamics can represent FWMAVs [3], [7], [20], [24].

The nonlinear EoM are derived in Appendix I and in short, the nonlinear EoM of the rigid body dynamics are summarized as

$$\begin{aligned} \dot{p} &= v, \\ \dot{v} &= gz^{\mathcal{I}} - m^{-1}Rf_b, \\ \dot{R} &= R\widetilde{\omega}_b, \\ \dot{\omega}_b &= -J^{-1}\widetilde{\omega}_b J\omega_b + J^{-1}\tau_b. \end{aligned} \tag{3-2}$$

An appropriate set of DoF is defined for which the attitude is parameterized by a matrix R rather than quaternions or Euler angles. The tilde notation $\widetilde{(\cdot)}$ maps a vector to a skew-symmetric matrix $\in \mathbb{R}^3 \to so(3)$ as

$$\tilde{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{3-3}$$

This preference is founded on the ambiguity-free and global characteristics of a rotation matrix representation. In Section 3.3 the rigid body dynamics will be coupled to the aerodynamic model and in Chapter 5 this model will be used for the trajectory control design.

3.3. Flapping-wing aerodynamics

Although there is much technological development for the Nimble, there is still lack of understanding the aerodynamic force mechanisms and lack of a full dynamic model. The dynamics and force generation of flapping wings are complex. Commonly, aerodynamic forces and moments are expressed in linear terms or polynomials with time-invariant parameters quantifying the contribution to the aerodynamic force or moment components. In particular, quasi steady dynamics are widely discussed in the literature [2], [11], [15]–[19], [25]–[27]. Quasi steady dynamics can be used when the bandwidth of the flapping dynamics is much higher than the desired motion of the Nimble. This implies that the forces generated by the wings are averaged over one stroke. This is a common approach to analyze flapping dynamics, which has been discussed mathematically and is accepted by physical intuition. The work of Huo et al. (2015) assumes that the mean translational aerodynamic forces, depending on the velocity of the wing and body, apply on the centers of pressure of the wings. This method was applied and validated for the longitudinal (2D) dynamics. As a matter of fact, the x-, y- and z-components of the FCFs were demonstrated analytically by Cheng and Deng (2011). By using this justification, in this section the concept is extended to the full dynamics in $\mathbb{R}^6$ and incorporated in the 6 DoF Equations of Motion (EoM) of conventional rigid bodies.

3.3.1. Linear drag model

The stroke-averaged drag force at the center of pressure (CoP) of the wing can be written by the translational force component of the quasi-steady assumption as [2], [3]

$$\bar{D} = \frac{1}{T}\int_0^T \frac{1}{2}\rho C_D\big(\alpha(t)\big)SU^2(t)dt, \tag{3-4}$$

in which $T = 1/f$ is the wingbeat period, ρ the density of air, C_D a drag coefficient at a certain angle of attack α, S the wing area and U the speed of the wing with respect to the body at time t. Furthermore, α and U are assumed to be constant as the wings flap back and forth at a certain wingbeat frequency f (quasi-steady assumption). The speed of the wing due to flapping can be written as

$$U = 2\Phi l_\gamma f, \tag{3-5}$$

with l_γ the length from the center of mass (CoM) to the CoP of the wing and Φ the flapping amplitude. Subsequently, the drag force can be decomposed in the up-stroke and down-stroke [7], [10]. When the wing speed is higher than the body speed u, i.e. $U > u$, equation (3-4) can be rewritten as

$$\begin{aligned} \bar{d} &= \frac{1}{2}\bar{d}_{down} + \frac{1}{2}\bar{d}_{up}, \\ &= \frac{1}{4}\rho C_D(\alpha)S(-(U+u)^2 + (U-u)^2), \\ &= b\Phi f u. \end{aligned} \tag{3-6}$$

The flapping amplitude is constant, so the linear drag can be represented by

$$\begin{cases} \bar{d} = -b[U^2 + u^2] & \text{if } u > U, \\ \bar{d} = -bfu & \text{if } u < U, \end{cases} \tag{3-7}$$

in which $\bar{d}$ is the wingbeat-cycle-averaged drag force, f the flapping frequency, b a constant force coefficient (given the assumption of the angle of attack α to be constant) and u the body velocity. Equation (3-7) shows that the drag force is dependent on the flapping frequency, body speed and a constant which can be estimated by parameter estimation. Although these assumptions are not fully applicable, it was observed that the model was able to account for the behavior during measurements in the presence of wind [7]. Let alone, the assumption of Equation (3-7) will improve in the foreseeable future as the flapping frequency increases.

The Nimble has four control inputs, namely the two flapping frequencies of the flapping wings, the dihedral angle γ and the wing root angle β. In order to map the control inputs to the forces and moments about the x-, y- and z-axes as

$$\begin{bmatrix} f_{b,z} \\ \tau_{b,x} \\ \tau_{b,y} \\ \tau_{b,z} \end{bmatrix} = \mathcal{F}\left(\begin{bmatrix} f_l \\ f_r \\ \gamma \\ \beta \end{bmatrix}\right), \tag{3-8}$$

an aerodynamic analysis is carried out. For convenience, this analysis will be explained for the left wing and right wing separately. The concept is projected on the body of the Nimble in Figure 3-2. The drag force depends on the velocity of the center of pressure of the wing. The CoP translates with respect to the CoM on the black circle by varying the control input γ, and will be treated in Section 3.3.2. The combination of the control inputs γ and β changes the orientation of the CoP, and will be treated in Section 3.3.3. The translation and rotation of the CoP regulate the damping and thrust force vectors and ultimately regulates the motion of the body.

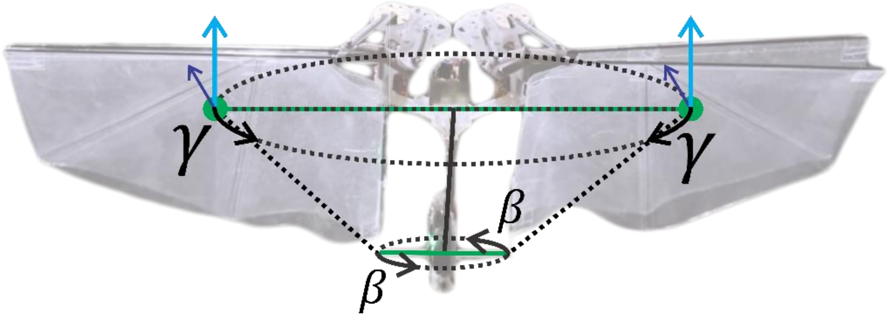

Figure 3-2: Concept of the aerodynamic model of the DelFly Nimble portrayed on the body while hovering. The blue and purple vectors represent the thrust and damping forces of the CoP, the green lines represent the variable length parameters, and the black lines define the path on which the length parameters are constrained.

3.3.2. Position of the centers of pressure

The concept of the translation of the CoP is shown in Figure 3-3. The forces acting on the left wing are represented by the thrust force F_l and drag force, which are split in the x-, y- and z-components attached to the CoP of the left wing. The location of the CoP changes by the control input γ with respect the CoM, which is represented by the length parameters $l_{l,x}$, $l_{l,y}$ and $l_{l,z}$, as depicted in Figure 3-3. In Figure 3-4, the FBD of the Nimble is projected in the yz-plane around the x-axis, the zx-plane around the y-axis and the xy-plane around the z-axis.

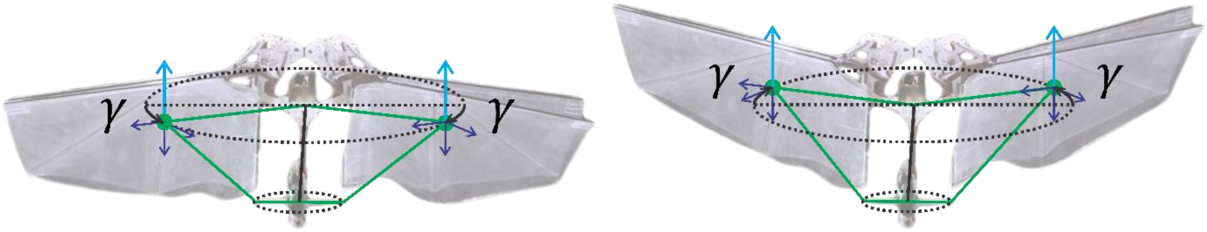

Figure 3-3: Concept of the aerodynamic model of the DelFly Nimble portrayed on the body. Positive and negative pitch are shown in the left and right figures respectively. The blue and purple vectors represent the thrust and damping forces of the CoP, the green lines represent the variable length parameters, and the black lines define the path on which the length parameters are constrained.

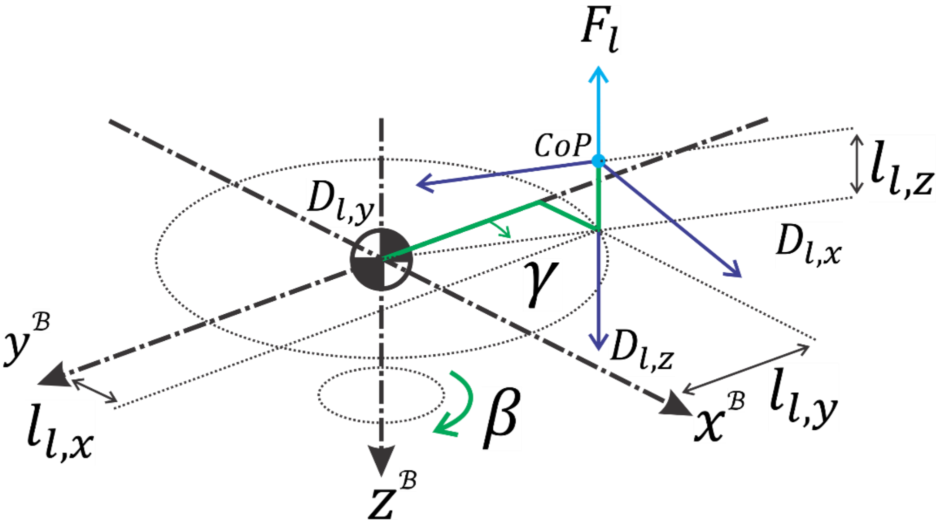

Figure 3-4: FBD of the Nimble for the left wing. The force acting on the CoP of the wing is decomposed in drag forces in the directions of the body frame (purple) and the thrust force (blue).

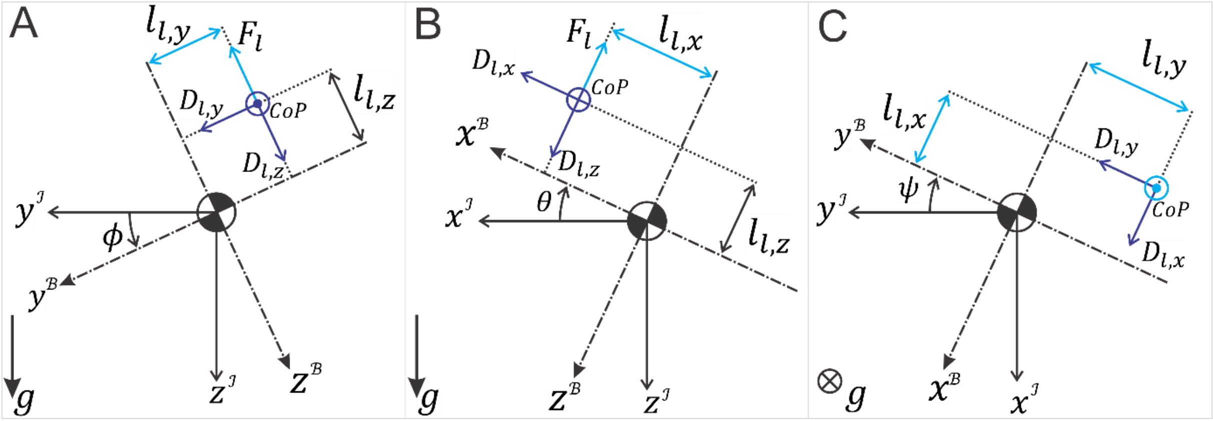

Figure 3-5: Projections of the FBD for the left wing. The FBD is projected in (A) the yz-plane around the x-axis, (B) the zx-plane around the y-axis and (C) the xy-plane around the z-axis.

The damping has a linear dependence on the body translational and angular velocities and increases with wing-stroke amplitude Φ, and wing frequencies f_l and f_r (see Equation (3-6)). Therefore, the drag force D_l of the left wing in the x-, y- and z-directions, with $d_l = \left[d_{l,x}, d_{l,y}, d_{l,z}\right]^T$ the damping force coefficient and $V_{l,CoP} = \left[u_{l,CoP}, v_{l,CoP}, w_{l,CoP}\ \right]$ the velocity of the CoP of the left wing, is modelled as

$$D_l = -d_l f_l V_{l,CoP}. \tag{3-9}$$

The velocities of the CoP of the left wing are expressed as

$$V_{l,CoP} = v_b + \dot{l}_l + \tilde{\omega}_b l_l, \tag{3-10}$$

in which $l_l = \left[l_{l,x}, -l_{l,y}, -l_{l,z}\right]^T$ is the location of the CoP with respect to the CoM of the left wing in body frame, which depend on the dihedral angle γ and the tilde map $\widetilde{(\cdot)}$ represents the cross product. The time derivatives of the length parameters of the CoP are also included. It was found that they change over time and cannot be neglected in the closed loop [2]. The thrust vector F_L linearly depends on the flapping frequency f_l of the wing. By combining Equations (3-9) and (3-10), the forces and torques in the body frame are given by Equations (3-11) and (3-12).

$$\begin{aligned} f'_{b,x} &= D_{l,x} = -d_{l,x} f_l\left(u - l_{l,z} q + l_{l,y} r + \dot{l}_{l,x}\right) \\ f'_{b,y} &= D_{l,y} = -d_{l,y} f_l\left(v + l_{l,x} r + l_{l,z} p + \dot{l}_{l,y}\right) \\ f'_{b,z} &= D_{l,z} - F_L(f_l) = -d_{l,z} f_l\left(w - l_{l,x} q - l_{l,y} p + \dot{l}_{r,z},\right) - F_L(f_l) \end{aligned} \tag{3-11}$$

$$\begin{aligned} \tau'_{b,x} &= D_{l,y} l_{l,z} - \left(D_{l,z} - F_L\left(f_l\right)\right) l_{l,y} \\ &= -d_{l,y} f_l\left(v + l_{l,x} r + l_{l,z} p + \dot{l}_{l,y}\right) l_{l,z} - \left(-d_{l,z} f_l\left(w - l_{l,x} q - l_{l,y} p + \dot{l}_{r,z}\right) - F_L(f_l)\right) l_{l,y} \\ \tau'_{b,y} &= -D_{l,x} l_{l,z} - \left(D_{l,z} - F_L\left(f_l\right)\right) l_{l,x} \\ &= d_{l,x} f_l\left(u - l_{l,z} q + l_{l,y} r + \dot{l}_{l,x}\right) l_{l,z} - \left(-d_{l,z} f\left(w - l_{l,x} q - l_{l,y} p + \dot{l}_{r,z}\right) - F_L(f_l)\right) l_{l,x} \\ \tau'_{b,z} &= D_{l,y} l_{l,x} + D_{l,x} l_{l,y} \\ &= -d_{l,y} f_l\left(v + l_{l,x} r + l_{l,z} p + \dot{l}_{l,y}\right) l_{l,x} - d_{l,x} f_l\left(u - l_{l,z} q + l_{l,y} r + \dot{l}_{l,x}\right) l_{l,y} \end{aligned} \tag{3-12}$$

Next, the right wing is incorporated in the model. The FBD, describing the right wing, is shown in Figure 3-6 and the projections in Figure 3-7. The drag force D_r of the right wing in the x-, y- and z-directions with $d_r = \left[d_{r,x}, d_{r,y}, d_{r,z}\right]^T$ the damping force coefficient and $V_{r,CoP} = \left[u_{r,CoP}, v_{r,CoP}, w_{r,CoP}\ \right]$ the velocity of the CoP of the right wing, is modelled as

$$D_l = -d_r f_r V_{r,CoP}. \tag{3-13}$$

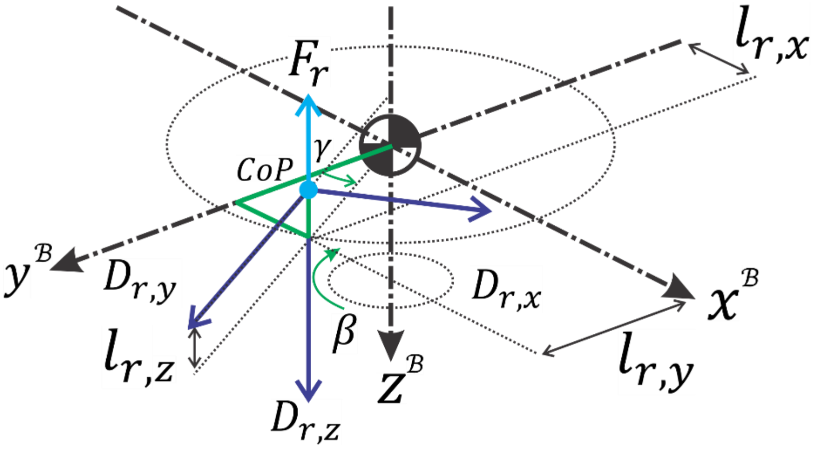

Figure 3-6: FBD of the Nimble for the right wing. The force acting on the CoP of the wing is decomposed in drag forces and the thrust force.

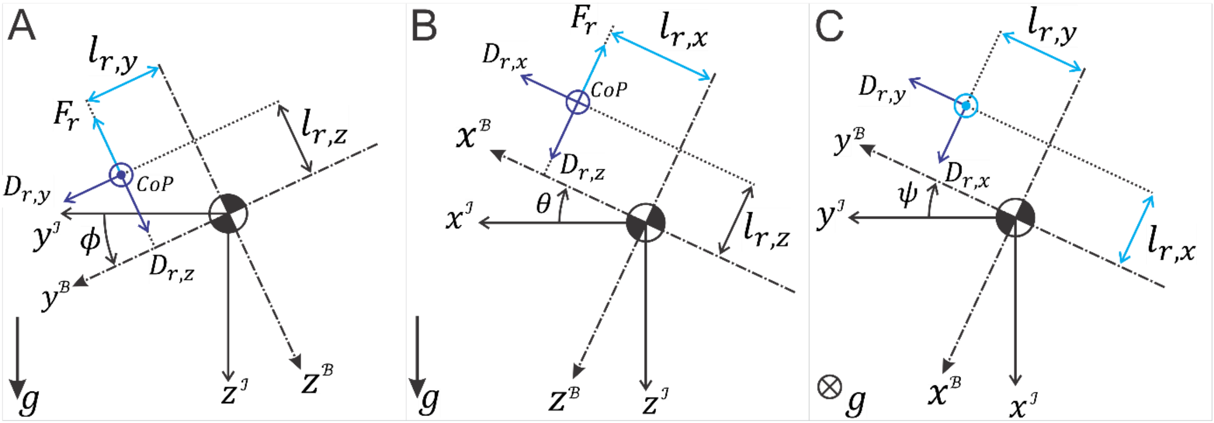

Figure 3-7: Projections for the right wing. The FBD is projected in (A) the yz-plane around the x-axis, (B) the zx-plane around the y-axis and (C) the xy-plane around the z-axis. The control inputs and the drag forces are depicted in light blue and dark blue respectively.

The velocities of the CoP of the right wing are expressed as

$$V_{r,CoP} = v_b + \dot{l}_r + \widetilde{\omega}_b l_r, \tag{3-14}$$

in which $l_r = \left[l_{r,x}, l_{r,y}, -l_{r,z}\right]^T$ are the horizontal, longitudinal and vertical CoP locations respectively of the right wing in body frame, which depend on the dihedral angle γ. Again, by combining Equations (3-13) and (3-14), the force and torques in the body frame for the right wing are given by Equations (3-15) and (3-16).

$$\begin{aligned} f'_{b,x} &= D_{r,x} = -d_{r,x} f_r\left(u - l_{r,z} q - l_{r,y} r + \dot{l}_{r,x}\right) \\ f'_{b,y} &= D_{r,y} = -d_{r,y} f_r\left(v + l_{r,x} r + l_{r,z} p + \dot{l}_{r,y}\right) \\ f'_{b,z} &= D_{r,z} - F_R(f_l) = -d_{r,z} f_r\left(w - l_{r,x} q + l_{r,y} p + \dot{l}_{r,z}\right) - F_R(f_r) \end{aligned} \tag{3-15}$$

$$\begin{aligned} &\tau'_{b,x} = D_{r,y} l_{r,z} + \left(D_{r,z} - F_R\left(f_r\right)\right) l_{r,y} \\ &= -d_{r,y} f_r\left(v + l_{r,x} r + l_{r,z} p + \dot{l}_{r,y}\right) l_{l,x} + \left(-d_{r,z} f_r\left(w - l_{r,x} q + l_{r,y} p + \dot{l}_{r,z}\right) - F_R(f_r)\right) l_{r,y} \\ &\tau'_{b,y} = -D_{r,x} l_{r,z} + \left(-D_{r,z} + F_R\left(f_r\right)\right) l_{r,x} \\ &= d_{r,x} f_r\left(u - l_{r,z} q - l_{r,y} r + \dot{l}_{r,x}\right) l_{r,z} + \left(d_{r,z} f_r\left(w - l_{r,x} q + l_{r,y} p + \dot{l}_{r,z}\right) + F_R(f_r)\right) l_{r,x} \\ &\tau'_{b,z} = D_{r,y} l_{r,x} - D_{r,x} l_{r,y} \\ &= -d_{r,y} f_r\left(v + l_{r,x} r + l_{r,z} p + \dot{l}_{r,y}\right) + d_{r,x} f_r\left(u - l_{r,z} q - l_{r,y} r + \dot{l}_{r,x}\right) l_{r,y} \end{aligned} \tag{3-16}$$

Simplifications can be made to reduce the number of parameters in the model. Length parameters on right and left side are dependent on the dihedral angle as in Equation (3-17) and the Nimble is symmetric. l_γ represents the length from the CoM to the CoP as,

$$\begin{cases} l_\gamma = \sqrt{l_x^2 + l_y^2} \\ l_x = s_\gamma l_w \\ l_y = c_\gamma l_w \end{cases} \tag{3-17}$$

Furthermore, the drag force coefficients of both CoPs and the length parameters of the left and right wing are the same due to symmetry of the wings as

$$\begin{cases} D = D_l = D_r \\ l = l_l = l_r \\ \dot{l} = \dot{l}_l = \dot{l}_r \end{cases} \tag{3-18}$$

In addition, aerodynamic damping can be considered for the body of the Nimble. The body damping is often neglected or modelled depending on the body its velocity (3-4). Notably, nonlinear damping forces have a passive nature. These damping forces can be written as

$$f_d = -D_b v_b^2, \tag{3-19}$$

with $D_b = \left[d_{b,x}, d_{b,y}, d_{b,z}\right]^T$ as the aerodynamic drag force coefficient encountered by the body in body frame, following from Equation (3-4).

Using the analogy of the wings flapping, i.e. the geometrical center of a wing pair on the same side always stays in the in a horizontal $x^{\mathcal{B}}y^{\mathcal{B}}$-plane, the parameter l_z is modelled as a constant and $\dot{l}_z = 0$. Combining Equations (3-11), (3-12), (3-15) and (3-16) the total forces and torques in the body frame for the right wing can be written as in Equations (3-20) and (3-21):

$$f_b(f_r, f_l, \gamma, \beta, X, \Phi) = \left[f_{b,x}\ f_{b,y}\ f_{b,z}\right]^T,$$

$$\begin{aligned} f'_{b,x} &= D_{r,x} + D_{l,x} = -d_x\left(f_r\left(u - l_z q - l_y r + \dot{l}_x\right) + f_l\left(u - l_z q + l_y r + \dot{l}_x\right)\right) - d_{b,x}\mathrm{u}^2, \\ f'_{b,y} &= D_{r,y} + D_{l,y} = -d_y\left(v + l_x r + l_z p + \dot{l}_y\right)(f_r + f_l) - d_{b,y}\mathrm{v}^2, \\ f'_{b,z} &= D_{r,z} + D_{l,z} - F_L(f_l) - F_R(f_l) - d_{b,z}\mathrm{w}^2 \\ &= -d_z\left(f_r\left(w - l_x q + l_y p\right) + f_l\left(w - l_x q - l_y p\right)\right) - F_L(f_l) - F_R(f_r) - d_{b,z}\mathrm{w}^2, \end{aligned} \tag{3-20}$$

$$\tau_b(f_r, f_l, \gamma, \beta, X, \Phi) = \left[\ \tau_{b,x}\ \tau_{b,y}\ \tau_{b,z}\right]^T,$$

$$\begin{aligned} \tau'_{b,x} &= -d_y l_z\left(v + l_x r + l_z p + \dot{l}_y\right)\left(f_r + f_l\right) \\ &\qquad + \left(-d_z\left(f_r\left(w - l_x q + l_y p\right) - f_l\left(w - l_x q - l_y p\right)\right) + F_L\left(f_l\right) - F_R\left(f_r\right)\right) l_y, \\ \tau'_{b,y} &= d_x l_z\left(f_r\left(u - l_z q - l_y r + \dot{l}_x\right) + f_l\left(u - l_z q + l_y r + \dot{l}_x\right)\right) \\ &\qquad + \left(d_z\left(f_r\left(w - l_x q + l_y p\right) + f_l\left(w - l_x q - l_y p\right)\right) + F_L\left(f_l\right) + F_R\left(f_r\right)\right) l_x, \\ \tau'_{b,z} &= -d_y(f_r + f_l)\left(v + l_x r + l_z p + \dot{l}_y\right) l_x \\ &\qquad - d_x(f_l\left(u - l_z q + l_y r + \dot{l}_x\right) - f_r\left(u - l_z q - l_y r + \dot{l}_x\right)\ ) l_y. \end{aligned} \tag{3-21}$$

3.3.3. Orientation of the wing frame

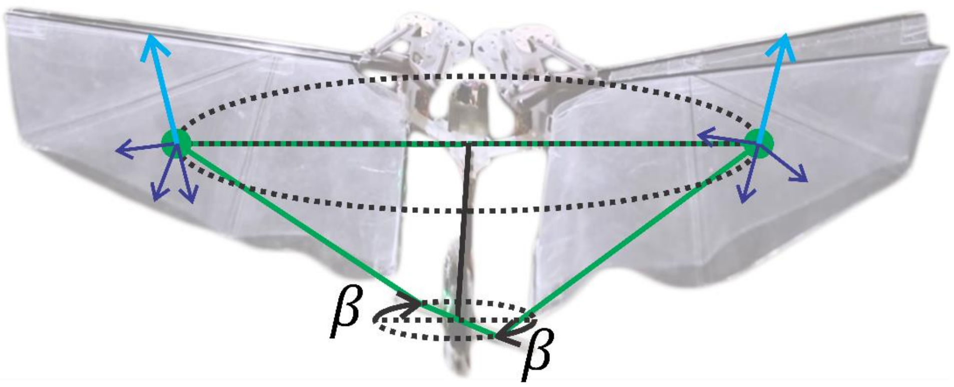

Figure 3-8: Concept of the aerodynamic model of the DelFly Nimble portrayed on the body. By actuating the wing root angle β, the thrust and damping vectors rotate with respect to each other. The blue and purple vectors represent the thrust and damping forces acting at the CoP, the green lines represent the variable length parameters, and the black lines define the paths on which the length parameters are constrained.

Until now, only the control inputs f_l, f_r and γ are accounted for. Figure 3-9 shows the effect of when the control inputs γ and β change with respect to each other. The orientation of the wing changes which affects the thrust and drag force directions. The orientation is portrayed on the body of the Nimble in Figure 3-8. For clarity, the force vector is only depicted for the left side. In the longitudinal model, when the angle γ is changing, the variables l_x and l_y change, but the orientation of the force vector remains the same. On the contrary, by varying the combination of the angles γ and β, the force vectors are rotated. As can be observed, there are two circles, one describing the upper orbit in which the CoP translates by the dihedral angle γ, and one describing the lower orbit in which the wing root shaft moves by the wing root angle β. The purple vectors describe the damping forces of the wing frame. The cyan line describes the thrust vector generated by the CoP.

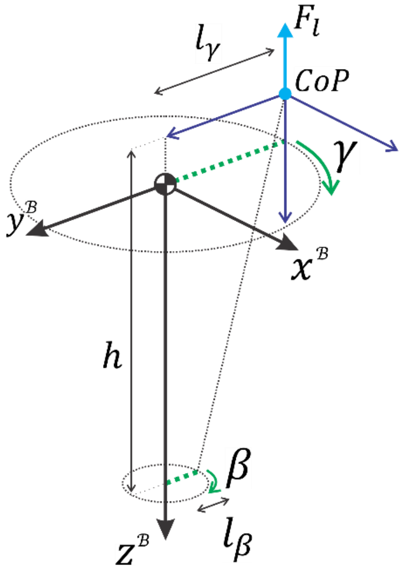

Figure 3-9: The force vectors rotate in the wing frame with respect to the body frame when the wing root angle β or dihedral angle γ change.

The rotation matrix of the thrust and damping vectors in wing frame with respect to the body frame is fully determined by choosing the control input angles γ and β and the length parameters $h,\ l_\gamma$ and l_β as in

$$\begin{aligned}
R'_L(\varsigma_L, \varrho_L, \gamma) &= \begin{bmatrix} c_{\varrho_L}c_\gamma & s_{\varsigma_L}s_{\varrho_L}c_\gamma - c_{\varsigma_L}s_\gamma & s_{\varsigma_L}s_\gamma + c_{\varsigma_L}s_{\varrho_L}c_\gamma \\ c_{\varrho_L}s_\gamma & s_{\varsigma_L}s_{\varrho_L}s_\gamma + c_{\varsigma_L}c_\gamma & c_{\varsigma_L}s_{\varrho_L}s_\gamma - s_{\varsigma_L}c_\gamma \\ -s_{\varrho_L} & s_{\varsigma_L}c_{\varrho_L} & c_{\varsigma_L}c_{\varrho_L} \end{bmatrix}, with \\
\varsigma_L &= -\tan^{-1}\left(\frac{\dfrac{l_\gamma - l_\beta\big(cos(\gamma - \beta)\big)}{h}}{cos\left(\tan^{-1}\left(\dfrac{l_\beta sin(\gamma - \beta)}{h}\right)\right)} \right), \\
\varrho_L &= -\tan^{-1}\left(\frac{l_\beta sin(\gamma - \beta)}{h}\right),
\end{aligned} \tag{3-22}$$

$$and$$

$$\begin{aligned}
R'_R(\varsigma_R, \varrho_R, -\gamma) &= \begin{bmatrix} c_{\varrho_L}c_\gamma & s_{\varsigma_L}s_{\varrho_L}c_\gamma - c_{\varsigma_L}s_\gamma & s_{\varsigma_L}s_\gamma + c_{\varsigma_L}s_{\varrho_L}c_\gamma \\ c_{\varrho_L}s_\gamma & s_{\varsigma_L}s_{\varrho_L}s_\gamma + c_{\varsigma_L}c_\gamma & c_{\varsigma_L}s_{\varrho_L}s_\gamma - s_{\varsigma_L}c_\gamma \\ -s_{\varrho_L} & s_{\varsigma_L}c_{\varrho_L} & c_{\varsigma_L}c_{\varrho_L} \end{bmatrix}, \\
& with \\
\varsigma_R &= \tan^{-1}\left(\frac{\dfrac{l_\gamma - l_\beta\big(cos(\gamma + \beta)\big)}{h}}{cos\left(\tan^{-1}\left(\dfrac{l_\beta sin(\gamma + \beta)}{h}\right)\right)} \right), \\
\varrho_R &= -\tan^{-1}\left(\frac{l_\beta sin(\gamma + \beta)}{h}\right).
\end{aligned} \tag{3-23}$$

The rotation is determined by rotating around the local x-, y-, and z-axes of the wing frame respectively, followed with a fixed rotation (ς_{L_s} and ς_{R_s}) to align the thrust vectors parallel to the $z^{\mathcal{B}}$-axis of the body frame in hover state as

$$\begin{aligned}
R_{L_s}\big(\varsigma_{L_s}, 0{,}0\big) &= \begin{bmatrix} c_{\varrho_{L_s}} & 0 & 0 \\ 0 & c_{\varsigma_{L_s}} & -s_{\varsigma_{L_s}} \\ 0 & s_{\varsigma_{L_s}} & c_{\varrho_{L_s}} \end{bmatrix}, with \\
\varsigma_{L_s}(\gamma, \beta) &= tan^{-1}\left(\frac{l_\gamma - l_\beta}{h}\right)\ and \\
R_{R_s}\big(\varsigma_{R_s}, 0{,}0\big) &= \begin{bmatrix} c_{\varrho_{R_s}} & 0 & 0 \\ 0 & c_{\varsigma_{R_s}} & -s_{\varsigma_{R_s}} \\ 0 & s_{\varsigma_{R_s}} & c_{\varrho_{R_s}} \end{bmatrix}, with \\
\varsigma_{R_s}(\gamma, \beta) &= -tan^{-1}\left(\frac{l_\gamma - l_\beta}{h}\right).
\end{aligned} \tag{3-24}$$

Finally, the resultant rotations of the left and right wings are given by

$$\begin{aligned}
R_L(\gamma, \beta) &= R'_L R_{L_s} \\
R_R(\gamma, \beta) &= R'_R R_{R_s}
\end{aligned} \tag{3-25}$$

The torques and forces, or wrenches, are co-vectors $\in se^{*}(3)$. The resulting wrenches are transformed by using the dual adjoint mapping [4] as

$$\begin{aligned} (W_b)^T &\equiv Ad^T_{H^b_{b'}} (W'_b)^T \\ \begin{bmatrix} \tau_b \\ f_b \end{bmatrix}^T &= \begin{bmatrix} R^b_{b'} & -R^b_{b'} \tilde{p}^b_{b'} \\ 0 & R^b_{b'} \end{bmatrix} \begin{bmatrix} \tau'_b \\ f'_b \end{bmatrix}^T \end{aligned} \tag{3-26}$$

3.4. Inverse aerodynamics

The control inputs of the Nimble are constrained due to physical limitations such as the operating range of the actuator or collision of the wings. The control inputs of the Nimble are constrained to

$$\begin{aligned} -25° \leq \gamma \ &\leq 25°, \\ -50° \leq \beta \ &\leq 50°\,, \\ 0 \leq f_L &\leq 22\ Hz, \\ 0 \leq f_R &\leq 22\ Hz. \end{aligned} \tag{3-27}$$

From the aerodynamic equations, the forces and torques in body frame can be computed when the linear and angular velocities in body frame and the control inputs of the Nimble are known. In order to control the Nimble, $\mathcal{V} = \left[f_{b_z}\, \tau_{b_x}\, \tau_{b_y}\, \tau_{b_z} \right]^T$ are chosen as generalized forces. The mapping from the generalized forces to the control inputs of the Nimble is described by

$$\begin{bmatrix} \gamma \\ \beta \\ f_l \\ f_r \end{bmatrix} = \mathcal{F}\left[f_{b_z}\, \tau_{b_x}\, \tau_{b_y}\, \tau_{b_z} \right]. \tag{3-28}$$

In order to compute the necessary control inputs of the Nimble when the generalized forces are known, $\mathcal{F}$ has to be full rank. In other words, the inverse of the aerodynamics has to be unique and invertible in the operating domain as given by Equation (3-27). The map $\mathcal{F}$ is a multivariate nonlinear function. Nevertheless, it turns out that the Jacobian $\mathcal{J}$ of the generalized forces $\mathcal{V}$ with respect to the control inputs $\mathcal{U} = [\gamma\ f_L\ f_R\ \beta]^T$ of the Nimble has full rank as

$$rank\big(\mathcal{J}(\gamma, f_L, f_R, \beta)\big) = rank\left(\frac{\partial \mathcal{V}}{\partial \mathcal{U}}\right) = 4, \qquad \forall \begin{bmatrix} \gamma \\ \beta \\ f_l \\ f_r \end{bmatrix}.$$

3.5. Concluding remarks of the aerodynamic model

In this section, a white-box dynamic model is proposed for the DelFly Nimble. The Nimble is modelled as a rigid body in $\mathbb{R}^6$, together with an aerodynamic model. The Nimble regulates its thrust along its body-fixed z-direction and regulates its torques around the three body-fixed axes. The translational and rotational motion of the Nimble has six degrees of freedom and four degrees of actuation. The rigid body dynamics capture the underactuated system, its intrinsic instability, coupling between the translational and rotational dynamics and the strong nonlinearities. The body forces and torques depend on the mean translational aerodynamic forces, which are applied on the centers of pressure of the wings and capture the FCFs and torques. Furthermore, the map between the generalized forces $\left[f_{b,z}\, \tau_{b,x}\, \tau_{b,y}\, \tau_{b,z}\right]^T$ and the control inputs $[\gamma\, f_l\, f_r\, \beta]^T$ is a multivariable nonlinear function and has full rank. The model parameters are summarized in Table 3-1.

Table 1: Parameters of the aerodynamic model.

Symbol	Description
l_γ	The length from the CoM straight to the CoP in the horizontal xy-plane
l_z	The length from the CoM straight to the CoP in the z-direction
l_β	Horizontal length from the CoM straight to the wing root rod attachment
h	Vertical length from the CoP to the wing root rod attachment
d_x, d_y, d_z	The damping coefficients of the drag forces
m	Mass of the Nimble
I	Mass moment of inertia of the Nimble
c_1, c_2	Coefficient used to map the flapping frequencies to the thrust force of the CoPs
$d_{b,x}, d_{b,y}, d_{b,z}$	Damping coefficients of the body

The translation of the wing frames is regulated by the control input γ. The total body forces and torques are expanded by the rotation matrices R_L (for the left wing) and R_R (for the right wing). The rotation of the wing frames is regulated by a combination of both control inputs γ and β. In spite of the fact that this approximates the rather complicated aerodynamics, this is an appealing way of modeling the flapping-cycle-averaged forces with relatively few variables, being computationally efficient and useful for insight in the dynamics and parameter changes of the DelFly Nimble. This may give insight in answering questions that cannot always be validated by experiments with animals or tethered robots.

The modeling choice is motivated by the use of the geometry and physical constraints of the Nimble. There are several differences by expanding the model from the longitudinal dynamics to the 3D dynamics. In contrast to the longitudinal model as described by Huo et al. (2015), f_b is rotated around the $x^{\mathcal{B}}$-, $y^{\mathcal{B}}$- and $z^{\mathcal{B}}$-axes, even when the control input β remains zero. Secondly, as described by Fei et al. (2019), the control input β can be modelled proportional to the yaw torque, as an individual rotation around the $y^{\mathcal{B}}$- axis. However, these individual rotation matrices are necessary, since the forces of the left and right rotation around the CoPs only cancel each other when the control input γ is zero. Moreover, according to the aerodynamic model, the mapping between β and τ_z is nonlinear and not proportional. In addition, the thrust vectors can be rotated in the body frame. This induces forces in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-direction as well. Regardless of the substantial variation in size, mass, wing kinematics and wing morphology among FWMAVs, they all undergo similar passive aerodynamic damping and therefore, this model could potentially also be used for other FWMAVs. This model can be used to gain insight in the physics and system dynamics, for optimization purposes or testing hypotheses for the Nimble. In the next section validation of the model will be demonstrated.

4. Model validation

Validation is an important aspect of the development of the dynamic model to ensure the model adequately represents the Nimble system. Therefore, a methodical framework is followed to verify the model [28]. According to this framework, validation schemes are proposed to validate the model structure by using tests. These tests are used to test the structure, suitability and consistency of a model. In this section, validation schemes, both quantitative and qualitative, are employed. The power of modeling techniques lies in validity, credibility and generality. In general, the simplest model is the most rigorous one, although no particular validity test will allow to assert validation of the dynamic model. The more tests pass, the higher the level of confidence will be. To simulate the dynamics of the Nimble, the Runge-Kutta 4 method (RK4) is used and simulated with MATLAB [29]. The truncation error of the RK4 method is $O(\Delta t^5)$ and a step size of $\Delta t = 1e - 3$ is chosen. The structure-verification and face validity of the dynamic model is discussed, at the levels of the rigid body dynamics in Section 4.1. In Section 4.2, the aerodynamics of the Nimble are discussed. First of all, tests are carried out by validating whether the right control inputs correspond with simple maneuvers. Secondly, the Jacobian of the aerodynamic model is used to show what happens as control inputs are varied. Thereafter, the forces and torques are discussed by varying the control inputs and body speeds. Lastly, some validation tests are summarized such as testing the dimensional-consistency, energy conservation, predicting the behavior, family member test and parameter estimation. The closed-loop simulations are shown in Chapter 6.

4.1. Verification and validity of the rigid body dynamics

The rigid body dynamics should produce the motion of the Nimble under the applied forces and torques, depending on the mass and inertia of the body. The rigid body dynamics are modelled in 3D and capture the nonlinear behavior and dynamic coupling between the attitude and translational dynamics. In simulations the rigid body dynamics are verified together with the aerodynamics in Chapter 6. In the derivation of the equations of motion, the following assumptions were used. The assumption that the Nimble is modelled as a rigid body is viable, since the flapping body has a bilateral symmetric structure. The flapping wings contribute less than 8% to the body mass and the force generation is represented by wingbeat-cycle-average aerodynamics. In other words, this requires the time scale of the wing motion to be much larger than the body motion and encompasses the largest eigenvalue to be much smaller than the circular frequency of the wingbeats. This is due to the fact that the accelerations in the upstroke and downstroke approximately cancel each other out in one flapping cycle. By assuming that the wing mass is negligible compared to the body weight, the equations of motion are the same as that of a rigid flying body. Moreover, it is likely that in the future the actuators will obtain higher performance and that the Nimble will reduce in size. Therefore, this assumption will become more applicable. Second, the wing state cannot be used for the control approach and multi-body representations would not be suitable for being used onboard. Not only is this a convenient way of modeling, but it might even be necessary to employ a trajectory controller. Third, elastic deformations or shocks to the vehicle are neglected and therefore the mass moments of inertia are time-invariant. Lastly, based on test simulations and based on the literature, the 2D rigid body dynamics are well captured [7].

4.2. Verification and validity of the aerodynamics

In this section tests are discussed for the aerodynamics. In Section 4.2.1 simulations validate that the standard control inputs result in the desired motion of the system. Thereafter, Section 4.2.2 elaborates on the Jacobian of the aerodynamic model. Lastly, Section 4.2.3 discusses the force torque generation when the control inputs are varied.

4.2.1. Hover, pitch, yaw and roll

The next aspect is the aerodynamic model of the Nimble. The aerodynamic model determines how much lift or drag is generated and determines the motion of the Nimble combined with the rigid body. Naturally, when the Nimble hovers in simulation, the dihedral and wing root angle are zero and the flapping frequency is around 15.47 Hz, like for the real system. Increasing or decreasing the dihedral angle gives a positive or negative pitch maneuver, respectively. Increasing or decreasing the flapping frequencies f_L and f_R accelerate or decelerate the vertical motion. This behavior is also observed in real life. When the Nimble pitches forward (negative γ), the nose points down and the Nimble starts increasing its longitudinal speed. Due to the speed, flapping counter forces increase and cause a nose up rotation. Increasing or decreasing $f_L$with respect to f_R results in a positive or negative roll maneuver. Lastly, increasing or decreasing β results in a negative or positive yaw maneuver. Simulations, shown in Section 6.2 and 6.3, validate this kind of simple maneuvers in closed loop.

Simulation shows that actuating the angle β, γ and the flapping frequencies f_L and f_R correspond to yaw, pitch, roll and vertical motion respectively (Appendix III.b). The aerodynamic model shows the nonlinear behavior, which results from the rotated CoPs and linear or angular speeds. These behaviors can be visualized by the 3D MATLAB simulation (Appendix III.c). Given the control inputs and length parameters of the Nimble, the visualization of the dynamic model is obtained. For example, in Figure 4-1, a simulation is given for $\gamma = 45°$ and $\beta = 0°$.

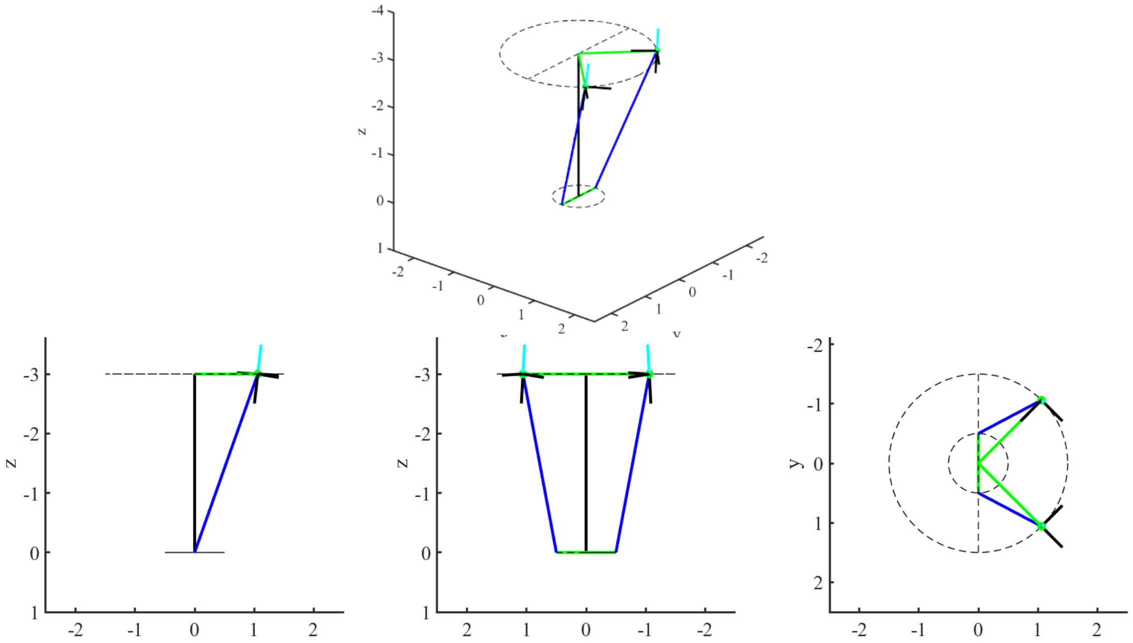

Figure 4-1: Aerodynamic model visualization for $\gamma = 45^\circ$ and $\beta = 0^\circ$.The green lines represent the variable length parameters (γ on top and β below), the black and cyan lines represent the damping and thrust forces of the CoPs, the dashed lines represent the constrained kinematics and the blue lines represent the constraint between the wing root and the CoPs.

4.2.2. Jacobian of the aerodynamic model

Another crucial part for the aerodynamic system is that the model should capture the FCFs. In several ways, the FCFs are observed from simulations. One way to do this is to investigate the Jacobian of the aerodynamic model. The Jacobian gives insight in the difference of operating regions of the aerodynamic model as the body speed and control input change. The Jacobian is given by $\mathcal{J}(\gamma, f_L, f_R, \beta) = \frac{\partial \nu}{\partial u} \in R^{4\times4}$. In Appendix III.a the Jacobian of individual cases show the FCFs are present. The higher the linear or angular velocity, the higher the FCFs.

One of these figures is depicted below, in Figure 4-2. The dihedral angle γ is varied while β remains zero. In the four subplots, the magnitude of the numeric values of $\mathcal{J}(\gamma, f_L, f_R, \beta)$ are shown. The magnitudes are normalized with respect to the maximum attainable control input value, in order to put the graphs into perspective. Figure 4-2 illustrates that $f_{b,z}$ is mainly coupled with the flapping frequencies and just a small portion of the control input angle γ. As the angle γ increases, $f_{b,z}$ decreases by slightly rotating the thrust. $\tau_{b,x}$ is mainly coupled with the flapping frequencies and a small portion of the control input angle γ. By increasing the flapping frequencies f_L and f_R increase and decrease the roll torque respectively with the same magnitude. As the thrust vectors are being rotated by γ, the magnitude of the flapping frequencies decreases. $\tau_{b,y}$ is coupled with the flapping frequencies and angle γ. The flapping frequencies are similar and have more impact the when the angle γ is larger. Lastly, $\tau_{b,z}$ is mainly coupled with the flapping frequencies and angle β, although the values are in the order of 10^{-3}. Again as with the roll torque the flapping frequencies have equal magnitude, but opposite in sign. More graphs are shown in Appendix III.a, where insight is obtained by varying other control inputs and body speeds. The summary of these results is given in Table 4-1.

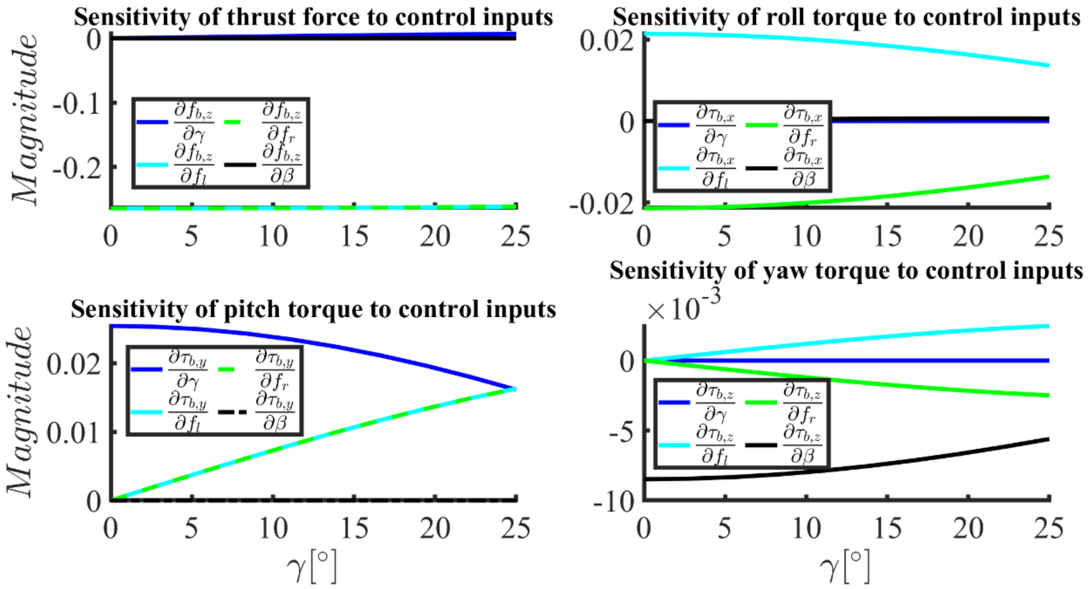

Figure 4-2: Sensitivity of $J(\gamma, f_L, f_R, \beta)$ by varying the control input angle γ in the range of $[0\ 25°]$ while $\beta = 0°$. The frequencies are held constant with a value of $15\ Hz$. The Jacobian values are multiplied with the maximum attainable control input values.

Table 4-1: Sensitivity of $J(\gamma, f_L, f_R, \beta)$ by varying the control inputs.

Sensitivity of $J(\gamma, f_L, f_R, \beta)$ by varying the control input	**Expected outcome**	**Actual outcome**	**Match**
angle γ *in the range of $[0\ 25°]$ while $\beta = 0°$ and $f_L = f_R = 15Hz$.* *(Figure III-1)*	*While providing thrust, this will induce an increasing positive pitch torque by tilting the thrust forces.*	*Next to the expected outcome the increase of γ induces* *1. a slightly increasing positive thrust force,* *2. decreasing roll and yaw torques caused by f_L and f_R with opposite signs,* *3. a slightly decreasing positive roll torque by γ,* *4. a decreasing pitch torque by γ and increasing pitch torque by $f_L$and f_R and* *5. a decreasing negative yaw torque by β.*	✓
angle β *in the range of $[0\ 50°]$ while $\gamma = 0°$ and $f_L = f_R = 15Hz$.* *(Figure III-2)*	*While providing thrust, this will induce an increasing negative yaw torque by tilting the thrust forces.*	*Next to the expected outcome the increase of β induces* *1. a slightly increasing positive thrust force by β,* *2. a positive and a negative decreasing roll torque caused by f_L and f_R respectively,* *3. a slightly increasing positive roll torque by γ,* *4. a slightly decreasing pitch torque by γ,* *5. a decreasing negative yaw torque by β and* *6. an increasing negative yaw torque by f_L and f_R.*	✓
angles γ and β *from 0% $(\gamma = 0°, \beta = 0°)$ to 100% $(\gamma = 25°, \beta = 50°)$ simultaneously while $f_L = f_R = 15Hz$.* *(Figure III-3)*	*While providing thrust, this will induce an increasing positive combined pitch and negative yaw torque by tilting the thrust forces.*	*Next to the expected outcome the increase of γ and β induce* *1. a slightly increasing positive thrust force by γ and β,* *2. a slightly decreasing negative thrust force by f_L and f_R,* *3. a decreasing roll torque caused by f_L and f_R with opposite signs,* *3. a slightly increasing positive roll torque by γ and β,* *4. an increasing positive pitch torque by f_L and f_R,* *5. a slightly decreasing pitch torque by γ and β,* *5. a decreasing negative yaw torque by β and increasing positive yaw torque by γ and* *6. an increasing negative yaw torque by f_L and f_R (although with different magnitudes due to the combination of increasing γ and β).*	✓

4.2.3. Force and torque generation

In Appendix III.b, the forces and torque outputs are shown by varying the control inputs of the Nimble. In Figure 4-3, the graphs are obtained by varying the angle γ. These graphs give insight into what extent the nonlinearities take hold of the system. For this figure, the velocities are equal to zero and the flapping frequencies of the left and right wings are $20\,Hz$. The higher the magnitude of the angle γ, the higher the generated pitch torque. Due to the rotation of the wings, the thrust force becomes less when γ increases and causes a force in the x-direction as well.

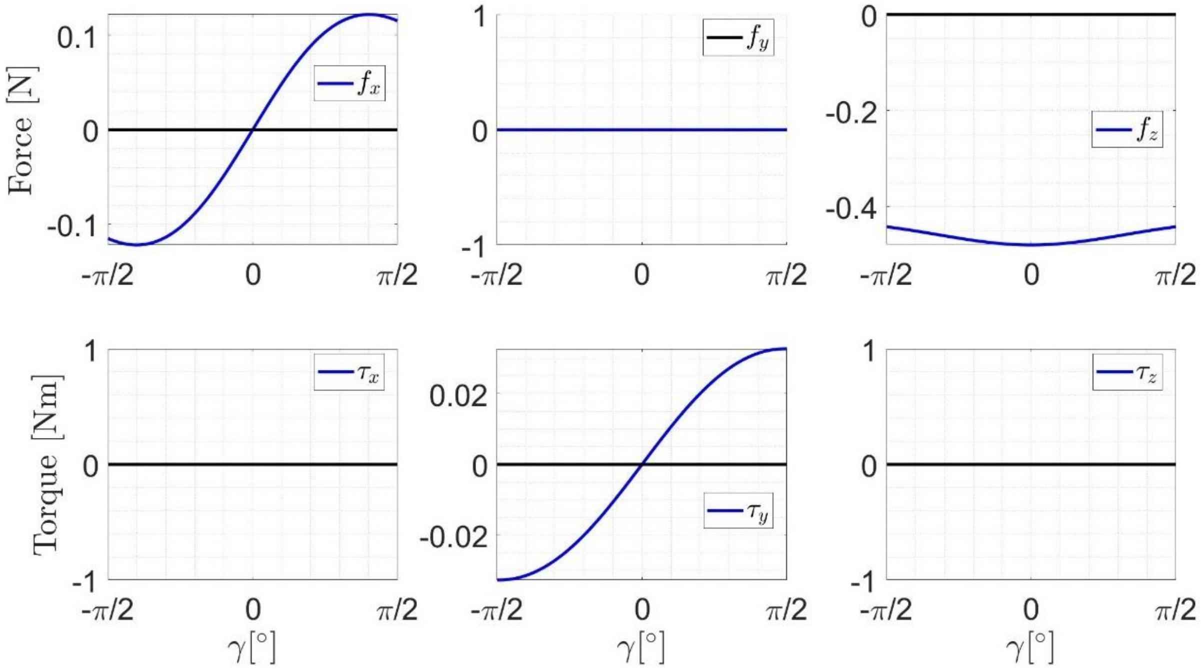

Figure 4-3: Varying γ in the range of $[-90°\ 90°]$ in order to generate pitch motion.

More graphs are shown in Appendix III.b, where control inputs and body speeds are varied. One of the outcomes of these figures is that when there is no velocity, the control input β only affects the yaw torque. On the other hand, when the Nimble experiences a forward velocity, pitch and roll forces are generated as well when the angle β changes. In [7], a maximum longitudinal velocity of $7\,m/s$ was obtained. Thereafter the Nimble started to oscillate. According to the aerodynamic model, a counter flapping force of $1\,N$ is encountered at $7\,m/s$ (Figure III-13). Logically and in simulation, this generates a positive pitch torque, which then has to be compensated and results in an unstable oscillating behavior. A summary of Appendix III.b is given in Table 4-2.

Table 4-2: Dynamical behavior by varying the control inputs.

Varying control input	***Outcome***	***Considered a match***
$\boldsymbol{\gamma}$ *in the range of* $[-90° \, 90°]$ *in order to generate pitch motion with* $f_L = f_R = 20 \, Hz$ *and* $\beta = 0°$ *(Figure III-10).*	*Increasing* γ *from* 0 *to* $90°$ *induces* *1. a positive force in the* $x^{\mathcal{B}}$*-direction,* *2. a decrease in thrust force and* *3. a positive pitch torque.* *Decreasing* γ *from* 0 *to* $-90°$ *induces* *1. a negative force in the* $x^{\mathcal{B}}$*-direction,* *2. a decrease in thrust force and* *3. a negative pitch torque.*	✓
$\boldsymbol{f_L}$ ***and*** $\boldsymbol{f_R}$ *in the range of* $[15 - 22Hz]$ *and* $[22 - 15Hz]$ *respectively with respect to each other to generate roll motion with* $\gamma = \beta = 0°$ *(Figure III-11).*	*The increase of* f_L *in contrast to* f_R *induces* *1. a positive roll torque and* *2. a negative thrust force.* *The increase of* f_R *in contrast to* f_L *induces* *1. a negative roll torque and* *2. a negative thrust force.*	✓
$\boldsymbol{f_L}$ ***and*** $\boldsymbol{f_R}$ *in the range of* $[0 - 22Hz]$ *in order to generate thrust and vertical motion*	*A positive increase of* f_L *and* f_R *simultaneously induces* *1. an increase in the thrust force.* *Due to the mapping between the flapping frequencies and the thrust force, the thrust force becomes positive when* $f < 3.27Hz$*.* *Physically, below* $f = 10 \, Hz$*, the thrust force behaves not linearly anymore as increasingly less thrust will be generated.*	✓
$\boldsymbol{\beta}$ *in the range of* $[0 - 90°]$ *in order to generate yaw motion (Figure III-13).*	*A positive increase of* β *induces* *1. a decrease in thrust force and* *2. a negative yaw torque.* *A negative increase of* β *induces* *1. a decrease in thrust force and* *2. a positive yaw torque.*	✓
$\boldsymbol{\beta}$ *in the range of* $[0 - 90°]$ *while* $\gamma = 25°$ *(Figure III-14).*	*A positive increase of* β *induces* *1. a decrease of the positive force* $f_{b,x}$*,* *2. a decrease of the force* $f_{b,y}$*,* *3. a decreasing negative thrust force,* *4. a small negative roll torque (due to the asymmetry of the wing root angle),* *5. a decreasing positive pitch torque and* *6. a negative yaw torque.* *A negative increase of* β *induces* *1. a decrease of the positive force* $f_{b,x}$*,* *2. an increase of the force* $f_{b,y}$*,* *3. a decreasing negative* $f_{b,z}$*,* *4. a small positive roll torque (due to the asymmetry of the wing root angle),* *5. a decreasing positive pitch torque and* *6. a positive yaw torque.*	✓

Lastly, in Appendix III.a several numerical outcomes for different FCFs are shown for the longitudinal, lateral and vertical directions. As expected, a longitudinal linear or angular velocity causes a FCF along the $x^{\mathcal{B}}$-axis and a FCT around the $y^{\mathcal{B}}$-axis. A lateral linear or angular velocity causes a FCF along the $y^{\mathcal{B}}$-axis and a FCT around the $x^{\mathcal{B}}$-axis. A vertical linear or angular velocity causes a FCF along the $z^{\mathcal{B}}$-axis and a FCT around the $z^{\mathcal{B}}$-axis. When $\beta = 0$, no FCT is observed as the FCF has no moment arm with respect to the CoM. Lastly, when the Nimble yaws, a FCT is caused around the $z^{\mathcal{B}}$-axis.

Furthermore, it is important to validate that the model makes sense when it is subjected to extreme but possible values. Even the conditions which have never been observed by the real system should result in an expected fashion. Even though the control inputs have physical constraints, the dynamic model is able to compute outputs outside the operating region, which can be observed in Appendix III.b. Furthermore, due to the rotation of the CoPs, the thrust force has its highest magnitude when $\gamma = \beta = 0°$. The highest torques are obtained by the extremities of the control inputs. In Table 4-3 some other tests are shown that have been explored.

Table 4-3: Validation tests.

Validation test	Description
Dimensional-Consistency	The dimensions of the variables are balancing each other.
Energy Conservance	No power is lost or created in the model due to actuation or damping, the dynamics show respective behavior. The energy of the system is conserved.
Behavior-Prediction Test	The model generated qualitatively expected patterns.
Family Member Test	This model is for a general class of FWMAVs. The general theory is depicted in the structure. By choosing different parameters, other situations can be applied as well. Moreover, this model is an extension from the 2D longitudinal dynamics. This model could for example also be applied for the 2D lateral dynamics.
Parameter estimation and comparison to measurements (see Appendix I)	Parameter estimation is used for estimating parameters for the lateral dynamics. The simulation results are compared with real life measurements and showed good results with high correlations and low RMSE. The estimated parameters match the estimated parameters of the longitudinal dynamics [7].

5. Nonlinear Control design

In order to let the Nimble fly autonomously, a trajectory control design is proposed. A trajectory control design is desirable in order to carry out accurate measurements and perform aerobatic maneuvers, which can be used for research, demonstrations and recreation. At the present moment, the Nimble has a PD controller that stabilizes the attitude. By using this controller, the Nimble can be steered manually by using a joystick [8], [10]. This joystick assigns set points for the PD control in order to carry out maneuvers. In previous work, manually tuned PD-controllers for controlling the orientation of the Nimble encountered problems [7], [30]. Since the flight dynamics are highly nonlinear, gain scheduling schemes would have to be used for the full flight envelope when using linear controllers. The maximum longitudinal and lateral speeds at full power are 7 m/s and 4 m/s respectively. After that, the Nimble starts to oscillate and destabilizes. However, since a dynamic model is investigated, model-based control techniques can be applied. Besides the attitude controller, a trajectory controller has not yet been developed.

Consequently, a nonlinear model-based attitude control law and a trajectory control design are proposed in this section. Aerobatic complex flight maneuvers can be achieved by using this hybrid control design and defining a concatenation of different reference commands. The main interesting aerobatic maneuvers are recovering from an upside-down orientation (Section 6.2), a $360°$ flip (Section 6.4), and following an elliptic helix trajectory (Section 6.5). The nonlinear controller design enables the Nimble to perform these maneuvers and to switch between tracking the attitude, position or velocity to perform complex aerobatic maneuvers. The controller is robust when switching the reference commands due to its global stability properties.

The overall control structure that is inspired by quadcopter control is explained in Section 5.1. Employing the geometric-based attitude controller defined in Section 5.2, the Nimble exhibits almost global exponential stability when the initial attitude error is less than $180°$. Employing the geometric-based attitude controller defined in Section 5.3, the Nimble exhibits exponential stability when the initial attitude error is less than $90°$, and yields almost global exponentially attractiveness when the initial attitude error is less than $180°$. Finally, Section 5.4 elaborates on the control gain parameters.

5.1. Control strategy for the Nimble

The Nimble is similar to the quadcopter, in the sense that they both are flying 6DoF underactuated systems with four control inputs and are dynamically coupled between the translational and rotational dynamics. Various control approaches have been carried out in the past decades. They are heavily researched for many applications due to their outstanding capabilities, among which: inspection and mapping, search and rescue, disaster operations, environmental monitoring, package delivery, surveillance, and cooperative manipulation. As sensors and microcomputer technology become more advanced, small MAVs become a reality. Small sizes result in high maneuverability, but pose significant challenges. Sensors become less accurate and environmental effects have more impact. There are some important design choices. In Section 5.1.1 parameterization choices for the orientation are elaborated. In Section 5.1.2 different control strategies are discussed. Finally, in Section 5.1.3 a suitable geometric control framework is introduced.

5.1.1. Parameterization of the orientation

Several nonlinear trajectory control models are investigated for quadcopters in [31]–[40]. The control of rigid bodies is motivated by attitude maneuvers and stabilization for the Nimble. The parameterization of the orientation differs between various studies which are Euclidean parameterizations such as Euler angles, lying in $\mathbb{R}^3$, and non-Euclidean parametrizations such as quaternions, lying in $\mathbb{S}^3$. An issue in controller designs for UAVs is that they are based on local coordinates. This gives rise to complicated expressions and singularities in representing the orientation of the body and therefore causes restrictions to achieve aerobatic maneuvers. Thus, these parameterizations have flaws to represent the orientation of a rigid body uniquely or globally.

Euler angles have a kinematic singularity, referred to as Gimbal Lock [34]. Using this parameterization, continuous control laws are only able to locally define attitude maneuvers [34]. Moreover, ambiguity is introduced since there are 24 possible sets of Euler angles with different conventions. Another drawback is that the integration of the Euler angles' time derivation is not globally defined. Moreover, with large angular errors, the Euler angles are not a good metric to define the orientation error [41].

Furthermore, quaternions are able to adopt every attitude and attitude rate, only every physical attitude is represented by a pair of antipodal unit quaternions. To be more specific, the attitude $R \in SO(3)$ is constituted by a pair of antipodal quaternions $\pm q \in SO(3)$. Therefore, it cannot be guaranteed that almost global asymptotic stability of the closed-loop vector field on $\mathbb{S}^3 \times \mathbb{R}^3$ to the space $SO(3) \times \mathbb{R}^3$ holds. Another problem which may arise by using quaternions is unwinding. This is a phenomenon that for particular starting states, the trajectories of the closed-loop system begin near the demanded orientation equilibrium and nevertheless travel a wide distance before returning to the demanded orientation. This may result in unnecessary cost of control effort.

It turns out that by using solely orthogonal rotation matrices, the attitude representation is both globally and uniquely described. In order to be able to track aerobatic maneuvers, in which the system evolves in $SE(3)$, parametrization problems are waived and topological obstructions can be taken into account by design [31]–[40].

5.1.2. Linear and nonlinear control techniques

Like the Nimble, quadrotors have similar system dynamics. These unmanned aerial vehicles (UAVs) are underactuated and regulate their thrust vector in order to translate through space. Autonomous operation of UAVs is feasible by on-board stabilization and trajectory tracking controllers. There is vast literature on trajectory control approaches for quadrotors that have been carried out in the past decades. A variety of control laws have been proposed to control the position and attitude dynamics among PID [42]–[44], LQR [45]–[47], feedback linearization [48], [49], backstepping [50]–[53], sliding mode [54]–[56] and adaptive [57], [58], fuzzy and geometric control methods of UAVs [31]–[40].

Many previous works have demonstrated linear control techniques, linearizing the dynamics around hovering. Linear control strategies close to equilibria are generally applied to avoid complexity that comes with modeling and controlling the nonlinear system. Employment of local control designs would not be suitable for nonlinear plants such as the Nimble, due to that linear control limits the system to small angle movements and employ the global operating region. Linear methods can only be applied for small angular maneuvers. Often the conclusion is drawn that the linear approach sufficiently provides for trajectory tracking. Nevertheless, in these papers, methods show small regions of the total configuration space.

Altogether, aggressive maneuvers cannot be obtained by using linear control methods and hence nonlinear control approaches are investigated. Wider flight envelopes, higher performance of stability and accuracy are achieved by the employment of nonlinear control techniques. Nonlinear systems governed by nonlinear differential equations are capable of representing the dynamics in a more realistic manner. In spite of the substantial undivided attention of UAVs, there is little regard for designing nonlinear control systems capable of complex aerobatic maneuvers. Moreover, the coupling between the rotational and translational dynamics is sometimes not considered. Finally, several studies fall short regarding analytic stability analysis and hence fail to assure the performance of the closed-loop system [59].

Nonlinear Geometric Control, as explicated in [36],[60], can be utilized for dynamic and control systems evolving over nonlinear manifolds that cannot be globally expressed with Euclidean spaces [34]. Nonlinear Geometric Control is a nonlinear model-based control method established on concepts of differential geometry. Rigid body dynamics are fully actuated by employing Lie groups, achieving almost global asymptotic stability. This representation also does not need reinterpretation of the attitude when implementing open-loop maneuvers or continuous feedback control [34]. Opposed to local attitude controllers, global attitude controllers emerge when arbitrary changes in the rigid body attitude and angular velocity are allowed. This means that in advance, there are no confinements on the potential rotational motion, which gives rise to nonlinear methods to address these rotational motions. These nonlinear approaches have to be interpreted on configuration space $SO(3)$ of the Nimble. For attitude estimation of rigid bodies, the employment of $SO(3)$ is simpler in representation and derivation, computationally efficient and it shows better tracking convergence than other parameterizations in the family of Gaussian filters [61]. Not to mention, since the attitude evolves over $SO(3) \times \mathbb{R}^3$, geometrical numerical integration schemes warrant that the computations lie in the appropriate manifold and can therefore be applied without the use of projection or normalization methods [34].

5.1.3. Geometric trajectory tracking control design

The position control loop requires the analysis of both the translational and rotational dynamics. The rotational dynamics of the Nimble are fully actuated, although the translational dynamics are underactuated. The rotational and translational motion of the Nimble has six DoF, but has solely four control inputs. Therefore, at most four outputs can be tracked. The attitude control loop can be used independently of the position control loop. For the attitude control loop, the outputs are the attitude of the Nimble as shown in Figure 5-1. Given a desired attitude R_d and angular rotation $\omega_{b,d}$, the nonlinear attitude controller computes the necessary torque $\tau_{b,v}$ in order to reach the desired attitude.

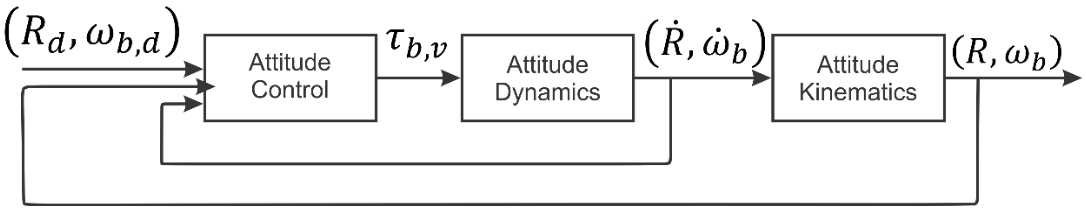

Figure 5-1: Block diagram of the tracking control structure for the attitude loop.

The structure of the position tracking control problem is split into an outer loop, the position control loop and an inner loop, the attitude control loop as is shown in Figure 5-2. The Nimble follows a quadcopter model, whereby longitudinal flight motion is controlled by pitching the body and lateral motion is achieved by rolling its body. Hereby, the rigid body regulates the thrust force along the body-fixed $z^{\mathcal{B}}$-axis and the attitude control torque along the three body-fixed axes. The reference trajectory, consisting of the desired positions p_d, yaw angles x_d^{β} and their derivatives, generate the control rotation matrix R_c and virtual thrust force $f_{b,v}$ in order to track the desired trajectory. This is a common approach, since the rotational motion affects the translational motion, but not vice versa [35]. Note that the desired rotation matrix R_d is chosen by the operator when the attitude controller is being used as a stand-alone controller. The control rotation matrix R_c follows from the position controller in which only $x_d^{\mathcal{B}}$ is chosen by the operator. The position control loop can be used as a velocity control loop as well, in order to asymptotically track the desired time dependent velocity vector of the CoM of the Nimble.

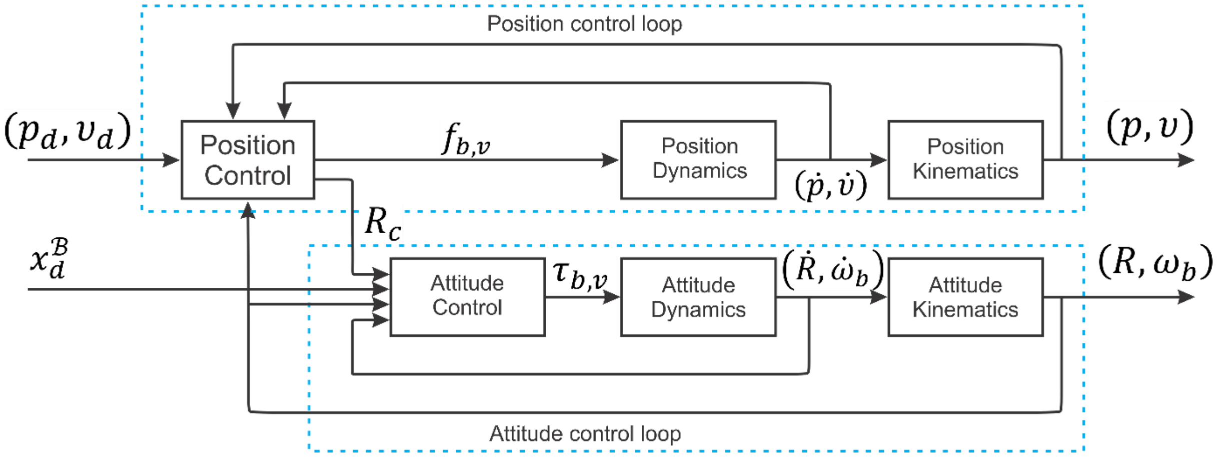

Figure 5-2: Block diagram of the nonlinear tracking control structure.

5.1.4. Closed-loop system with aerodynamics

The nonlinear controller generates the desired virtual inputs thrust force $f_{b,v}$ and torques $\tau_{b,v}$ that the Nimble needs to produce. From here the desired control input commands can be computed by using the aerodynamic mapping $\mathcal{F}$ as

$$\begin{bmatrix} \gamma \\ \beta \\ f_l \\ f_r \end{bmatrix} = \mathcal{F}\left[f_{b_z}\ \tau_{b_x}\ \tau_{b_y}\ \tau_{b_z}\right], \tag{5-1}$$

where it is assumed that the control input angles and frequencies are directly controlled (see Section 3.3 and Section 3.4). The inverse aerodynamic mapping $\mathcal{F}^{-1}$ is unique and therefore for given virtual inputs, the control inputs can be computed. The control inputs can be applied by the Nimble through its motors by using the inverse aerodynamic model. This capacitates the Nimble to perform the desired aerobatic maneuvers and other trajectory commands.

This mapping in Equation (5-1) introduces restrictions on the domain of achievable forces and torques, since the inputs are bounded by

$$\begin{aligned} -25° \leq \gamma\ &\leq 25°, \\ -50° \leq \beta\ &\leq 50°, \\ 0 \leq f_L &\leq 22\ Hz, \\ 0 \leq f_R &\leq 22\ Hz. \end{aligned} \tag{5-2}$$

The control design is unaware of these physical constraints and therefore this should be taken into account when designing trajectories.

5.2. Controller design for the attitude loop

The aim of the attitude controller is to rotate the rigid body from the initial orientation and angular velocity to the desired orientation and angular velocity in a finite time period. The desired attitude is generated by the position trajectory controller or assumed to be given by the joystick it is currently piloted by. Among Lee et al. discusses control of complex flight maneuvers for quadrotor UAVs [32]. The control architecture and proofs, which is used for quadcopters, are inspired on this paper and applied for the DelFly Nimble. The nonlinear controller for the attitude loop achieves almost global exponential stability and can also be used to achieve asymptotic tracking of the position closed loop in Section 5.3. This is a major improvement with respect to the linear PD controller that is presently used due to the following reasons. Firstly, the linear PD controller has not a stability analysis to demonstrate the performance and stability region of the controller. Secondly, the nonlinear attitude controller allows for almost a global region of attraction. Linear methods can only be applied for small angular maneuvers. Aggressive maneuvers cannot be obtained by using linear control methods and hence nonlinear control approaches need to be used. Lastly, the model-based nonlinear attitude controller can provide attractive features such as faster convergence, enhanced accuracy and reduced control effort.

An illustration of the closed-loop system for the attitude tracking block diagram is shown in Figure 5-1. In this section the rotational equations of motion (see Section 3.2) are considered as

$$\begin{gathered} J\dot{\omega}_b + J\omega_b \times \omega_b = \tau_{b,v}, \\ \dot{R} = R\tilde{\omega}_b, \end{gathered} \tag{5-3}$$

in which the virtual control input $\tau_{b,v}$ is chosen as the torque of the rigid body, J is the inertia matrix in body frame, ω_b and $\omega_{b,d}$ are the current and desired angular velocity in body frame $\mathcal{B}$, and R and R_d are the current and desired rotation matrix from body frame $\mathcal{B}$ to inertial frame $\mathcal{I}$. The tilde map $\widetilde{(\cdot)}$ represents the cross product. Note that in the general case for long durations it is important that the rotation matrices remain orthonormal when integrating the equations. Therefore, the goal of the attitude controller is to select a feedback function $\tau_{b,v}: SO(3) \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that asymptotically stabilizes a desired attitude equilibrium. R both globally and uniquely describes the parameterization of the Nimble its orientation. By introducing the desired attitude of the Nimble, $R_d \in SO(3)$, the control objective can be characterized as rotating the Nimble from $R(t) \rightarrow R_d(t)$ as $t \rightarrow \infty$. In this section a nonlinear attitude controller is presented in which the Nimble exhibits exponential stability when the initial attitude error is less than $180°$. In addition, in Appendix II the linearized attitude dynamics with nonlinear input are analyzed. The linear analysis is inconclusive, since there are eigenvalues with value zero. Therefore, a nonlinear analysis is needed to conclude about stability.

The attitude errors that are associated with the attitude error dynamics are formulated in Section 5.2.1. Subsequently, in Section 5.2.2, the nonlinear attitude control input is presented. Thereafter, in order to prove almost global exponential stability of the attitude error dynamics, that is for all angles except $180°$, a Lyapunov analysis is carried out in Section 5.2.3. Finally, next to the presented attitude tracking, altitude tracking is discussed in Section 5.2.4.

5.2.1. Attitude error dynamics

In this section attitude error functions are presented that are associated with the attitude error dynamics. The trajectory tracking control problem demands tracking errors, a measure of the difference between the current states and the desired states. Moreover, the error dynamics that evolve on nonlinear manifolds are necessary for defining the nonlinear attitude control input. Properties that follow will eventually be used for the Lyapunov analysis for the attitude control loop. Given an arbitrary smooth desired attitude tracking command $R_d(t) \in SO(3)$, the angular velocity command for the Nimble is represented by

$$\tilde{\omega}_{b,d} = R_d^T \dot{R}_d. \tag{5-4}$$

The desired reference attitude can be tracked by introducing an error function $e_R(R_d, R)$. One cannot simply take the difference of two rotation matrices, since this would not yield another orthogonal rotation matrix. The attitude error should give a measure of the magnitude required to go from the current orientation to the desired orientation. The attitude error is defined as $R_d^T R$, which is a rotation matrix by itself. An attitude error function on $SO(3)$ is defined as

$$\Psi(\mathrm{R}, \mathrm{R_d}) = \frac{1}{2} tr\left[I - R_d^T R\right]. \tag{5-5}$$

The attitude error function $\Psi(\mathrm{R}, \mathrm{R_d})$ is locally positive definite about $R = R_d$ within the region in which the rotation angle between R and R_d is lower than $180°$ or $\Psi \leq \Psi_2 < 2$, nearly covering $SO(3)$ [36]. The attitude error function Ψ represents the Euclidean distance of $R_d^T R$. Using the Rodriguez formula, the error function Ψ can be written as $1 - \cos\vartheta$, where ϑ represents the axis-angle between R_d and R [36]. The axis-angle parameterization uses a unit vector, which indicates the direction of the axis of rotation, and the axis-angle ϑ, which indicates the magnitude of the rotation around the axis of rotation.

Likewise to the attitudes R and R_d, $\dot{R}$ and $\dot{R}_d$ cannot be simply compared with each other, since they lie in distinct tangent space [32]. $\dot{R}_d$ can be transformed to the same tangent space and compared with $\dot{R}$ as

$$\begin{aligned} \dot{R} - \dot{R}_d\left(R_d^T R\right) &= R\left(\omega_b - \widetilde{R^T R}_d \omega_{b,d}\right), \\ &= R\tilde{e}_{\omega_b}, \end{aligned} \tag{5-6}$$

where the tracking error vector for the angular velocity $e_{\omega_b} \in \mathbb{R}^3$ is selected as

$$e_{\omega_b} = \omega_b - R^T R_d \omega_{b_d} \in \mathbb{R}^3. \tag{5-7}$$

e_{ω_b} can be interpreted as a vector of the relative rotation matrix $R_d^T R$ [32]. In more detail, e_{ω_b} is the angular velocity, in body frame, of $R_d^T R$, since [62]

$$\frac{d\left(R_d^T R\right)}{dt} = \left(R_d^T R\right) e_{\omega_b}. \tag{5-8}$$

The angular error function can be derived by differentiating $\Psi(\mathrm{R}, \mathrm{R_d})$ as shown in (5-9). The derivative of the Ψ is given by:

$$\dot{\Psi} = e_R e_{\omega_b}, \tag{5-9}$$

(see [62]) in which the tracking error vector of the attitude is selected as

$$e_R = \frac{1}{2}\left(R_d^T R - R^T R_d\right)^{\vee} \in \mathbb{R}^3. \tag{5-10}$$

The *vee map* $(\cdot)^{\vee}$ is the reverse operation of the tilde map $\widetilde{(\cdot)}$: $\mathfrak{so}(3) \to \mathbb{R}^3$. By using the interpretation of the Rodriguez formula, the tracking error of the attitude e_R represents $\sin\vartheta$ [62]. The derivative of e_R can be rewritten as [62]

$$\begin{aligned} \dot{e}_R &= \frac{1}{2}\left(R_d^T R \tilde{e}_{\omega_b} + \tilde{e}_{\omega_b} R^T R_d\right)^{\vee}, \\ &= \frac{1}{2}(tr[R^T R_d] - R^T R_d) e_{\omega_b} \equiv C\left(R_d^T R\right) e_{\omega_b}. \end{aligned} \tag{5-11}$$

Furthermore, it can be shown that $\left\|C\left(R_d^T R\right)\right\|_2 \leq 1$ for all $R_d^T R \in SO(3)$ [32]. Therefore

$$\|\dot{e}_R\| \leq \left\|e_{\omega_b}\right\|, \tag{5-12}$$

guaranteeing that $\dot{e}_R$ is bounded whenever e_{ω_b} is bounded $\forall\, R_d^T R \in SO(3)$. By using $\dot{R}_d = R_d \widetilde{\omega}_{b,d}$, $\dot{R} = R\widetilde{\omega}_b$ and that $\widetilde{\omega}_{b,d}\omega_{b,d} = 0\ \forall\ \omega_{b,d} \in \mathbb{R}^3$, the derivative of e_{ω_b} is derived as

$$\begin{aligned} \dot{e}_{\omega_b} &= \frac{d}{dt}\left(\omega_b - R^T R_d \omega_{b,d}\right), \\ &= \dot{\omega}_b - \dot{R}^T R_d \omega_{b,d} - R^T \dot{R}_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}, \\ &= \dot{\omega}_b - \widetilde{\omega}_b^T R^T R_d \omega_{b,d} - R^T R_d \widetilde{\omega}_{b,d} \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}, \\ &= \dot{\omega}_b + \widetilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}. \end{aligned} \tag{5-13}$$

Finally, from here the attitude error dynamics is represented by

$$J\dot{e}_{\omega_b} = J\dot{\omega}_b + J\left(\widetilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\right). \tag{5-14}$$

5.2.2. Nonlinear virtual attitude control input

A nonlinear control input which asymptotically stabilizes the orientation of the Nimble to the desired attitude $R_d \in SO(3)$ is described by

$$\tau_{b,v} = -k_R e_R - k_{\omega_b} e_{\omega_b} + \omega_b \times J\omega_b - J\big(\tilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\big), \tag{5-15}$$

where the control gains $k_R, k_\omega > 0$. By combining the rigid body equations and the nonlinear control input of (5-15) the attitude error dynamics can be further rewritten as

$$\begin{aligned} J\dot{\omega}_b + J\omega_b \times \omega_b &= -k_R e_R - k_{\omega_b} e_{\omega_b} + \omega_b \times J\omega_b - J\big(\tilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\big), \\ J\dot{\omega}_b &= -k_R e_R - k_{\omega_b} e_{\omega_b} - J\big(\tilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\big), \end{aligned} \tag{5-16}$$

Finally, by using Equation (5-14), the attitude error dynamics are given by

$$J\dot{e}_{\omega_b} = -k_R e_R - k_{\omega_b} e_{\omega_b}. \tag{5-17}$$

The virtual control input $\tau_{b,v}$ is feedback dependent on the current and desired attitude and angular velocity and on the desired angular acceleration. This controller can be read as introducing a potential expression through the orientation, a dissipation expression through the angular velocity and a feedforward term. Note that the translational motion of the Nimble does not have to be considered.

5.2.3. Lyapunov analysis for the attitude control loop

For exponential stability of the attitude dynamics, a Lyapunov analysis is carried out. The method can be thought of as thinking of a physical mass damper system and considering the energy of this system. When the system loses energy over time, the system eventually reaches a final resting state, the attractor. These kinds of systems are found all over nature in biological, mathematical and economic systems. For the attitude error dynamics of the Nimble, the quadratic Lyapunov function $\mathcal{V}_R$ will demonstrate that $(e_R, e_{\omega_b}) = (0{,}0)$ is an almost global exponentially stable equilibrium. In order to prove this, first it is shown that $\mathcal{V}_R$ is zero if and only if $(e_R, e_{\omega_b}) = (0{,}0)$ and that $\mathcal{V}_R$ is positive if and only if $(e_R, e_{\omega_b}) \neq (0{,}0)$. Subsequently, the proof is given that $\dot{\mathcal{V}}_R$ is semi-negative definite and is only zero when $(e_R, e_{\omega_b}) = (0{,}0)$. To begin with, define a Lyapunov function $\mathcal{V}_R$ with a constant $\epsilon_R > 0$ as

$$\mathcal{V}_R = \frac{1}{2} e_{\omega_b}^T J e_{\omega_b} + k_R \Psi(\mathrm{R}, \mathrm{R_d}) + \epsilon_R e_R^T e_{\omega_b}. \tag{5-18}$$

The attitude error e_R can be written in terms of $\Psi(\mathrm{R}, \mathrm{R_d})$ as

$$\begin{aligned} \|e_R\|^2 &= \sin^2\|\vartheta\| = (1 + \cos\|\vartheta\|)\Psi(\mathrm{R}, \mathrm{R_d}), \\ &= (2 - \Psi(\mathrm{R}, \mathrm{R_d}))\Psi(\mathrm{R}, \mathrm{R_d}). \end{aligned} \tag{5-19}$$

It can be shown that given the attitude errors Ψ and e_{ω_b} lie in the sublevel sets

$$\mathcal{A}_R = \{\mathrm{R} \in \mathrm{SO}(3) \mid \Psi(R(0), R_d(0)) \leq \Psi_2 < 2\}, \tag{5-20}$$

$$\mathcal{A}_{\omega_b} = \left\{\omega_b \in \mathbb{R}^3 \,\middle|\, \|e_{\omega_b}(0)\|^2 \leq \frac{2}{\lambda_{min}(J)} k_R \left(\Psi_2 - \Psi(R(0), R_d(0))\right)\right\}, \tag{5-21}$$

where $\Psi_2 = \frac{1}{k_R} \mathcal{V}_R|_{\epsilon_R = 0}(0)$. The error function $\Psi(\mathrm{R}, \mathrm{R_d})$ has bounds

$$\frac{1}{2}\|e_R\|^2 \leq \Psi(\mathrm{R}, \mathrm{R_d}) \leq \frac{1}{2 - \Psi_2}\|e_R\|^2. \tag{5-22}$$

$\lambda_{min}(J)$ represent the smallest and highest eigenvalues of the inertia tensor matrix and is positive by definition. Therefore $\Psi(\mathrm{R}, \mathrm{R_d})$ is positive definite and decreasing [32]. By choosing

$$0 < \epsilon_R < \sqrt{k_R \lambda_{min}(J)}, \tag{5-23}$$

the Lyapunov function $\mathcal{V}_R$ is bounded by

$$
\begin{gathered}
z_R \mathcal{M}_{R1} z_R^T \le \mathcal{V}_R \le z_R \mathcal{M}_{R2} z_R^T, \\
with \\
\mathcal{M}_{R1} = \frac{1}{2}\begin{bmatrix} k_R & -\epsilon_R \\ -\epsilon_R & \lambda_{min}(J) \end{bmatrix} \succ 0 \in \mathbb{R}^{2\times 2}, \\
\mathcal{M}_{R2} = \frac{1}{2}\begin{bmatrix} \frac{2k_R}{2-\Psi_2} & -\epsilon_R \\ -\epsilon_R & \lambda_{max}(J) \end{bmatrix} \succ 0 \in \mathbb{R}^{2\times 2},
\end{gathered}
\tag{5-24}
$$

with $z_R = (\|e_R\| \; \|e_{\omega_b}\|)$. Next, the time derivative of $\mathcal{V}_R$ is computed. By using (5-9), (5-11) and (5-16), the time derivative of V_R is given by

$$
\begin{aligned}
\dot{\mathcal{V}}_R &= e_{\omega_b}^T J \dot{e}_{\omega_b} + k_R e_R^T e_{\omega_b} + \epsilon_R \dot{e}_R^T e_{\omega_b} + \epsilon_R e_R^T \dot{e}_{\omega_b}, \\
\dot{\mathcal{V}}_R &= e_{\omega_b}^T\left(-k_R e_R - k_{\omega_b} e_{\omega_b}\right) + k_R e_R^T e_{\omega_b} + \epsilon_R C\left(R_d^T R\right) e_{\omega_b}^T e_{\omega_b} \\
&\qquad + \epsilon_R e_R^T J^{-1}\left(-k_R e_R - k_{\omega_b} e_{\omega_b}\right), \\
\dot{\mathcal{V}}_R &= -e_{\omega_b}^T k_{\omega_b} e_{\omega_b} + \epsilon_R C\left(R_d^T R\right) e_{\omega_b}^T e_{\omega_b} - \epsilon_R J^{-1} k_R e_R^T e_R - \epsilon_R J^{-1} k_{\omega_b} e_R^T e_{\omega_b}.
\end{aligned}
\tag{5-25}
$$

Since $\|C(R_d^T R)\| \le 1$, and by choosing

$$
0 < \epsilon_R < \min\left\{k_{\omega_b}, \frac{4k_{\omega_b} k_R \lambda_{min}(J)^2}{k_\omega^2 \lambda_{max}(J) + 4k_R \lambda_{min}(J)^2}\right\},
\tag{5-26}
$$

the time derivative of the Lyapunov function $\mathcal{V}_R$ is bounded by

$$
\begin{aligned}
\dot{\mathcal{V}}_R &\le -z_R \begin{bmatrix} \frac{\epsilon_R k_R}{\lambda_{max}(J)} & -\frac{\epsilon_R k_{\omega_b}}{2\lambda_{min}(J)} \\ -\frac{\epsilon_R k_{\omega_b}}{2\lambda_{min}(J)} & k_{\omega_b} - \epsilon_R \end{bmatrix} z_R^T, \\
\dot{\mathcal{V}}_R &\le -z_R \, \mathcal{W}_R z_R^T, \\
\dot{\mathcal{V}}_R &\le -\frac{\lambda_{min}(\mathcal{W}_R)}{\lambda_{max}(\mathcal{M}_{R2})} V_R. \\
with \;\; \mathcal{W}_R &= \begin{bmatrix} \frac{\epsilon_R k_R}{\lambda_{max}(J)} & -\frac{\epsilon_R k_{\omega_b}}{2\lambda_{min}(J)} \\ -\frac{\epsilon_R k_{\omega_b}}{2\lambda_{min}(J)} & k_{\omega_b} - \epsilon_R \end{bmatrix} \succ 0 \in \mathbb{R}^{2\times 2}.
\end{aligned}
\tag{5-27}
$$

In Equation (5-25), λ_{min} and λ_{max} represent the smallest and highest eigenvalues of the inertia tensor matrix and are positive by definition. Therefore $\dot{\mathcal{V}}_R$ is negative definite. On top of that, Ψ exponentially decreases and it is guaranteed that $\Psi < 2$, since

$$
\begin{aligned}
(2-\Psi_2)\lambda_{min}(\mathcal{M}_{R1})\Psi &\le \lambda_{min}(\mathcal{M}_{R1})\|e_R\|^2 \\
&\le \lambda_{min}(\mathcal{M}_{R1})\|z_R\|^2 \\
&\le \mathcal{V}_R(t) \le \mathcal{V}_R(0) e^{-\frac{\lambda_{min}(\mathcal{W}_R)}{\lambda_{max}(\mathcal{M}_{R2})}t}.
\end{aligned}
\tag{5-28}
$$

The region of attraction $\mathcal{A}_R$ is almost the entire space of rotations with the exception of $\mathit{\Psi} = 2$, or an initial attitude error les than $180°$ between R_d and R, and the region of attraction $\mathcal{A}_{\omega_\mathrm{b}}$ has a large set of angular velocities. Equation (5-20) specifies the set of rotation errors and Equation (5-21) specifies the range of angular velocity errors from which the body can converge to the desired configuration. The smaller λ_{min}, the larger the value at the right-hand side, which increases the set of angular velocities from which the Nimble is still able to converge to the hover state. This is a great motivation to build small FWMAVs. Furthermore, the higher the control gain k_R, the larger the region of attraction of the set of rotational velocity error. With a lower mass moment of inertia, the region of attraction $\mathcal{A}_{\omega_\mathrm{b}}$ in Equation (5-21) will increase to enhance the controller performance.

Proposition 1: Almost global exponential stability of the attitude control design

Choosing feasible constants for ϵ_R according to (5-26), positive constant for k_R and k_{ω_b} and assuming $\mathit{\Psi} \leq \Psi_2$ and $\omega_b \in \mathcal{A}_{\omega_b}$. Then, since $\mathcal{V}_R = 0 \Leftrightarrow (e_R,\ e_{\omega_b}) = (0{,}0)$, $\mathcal{V}_R > 0 \Leftrightarrow (e_R,\ e_{\omega_b}) \neq (0{,}0)$ and $\dot{\mathcal{V}}_R \leq \mathcal{V}_R(0)e^{-\frac{\lambda_{min}(\mathcal{W}_R)}{\lambda_{max}(\mathcal{M}_{R2})}t} \Leftrightarrow$ $(e_R,\ e_{\omega_b},) \neq (0{,}0)$, the controller design for the attitude closed loop is almost global exponentially stable.

5.2.4. An auxiliary goal: altitude tracking

Since the Nimble has 4 control inputs, the desired altitude can be tracked as an auxiliary goal in different ways. An illustration of the closed-loop system for the position tracking control loop is shown in Figure 5-3.

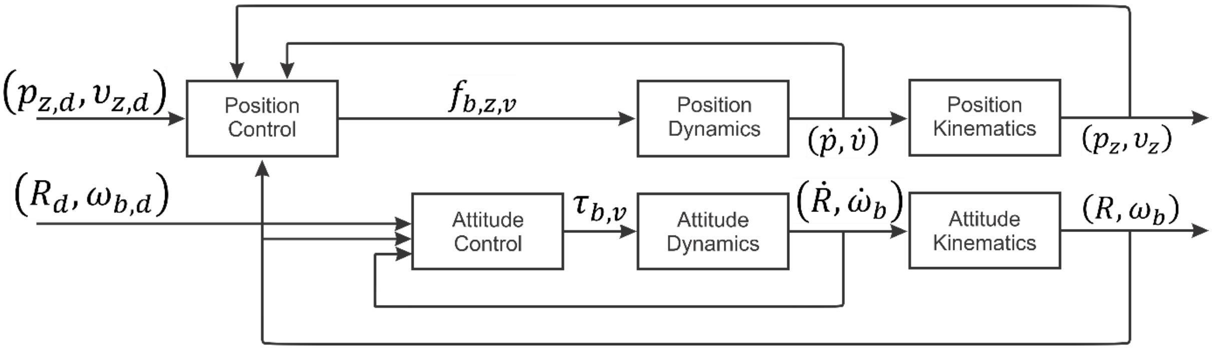

Figure 5-3: Block diagram of the tracking control structure for the attitude loop and vertical position.

Now for the vertical translational error dynamics, the position error $e_{p,z}$ is defined as the difference between the current position p_z and the desired position $p_{z,d}(t) \in \mathbb{R}^3$ as given by

$$e_{p,z} = p_z - p_{z,d} \tag{5-29}$$

The velocity error $e_{z,v}$ is defined as the difference between the current velocity v_z and the desired velocity $v_{z,d}(t)$ as given by

$$e_{v_z} = v_z - v_{z,d} \tag{5-30}$$

The control law for the vertical position loop is essentially a proportional plus derivative control law. The magnitude of the thrust of the Nimble depends on the vertical position and vertical velocity errors and the desired vertical translational acceleration. In order to track the vertical position, the commanded thrust vector of the Nimble $f_{b,z}$ is given by

$$f_{b,z,v} = \frac{k_p(p_z - p_{z,d}) + k_v(v_z - v_{z,d}) + mg - m\ddot{p}_{z,d}}{z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}}, \tag{5-31}$$

where $z^{\mathcal{I}}$ represents the z-axis in inertial frame, $\ddot{p}_{z,d}$ the desired vertical acceleration and with k_p, k_v positive constants. $z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}$ is singular when physically the Nimble attains a roll or pitch angle of $\frac{\pi}{2}$ and cannot generate a vertical thrust force. Therefore, it is assumed that $z^{\mathcal{I}} \cdot Rz^{\mathcal{I}} \neq 0$. This attitude control design can be used for short time periods in which (aerobatic) attitude maneuvers are performed. The stability of the vertical translation together with the full translational motion of the Nimble is analyzed in Section 5.3.

5.3. Controller design for the position loop

The general aim of the Nimble is to be able to have the required intelligence for autonomously performing tasks in unknown environments. Therefore, it is desirable to implement a trajectory controller to bring autonomous flight a step closer. In this section, a nonlinear controller for the position control problem is derived by using the control framework created by Lee et al. (2010) [32]. The position loop is coupled to the attitude loop and results from Section 5.1.3 are directly applied in this section. Typically, a trajectory with its derivatives up to the second order and the desired yaw angle are provided. Let the position tracking control problem formulation be the following. Assuming a feasible desired reference trajectory $\sigma_d(t) = \{p_d, v_d, R_c, \omega_{b,c}\}$ and a current state of the Nimble $\sigma_0(t) = \{p, v, R, \omega_b\}$, a state-feedback tracking control law should thrive the error measure between the actual and reference states in the closed-loop system to zero as

$$\lim_{t\to\infty} \varepsilon(t, \sigma_0, \sigma_d) = \left(e_p, e_v, e_R, e_{\omega_b}\right) = (0, 0, 0, 0)\,, \tag{5-32}$$

with some error measure $\varepsilon(t, \sigma_0, \sigma_d)$ between the reference and the actual states. An illustration of the closed-loop system for the position tracking control loop is shown in Figure 5-4. The position loop is interconnected with the attitude loop in which the control attitude R_c, generated by the position controller, is followed by the attitude controller. R_c is determined by chosing the desired direction $x_d^{\mathcal{B}}$ (yaw angle) and thrust force direction $z_d^{\mathcal{B}}$. The position controller would perform perfectly in the ideal case that the rotation matrix is equal to the control rotation matrix at all times, i.e. $R(t) = R_c(t)$.

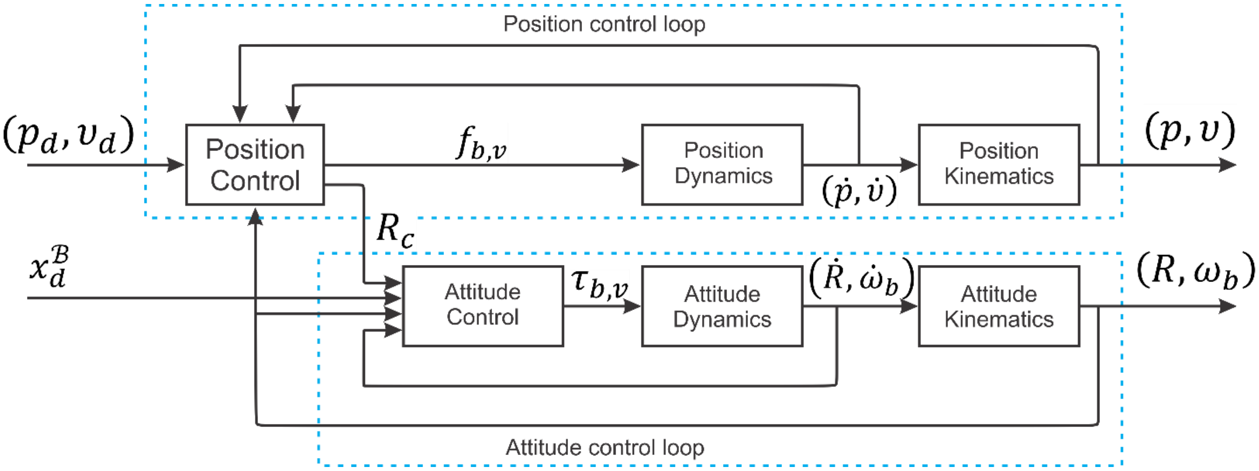

Figure 5-4: Block diagram of the nonlinear tracking control structure.

In this section the combined equations of motion (see Section 3.2) are considered as

$$\begin{aligned} \dot{p} &= v, \\ \dot{v} &= gz^{\mathcal{I}} - \frac{Rf_{b,v}}{m}, \\ \dot{R} &= R\widetilde{\omega}_b, \\ \dot{\omega}_b &= -J^{-1}\widetilde{\omega}_b J\omega_b + J^{-1}\tau_b \end{aligned} \tag{5-33}$$

in which the virtual control inputs $f_{b,v}$ and $\tau_{b,v}$ are chosen as the thrust force and torque of the rigid body, J is the inertia matrix in body frame, g is the gravitational acceleration, m is the mass of the Nimble, ω_b is the angular velocity in body frame $\mathcal{B}$, p the position vector of the center of mass in the inertial frame $\mathcal{I}$, v the velocity vector of the center of mass in the inertial frame $\mathcal{I}$ and R is the rotation matrix from body frame $\mathcal{B}$ to inertial frame $\mathcal{I}$. In this section a nonlinear position controller is presented that achieves exponential stability when the attitude error is less than $90°$ and almost global exponential attractiveness when the attitude error is less than $180°$.

The position and attitude errors that are associated with the position closed-loop error dynamics are formulated in Section 5.3.1. Subsequently, in Section 5.3.2, the nonlinear position control input is presented. The generation of R_c is explained in Section 5.3.3. Thereafter, the stability properties of the closed-loop position dynamics are discussed with a Lyapunov analysis in Section 5.3.4.

5.3.1. Translational error dynamics

In this section position error functions are presented that are associated with the translational error dynamics. The translational error dynamics are necessary for defining the nonlinear control design and properties that follow will eventually be used for the Lyapunov analysis for the position closed loop. For the translational error dynamics, the position error e_p is defined as the difference between the current position p and the desired position $p_d(t) \in \mathbb{R}^3$ and the velocity error e_v is defined as the difference between the current velocity v and the desired velocity $v_d(t)$ as

$$\begin{aligned} e_p &= p - p_d, \\ e_v &= v - v_d. \end{aligned} \tag{5-34}$$

The derivatives of the translational error dynamics are given by

$$\begin{aligned} \dot{e}_p &= e_v, \\ \dot{e}_v &= g z^{\mathcal{I}} - m^{-1} R f_{b,v} - \dot{v}_d. \end{aligned} \tag{5-35}$$

The tracking error vector of the attitude e_R and the tracking error vector for the angular velocity e_{ω_b} are selected as

$$\begin{aligned} e_R &= (R_c^T R - R^T R_c)^{\vee} \in \mathbb{R}^3, \\ e_{\omega_b} &= \omega_b - R^T R_c \omega_{b,c} \ \in \mathbb{R}^3. \end{aligned} \tag{5-36}$$

Essentially the tracking errors are the same as in Equations (5-7) and (5-10), with the difference that R_c is now computed by the position controller. In order to write e_v in terms of e_R, the term $f_{b,z}\left(z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}\right)^{-1} R z^{\mathcal{I}}$ is added and subtracted from the velocity error dynamics as

$$m\dot{e}_v = mgz^{\mathcal{I}} - m\dot{v}_d - \frac{f_{b,z}}{z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}} R z^{\mathcal{I}} \underbrace{- \frac{f_{b,z}}{z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}}\left(\left(z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}\right) R z^{\mathcal{I}} - R_c z^{\mathcal{I}}\right)}_{X}. \tag{5-37}$$

The terms $z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}$ and $\left(z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}\right) R z^{\mathcal{I}} - R_c z^{\mathcal{I}}$ represent the cosine and sine of the axis-angle between $z^{\mathcal{B}} = R z^{\mathcal{I}}$ and $z_c^{\mathcal{B}} = R_c z^{\mathcal{I}}$. Therefore, $0 < z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}} \leq 1 - \Psi$ and $\frac{f_{b,z}}{z^{\mathcal{I}^T} R_c^T R z^{\mathcal{I}}}$ is well-defined.

5.3.2. Nonlinear virtual position control input

In order to track the desired trajectory, the nonlinear control input of the Nimble $f_{b,v}$ is introduced as

$$f_{b,v} = \left(k_p e_p + k_v e_v + mgz^{\mathcal{I}} - m\dot{v}_d\right) \cdot Rz^{\mathcal{I}}, \tag{5-38}$$

where the control gains $k_p, k_v > 0$. The control law for the outer position loop is essentially a proportional plus derivative control law. In addition, there is a feedforward term, describing the desired acceleration of the specified trajectory and compensating for the gravity vector. By combining the rigid body equations and the nonlinear control input of (5-39), the translational error dynamics can be further rewritten as

$$\begin{aligned} \dot{e}_p &= e_v, \\ m\dot{e}_v &= -k_p e_p - k_v e_v - X. \end{aligned} \tag{5-39}$$

If the system was fully actuated, in other words there would be an actuator for every DoF, this could solve the control problem. However, the virtual force input $f_{b,v}$ is a scalar quantity. In fact, the only direction in which the thrust can be applied is along the z-axis of the body frame. Because of that, this vector is projected along the body fixed z-axis. Next, for the attitude control problem, the orientation of the Nimble has to be aligned with the desired orientation in order to track the position. Aligning the thrust vector and choosing the yaw angle, the error in rotation can be computed by comparing the control orientation with the current orientation. The nonlinear control design given by Equation (5-40) can be interpreted as backstepping. The virtual control input $\tau_{b,v}$ depends on the error in orientation, the error in angular velocity, plus a feedforward term as defined in Section 5.2, with the control attitude R_c instead of the desired attitude R_d.

$$\tau_{b,v} = -k_R e_R - k_{\omega_b} e_{\omega_b} + \widetilde{\omega}_b J \omega_b - J\left(\widetilde{\omega}_b R^T R_c \omega_{b,c} - R^T R_c \dot{\omega}_{b,c}\right) \tag{5-40}$$

5.3.3. Generation of R_c by the position controller

Given the prior definitions of e_R and e_ω in Equations (5-7) and (5-10), the control attitude $R_c(t) \in SO(3)$ and control angular velocity $\omega_{b,c} \in \mathbb{R}^3$ are determined as

$$\begin{aligned} R_c &= [x_c^{\mathcal{B}}, z_c^{\mathcal{B}} \times x_c^{\mathcal{B}}, z_c^{\mathcal{B}}]^T, \\ \widetilde{\omega}_{b,c} &= R_c^T \dot{R}_c. \end{aligned} \tag{5-41}$$

The Nimble has to rotate the direction of its axes $z^{\mathcal{B}}$, fixed and parallel to the its thrust vector $f_{b,z}$. Therefore R_c is chosen such that the direction of $-z_c^{\mathcal{B}}$ equals $f_{b,z,c}$, i.e.

$$z_c^{\mathcal{B}} = \frac{-k_p e_p - k_v e_v - mgz^{\mathcal{I}} + m\dot{v}_d}{\left\|-k_p e_p - k_v e_v - mgz^{\mathcal{I}} + m\dot{v}_d\right\|} = -\frac{A}{\|A\|}. \tag{5-42}$$

It is assumed that $\|A\| \neq 0$ and that the commanded acceleration is uniformly bounded as $\left\|-mgz^{\mathcal{I}} + m\dot{v}_d\right\| < B$. In turn, the thrust vector of the Nimble rotates as the attitude controller follows the commanded attitude R_c. Ultimately, when the orientation of the Nimble equals the commanded attitude, $f_{b,z} = \left\|f_{b,z,c}\right\|$. R_c can be viewed as the required attitude to asymptotically track the position command, namely $R(t) \to R_c(t)$ as $t \to \infty$. In addition, another DoF is free to track, since the position controller tracks the position in $\mathbb{R}^3$ and the Nimble has four control inputs. The direction $x_c^{\mathcal{B}}$ is chosen in order to control the yaw angle and can be constrained as follows. Given the desired direction $x_d^{\mathcal{B}} \in S^2$, $x_c^{\mathcal{B}}$ is computed as the normalized projection of $x_d^{\mathcal{B}}$ onto the plane spanned by $x_d^{\mathcal{B}}$ and $z_c^{\mathcal{B}}$ and perpendicular to $z_c^{\mathcal{B}}$ as depicted in Figure 5-5. When $\dot{v}_d = 0$, $z^{\mathcal{B}}$ converges to $z^{\mathcal{I}}$ and $x_c^{\mathcal{B}}(t) \to Proj\left[x_d^{\mathcal{B}}(t)\right] = x_d^{\mathcal{B}}(t)$ as $t \to \infty$. In short, the expressions for R_c are given by

$$z_c^{\mathcal{B}} = -\frac{A}{\|A\|}, \quad y_c^{\mathcal{B}} = -\frac{C}{\|C\|} = -\frac{z_c^{\mathcal{B}} \times x_d^{\mathcal{B}}}{\left\|z_c^{\mathcal{B}} \times x_d^{\mathcal{B}}\right\|}, \quad x_c^{\mathcal{B}} = y_c^{\mathcal{B}} \times z_c^{\mathcal{B}} \tag{5-43}$$

The expressions for the derivatives of the angular velocities can be found in (I-1).

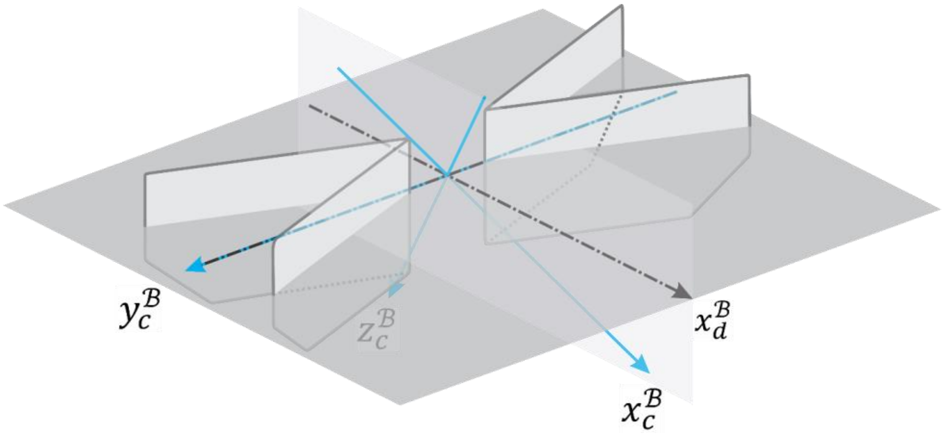

Figure 5-5: Concept of the control orientation R_c. The dark grey plane represents the plane normal to $z_c^{\mathcal{B}}$ and the light grey plan represents the plane spanned by $x_d^{\mathcal{B}}$ and $z_c^{\mathcal{B}}$. Given the computed direction $z_c^{\mathcal{B}}$ and freely chosen $x_d^{\mathcal{B}}$, $x_c^{\mathcal{B}}$ is the projection on the plane perpendicular to $z_c^{\mathcal{B}}$ and $x_d^{\mathcal{B}}$.

5.3.4. Lyapunov analysis of the position control loop

For the translational error dynamics of the Nimble, the quadratic Lyapunov function $\mathcal{V}$ will demonstrate that $(e_p, e_v, e_R, e_{\omega_b}) = (0,0,0,0)$ is an exponentially stable equilibrium when $\Psi(R(0), R_c(0)) < 1$ and exponentially attractive when $1 \leq \Psi(\mathrm{R}(0), \mathrm{R_c}(0)) < 2$. In order to prove this, first it is shown that $\mathcal{V}$ is zero if and only if $(e_p, e_v, e_R, e_{\omega_b}) = (0,0,0,0)$ and that $\mathcal{V}$ is positive if and only if $(e_p, e_v, e_R, e_{\omega_b}) \neq (0,0,0,0)$, i.e. $\mathcal{V}$ is positive definite. Subsequently, the proof is given that $\dot{\mathcal{V}}$ is negative definite. To show that the trajectory controller achieves almost global asymptotic tracking, consider the Lyapunov function $\mathcal{V}_p$ defined in Equation (5-44).

$$\mathcal{V}_p = \frac{1}{2}\left(k_p e_p^T e_p + m e_v^T e_v\right) + \epsilon_p e_p^T e_v \tag{5-44}$$

By choosing

$$0 < \epsilon_p < \sqrt{k_p m}, \tag{5-45}$$

the Lyapunov function $\mathcal{V}_p$ is bounded by

$$\begin{gathered} z_p \mathcal{M}_{p1} z_p^{\mathrm{T}} \leq \mathcal{V}_p \leq z_p \mathcal{M}_{p2} z_p^{\mathrm{T}} \\ \textit{with} \\ \mathcal{M}_{p1} = \frac{1}{2}\begin{bmatrix} k_p & -\epsilon_p \\ -\epsilon_p & m \end{bmatrix} \succ 0 \in \mathbb{R}^{2\times 2}, \qquad \mathcal{M}_{p2} = \frac{1}{2}\begin{bmatrix} k_p & \epsilon_p \\ \epsilon_p & m \end{bmatrix} \succ 0 \in \mathbb{R}^{2\times 2}, \end{gathered} \tag{5-46}$$

with $z_p = (\|e_p\| \; \|e_v\|)$. By substituting the translational error dynamics from Equation (5-39), the time derivative of $\mathcal{V}_p$ is given by

$$\begin{aligned} \dot{\mathcal{V}}_p &= k_p e_p^T e_v + m e_v^T \dot{e}_v + \epsilon_p\left(e_v^T e_v + e_p^T \dot{e}_v\right) \\ &= k_p e_p^T e_v + e_v^T\left(-k_p e_p - k_v e_v + X\right) + \epsilon_p\left(e_v^T e_v + m^{-1} e_p^T\left(-k_p e_p - k_v e_v + X\right)\right) \\ &= -k_v \epsilon_p m^{-1} e_p^T e_v + \left(-k_v + \epsilon_p\right) e_v^T e_v - m^{-1}\epsilon_p k_p e_p^T e_p + X\left(m^{-1}\epsilon_p e_p + e_v\right). \end{aligned} \tag{5-47}$$

By using that X is bounded by $\left(k_p\|e_p\| + k_v\|e_v\| + mgz^{\mathcal{J}} + m\dot{v}_d\right)\|e_R\|$ [63], $\dot{\mathcal{V}}_p$ can be further written as

$$\begin{aligned}
\dot{\mathcal{V}}_p \leq & -k_v\epsilon_p m^{-1}(1+\|e_R\|)e_p^T e_v + \left(-k_v(1-\|e_R\|)+\epsilon_p\right)e_v^T e_v \\
& + \left(-\epsilon_p k_p m^{-1}\right)(1-\|e_R\|)e_p^T e_p \\
& + \|e_R\|\left(k_p\|e_p\|\|e_v\| + B\left(\epsilon_p m^{-1}\|e_p\| + \|e_v\|\right)\right) \\
\leq & -z_p\mathcal{W}_p z_p^T + z_p\mathcal{W}_{pR} z_R^T, \\
& \qquad\qquad with \\
\mathcal{W}_p = & \begin{bmatrix} \left(\epsilon_p k_p m^{-1}\right)(1-\|e_R\|) & -\dfrac{k_v\epsilon_p m^{-1}}{2}(1+\|e_R\|) \\ -\dfrac{k_v\epsilon_p m^{-1}}{2}(1+\|e_R\|) & \left(k_v(1-\|e_R\|)-\epsilon_p\right) \end{bmatrix} \in \mathbb{R}^{2\times 2}, \\
\mathcal{W}_{pR} = & \begin{bmatrix} k_p e_{v_{max}} + B\epsilon_p m^{-1} & 0 \\ B & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}.
\end{aligned} \tag{5-48}$$

Employing Equations (5-18) and (5-44), a Lyapunov function for both the position and attitude systems is

$$\begin{aligned}
\mathcal{V} &= \mathcal{V}_R + \mathcal{V}_p \\
&= \frac{1}{2}e_{\omega_b} J e_{\omega_b} + k_R\Psi + \epsilon_R e_R e_{\omega_b} + \frac{1}{2}\left(k_p e_p^T e_p + e_v^T e_v\right) + \epsilon_p e_p^T e_v.
\end{aligned} \tag{5-49}$$

Making use of Equations (5-24) and (5-46), the Lyapunov function $\mathcal{V}$ is bounded by

$$z_R\mathcal{M}_{R1}z_R^{\mathrm{T}} + z_p\mathcal{M}_{p1}z_p^T \leq \mathcal{V} \leq z_R\mathcal{M}_{R2}z_R^T + z_p\mathcal{M}_{p2}z_p^T. \tag{5-50}$$

Making use of Equations (5-25) and (5-48), the time derivative of $\mathcal{V}$ is given by

$$\dot{\mathcal{V}} \leq -z_p\mathcal{W}_p z_p^T + z_p\mathcal{W}_{pR}z_R^T - z_R\mathcal{W}_R z_R^T. \tag{5-51}$$

By choosing a feasible constant for ϵ_p

$$\begin{aligned}
& 0 < \epsilon_p < \min\left\{k_v(1-\|e_R\|), \frac{4mk_p k_v(1-\|e_R\|)}{k_v^2\|e_R\|^2 + 2k_v^2\|e_R\| + k_v^2 + 4mk_p}\right\}, \\
& \lambda_{min}(W_R) > \frac{4\left\|W_{pR}\right\|^2}{\lambda_{min}\left(W_p\right)},
\end{aligned} \tag{5-52}$$

and by assuming that

$$\begin{aligned}
& \left\|mgz^{\mathcal{J}} + m\dot{v}_d\right\| < B, \\
& \left\|e_{v_{max}}\right\| \leq \max\left\{e_v(0), \frac{B}{k_v(1-\|e_R\|)}\right\}, \\
& \Psi\left(R(0), R_c(0)\right) \leq \Psi_1 < 1,
\end{aligned} \tag{5-53}$$

the time derivative of $\mathcal{V}_p$ is negative definite.

Proposition 2: Almost Global Exponential Stability of the position control design

Choosing feasible constants for ϵ_p *and* ϵ_R *according to (5-23), (5-26), (5-45) and (5-52), positive constant for* k_p, k_v, k_R *and* k_{ω_b} *and assuming* $\Psi\big(R(0), R_c(0)\big) \leq \Psi_1$, $\omega_b \in \mathcal{A}_{\omega_b}$ *and* $\|e_{v_{max}}\| \leq max\left\{e_v(0), \frac{B}{k_v(1-\|e_R\|)}\right\}$*. Then, since* $\mathcal{V} = 0 \Leftrightarrow (e_R,\ e_{\omega_b},\ e_p,\ e_v) = (0,0,0,0)$*,* $\mathcal{V} > 0 \Leftrightarrow (e_R,\ e_{\omega_b},\ e_p,\ e_v) \neq (0,0,0,0)$ *and* $\dot{\mathcal{V}} \leq -z_p \mathcal{W}_p z_p^T + z_p \mathcal{W}_{pR} z_R^T - z_R \mathcal{W}_R z_R^T \ \Leftrightarrow\ (e_R,\ e_{\omega_b},\ e_p,\ e_v) \neq (0,0,0,0)$*, the controller design for the position closed loop is exponentially stable.*

When the attitude rotation error is equal or higher than 90°, i.e. $1 \leq \Psi\big(R(0), R_c(0)\big) < 2$, Proposition 1 is applied in which $\Psi\big(R(0), R_c(0)\big)$ exponentially decreases in a finite time $ƭ$ until it enters the region of Proposition 2 again. The goal is to prove that the error z_p in $t \in [0, ƭ]$ has no finite escape time and thus is confined. Exponential attractiveness can be proved if z_p remains bounded. When z_p is bounded, $\big[\|z_p\|, \|z_R\|\big]$ is uniformly bounded in $t \in [0, ƭ]$ and exponentially decreases when $t > ƭ$ [63].

Definition: Exponential attractiveness [63]:
An equilibrium point $z = 0$ of a dynamic system is exponentially attractive if, for some $\delta > 0$, there exist constants $\alpha(\delta) > 0$, $\beta > 0$, such that $\|z(0)\| < \delta$ implies $\|z(t)\| \leq \alpha(\delta) e^{-\beta t}\ \forall\, t > 0$. This is different from exponential stability, since this yields $\|z(t)\| \leq \alpha(\delta) z(0) e^{-\beta t}$.

Consider the positive definite function

$$\mathcal{V}_p' = \frac{1}{2}\big(e_p^T e_p + m e_v^T e_v\big). \tag{5-54}$$

Then, $\|e_p\| \leq \sqrt{2\mathcal{V}_p'}$ and $\|e_v\| \leq \sqrt{2m^{-1}\mathcal{V}_p'}$. By using Equations (5-38) and (5-53), the time derivative of $\mathcal{V}_p'$ is given by

$$\begin{aligned} \dot{\mathcal{V}}_p' &= e_p^T e_v + e_v^T\big(mgz^{\mathcal{I}} - f_{b,v} R z^{\mathcal{I}} - m\dot{v}_d\big) \\ &\leq \|e_p\|\|e_v\| + \|e_v\|\big\|mgz^{\mathcal{I}} - m\dot{v}_d\big\| + \|e_v\|\big\|Rz^{\mathcal{I}}\big\|\big\|f_{b,v}\big\| \\ &= k_v\|e_v\|^2 + \big(2B + \big(k_p + 1\,\big)\|e_p\|\big)\|e_v\| \\ &\leq d_1\mathcal{V}_p' + d_2\sqrt{\mathcal{V}_p'}, \end{aligned} \tag{5-55}$$

where $d_1 = 2m^{-1}k_v + 2\big(k_p + 1\big)m^{-0.5}$, $d_2 = 2B\sqrt{2m^{-1}}$. When $\mathcal{V}_p' > 1$ for a time interval $[t_a, t_b] \subset [0, ƭ]$, $\sqrt{\mathcal{V}_p'} \leq \mathcal{V}_p'$ and therefore,

$$\dot{\mathcal{V}}_p' \leq (d_1 + d_2)\mathcal{V}_p' \quad \Rightarrow \quad \mathcal{V}_p'(t) \leq \mathcal{V}_p'(t_a) e^{(d_1+d_2)(t-t_a)} \tag{5-56}$$

Altogether, $\mathcal{V}_p'$ is bounded for when $\mathcal{V}_p' \geq 1$, which implies that $\mathcal{V}_p'$ is bounded for $0 \leq t \leq \mathcal{t}$. In conclusion, the equilibrium of the tracking error is exponentially attractive.

Proposition 3: Almost global exponential attractiveness of the position control design

Choosing positive constant for k_p, k_v, k_R *and* k_{ω_b} *and assuming* $1 \leq \Psi\big(R(0), R_c(0)\big) < 2$, $\omega_b \in \mathcal{A}_{\omega_b}$ *and* $\|e_{v_{max}}\| \leq max\left\{e_v(0), \frac{B}{k_v(1-\|e_R\|)}\right\}$*. Then, the zero equilibrium* $(e_R, e_{\omega_b}, e_p, e_v) = (0,0,0,0)$ *is exponentially attractive.*

5.4. Control gain parameters

5.4.1. Attitude control gain parameters

The closed-loop system depends on four control gains. Two gains of the position loop, k_p and k_v and two gains of the attitude loop k_R and k_{ω_b}. The mass and the inertia of the rigid body equations influence the gain parameters. By choosing the gains as $k_R = 0.011$ and $k_{\omega_b} = 0.0019$ for Equation (5-18), the attitude closed-loop system is critically damped. As observed in Figure 5-7, the stiff controller reduces the error faster and the damped controller has as slower response in comparison to the nominal controller. On the other hand, higher control effort, transients and overshoot are caused by using the stiff controller. Moreover, the nominal controller barely has a lower settling time. The region of attraction $\mathcal{A}_{\omega_b}$, as defined in Equation (5-20), has a large set of angular velocities as is visualized in Figure 5-6 with the chosen gain k_R and the inertia matrix tensor J.

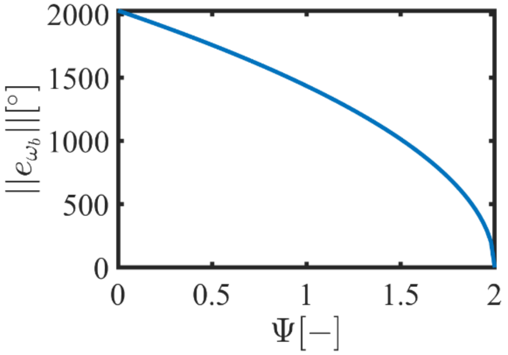

Figure 5-6: Visualization of $\mathcal{A}_{\omega_b}$.

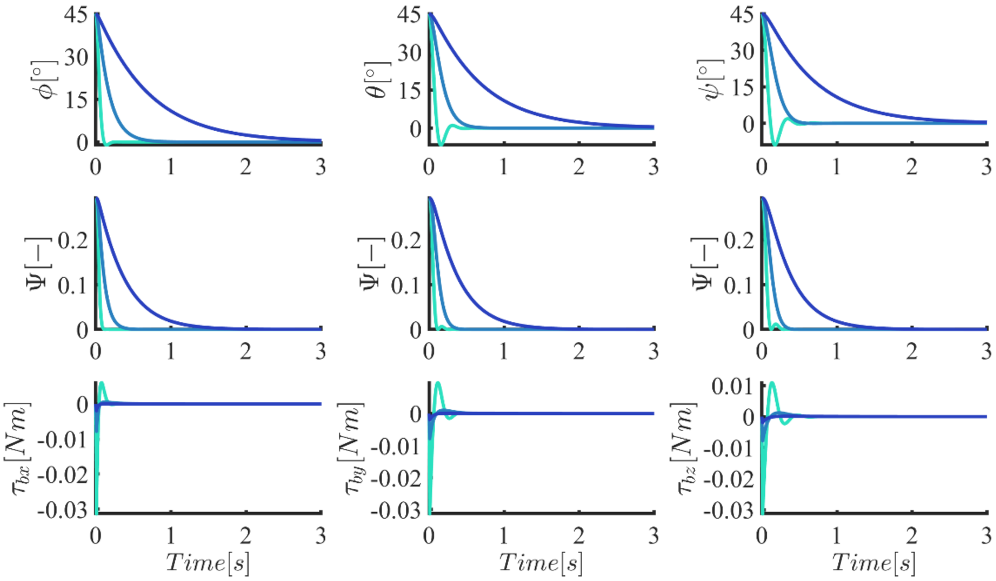

Figure 5-7: The Euler angles, angular error Ψ and control torques are shown for three controllers cases, namely a stiff controller ($4k_R$), nominal controller (k_R) and a damped controller ($0.25k_R$).

The characteristics of the nominal attitude controller are shown in Table 5-1.

Table 5-1: Characteristic parameters of the nominal controller by using the attitude control loop.

	$\boldsymbol{\phi}$	$\boldsymbol{\theta}$	$\boldsymbol{\psi}$
Rise time [s]	0.33	0.29	0.27
Overshoot [m]	0%	0%	0%
Settling time [s]	0.48	0.42	0.39

5.4.2. Position control gain parameters

By choosing the gains as $k_p = 16 \cdot m = 0.48,\ k_v = 7 \cdot m = 0.21$ for Equation (5-38), the position closed-loop system is critically damped. Once more observed in Figure 5-8, the stiff controller reduces the error faster and the damped controller has as slower response in comparison to the nominal controller. On the other hand, higher control effort, transients and overshoot are caused by using the stiff controller. Moreover, the nominal controller barely has a lower settling time and tuning k_p too high will result in an unstable closed-loop system. The attitude, position and velocity errors nicely converge to zero. No yaw torque is necessary for altitude tracking and therefore the thrust force is shown. The Nimble is actuated by the gravity force to lower the altitude until the thrust force attains stationary hovering behavior.

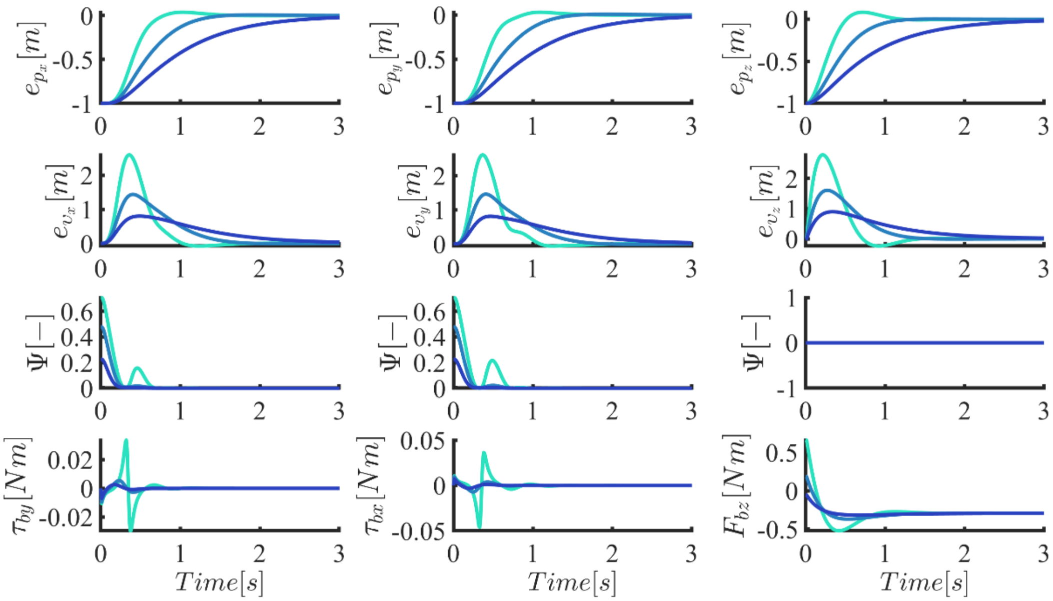

Figure 5-8: The position and velocity errors and control torques and forces are shown for three controllers cases, namely a stiff controller $(2k_p)$*, nominal controller* (k_p) *and a damped controller* $(0.5k_p)$*.*

The characteristics of the nominal attitude controller are shown in Table 5-2.

Table 5-2: Characteristic parameters of the nominal controller by using the position control loop.

	p_x	p_y	p_z
Rise time [s]	0.8	0.8	0.68
Overshoot [m]	0%	0%	0%
Settling time [s]	1.22	1.23	0.97

5.5. Concluding remarks of the nonlinear control design

As discussed in the problem statement, the Nimble does not have a trajectory control design or a nonlinear attitude control design. In this section a trajectory tracking control design is proposed for the DelFly Nimble based on the quadrotors and common control structures. The tracking errors are developed on $SE(3)$ which avoids singularities and ambiguities when using quaternions or Euler angels. Since the Nimble has four DoF, the attitude control loop can be used to track the desired states ϕ_d, θ_d and ψ_d, the desired altitude $z_d^{\mathcal{B}}$ and their derivatives. The position control loop can be used to track the desired states $x_d^{\mathcal{B}}$, $y_d^{\mathcal{B}}$, $z_d^{\mathcal{B}}$, the desired yaw angle ψ_d and their derivatives. By using a Lyapunov analysis, the stability properties are examined and proven to be almost global. Subsequently, the virtual inputs can be mapped to the control inputs by making use of the aerodynamic model. Altogether, when the virtual inputs are met, the equilibrium of the closed-loop attitude error dynamics is exponentially stable when $\varPsi\big(R(0), R_d(0)\big) < 2$. Furthermore, the equilibrium of the closed-loop position error dynamics is exponentially stable when the attitude error $\Psi\big(R(0), R_C(0)\big) < 1$ and shows almost global exponentially attractiveness within a region of attraction that covers the complete configuration space except for when the attitude error $\Psi\big(R(0), R_C(0)\big) < 2$. The attitude and position control loops can be used independently of each other and thereby can be employed to perform aerobatic maneuvers. The intended aerobatic maneuvers will be presented in Chapter 6.

6. Closed-loop system simulations

In this section simulations of the closed-loop system are presented that have been implemented in MATLAB. The tracking performance for the closed-loop system is discussed by analyzing the desired and actual trajectories. The simulations demonstrate performance of simple flight commands to show the performance of the control loops and highlights the performance of complex flight maneuvers by the DelFly Nimble. We distinguish between the closed-loop system with and without the aerodynamic model (as defined in Section 3.3 and Section 3.4). The closed-loop system of the rigid body assumes that the necessary control thrust force and torques are feasible and that no forces in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions are generated by the wings. On the other hand, closed-loop simulations incorporating the aerodynamic model impose damping forces in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions and potential input saturation. In particular, the control inputs are mapped to the thrust force and torques as in Equation (6-1).

$$\begin{bmatrix} \gamma \\ \beta \\ f_l \\ f_r \end{bmatrix} = \mathcal{F}\left[f_{b_z}\, \tau_{b_x}\, \tau_{b_y}\, \tau_{b_z}\right] \tag{6-1}$$

Therefore, the FCFs generated in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions are not accounted for. For clarity the orientation of the Nimble is expressed with Euler angles, even though the dynamic model and controller design makes use of rotation matrices. Lastly, note that the aim of designing a trajectory controller is different than trajectory generation. In this section a heuristic trajectory generation method is used.

First, in 6.1 the generation of trajectories is discussed. Thereafter, simple control trajectories are shown for both attitude and position commands in Sections 6.2 and 6.3. This enlarges the validity of the model and demonstrates the capabilities of the trajectory control design. Thereafter, in Section 6.4 a lateral aerobatic flip is demonstrated using the attitude control loop. In Section 6.5 the Nimble follows an elliptical helix using the position control loop. Lastly, in Section 6.6 the relation between input saturation and control gain parameters is investigated.

6.1. Trajectory generation

There exist two common approaches for trajectory generation [64]: 1. polynomial representations and 2. discretized state space formulations. Polynomial representations utilize the differentially-flat output states of the Nimble. These representations are computationally efficient in which trajectories go through multiple waypoints by concatenating polynomial segments and minimal-time solutions are provided given time segments and boundary conditions. Herewith minimum-snap trajectories can be established by minimizing the 4th derivative of the position to ensure smooth trajectories, making a trade-off between smoothness and minimum time. Discretized state space formulations use nonlinear optimization routines to plan trajectories described by a time-discretized state space of the system dynamics with input constraints. For the simulations the first approach is used.

Consider the trajectory generation problem for the Nimble. The dynamics with the four virtual inputs are differentially flat. Therefore, smooth trajectories can be generated and followed by the Nimble. A trajectory $\sigma(t)$ is defined to be a smooth curve in the space of flat outputs as

$$\sigma(t): [t_i, t_e] \rightarrow \mathbb{R}^3 \times SO(3). \tag{6-2}$$

Since the second derivative of position depends on the virtual force input and the fourth order derivative depends on the virtual torque input, the position control loop is a fourth order system. Therefore, the position trajectories with derivatives up to the fourth order are used (I-1). Attitude trajectories depend on the second order derivative of the virtual torque input; hence second order trajectories are used. Smooth trajectories are built by minimizing the state derivatives up to snap with a variable time span. Such a particular trajectory is minimized as the integral of the square of snap (note that $\mathcal{L}$ is not the Lagrangian in mechanics)

$$\sigma^{\star}(t) = \underset{\sigma(t)}{\operatorname{argmin}} \int_0^t \mathcal{L}(\sigma^n, \sigma^{n-1}, \dots, \dot{\sigma}, \sigma, t) dt \,|_{n=4} = \underset{\sigma(t)}{\operatorname{argmin}} \int_0^t \ddddot{\sigma}^2 dt\,. \tag{6-3}$$

Equation (6-3) can be employed in between trajectory fragments at specified times. Moreover, the optimal trajectories have smooth transitions. The design of the minimum snap trajectory is found by the Euler-Lagrange equation as,

$$\frac{\delta\mathcal{L}}{\delta\sigma} - \frac{d}{dt}\left(\frac{\delta\mathcal{L}}{\delta\dot{\sigma}}\right) + \frac{d^2}{dt^2}\left(\frac{\delta\mathcal{L}}{\delta\ddot{\sigma}}\right) - \frac{d^3}{dt^3}\left(\frac{\delta\mathcal{L}}{\delta\dddot{\sigma}}\right) + \frac{d^4}{dt^4}\left(\frac{\delta\mathcal{L}}{\delta\ddddot{\sigma}}\right) = \frac{d^4}{dt^4}\left(\frac{\delta\mathcal{L}}{\delta\ddddot{\sigma}}\right) = 0. \tag{6-4}$$

This can be solved by a seventh order polynomial in time with eight coefficients as

$$\begin{aligned} \sigma(t) &= c_o + c_1 t + \cdots + c_{2n-1} t^{2n-1}|_{n=4} \\ &= c_o + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 + c_6 t^6 + c_7 t^7 \end{aligned} \tag{6-5}$$

In order to solve for the coefficients, boundary conditions need to be specified. Subsequently, the minimum snap trajectory is obtained by using a quadratic program

(QP). A straight forward way to use the trajectories is to use piecewise continuous trajectories as in Equation (6-6).

$$\sigma(t) = \begin{cases} \sigma_1(t), & t_0 \leq t_1 \\ \sigma_2(t), & t_1 \leq t_2 \\ & \vdots \\ \sigma_k(t), & t_{k-1} \leq t_k \end{cases} \tag{6-6}$$

In addition, this method then can be used for generating the x, y and z-coordinates and the yaw angle ψ for the position loop by decoupling the Euler-Lagrange equations. Safely flying through space by avoiding obstacles can be implemented as well by using constrained QP as explained in [65].

6.2. Attitude recovery

In this section, attitude recoveries are discussed by using the attitude control loop design. Next to the attitude, the altitude is tracked as an auxiliary goal unless stated otherwise. Simulations are distinguished by using the closed-loop system with and without taking the aerodynamics of the Nimble into account. The nonlinear geometric controllers are capable to deal with large attitude errors and allow for aggressive maneuvering. Figure 6-1 shows attitude recovery from $60°$, $120°$ and $178°$ for ϕ, θ and ψ seperately in which the other angles are zero. The recovering trajectory from $\vartheta > 90°$ results in a large attitude error $\Psi\big(R(0), R_d(0)\big) > 1$. To begin with, the closed-loop system without input saturation shows almost global asymptotical behavior as can be seen from Figure 6-1 and Table 6-1. In other words, in agreement with Proposition 1, the system is able to exponentially recover from every orientation except for when the angular error $\Psi\big(R(0), R_d(0)\big) = 2$.

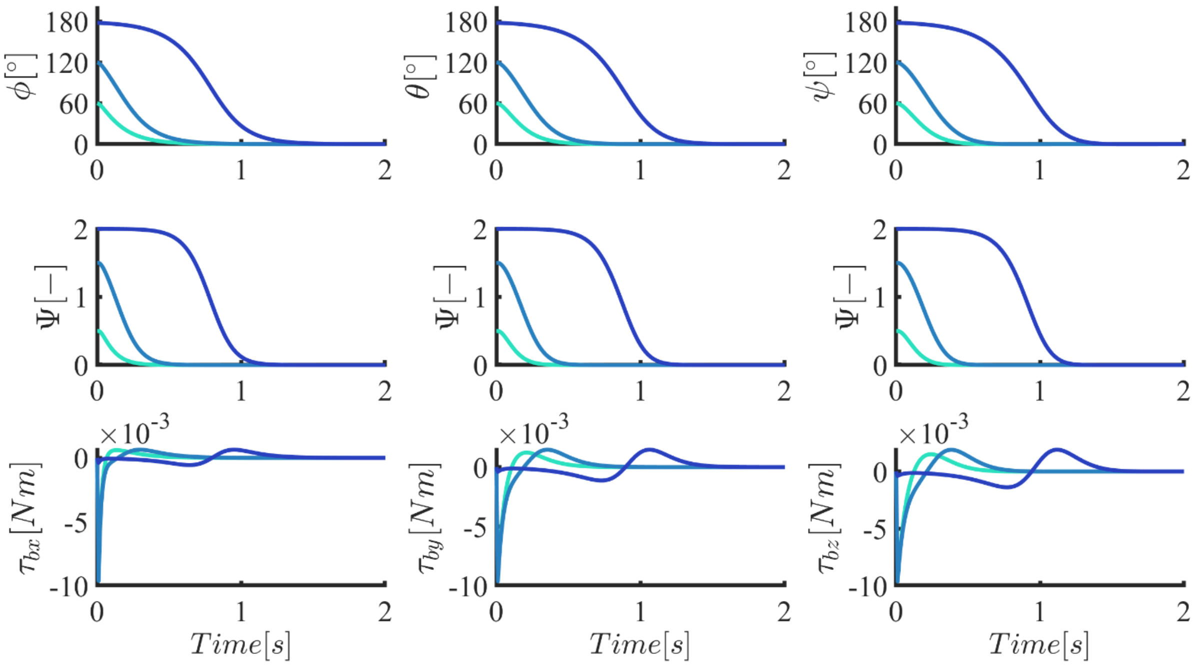

Figure 6-1: Attitude recovery from $[60°, 120°, 178°]$ *for the angles* ϕ, θ *and* ψ *seperately.*

Table 6-1: Characteristic parameters by using the attitude closed loop without aerodynamics.

	$\boldsymbol{\phi}$			$\boldsymbol{\theta}$			$\boldsymbol{\psi}$		
	60°	**120°**	**178°**	**60°**	**120°**	**178°**	**60°**	**120°**	**178°**
Rise time [s]	0.35	0.42	0.62	0.31	0.39	0.62	0.30	0.37	0.62
Overshoot [%]	0	0	0	0	0	0	0	0	0
Settling time [s]	0.50	0.58	1.17	0.44	0.55	1.21	0.42	0.53	1.23

Figure 6-2 shows attitude recoveries around the x-axis. The performance characteristics are given in Table 6-2.

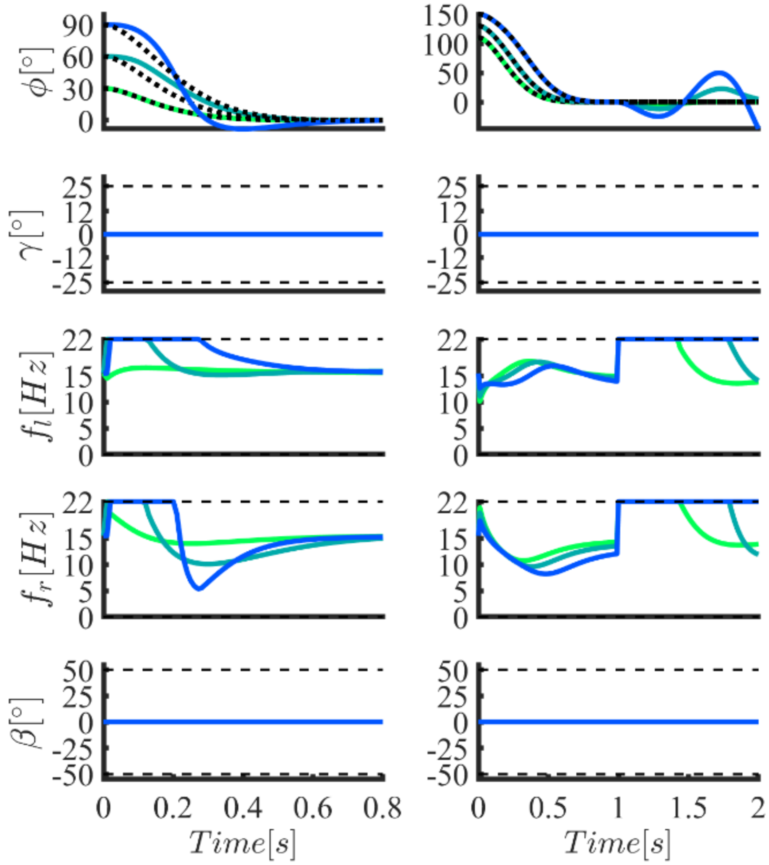

Figure 6-2: Attitude recovery from $\phi = [\,30°, 60°, 90°]$ *on the left and* $\phi = [110°, 130°, 150°]$ *on the right. The closed-loop behavior is indicated with (colored graphs) and without (black dotted graphs) the aerodynamic model. The control input bounds are indicated by the black dotted graphs.*

Moreover, the roll stabilization shows logical behavior of the control inputs. In order to stabilize a positive roll error, the frequency for the right wing increases and the frequency of the left wing decreases. This behavior is clearly visible when the Nimble recovers from $\phi = 30°$. After that, both wing frequencies increase to track the altitude. Furthermore, the control input angles γ and β remain unaffected.

By using the closed-loop system with control input saturation, the same desired behavior is obtained by recovering from $\phi = 30°$. When the Nimble recovers from higher angles ($\phi = 60°$ and $\phi = 90°$), control input saturation occurs and the recovery trajectories show different behavior. Nonetheless, the attitudes are still stabilized.

When recovering from higher angles ($\phi > 90°$), input saturation occurs as due to that the attitude controller demands negative thrust. This occurs due to that Equation (5-31) becomes negative as the term $z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}$ becomes negative. Therefore, the system cannot be stabilized anymore, since the Nimble is only able to generate positive thrust with flapping frequencies between $0\,Hz \leq f_l, f_r \leq 22\,Hz$. Even more, the $z^{\mathcal{B}}$-axis aligns with the horizontal, which is not allowed according to Equation (5-31).

To account for this, the thrust force can be manually set for a short time period $0 \leq t \leq 1$. In this case the thrust force is set to $0.2985\ N$. This is motivated by the fact that this thrust force corresponds to the flapping frequency necessary for hovering. Herewith, there is still enough room to provide differential flapping frequency ($f_l \ll 15Hz \ll f_R$) and prevent input saturation. Moreover, the higher the provided thrust force as it is rolling in the air, the further the Nimble flies away from the initial position. In summary $f_{b,z,v}$ can be divided into two domains as

$$f_{b,z,v} \coloneqq \begin{cases} \dfrac{k_p\left(p_z - p_{z,d}\right) + k_v\left(v_z - v_{z,d}\right) + mg - m\ddot{p}_{z,d}}{z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}}, & for \quad \phi, \theta < 90° \\ 0.2985, & for \quad \phi, \theta \geq 90° \end{cases} \qquad (6\text{-}7)$$

Following this approach shows the same behavior as the desired closed-loop behavior, stabilizing roll angles from up to $\phi = 180°$. From $t = 1\ s$, the z-position is tracked again. Immediately after that, input saturation of the flapping frequency occurs due to the high attained speeds of the body. Attitude recovery is still viable for $\phi = 110°$. Attitude recoveries for $\phi = 130°$ and $\phi = 150°$ show oscillatory behavior of the system due to saturation of the flapping-wing motors.

Apart from that ϕ or θ become higher than $90°$, $f_{b,z,v}$ can become too large already as the term $z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}$ becomes small. This causes input saturation of the flapping frequencies as well even when $k_p, k_v = 0$, due to that the feedforward term compensating for gravity $mg\left(z^{\mathcal{I}} \cdot Rz^{\mathcal{I}}\right)^{-1}$ becomes too large.

Table 6-2: Characteristic parameters for Figure 6-2. (Note that the settling time is given for $t < 1$)

$\boldsymbol{\phi}$ [°]	**Rise time [s]**	**Settling time [s]**	**Overshoot [%]**
30	0.28	0.39	0.0
60	0.30	0.47	0.0
90	0.16	0.49	6.5
110	0.36	0.49	0.0
130	0.40	0.56	0.0
150	0.47	0.65	0.0

Figure 6-3 shows attitude recoveries around the y-axis. As well as for roll recoveries, the pitch recoveries show that the closed-loop system without the aerodynamics shows almost global asymptotical behavior. In addition, the pitch stabilization shows reasonable behavior of the control inputs. In order to stabilize a positive pitch error, the dihedral angle γ becomes negative. The wing frequencies simultaneously increase to generate more thrust and maintain altitude tracking. Furthermore, the control input angle γ decreases for a positive pitch and the control input angle β remains unaffected. Secondly, by using the closed-loop system with input saturation, the same desired behavior is obtained by recovering from $\theta = 30°$ and $\theta = 60°$. When the Nimble recovers from higher angles ($\theta = 90°$), input saturation occurs for the control inputs γ, f_l and f_r and the recovery trajectories show different behavior. Even so, the attitudes are stabilized.

When recovering from even higher angles ($\theta \geq 90°$), negative thrust is demanded as for roll recovery. Therefore, the system cannot stabilize anymore when the z-position and the orientation is tracked simultaneously. Once more the thrust force is set to $0.2985\,N$ as in Equation (6-7), for which the angles can be stabilized from up to $\theta = 180°$. From $t = 1\,s$, the z-position is tracked, giving rise to saturation due to the attained high speeds of the body. Attitude recovery shows oscillatory behavior for $\theta = 150°$ due to saturation of the flapping-wing motors. Lastly, the performance characteristics are given in Table 6-3.

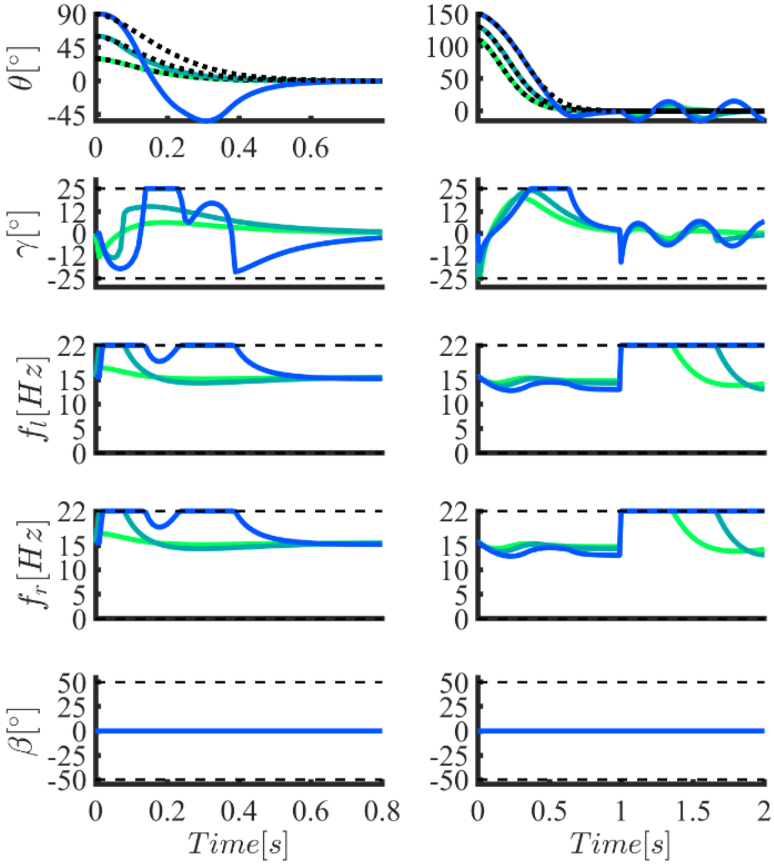

Figure 6-3: Attitude recovery from $\theta = [30°, 60°, 90°]$ on the left and $\theta = [110°, 130°, 150°]$ on the right. The closed-loop behavior is indicated with (colored graphs) and without (black dotted graphs) the aerodynamic model.

Table 6-3: Characteristic parameters for Figure 6-3. (Note that the settling time is given for $t < 1$)

$\boldsymbol{\theta}$ [°]	**Rise time [s]**	**Settling time [s]**	**Overshoot [%]**
30	0.30	0.41	0.0
60	0.27	0.39	0.0
90	0.09	0.62	53.2
110	0.37	0.51	0.0
130	0.41	0.57	0.0
150	0.40	0.77	7.9

Figure 6-4 shows attitude recoveries around the z-axis. Almost global asymptotical behavior is obtained both with and without taking the aerodynamics into account. The Nimble achieves yaw recovery with ease, since body speeds remain moderate. In addition, when $\gamma = 0°$, no forces are generated in the x- and y-directions due to symmetry. The frequencies slightly increase due to the tilt of the thrust vectors β changes. In order to stabilize a positive yaw error, the wing root angle β becomes positive and the dihedral angle γ remains unaffected. The trajectories recovering from $\psi = 60°$ and $\psi = 90°$ slightly deviate from the ideal case due to input saturation of the wing root angle. Lastly, the performance characteristics are given in Table 6-4.

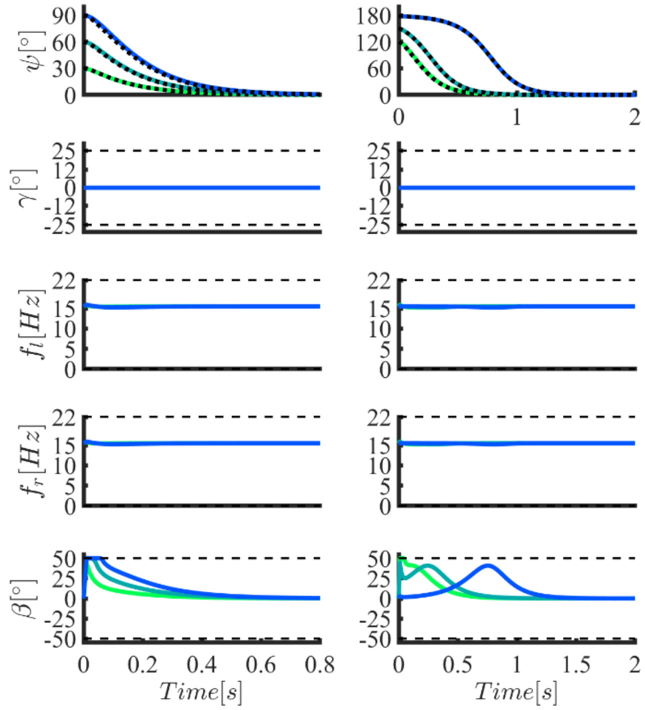

Figure 6-4: Attitude recovery from $\psi = [30°, 60°, 90°]$ on the left and $\psi = [120°, 150°, 178°]$ on the right. The closed-loop behavior is indicated with (colored graphs) and without (black dotted graphs) the aerodynamic model.

Table 6-4: Characteristic parameters for Figure 6-4. (Note that the settling time is given for $t < 1$)

ψ [°]	Rise time [s]	Settling time [s]	Overshoot [%]
30	0.34	0.47	0
60	0.35	0.49	0
90	0.37	0.53	0
110	0.41	0.58	0
130	0.48	0.68	0
150	0.62	1.16	0

In Figure 6-5 attitude recovery is shown from the angles $[\phi, \theta, \psi] = [140°, 40°, 10°]$ with $\Psi(R(0), R_d(0)) = 1.21$. At the same time the altitude is tracked as well. The simulation shows asymptotical performance and since there is no input saturation, the closed-loop behavior for the orientation is the same with and without considering the aerodynamics. Since the Nimble has four DoF, the attitude control loop can be used to track the desired states ϕ_d, θ_d and ψ_d, the desired altitude $z_d^{\mathcal{B}}$ and their derivatives. Therefore, the altitude is being tracked, but the states $x^{\mathcal{B}}$ and $y^{\mathcal{B}}$ are not. Due to the aerodynamics, the translational motion is damped by the FCFs and body damping in contrast to the simulation without aerodynamics. Note that since only the attitude and altitude are controlled in this case, the x- and y- positions do not converge.

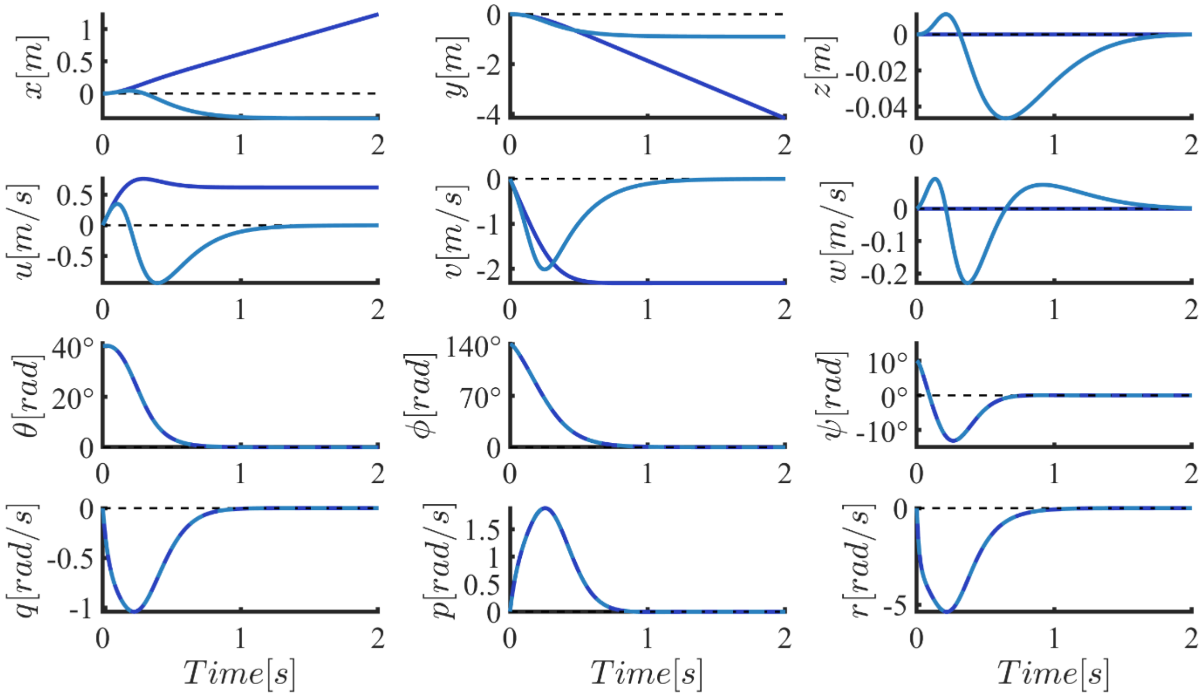

Figure 6-5: Attitude recovery from $[\phi, \theta, \psi] = [140°, 40°, 10°]$. The closed-loop behavior is indicated with (dark blue graphs) and without (light blue graphs) the aerodynamic model.

6.3. Position recovery trajectories

In this section, simple smooth trajectory for the position loop and attitude loop are discussed. First longitudinal trajectories are shown for different waypoints in Figure 6-6 with considering the aerodynamics. The same trajectories are shown in Appendix IV.a without considering the aerodynamics, and for different initial conditions to show the performance of the controller design. The states are initialized with the position and its derivatives equal to zero with waypoints of $x_e = 0.25\ m, 0.75\ m$ and $1.00\ m$ within a time frame of one second. The trajectory controller is able to track the desired trajectories well with negligible deviation from the desired trajectory when the aerodynamics are not taken into account. In addition, the trajectory can be tracked up to a certain steepness as the commanded acceleration as in Equation (5-53) is bounded as $\|mgz^{\mathcal{I}} + m\dot{v}_d\| < B$.

Due to the FCF in the $x^{\mathcal{B}}$-direction (Figure 6-8), the trajectories of the closed-loop system deviate from the desired trajectories. Furthermore, the trajectory with $x_e = 1.00\ m$ shows input saturation of the dihedral angle γ as can be observed in Figure 6-7. The FCF causes a nose up pitch for which the system cannot compensate until the velocity has decreased again. Not only the rise time and settling time may become longer but also become unpredictable as for example the target $p_x = 0.75m$ has a lower rise and settling time than the target $p_x = 0.25m$. The FCF D, defined in Section 3.3, is a multiplicative function of the control inputs, linear and angular body speeds as

$$D(\gamma, f_l, f_r, \beta, v_b, \omega_b) = f_1(\gamma, f_l, f_r, \beta) \cdot f_2(v_b, \omega_b). \qquad (6\text{-}8)$$

When the linear and angular body speeds go to zero, the FCF goes to zero, i.e. $v_b, \omega_b \to 0 \Rightarrow D \to 0$. Therefore, the state converges to the desired state.

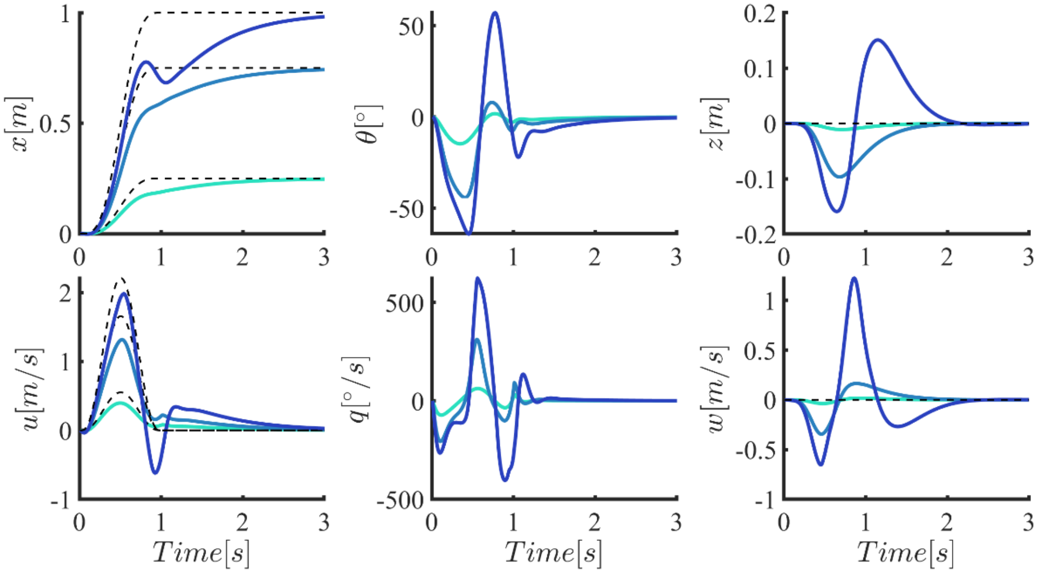

Figure 6-6: States of longitudinal trajectories with end positions $x_e = [0.25\ m\ 0.75\ m\ 1.00\ m]$. The black dotted graphs represent the desired trajectory, the green graphs represent the closed-loop behavior when $f_{b,x} = 0$ and the blue graphs show the closed-loop behavior when $f_{b,x} \neq 0$.

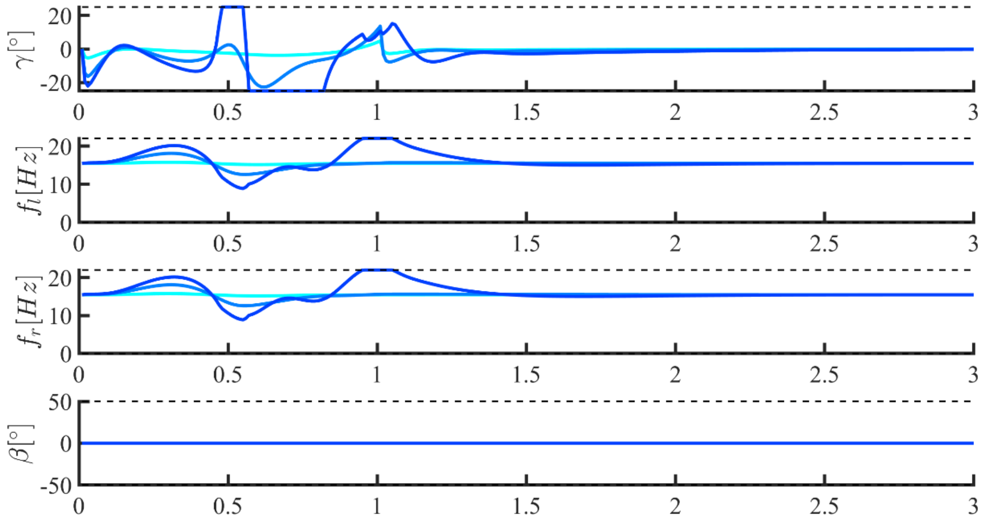

Figure 6-7: Control inputs of longitudinal trajectories with end positions $x_e = [0.25\,m\ 0.75\,m\ 1.00\,m]$. The black dotted graphs represent the control input constraints and the blue graphs show the closed-loop behavior when $f_{b,x} \neq 0$.

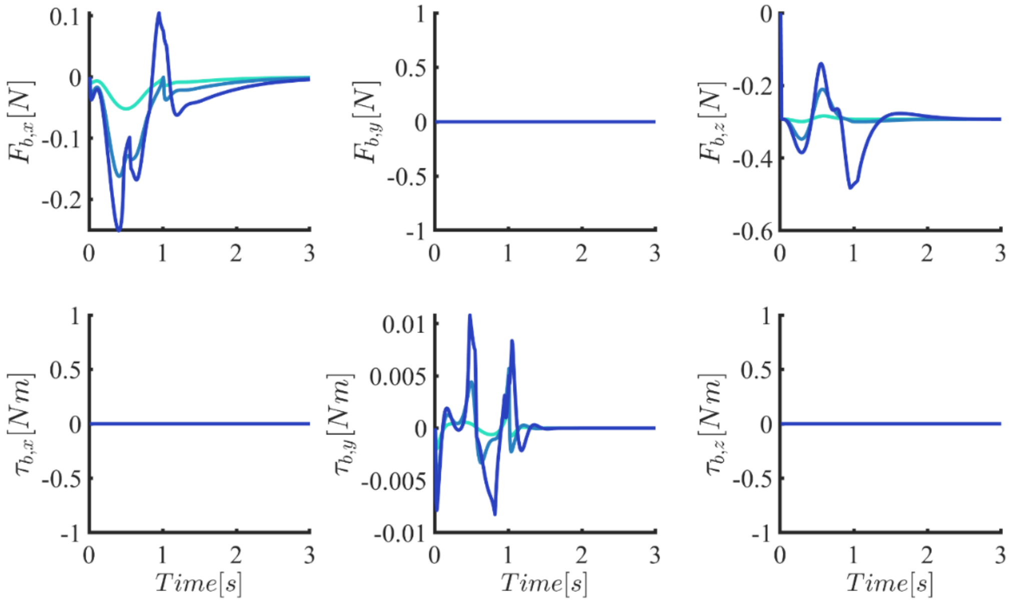

Figure 6-8: Forces and torques of longitudinal trajectories with end positions $x_e = [0.25\,m\ 0.75\,m\ 1.00\,m]$. The blue graphs show the closed-loop behavior when $f_{b,x} \neq 0$ and the green graphs represent the closed-loop behavior when $f_{b,x} = 0$.

Table 6-5: Characteristic parameters by using the position closed loop with aerodynamics for Figure 6-6.

	$\boldsymbol{p_x\ [m]}$		
	$\mathbf{0.25}$	$\mathbf{0.75}$	$\mathbf{1.00}$
Rise time [s]	1.27	1.17	1.51
Overshoot [%]	0.00	0.00	0.00
Settling time [s]	1.97	1.89	2.17

Secondly, lateral trajectories are shown for different waypoints in Figure 6-9. The states are initialized with the position and its derivatives equal to zero with waypoints of $y_e = 0.25\,m, 0.82\,m$ and $1.1\,m$ within a time frame of one second. As the Nimble is flying in the lateral direction, forces are generated in y-directions due to the FCF (Figure 6-11). In addition, the trajectory with $y_e = 1.1\,m$ shows input saturation for the flapping frequencies f_l and f_r as can be observed in Figure 6-10. Again, when the force in the y-direction is not taken into account, the trajectory controller is able to track the desired trajectories well with negligible deviation from the desired trajectory. The trajectory tracking shows good performance and the tracking error is zero when the tracking target is achieved. The trajectories are shown in Appendix I.a, where the graphs are shown for when $f_{b,\mathrm{y}} = 0$ only and for different initial conditions to show the performance of the controller design.

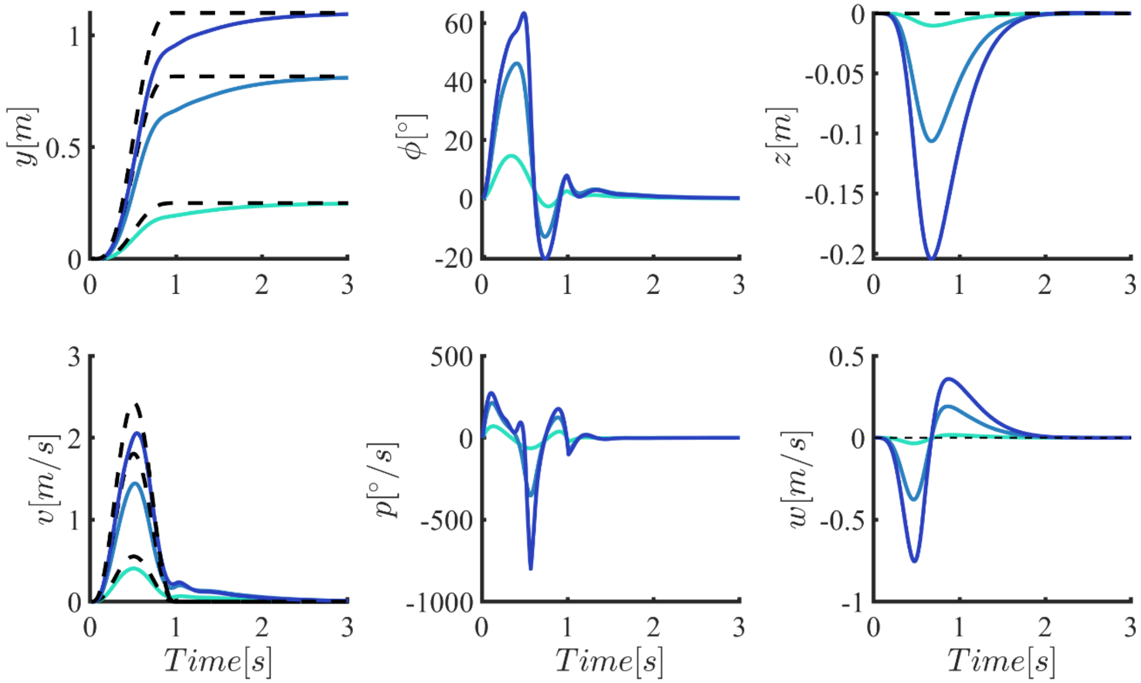

Figure 6-9: States of lateral trajectories with end positions $y_e = [0.25\,m\ 0.82\,m\ 1.1\,m]$. The black dotted graphs represent the desired trajectory, the green graphs represent the closed-loop behavior when $f_{b,y} = 0$ and the blue graphs show the closed-loop behavior when $f_{b,y} \neq 0$.

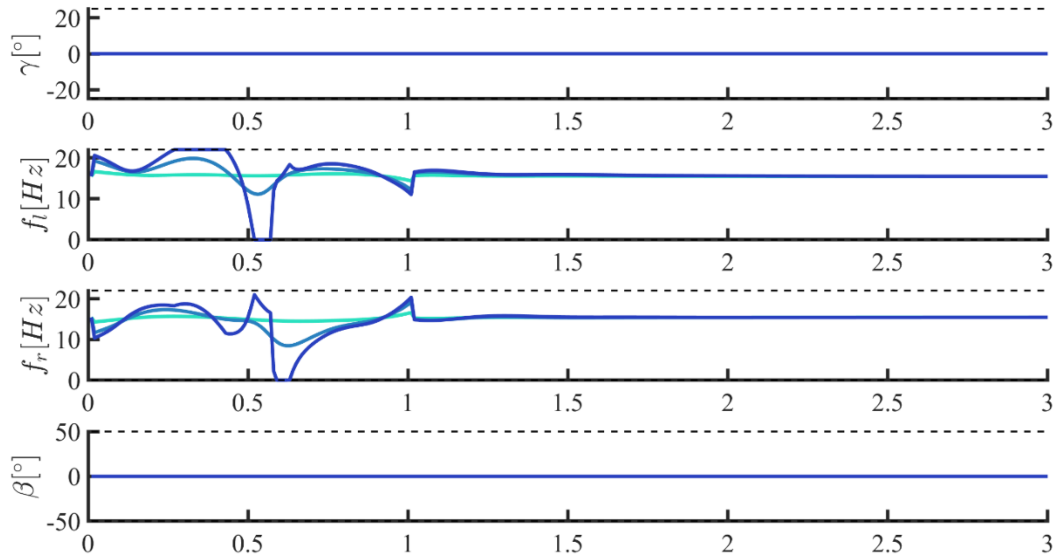

Figure 6-10: Control inputs of longitudinal trajectories with end positions $y_e = [0.25\,m\ 0.82\,m\ 1.1\,m]$. The black dotted graphs represent the control input constraints and the blue graphs show the closed-loop behavior when $f_{b,y} \neq 0$.

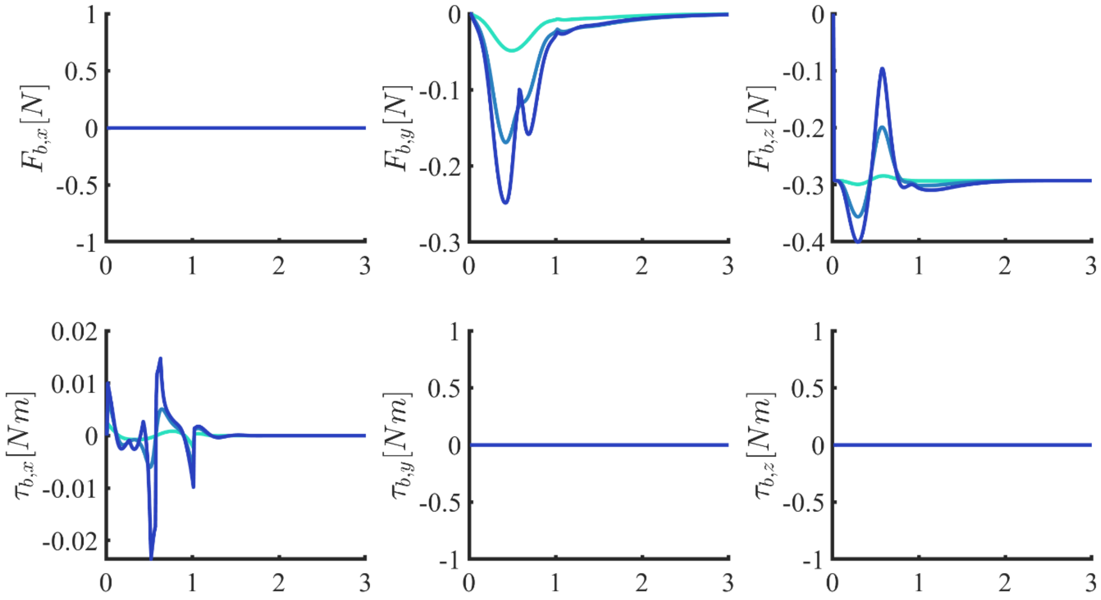

Figure 6-11: Forces and torques of lateral trajectories with end positions $y_e = [0.25\,m\ 0.82\,m\ 1.1\,m]$. The blue graphs show the closed-loop behavior when $f_{b,} \neq 0$ and the green graphs represent the closed-loop behavior when $f_{b,y} = 0$.

Table 6-6: Characteristic parameters by using the position closed loop with aerodynamics for Figure 6-9.

	$\boldsymbol{p_y}\ [\boldsymbol{m}]$		
	0.25	**0.82**	**1.10**
Rise time [s]	1.21	1.09	0.81
Overshoot [%]	0.00	0.00	0.00
Settling time [s]	1.90	1.80	1.56

6.4. 360° flip maneuver

The Nimble is able to perform 360° roll and pitch flips. In more detail, this maneuver is performed in open loop and consists of (1) gaining altitude while using attitude control, (2) initiate full positive roll/pitch torque until $\phi/\theta = 90°$, (3) no roll/pitch torque until $\phi/\theta = 270°$ with 26% throttle command, (4) recovering with attitude feedback to hover configuration with 94% throttle, and (5) return to hover the hover configuration [10]. In the real-life experiments, the Nimble performs a flip maneuver in $0.7\, s$ and reaches angular rates of $1000° s^{-1}$ for roll and $800° s^{-1}$ for pitch. Interestingly, these numbers are observed in simulation as well.

By using the nonlinear attitude controller, the Nimble is able to flawlessly perform flip maneuvers with great accuracy in closed loop. Nevertheless, the flip maneuvers are difficult to execute in simulation due to rapid occurring input saturations when the aerodynamics are taken into account. When the timespan is too short, input saturation occurs due to that the motors cannot generate the fast motion. On the other hand, when the timespan is too long, input saturation occurs due to the attained speeds of the Nimble. For lateral flips, input saturation does not occur for time spans $0.9 < t_e < 1.8$. As for recovering from certain orientations higher than $90°$, it is not possible to both stabilize attitude and altitude at the same time. Therefore, the Nimble has to fly upwards and roll as how flips are performed in demonstrations in open loop. In Figure 6-12 a lateral flip maneuver is shown. The desired states and real states are depicted by the black and blue graphs respectively. The Nimble first flies upwards, subsequently rolls and eventually stabilizes to the hover configuration. Due to that the Nimble cannot generate negative thrust, $f_{b,z}$ is set to the force necessary for hovering at $2.6 < t < 3.1$ seconds. The dihedral and wing root angles remain zero as no yaw or pitch torques are needed. Furthermore, the wing frequencies simultaneously enlarge to fly upwards, followed with a higher magnitude for f_l than f_r to generate a positive roll torque. The non-smooth control input behavior is causes by the switching of $f_{b,z}$. The deviation between the desired and simulated states are caused by input saturation.

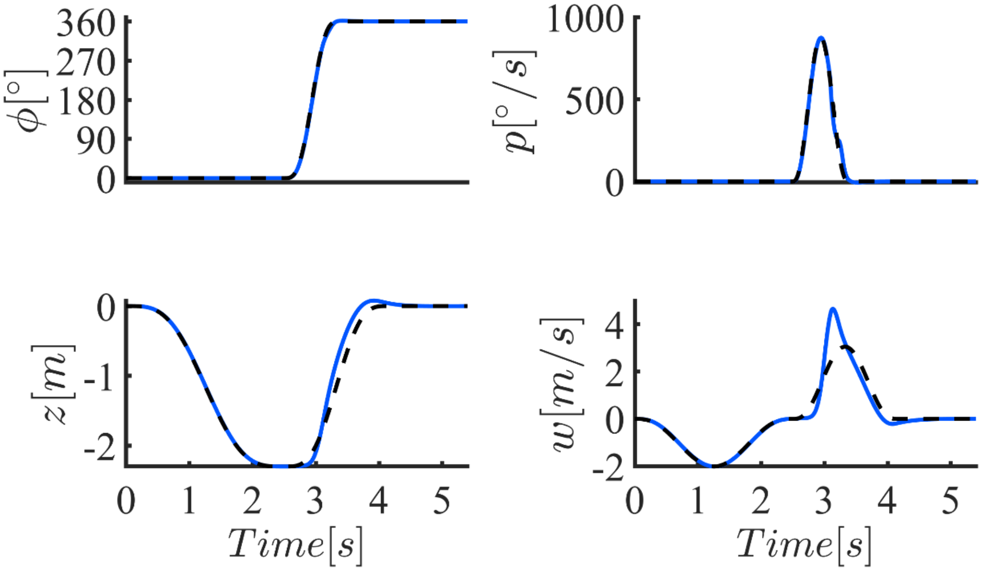

Figure 6-12: Lateral states of the lateral flip. The desired states and real states are depicted by the dotted black and blue graphs respectively. The desired state is not followed for $2.6 < t < 3.1$.

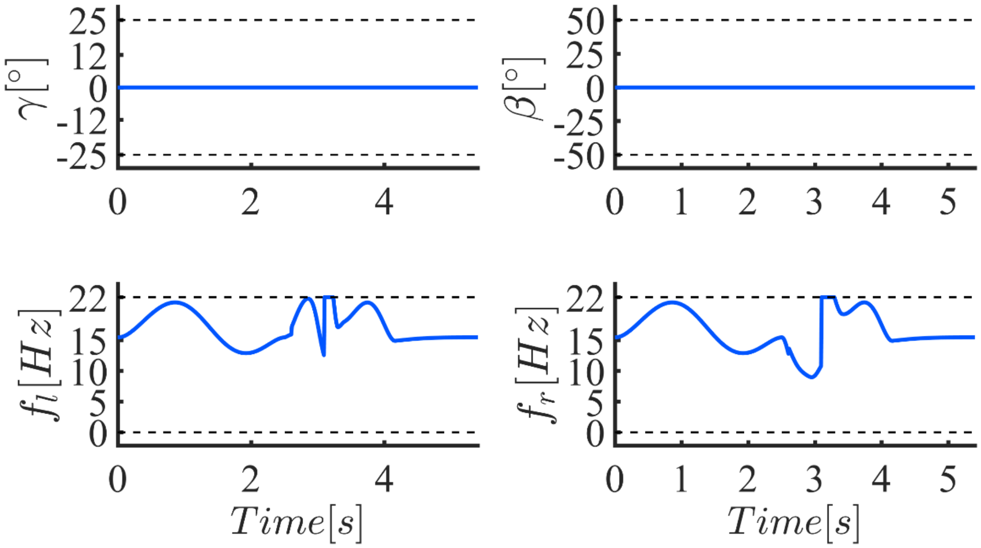

Figure 6-13: Lateral control inputs of the lateral flip.

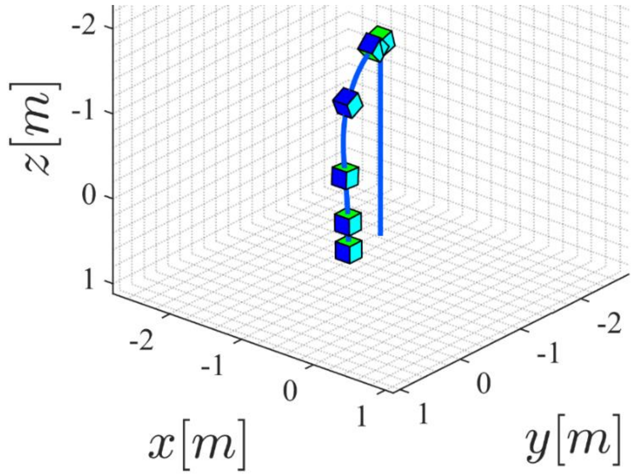

Figure 6-14: Lateral flip visualized in 3D. A cube is used to visualize the orientation of the Nimble. The green, light blue and dark blue colors represent the top, front and side planes respectively. The blue graph represents the traveled trajectory.

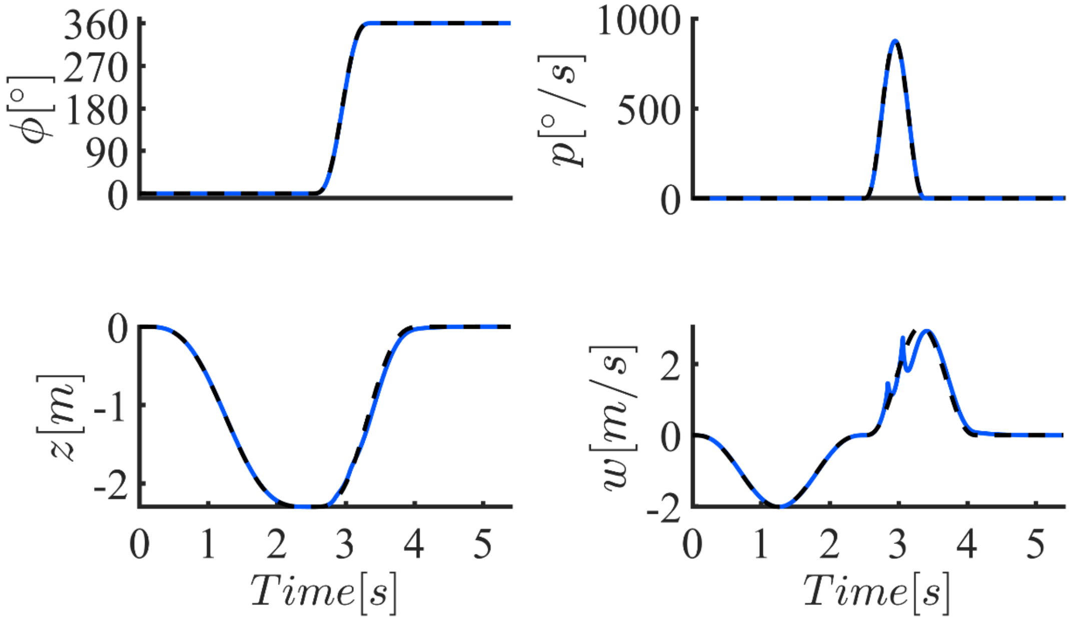

Figure 6-15: Lateral states of the lateral flip without taking the FCF into account. The desired states and real states are depicted by the dotted black and blue graphs respectively. The desired state is not followed for $2.6 < t < 3.1$*.*

6.5. Following an elliptic helix

In the figures below, a maneuver is depicted of following an elliptic helix. This maneuver demonstrates the position closed-loop design. The smooth trajectory is designed to just reach input saturation. Moreover, this trajectory is generated to investigate the response on tracking multiple outputs at the same time.

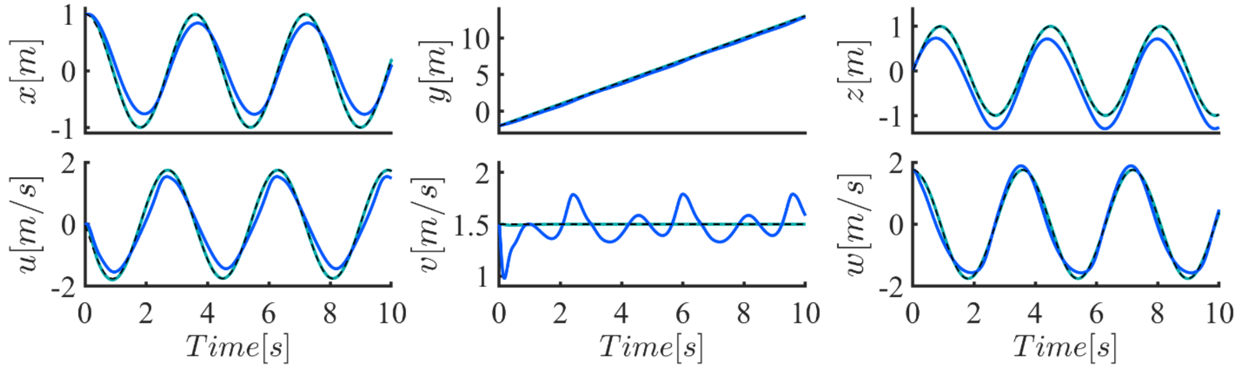

Figure 6-16: Position and linear velocities of the DelFly Nimble. The dotted black, green and blue graphs represent the desired closed-loop behavior and the closed-loop behavior with and without the aerodynamic model.

This flight trajectory requires that the Nimble excites all control inputs while following the position $\in \mathbb{R}^3$ and yaw command. The desired tracking commands are defined as

$$p_d = \begin{bmatrix} \cos(1.75t) \\ 1.7t - 2 \\ \sin(1.75t) \end{bmatrix}, \qquad x_c^{\mathcal{B}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \forall t. \tag{6-9}$$

The derivatives of p_d are smoothly defined for up to the fourth order and the derivatives of $x_c^{\mathcal{B}}$ are smoothly defined for up to the second order. The initial conditions are chosen as

$$p_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, v = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, R_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \omega_b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6-10}$$

The green graphs show the closed-loop performance when the aerodynamic model is not taken into account, i.e. $\mathrm{f}_{\mathrm{b,x}} = \mathrm{f}_{\mathrm{b,y}} = 0$. The initial attitude error $\Psi < 1$ and therefore, according to proposition 2, the position tracking error exponentially converges to zero. When the FCF is incorporated in the simulation, the Nimble is still able to follow the trajectory with some deviation until the Nimble comes to a stop.

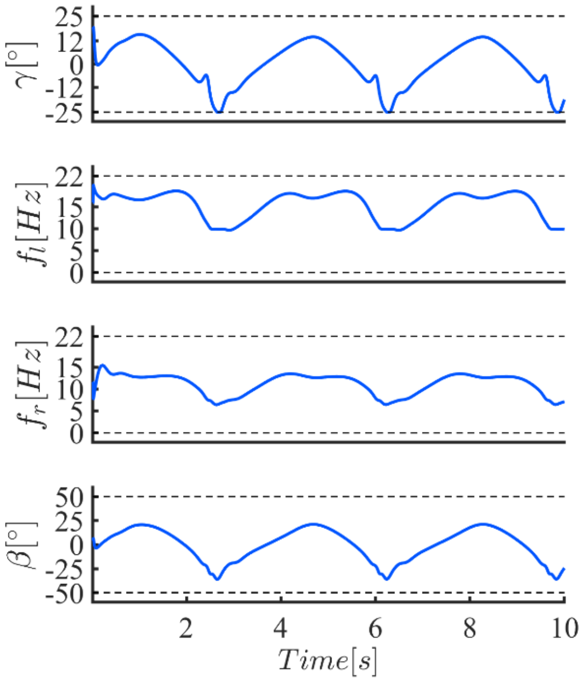

Figure 6-17: Control inputs of the DelFly Nimble. The black dotted graphs represent the control input bounds.

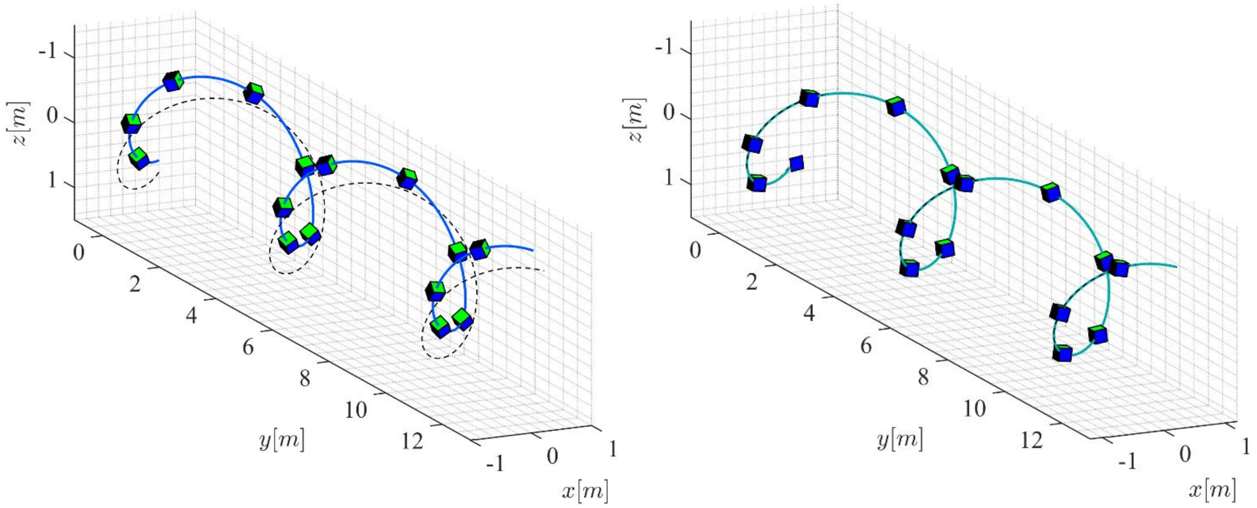

Figure 6-18: 3D trajectory of the DelFly Nimble. The dotted black, green and blue graphs represent the desired closed-loop behavior and the closed-loop behavior with and without the aerodynamic model. The green, light blue and dark blue colors of the cube represent the top, front and side planes respectively.

6.6. Input saturation and control gain parameters

The angular and linear acceleration errors are given by (as defined in Equations (5-16) and (5-39))

$$\begin{aligned} J\dot{e}_{\omega_b} &= -k_R e_R - k_{\omega_b} e_{\omega_b}, \\ m\dot{e}_v &= -k_p e_p - k_v e_v - X. \end{aligned} \tag{6-11}$$

It is desired to choose high gains for k_R, k_{ω_b}, k_p and k_v to guarantee fast exponential convergence of the error dynamics. The consequence on the other hand is that input saturation can occur due to that the controller reacts in an aggressive manner to errors in the position or attitude. Input saturation arises when physical limits are reached by any part of the feedback control system. Physical limitations are for example when certain forces or torques are required that the actuators cannot generate or that the Nimble cannot provide negative thrust. This decreases performance and even may render the closed-loop system unstable. Another method would be to implement a feasible trajectory generator. This would mean that feasible trajectories are generated in which errors remain sufficiently low to prevent input saturation. A trajectory generator for this system is out of the scope of this project. In this section the connection between input saturation and control gain parameters will be investigated. A trade-off can be accomplished by lowering the gains, that ensures that input saturation happens less quickly, though with slower convergence of the error dynamics. An example is given for tracking a lateral position trajectory and compared with Figure 6-9. For the dark blue graph, the gains k_p and k_v are multiplied by 0.5. The dark blue graph shows that no input saturation occurs anymore, but has higher rise and settling time as shown in Table 6-7.

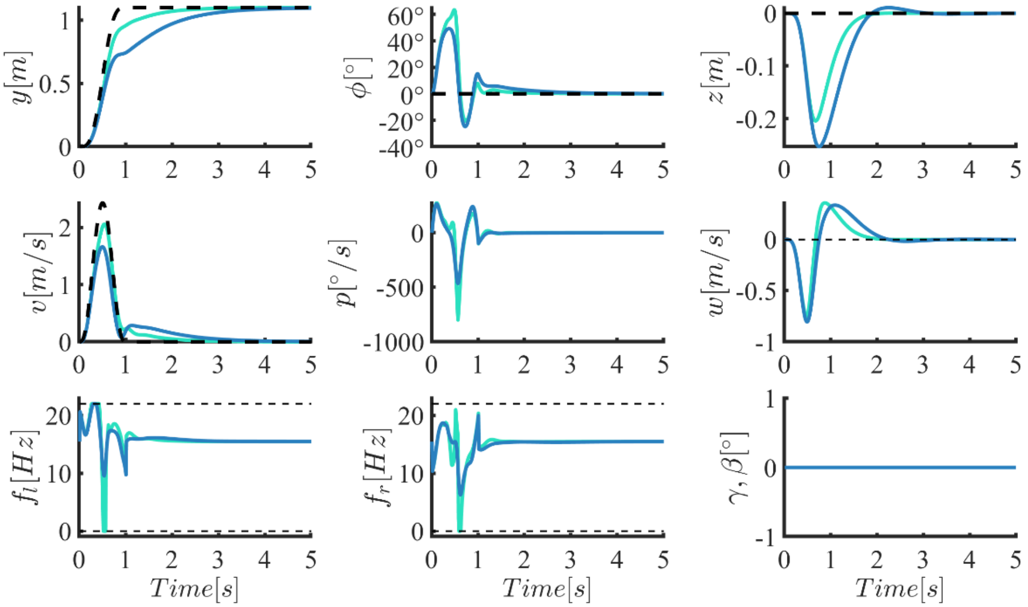

Figure 6-19: States of lateral trajectories with end positions $y_e = 1.1m$. The black dotted graphs represent the desired trajectory and the blue graphs show the closed-loop behavior. The gains k_p and k_v for the light blue graph are multiplied by 0.5.

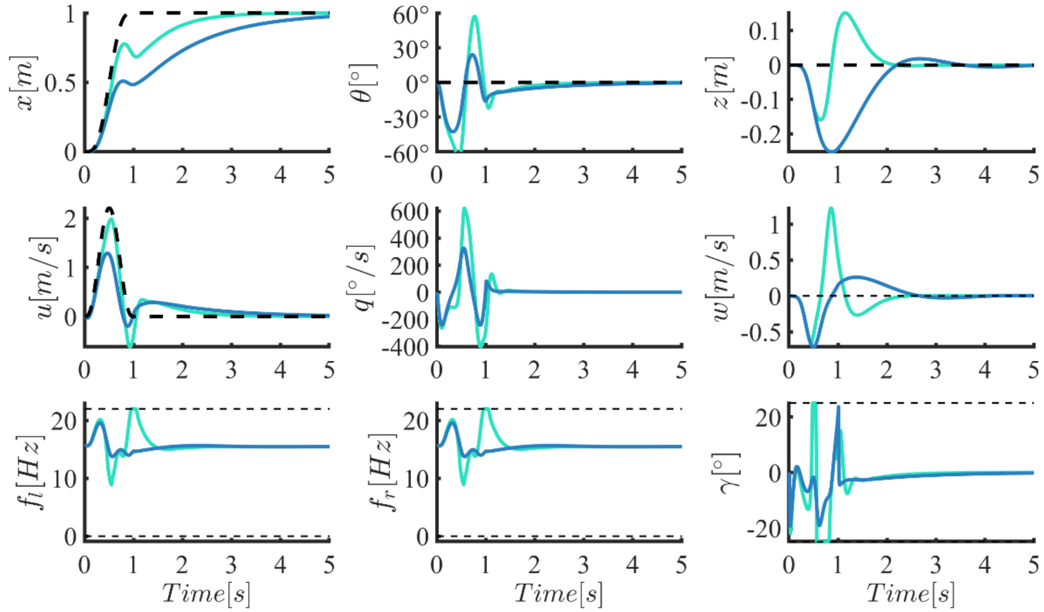

Figure 6-20: States of lateral trajectories with end positions $x_e = 1m$. The black dotted graphs represent the desired trajectory and the blue and green graphs show the closed-loop behavior. The gains k_p and k_v for the green graph are multiplied by 0.5.

Table 6-7: Characteristic parameters by using the position closed loop with aerodynamics for Figure 6-19 and Figure 6-20.

	p_x [m]		p_y [m]	
Multiplication gain [-]	0.5	1	0.5	1
Rise time [s]	2.69	1.60	1.79	0.81
Overshoot [%]	0.00	0.00	0.00	0.00
Settling time [s]	3.67	2.36	2.66	1.56

6.7. Concluding remarks of the closed-loop simulations

The attitude and the trajectory control designs were tested in simulation to verify the closed-loop behavior of the Nimble. Numeric simulations were shown both for with and without the aerodynamic model. The simulations without incorporated aerodynamics prove the nonlinear controller. The simulations with incorporated aerodynamics showed that the commanded virtual control inputs can be exceeded and revealed input saturation. Due to input saturation, the performance deteriorates. Next to input saturation, FCF are generated in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions, which are not accounted for in the control design.

Since the control inputs are computed from the desired virtual control inputs, the damping in the $z^{\mathcal{B}}$-direction and the flapping counter torques around the pitch, roll and yaw axes are accounted for. Only, the aerodynamic damping in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions are not considered in the control design, as the Nimble can only track 4 DoF. Therefore, in contrast in simulations where the aerodynamics are not considered, motion in the $x^{\mathcal{B}}$- and $y^{\mathcal{B}}$-directions is generated when the aerodynamics are considered in simulations. Nevertheless, as the linear and angular body speeds go to zero, the FCF also goes to zero and accordingly the trajectory target can be achieved.

The Nimble is able to recover from an attitude error, even from the upside-down orientation. However, when the altitude is tracked as an auxiliary goal, input saturation restricts the performance when the aerodynamic model is taken into account. In order to bypass this issue, altitude tracking is turned off for a short duration until the attitude error has recovered. Furthermore, the position recovery trajectories show that the translational error dynamics exponentially decrease to zero when the aerodynamics are not used. When the aerodynamics are used, FCFs are observed and performance decreases. Perhaps the used damping constants are too high as the FCF are substantial. Even so, the presented position recovery trajectories still converge to the desired equilibrium. Furthermore, a trade-off can be accomplished by lowering the controller gain values. This ensures that input saturation happens less quickly, although with slower convergence of the error dynamics.

The results show that the closed-loop system remains stable with sufficient tracking performance for both recovery tasks and aerobatic maneuvers. The aim was to perform aerobatic maneuvers such as a $360°$ flip and to follow an elliptic helix trajectory. By using the nonlinear attitude controller, the Nimble is able to flawlessly perform flip maneuvers with great accuracy in closed loop. In order to prevent input saturation, altitude was gained before carrying out the flip and the duration of the flip has to be in a certain time frame. When the timespan is too short, input saturation occurs due to that the motors cannot generate the fast motion. Conversely, when the timespan is too long, input saturation occurs due to the attained speeds of the Nimble. The elliptic helix, a trajectory exciting all outputs and control inputs in the position closed loop, showed excellent performance without using the aerodynamics. When the using the aerodynamics, the FCFs caused the Nimble to remain slightly behind the desired trajectory.

7. Conclusions and recommendations

This study researches the DelFly Nimble and the connection between biology and robotics. Very little is known about the dynamics and aerodynamics of the DelFly Nimble, especially in terms of how to physically model the behavior of the flow and the flapping wings. In order to perform autonomous tasks in unknown environments, knowledge of flight control and aerodynamics is required. For autonomously tracking certain trajectories, the design of appropriate tracking controllers is of the essence. This thesis addresses the modeling and tracking control problems for a particular kind of unmanned aerial vehicles: the DelFly Nimble. A nonlinear dynamic model and tracking control laws are proposed with as main purpose to perform simple to aggressive aerobatic maneuvers.

7.1. Conclusions

A nonlinear global dynamic model of the DelFly Nimble is developed.

- ***From the dynamic model it is evident that the translational and rotational dynamics of the DelFly Nimble are coupled and nonlinear.***
  The first contribution of this thesis is the physics-based global dynamic model of the DelFly Nimble that has been derived on the configuration space $SE(3)$. The research aim was split into the research of a dynamical model and an aerodynamic model for the DelFly Nimble. The dynamic model consists of a rigid body model in which the generalized forces and torques are outputs of the aerodynamic model. The attitude of the rigid body dynamics is parameterized by means of rotation matrices, whereby topological obstructions can be taken into account by design, resulting in a singularity free and global attitude representation. From the dynamic model, it is evident that the translational and rotational dynamics of the DelFly Nimble are coupled and nonlinear.
- ***The aerodynamic model is implemented in $\mathbb{R}^6$ by using a quasi-steady approach. The aerodynamics are derived by a drag model that is applied on the CoPs of the wings.***
  An understanding of the aerodynamic force mechanisms and a full dynamic model was still missing. The aerodynamic model is implemented in $\mathbb{R}^6$ by using a quasi-steady approach. The aerodynamics are derived by a drag model applied on the CoPs of the wings. The control inputs rotate and translate the wing frame and thereby control the thrust and damping forces encountered by the wings. This drag model captures the flapping counter forces and torques of the DelFly Nimble that are observed in real life experiments.

- ***The dynamic model is predictive and allows for exploiting the dynamics of the system in different ways. It allows for tuning physical and control parameters to improve flight performance or efficiency. It also provides understanding as to how the DelFly Nimble will behave in certain flight configurations depending on the control inputs and body speeds. Trajectories can be simulated and it can be tested whether these are feasible or not.***

  One aim of the nonlinear dynamic model of the DelFly Nimble was that it should allow for simulations and parameter optimization. Even trajectories can be tested that are not feasible in real life. Additionally, regardless of the substantial variation in size, mass, wing kinematics and wing morphology among FWMAVs, they all undergo similar passive aerodynamic damping and therefore, this model could potentially also be used for other FWMAVs.

A nonlinear trajectory tracking controller for the DelFly Nimble is proposed.

- ***Two output tracking controllers are proposed. When input saturation does not occur:***
  - ***The attitude error dynamics are almost global exponentially stable.***
  - ***The position error dynamics are exponentially stable when $\boldsymbol{\Psi}(\boldsymbol{R}(\mathbf{0}),\boldsymbol{R_c}(\mathbf{0})) < \mathbf{1}$ and exponentially attractive when $\mathbf{1} < \boldsymbol{\Psi}(\boldsymbol{R}(\mathbf{0}),\boldsymbol{R_c}(\mathbf{0})) < \mathbf{2}$.***

  After defining the dynamical model, the second contribution of this thesis is the nonlinear geometric control system for the DelFly Nimble. Another aim of the dynamic model was that it should allow for a model-based control design and stability analysis. The system dynamics are expressed on a nonlinear configuration manifold, which grants a globally defined, compact and unambiguous representation of the model. Thereby, a trajectory control algorithm evolving on the same nonlinear manifold is proposed. The DelFly Nimble has four degrees of actuation. Therefore, distinction is made between two output tracking problems, namely the outputs given by the attitude of the DelFly Nimble and the outputs given by the position of its center of mass. The objective of the attitude control design was to let a set of attitude and angular velocity tracking errors converge to zero. The objective of the position control design was to let as set of position and velocity errors converge to zero. The resulting tracking error dynamics are used to derive the nonlinear virtual control inputs, namely the generalized torques and thrust force. With the generalized thrust force and torques, the control inputs can be computed by using the inverse aerodynamic model. It has been shown through a Lyapunov stability analysis that, without considering the aerodynamics, (1) the attitude error dynamics are almost global exponentially stable, and (2) the position error dynamics are exponentially stable when $\Psi(R(0),R_c(0)) < 1$ and exponentially attractive when $1 < \Psi(\mathrm{R}(0),\mathrm{R_c}(0)) < 2$. These controller designs enable the DelFly Nimble to switch between tracking the attitude, position or velocity to perform complex aerobatic maneuvers.

- **Three aggressive aerobatic maneuvers can be achieved by the DelFly Nimble, namely: recovering from an upside-down orientation, a $360°$ flip and following an elliptic helix trajectory. For all three maneuvers applies that:**
  - ***Simulations without aerodynamics show excellent asymptotic tracking performance;***
  - ***Simulations with aerodynamics show that flapping counter forces and input saturation cause deviations from the reference trajectory.***

  The dynamic model and nonlinear geometric control design have been tested and validated by numeric simulations in MATLAB with satisfactory results. Simulations are conducted for the case of simple time-invariant attitude and position recovery trajectories. The results show that the DelFly Nimble is capable of autonomously performing recoveries from attitudes until $\Psi(\mathrm{R}(0), \mathrm{R_d}(0)) = 2$ with asymptotic convergence. However, deviations from the position reference trajectory are observed for the DelFly Nimble. The main cause for this effect is that only the generalized thrust force and torques can be used as virtual inputs and the forces in the x- and y-directions in body frame are not accounted for. Since the generalized torques are accounted for, the flapping counter torques are incorporated in the control design and no FCT are observed for the attitude trajectories. A second cause is that the commanded virtual control inputs may exceed the constraints of the control inputs. This leads to deviance of the desired trajectory or instability of the closed-loop system. Although the aerodynamics limits the performance, the simulations can be used whether the Nimble is still able to recover even when controls saturate.

  More challenging three-dimensional reference trajectories are conducted according to the three trajectory aims set for this thesis. Due to the fact that the nonlinear controller is almost-globally defined, the complex trajectories show satisfying results with superior performance to linear control laws. They showed that the DelFly Nimble is capable of autonomously tracking a $360°$ flip by using the attitude control loop or an elliptic helix by using the position control loop. Again, distinction is made between simulations with and without incorporating the aerodynamics. The simulations without aerodynamics showed excellent asymptotic tracking performance, whereas simulations with aerodynamics showed that damping forces caused deviations from the reference trajectory. Nevertheless, as the linear and angular body speeds go to zero, the FCF also goes to zero and accordingly, the trajectory target can be achieved.

7.2. Limitations and recommendations

Model parameters are obtained from literature or roughly estimated by parameter estimation. It is recommended to perform deliberate experiments, identify the parameters and validate the dynamic model. It is important that the input data excites all relevant system dynamics and that the relevant states and inputs are measured. More advanced identification methods than step signals are for example double or triplet inputs and multivariable splines that concentrate on a global aerodynamic framework considering the full flight envelope [30].

In order to improve the tracking performance of the closed-loop behavior of the DelFly Nimble, the FCFs in the x- and y-direction should be incorporated in the control description. Damping, in the general case, is often not modelled for UAVs, but is especially important for FWMAVs. One could add a feedforward structure to the proposed control design for compensating the damping on the wings and body.

The nonlinear attitude controller uses solely a gain while the mass moment of inertia is not the same around all directions. Accordingly, it is suggested to examine the stability properties by using non-diagonal matrices.

Next to damping, the presence of input saturation was evident. In this thesis point to point trajectories or heuristic minimum snap trajectories were used. It would be desirable to implement trajectory generation that accounts for saturation of the actuators. This goes hand in hand with the necessity that the thrust force $f_{b,z}$ should be larger than zero. The work of Foehn and Scaramuzza (2020) [64] implemented a time-optimal trajectory planner for quadrotors. This optimization formulation satisfied the system dynamics and input constraints, and minimized the trajectory time while passing through a sequence of waypoints.

It is assumed that control inputs are directly controlled. In real life this is not the case and therefore future work should focus on modeling the actuator dynamics. This has already been investigated by Kajak (2019) [7].

A global asymptotic stability is not possible due to the unstable equilibrium at an error angle of $180°$. Initial attitudes that live in the domain of attraction, asymptotically converge to the desired equilibrium. Nevertheless, solutions that live in the neighborhood of the stable manifold of one of the unstable equilibria, may remain near that stable manifold for an undesired period of time [34]. This can be resolved by using discontinuous feedback controllers. Only, discontinuous feedback suffers from chattering. Chattering can be resolved by the employment of hysteresis, yet this may render the Lyapunov theory inapplicable due to the non-Lipschitz nature of the closed-loop vector field [34]. The trajectory control design can be expanded by using hybrid control schemes. By employing these schemes, global robust exponential stability for attitude tracking was achieved [66].

In future work, disturbances can be taken into account. Almost global asymptotic stability with uncertainties in the quadrotor dynamics was achieved [33]. Moreover, this framework was extended by a Lyapunov-based adaptive law. This allows the control system to asymptotically follow attitude or position trajectories while tuning PID gains online [67]. Numerical simulations demonstrated superior performance by juxtaposing this method to an offline gain tuning method.

Lastly, it has been assumed that all states are directly available from measurements. In real life data will contain noise and not all states may be available. For example, not all states are available when flying outdoors as there are no cameras that measure the position of the DelFly Nimble. A full state feedback observer has to be designed to provide robustness.

Bibliography

[1] F. Fei, Z. Tu, Y. Yang, J. Zhang, and X. Deng, “Flappy hummingbird: An open source dynamic simulation of flapping wing robots and animals,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 9223–9229, 2019, doi: 10.1109/ICRA.2019.8794089.

[2] X. Huo *et al.*, “A hovering flapping-wing microrobot with altitude control and passive upright stability,” *J. Exp. Biol.*, vol. 305, no. 1, pp. 1–14, 2015, doi: 10.1109/TAC.2015.2407452.

[3] F. Fei, Z. Tu, J. Zhang, and X. Deng, “Learning extreme hummingbird maneuvers on flapping wing robots,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 109–115, 2019, doi: 10.1109/ICRA.2019.8794100.

[4] A. Roshanbin, H. Altartouri, M. Karásek, and A. Preumont, “COLIBRI: A hovering flapping twin-wing robot,” *Int. J. Micro Air Veh.*, vol. 9, no. 4, pp. 270–282, 2017, doi: 10.1177/1756829317695563.

[5] J. V. Caetano, C. C. de Visser, B. D. Remes, C. de Wagter, E. J. van Kampen, and M. Mulder, “Controlled Flight Maneuvers Of A Flappingwing Micro Air Vehicle: A step towards the delfly II identification,” *AIAA Atmos. Flight Mech. Conf.*, no. August, 2013, doi: 10.2514/6.2013-4843.

[6] P. Chirarattananon, K. Y. Ma, and R. J. Wood, “Adaptive control of a millimeter-scale flapping-wing robot,” *Bioinspiration and Biomimetics*, vol. 9, no. 2, 2014, doi: 10.1088/1748-3182/9/2/025004.

[7] K. M. Kajak, M. Karásek, Q. P. Chu, and G. C. H. E. De Croon, “A minimal longitudinal dynamic model of a tailless flapping wing robot for control design,” *Bioinspiration and Biomimetics*, vol. 14, no. 4, 2019, doi: 10.1088/1748-3190/ab1e0b.

[8] G. C. H. E. de Croon, M. Perçin, B. D. W. Remes, R. Ruijsink, and C. De Wagter, *The DelFly*. Delft: Springer, 2016.

[9] M. H. Dickinson, F. O. Lehmann, and S. P. Sane, “Wing rotation and the aerodynamic basis of insect right,” *Science (80-.).*, vol. 284, no. 5422, pp. 1954–1960, 1999, doi: 10.1126/science.284.5422.1954.

[10] M. Karasek, F. T. Muijres, C. De Wagter, B. D. W. Remes, and G. C. H. E. De Croon, “A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns,” *Science (80-.).*, vol. 361, no. 6407, pp. 1089–1094, 2018, doi: 10.1126/science.aat0350.

[11] J. R. Usherwood and C. P. Ellington, “The aerodynamics of revolving wings I. Model hawkmoth wings,” *J. Exp. Biol.*, vol. 205, no. 11, pp. 1547–1564, 2002.

[12] C. P. Ellington, “The Aerodynamics of Hovering Insect Flight . IV . Aeorodynamic Mechanisms,” *R. Soc. Publ.*, vol. 305, no. 1122, pp. 79–113, 2019, doi: https://doi-org.tudelft.idm.oclc.org/10.1098/rstb.1984.0052.

[13] L. C. Johansson, S. Engel, A. Kelber, M. K. Heerenbrink, and A. Hedenström, “Multiple leading edge vortices of unexpected strength in freely flying hawkmoth,” *Sci. Rep.*, vol. 3, pp. 1–5, 2013, doi: 10.1038/srep03264.

[14] Z. J. Wang, “Dissecting Insect Flight,” *Annu. Rev. Fluid Mech.*, vol. 37, no. 1, pp. 183–210, 2005, doi: 10.1146/annurev.fluid.36.050802.121940.

[15] J. M. Birch and M. H. Dickinson, “The influence of wing-wake interactions on the production of

aerodynamic forces in flapping flight," *J. Exp. Biol.*, vol. 206, no. 13, pp. 2257–2272, 2003, doi: 10.1242/jeb.00381.

[16] M. R. A. Nabawy and W. J. Crowther, "On the quasi-steady aerodynamics of normal hovering flight part II: Model implementation and evaluation," *J. R. Soc. Interface*, vol. 11, no. 94, 2014, doi: 10.1098/rsif.2013.1197.

[17] Q. Wang, J. F. L. Goosen, and F. Van Keulen, "A predictive quasi-steady model of aerodynamic loads on flapping wings," *J. Fluid Mech.*, vol. 800, pp. 688–719, 2016, doi: 10.1017/jfm.2016.413.

[18] M. R. A. Nabawy and W. J. Crowthe, "A quasi-steady lifting line theory for insect-like hovering flight," *PLoS One*, vol. 10, no. 8, pp. 1–18, 2015, doi: 10.1371/journal.pone.0134972.

[19] S. P. Sane and M. H. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *J. Exp. Biol.*, vol. 205, no. 8, pp. 1087–1096, 2002.

[20] J. Nijboer, S. F. Armanini, M. Karasek, and C. C. de Visser, "Longitudinal Grey-Box Model Identification of a Tailless Flapping-Wing MAV Based on Free-Flight Data," 2020, doi: 10.2514/6.2020-1964.

[21] S. F. Armanini, C. C. de Visser, G. C. H. E. de Croon, and M. Mulderz, "A time-scale separation approach for time-varying model identification of a flapping-wing micro aerial vehicle," *AIAA Atmos. Flight Mech. Conf.*, no. January, 2016, doi: 10.2514/6.2016-1529.

[22] S. F. Armanini, C. C. De Vissery, and G. C. H. E. De Croonz, "Black-box LTI modelling of flapping-wing micro aerial vehicle dynamics," *AIAA Atmos. Flight Mech. Conf. 2015*, no. August, 2015, doi: 10.2514/6.2015-0234.

[23] S. Stramigioli and H. Bruyninckx, "Geometry and Screw Theory for Robotics," *Matrix*, 2001.

[24] N. Gao, H. Aono, and H. Liu, "Perturbation analysis of 6DoF flight dynamics and passive dynamic stability of hovering fruit fly Drosophila melanogaster," *J. Theor. Biol.*, vol. 270, no. 1, pp. 98–111, 2011, doi: 10.1016/j.jtbi.2010.11.022.

[25] B. Cheng and X. Deng, "Translational and rotational damping of flapping flight and its dynamics and stability at hovering," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 849–864, 2011, doi: 10.1109/TRO.2011.2156170.

[26] Z. J. Wang, J. M. Birch, and M. H. Dickinson, "Unsteady forces and flows in low Reynolds number hovering flight: Two-dimensional computations vs robotic wing experiments," *J. Exp. Biol.*, vol. 207, no. 3, pp. 449–460, 2004, doi: 10.1242/jeb.00739.

[27] B. Cheng and X. Deng, "Near-hover dynamics and attitude stabilization of an insect model," *Proc. 2010 Am. Control Conf. ACC 2010*, pp. 39–44, 2010, doi: 10.1109/acc.2010.5530672.

[28] M. S. Martis, "Validation of simulation based models: A theoretical outlook," *Electron. J. Bus. Res. Methods*, vol. 4, no. 1, pp. 39–46, 2006.

[29] D. Tan and Z. Chen, "On A General Formula of Fourth Order Runge-Kutta Method," *J. Math. Sci. Math. Educ.*, vol. 7, no. 2, pp. 1–10, 2012.

[30] J. Caetano *et al.*, "Linear aerodynamic model identification of a flapping wing MAV based on flight test data," *Int. J. Micro Air Veh.*, vol. 5, no. 4, pp. 273–286, 2013, doi: 10.1260/1756-8293.5.4.273.

[31] T. Lee, M. Leok, and N. H. Mcclamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013, doi: 10.1002/asjc.567.

[32] T. Lee, M. Leok, and N. H. McClamroch, “Control of Complex Maneuvers for a Quadrotor UAV using Geometric Methods on SE(3),” no. i, 2010, [Online]. Available: http://arxiv.org/abs/1003.2005.

[33] F. Goodarzi, D. Lee, and T. Lee, “Geometric nonlinear PID control of a quadrotor UAV on SE(3),” *2013 Eur. Control Conf. ECC 2013*, pp. 3845–3850, 2013, doi: 10.23919/ecc.2013.6669644.

[34] S. Amit and N. H. Mcclamroch, “Rigid-Body Attitude Control: Using rotation matrices for continuous, singularity-free control laws,” *IEEE Control Syst.*, vol. 31, no. 3, pp. 30–51, 2011, doi: 10.1109/MCS.2011.940459.

[35] X. N. Shi, Y. A. Zhang, and D. Zhou, “A geometric approach for quadrotor trajectory tracking control,” *Int. J. Control*, vol. 88, no. 11, pp. 2217–2227, 2015, doi: 10.1080/00207179.2015.1039593.

[36] R. R. Warier, A. K. Sanyal, M. H. Dhullipalla, and S. P. Viswanathan, “Trajectory Tracking Control For Underactuated Thrust-Propelled Aerial Vehicles,” *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 555–560, 2018, doi: 10.1016/j.ifacol.2018.07.338.

[37] D. Cabecinhas, R. Cunha, and C. Silvestre, “A nonlinear quadrotor trajectory tracking controller with disturbance rejection,” *Control Eng. Pract.*, vol. 26, no. 1, pp. 1–10, 2014, doi: 10.1016/j.conengprac.2013.12.017.

[38] D. Invernizzi and M. Lovera, “Geometric tracking control of a quadcopter tiltrotor UAV,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11565–11570, 2017, doi: 10.1016/j.ifacol.2017.08.1645.

[39] Y. Liu, J. M. Montenbruck, P. Stegagno, F. Allgower, and A. Zell, “A robust nonlinear controller for nontrivial quadrotor maneuvers: Approach and verification,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2015-Decem, pp. 5410–5416, 2015, doi: 10.1109/IROS.2015.7354142.

[40] E. Lefeber, S. J. A. M. Van Den Eijnden, and H. Nijmeijer, “Almost global tracking control of a quadrotor UAV on SE(3),” *2017 IEEE 56th Annu. Conf. Decis. Control. CDC 2017*, vol. 2018-Janua, no. December, pp. 1175–1180, 2018, doi: 10.1109/CDC.2017.8263815.

[41] N. N. Vo, “Nonlinear Geometric Control of a Quadrotor with a Cable-Suspended load,” 2017.

[42] J. Li and Y. Li, “Dynamic analysis and PID control for a quadrotor,” *2011 IEEE Int. Conf. Mechatronics Autom. ICMA 2011*, pp. 573–578, 2011, doi: 10.1109/ICMA.2011.5985724.

[43] K. U. Lee, H. S. Kim, J. B. Park, and Y. H. Choi, “Hovering control of a quadrotor,” *Int. Conf. Control. Autom. Syst.*, pp. 162–167, 2012.

[44] B. Yongqiang, L. Hao, S. Zongying, and Z. Yisheng, “Robust control of quadrotor unmanned air vehicles,” *Chinese Control Conf. CCC*, pp. 4462–4467, 2012.

[45] S. Kurak and M. Hodzic, “Control and estimation of a quadcopter dynamical model,” *Period. Eng. Nat. Sci.*, vol. 6, no. 1, pp. 63–75, 2018, doi: 10.21533/pen.v6i1.164.

[46] A. Reizenstein, “Position and Trajectory Control of a Quadcopter Using PID and LQ Controllers,” 2017.

[47] D. C. Tosun, Y. Isik, and H. Korul, “Comparison of PID and LQR controllers on a quadrotor helicopter,” *Int. J. Syst. Appl. Eng. Dev.*, vol. 9, pp. 136–143, 2015.

[48] A. Roza and M. Maggiore, “Path following controller for a quadrotor helicopter,” *Proc. Am. Control Conf.*, no. August, pp. 4655–4660, 2012, doi: 10.1109/acc.2012.6315061.

[49] I. Palunko and R. Fierro, “Adaptive control of a quadrotor with dynamic changes in the center

of gravity," *IFAC Proc. Vol.*, vol. 44, no. 1 PART 1, pp. 2626–2631, 2011, doi: 10.3182/20110828-6-IT-1002.02564.

[50] A. A. Mian and D. Wang, "Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter," *Chinese J. Aeronaut.*, vol. 21, no. 3, pp. 261–268, 2008, doi: 10.1016/S1000-9361(08)60034-5.

[51] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.*, vol. 4, no. 11, pp. 2343–2355, 2010, doi: 10.1049/iet-cta.2009.0336.

[52] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," *IEEE Int. Conf. Intell. Robot. Syst.*, no. November, pp. 3255–3260, 2006, doi: 10.1109/IROS.2006.282433.

[53] X. Huo, M. Huo, and H. R. Karimi, "Attitude stabilization control of a quadrotor UAV by using backstepping approach," *Math. Probl. Eng.*, vol. 2014, 2014, doi: 10.1155/2014/749803.

[54] M. Önder Efe, "Sliding mode control for unmanned aerial vehicles research," *Stud. Syst. Decis. Control*, vol. 24, pp. 239–255, 2015, doi: 10.1007/978-3-319-18290-2_12.

[55] L. Besnard, Y. B. Shtessel, and B. Landrum, "Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer," *J. Franklin Inst.*, vol. 349, no. 2, pp. 658–684, 2012, doi: 10.1016/j.jfranklin.2011.06.031.

[56] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *Int. J. Control. Autom. Syst.*, vol. 7, no. 3, pp. 419–428, 2009, doi: 10.1007/s12555-009-0311-8.

[57] B. Emran, "Robust Nonlinear Composite Adaptive Control of Quadrotor," *Int. J. Digit. Inf. Wirel. Commun.*, vol. 4, no. 2, pp. 213–225, 2014, doi: 10.17781/p001100.

[58] P. De Monte and B. Lohmann, "Position trajectory tracking of a quadrotor helicopter based on L1 adaptive control," *2013 Eur. Control Conf. ECC 2013*, pp. 3346–3353, 2013, doi: 10.23919/ecc.2013.6669410.

[59] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," *Open J. Appl. Sci.*, vol. 04, no. 14, pp. 547–556, 2014, doi: 10.4236/ojapps.2014.414053.

[60] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," *Proc. IEEE Conf. Decis. Control*, pp. 5420–5425, 2010, doi: 10.1109/CDC.2010.5717652.

[61] H. A. Hashim, L. J. Brown, and K. McIsaac, "Nonlinear Pose Filters on the Special Euclidean Group SE(3) With Guaranteed Transient and Steady-State Performance," *IEEE Trans. Syst. Man, Cybern. Syst.*, no. 3, pp. 1–14, 2019, doi: 10.1109/tsmc.2019.2920114.

[62] Y. Zhu, X. Chen, and C. Li, "Some discussions about the error functions on SO(3) and SE(3) for the guidance of a UAV using the screw algebra theory," *Adv. Math. Phys.*, vol. 2017, 2017, doi: 10.1155/2017/1016530.

[63] T. Lee, M. Leok, and N. H. McClamroch, "Control of Complex Maneuvers for a Quadrotor UAV using Geometric Methods on SE(3)," pp. 1–14, 2010, [Online]. Available: http://arxiv.org/abs/1003.2005.

[64] P. Foehn, D. Scaramuzza, and R. O. Aug, "CPC : Complementary Progress Constraint for Time-Optimal Quadrotor Trajectories," pp. 1–12.

[65] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors,"

Proc. - IEEE Int. Conf. Robot. Autom., pp. 2520–2525, 2011, doi: 10.1109/ICRA.2011.5980409.

[66] T. Lee, “Global exponential attitude tracking controls on SO(3),” *IEEE Trans. Automat. Contr.*, vol. 60, no. 10, pp. 2837–2842, 2015, doi: 10.1109/TAC.2015.2407452.

[67] F. A. Goodarzi, “Self-Tuning Geometric Control for a Quadrotor UAV Based on Lyapunov Stability Analysis,” *Int. J. Robot. Autom.*, vol. 5, no. 3, pp. 1–15, 2016.

[68] T. Lee, M. Leok, and N. H. McClamroch, *Global Formulations of Lagrangian and Hamiltonian Dynamics on Embedded Manifolds*. 2017.

I. Rigid body dynamics

The Euler-Lagrange equations for mechanical systems as a rigid body that evolves on the nonlinear manifold $\mathbb{R}^3 \times SO(3)$ can be derived by using Hamilton's principle, the formulation of the principle of stationary action, as

$$\frac{\delta S}{\delta x(t)} = 0,$$

with S the action integral and $x(t)$ the generalized coordinates. For non-conservative systems S is equal to

$$\delta S = \int_{t_1}^{t_2} (\delta\mathcal{W} + \delta\mathcal{L})\, dt = \int_{t_1}^{t_2} (\delta W + \delta\mathcal{T} - \delta\mathcal{U})\, dt,$$

where $\mathcal{W}$ is the virtual work, $\mathcal{L}$ is the Lagrangian, $\mathcal{T}$ is the kinetic energy and $\mathcal{U}$ the potential energy of the system from time t_1 to time t_2. The states are

$$x = [p\ \dot{p}\ \omega_b\ R]^T.$$

The potential and kinetic energy of the system is given by

$$\begin{aligned}\mathcal{T} &= \frac{1}{2} m\dot{p}\cdot\dot{p} + \frac{1}{2} J\omega_b \cdot \omega_b,\\ \mathcal{U} &= mpg \cdot z^{\mathcal{I}}.\end{aligned}$$

with $J \in \mathbb{R}^{3\times3}$ is the moment of inertia tensor of the Nimble, g is the gravitational constant and m is the mass of the Nimble. The variation of the kinetic and potential energy and virtual work is written as

$$\begin{aligned}\delta\mathcal{T} &= \frac{\partial\mathcal{T}}{\partial\dot{p}}\delta\dot{p} + \frac{\partial\mathcal{T}}{\partial\omega_b}\delta\omega_b\\ &= m\dot{p}\cdot\delta\dot{p} + J\omega_b\cdot\delta\omega_b,\\ \delta\mathcal{U} &= \frac{\partial U}{\partial p}\delta p\\ &= mgz^{\mathcal{I}}\cdot\delta p,\\ \delta W &= f_{b,z}R\cdot\frac{\partial p}{\partial x}\delta p + \tau_{b,z}\cdot\frac{\partial\omega_b}{\partial\dot{x}}\delta\dot{p},\\ &= f_{b,z}R\cdot\delta p + \tau_{b,z}\cdot R^T\delta R,\end{aligned}$$

A variation of R can be described as $R^\epsilon(t) = R(t)e^{\epsilon\tilde{\eta}(t)}$, which is an exponential map and isomorphism between $\mathfrak{so}(3)$ and $\mathbb{R}^3$, where η and ϵ are differentiable curves that vanish at t_0 and t_f. $e^{\epsilon\tilde{\eta}(t)}$ defines a differentiable curve that takes values in the Lie group of R and is the identity matrix at t_0 and t_f. The time derivative of the variation of R^ϵ is given by

$$\dot{R}^\epsilon(t) = \dot{R}(t)e^{\epsilon\tilde{\eta}(t)} + \epsilon R(t)e^{\epsilon\tilde{\eta}(t)}\dot{\tilde{\eta}}(t).$$

The varied curve satisfies

$$\tilde{\eta}^{\epsilon} = R^{\epsilon^{T}} \dot{R}\epsilon = e^{-\epsilon\tilde{\eta}}\tilde{\eta}e^{\epsilon\tilde{\eta}(t)} + e^{\epsilon\tilde{\eta}(t)}\dot{\tilde{\eta}}$$

The variations that follow from differentiations on $SO(3)$ are given by [68]

$$\begin{aligned}
\delta R &= R\tilde{\eta} \\
\delta \dot{R} &= \dot{R}\tilde{\eta} + R\dot{\tilde{\eta}}, \\
\delta \tilde{\omega}_b &= \delta\left(R^T \dot{R}\right) \\
&= \delta R^T \dot{R} + R^T \delta \dot{R} \\
&= (R\tilde{\eta})^T \dot{R} + R^T\left(\dot{R}\tilde{\eta} + R\dot{\tilde{\eta}}\right) \\
&= \tilde{\eta}^T \tilde{\omega}_b + \tilde{\omega}_b \tilde{\eta} + \dot{\tilde{\eta}} \\
&= \widetilde{\tilde{\omega}_b}\eta + \dot{\tilde{\eta}} \\
\delta \omega_b &= \tilde{\omega}_b \eta + \dot{\eta}
\end{aligned}$$

By substituting the equations, the action integral can now be written as

$$\begin{aligned}
\delta S &= \int_{t_1}^{t_2} (\delta W + \delta \mathcal{T} - \delta \mathcal{U})\, dt \\
&= \int_{t_1}^{t_2} \left(\left(f_{b,z}R - mgz^{\mathcal{I}}\right) \cdot \delta p + m\dot{p} \cdot \delta \dot{p}\right)\, dt + \int_{t_1}^{t_2} \left(\tau_{b,z} \cdot R^T \delta R + J\omega_b \cdot \delta \omega_b\right) dt \\
&= \int_{t_1}^{t_2} \left(f_{b,z}R - mgz^{\mathcal{I}} + \frac{d}{dt}m\dot{p}\right) \cdot \delta p\, dt + \int_{t_1}^{t_2} \left(\tau_{b,z} \cdot R^T R\tilde{\eta} + J\omega_b \cdot (\dot{\eta} + \tilde{\omega}_b \eta)\right) dt \\
&= \int_{t_1}^{t_2} \left(f_{b,z}R - mgz^{\mathcal{I}} + \frac{d}{dt}m\dot{p}\right) \cdot \delta p\, dt + \int_{t_1}^{t_2} \left(-\tau_{b,z} \cdot R^T R + \frac{d}{dt}J\omega_b + J\omega_b \times \omega_b\right) \cdot \delta \eta\, dt
\end{aligned}$$

Setting each variation to zero according to Hamilton's principle, the equations of motion of the Nimble are given by

$$\begin{aligned}
&\dot{p} = v, \\
&\dot{v} = gz^{\mathcal{I}} - m^{-1}Rf_{b,z}, \\
&\dot{R} = R\tilde{\omega}_b, \\
&J\dot{\omega}_b + J\omega_b \times \omega_b = \tau_b.
\end{aligned}$$

These dynamics are compact and are globally expressed on the smooth manifolds $SO(3) \times SE(3)$.

II. Analysis of the linearized attitude dynamics

The Nimble is modelled as a rigid body. The attitude dynamics including the nonlinear control input are given by

$$\begin{aligned} J\dot{\omega}_b &= -k_R e_R - k_{\omega_b} e_{\omega_b} - J\left(\widetilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\right) \\ &= -k_R\left(R_d^T R - R^T R_d\right)^{\vee} - k_{\omega_b}\left(\omega_b - R^T R_d \omega_{b\,d}\right) - J\left(\widetilde{\omega}_b R^T R_d \omega_{b,d} - R^T R_d \dot{\omega}_{b,d}\right) \end{aligned} \tag{II-1}$$

Substituting the variations $\delta R = R\tilde{\eta}$ and $\delta\omega_b = \widetilde{\omega}_b \eta + \dot{\eta}$ in Equation (II-1) and by using the properties $\tilde{x}A + A^T\tilde{x} = tr[A]I - A\ \forall\ x \in \mathbb{R}^3, A \in \mathbb{R}^{3\times 3}$ and $\tilde{x}y = -\tilde{y}x\ \forall\ x \in \mathbb{R}^3$, $y \in \mathbb{R}^3$, gives

$$\begin{aligned} J\delta\dot{\omega}_b &= -k_R\left(R_d^T R\tilde{\eta}\ - \tilde{\eta}R^T R_d\right)^{\vee} - k_{\omega_b}\left(\delta\omega_b - \tilde{\eta}R^T R_d \omega_{b,d}\right) \\ &\qquad - J\left(\delta\widetilde{\omega}_b R^T R_d \omega_{b,d} + \widetilde{\omega}_b \tilde{\eta} R^T R_d \omega_{b,d} - \tilde{\eta} R^T \widetilde{R_d \dot{\omega}}_{b,d}\right) \\ &= -k_R(tr[R^T R_d]I - R^T R_d)\eta - k_{\omega_b}\delta\omega_b - k_{\omega_b}\left(R^T \widetilde{R_d \omega_{b,d}}\right)\eta \\ &\qquad + J\left(R^T \widetilde{R_d \omega_{b,d}}\right)\delta\omega_b + J\widetilde{\omega}_b\left(R^T \widetilde{R_d \omega_{b,d}}\right)\eta - J\left(R^T R_d \dot{\omega}_{b,d}\right)\eta \\ &= -k_R C_1 \eta - k_{\omega_b}\delta\omega_b - k_{\omega_b} C_2 \eta - J C_2 \delta\omega_b + J\widetilde{\omega}_b C_2 \eta - J C_3 \eta, \end{aligned} \tag{II-2}$$

where $C_1 = (tr[R^T R_d]I - R^T R_d)$, $C_2 = \left(R^T \widetilde{R_d \omega_{b,d}}\right)$ and $C_3 = \left(R^T \widetilde{R_d \dot{\omega}_{b,d}}\right)$. The linearized equations of motion are written as

$$\dot{x} = \begin{bmatrix} \dot{\eta} \\ \delta\dot{\omega}_b \end{bmatrix} = \begin{bmatrix} -\widetilde{\omega}_b & I \\ -k_R J^{-1} C_1 + \left(\widetilde{\omega}_b - J^{-1} k_{\omega_b}\right) C_2 - C_3 & -J^{-1} k_{\omega_b} - C_2 \end{bmatrix} \begin{bmatrix} \eta \\ \delta\omega_b \end{bmatrix} = Ax. \tag{II-3}$$

Next, the equilibrium solutions are investigated. By choosing $R_d = I$, $\omega_{b,d} = [0,0,0]^T, \dot{\omega}_{b,d} = [0,0,0]^T$. There are four equilibria, namely $(I, 0)$, $(diag(-1,1,1), 0)$, $(diag(1,-1,1), 0)$ and $(diag(1,1,-1), 0)$. The existence of the undesirable equilibria is due to the nonlinear topological structure of SO(3) and cannot be avoided by the construction of a different continuous control system. The eigen-structure of the equilibria are examined using Equation (II-2) below.

1) Equilibrium $(I, 0)$: The eigenvalues of the matrix A at the equilibrium $(I, 0)$ are given by

$$\begin{aligned} \lambda_{1,2} &= -9.25 \pm 11.49i, \\ \lambda_3 &= -17.52, \\ \lambda_4 &= -35.78, \\ \lambda_{5,6} &= -11.99 \pm 11.76i. \end{aligned}$$

This equilibrium is an asymptotically stable focus.

2) Equilibrium $(diag(-1,1,1),0)$: The eigenvalues of the matrix A at the equilibrium $(I,0)$ are given by

$$\begin{aligned} \lambda_{1,2} &= -9.25 \pm 11.49, \\ \lambda_{3,4} &= \quad 0.00, \\ \lambda_5 &= -23.97, \\ \lambda_6 &= -53.31. \end{aligned}$$

3) Equilibrium $(diag(1,-1,1),0)$: The eigenvalues of the matrix A at the equilibrium $(I,0)$ are given by

$$\begin{aligned} \lambda_{1,2} &= -11.99 \pm 11.76i, \\ \lambda_{3,4} &= \quad 0.00, \\ \lambda_5 &= -18.50, \\ \lambda_6 &= -53.31. \end{aligned}$$

4) Equilibrium $(diag(1,1,-1),0)$: The eigenvalues of the matrix A at the equilibrium $(I,0)$ are given by

$$\begin{aligned} \lambda_1 &= -17.52, \\ \lambda_2 &= -35.79, \\ \lambda_{3,4} &= \quad 0.00, \\ \lambda_5 &= -18.50, \\ \lambda_6 &= -23.98. \end{aligned}$$

We cannot infer how the nonlinear system behaves at the origin though linearization, since there are eigenvalues with value zero. Therefore, a nonlinear analysis is needed to conclude about stability.

III. Aerodynamics

a. Sensitivity of the Jacobian

In this appendix section, some insights of the Jacobian are obtained by varying for the range of control inputs of the Nimble. The aerodynamic map depends on the four control inputs and on the translational and rotational speed of the Nimble. Since, the number of dimensions of the structure is difficult to investigate, the speeds are set to zero and varied separately.

In Figure III-1, the dihedral angle γ is varied while β remains zero. In the four subplots, the magnitude of the numeric values of $\mathcal{J}(\gamma, f_L, f_R, \beta)$ are shown. The magnitudes are normalized with respect to the maximum attainable control input value, in order to put the graphs into perspective. Figure 4-2 illustrates that $f_{b,z}$ is mainly coupled with the flapping frequencies and just a small portion of the control input angle γ. As the angle γ increases, $f_{b,z}$ decreases by slightly rotating the thrust. $\tau_{b,x}$ is mainly coupled with the flapping frequencies and a small portion of the control input angle γ. By increasing the flapping frequencies f_l and f_R increase and decrease the roll torque respectively with the same magnitude. As the thrust vectors are being rotated by γ, the magnitude of the flapping frequencies decreases. $\tau_{b,y}$ is coupled with the flapping frequencies and angle γ. The flapping frequencies are similar and have more impact the when the angle γ is larger. Lastly, $\tau_{b,z}$ is mainly coupled with the flapping frequencies and angle β, although the values are in the order of 10^{-3}. Again as with the roll torque the flapping frequencies have equal magnitude, but opposite in sign.

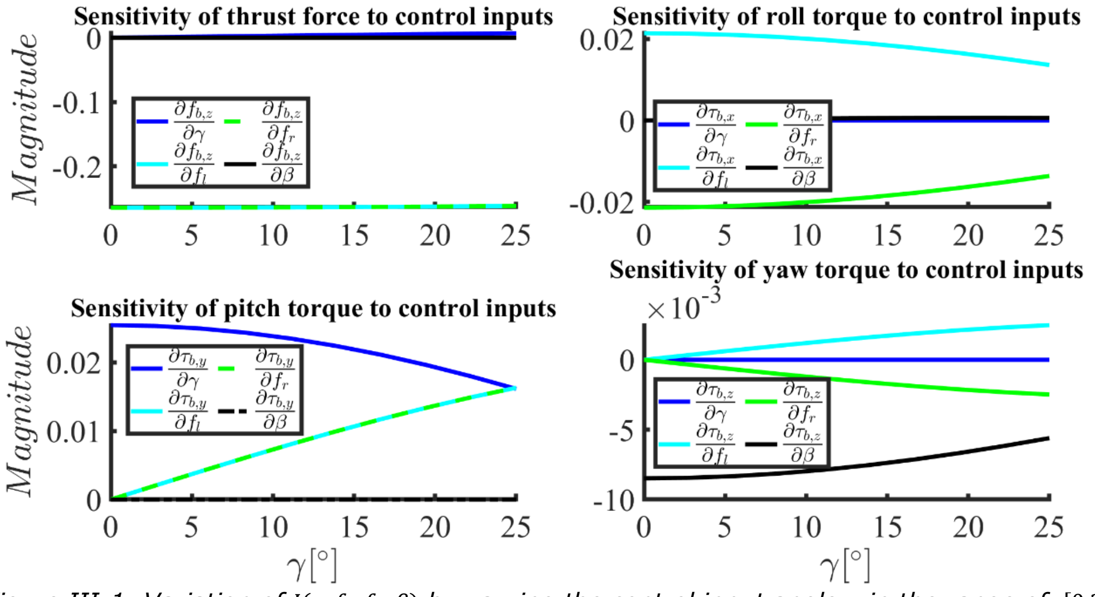

Figure III-1: Variation of $J(\gamma, f_L, f_R, \beta)$ by varying the control input angle γ in the range of $[0\ 25°]$ while $\beta = 0°$. The frequencies are held constant with a value of $15\ Hz$. The Jacobian values are multiplied with the maximum attainable control input values.

It is assumed that the control inputs of the Nimble generating the pitch, roll, yaw and thrust motions are decoupled. Nevertheless, according to the model there are some notable coupling terms. $f_{b,z}$ decreases when the control input γ is increased by a higher value as the control input f_L/f_R. This observation can be compared with Figure III-10, in which the reduction of $f_{b,z}$ is observed as well. In Figure III-1, the dihedral angle β is varied while γ remains zero. In the four subplots the magnitude of the numeric values of $J(\gamma, f_L, f_R, \beta)$ are shown. Figure III-2 illustrates that $f_{b,z}$ is mainly coupled with the frequencies and just a really small portion of the control input angle γ, $\tau_{b,x}$ is mainly coupled with the frequencies and just a really small portion of the control input angle γ, $\tau_{b,y}$ is coupled with the angle γ and flapping frequencies f_L and f_R, and $\tau_{b,z}$ is coupled with the frequencies and angle β.

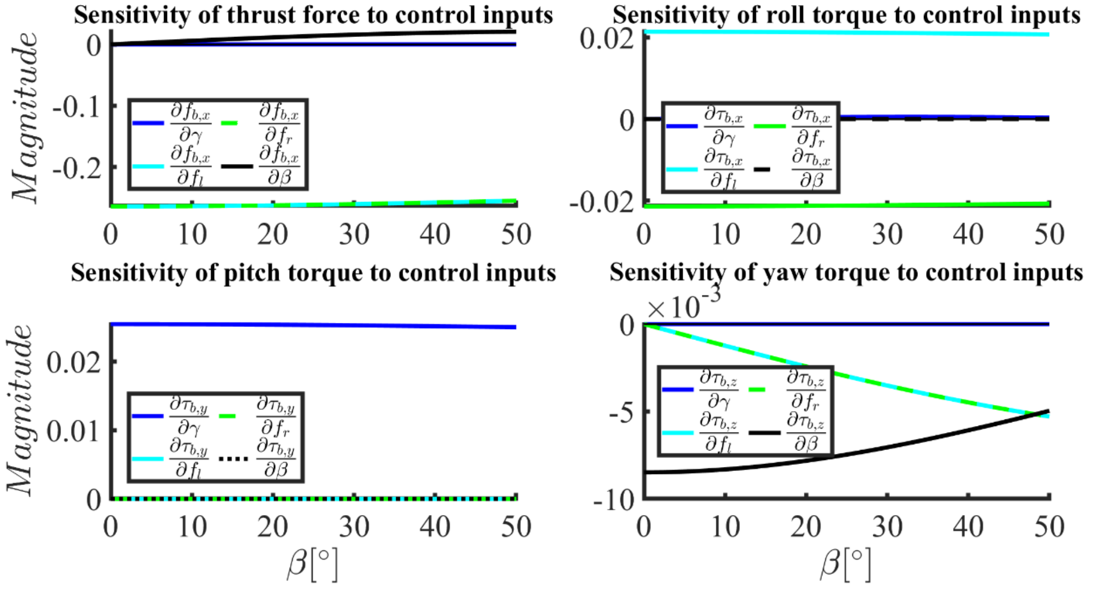

Figure III-2: Variation of $J(\gamma, f_L, f_R, \beta)$ by varying the control input angle β in the range of $[0\ 50°]$ while $\gamma = 0°$. The frequencies are held constant with a value of $15\ Hz$. The Jacobian values are multiplied with the maximum attainable control input values.

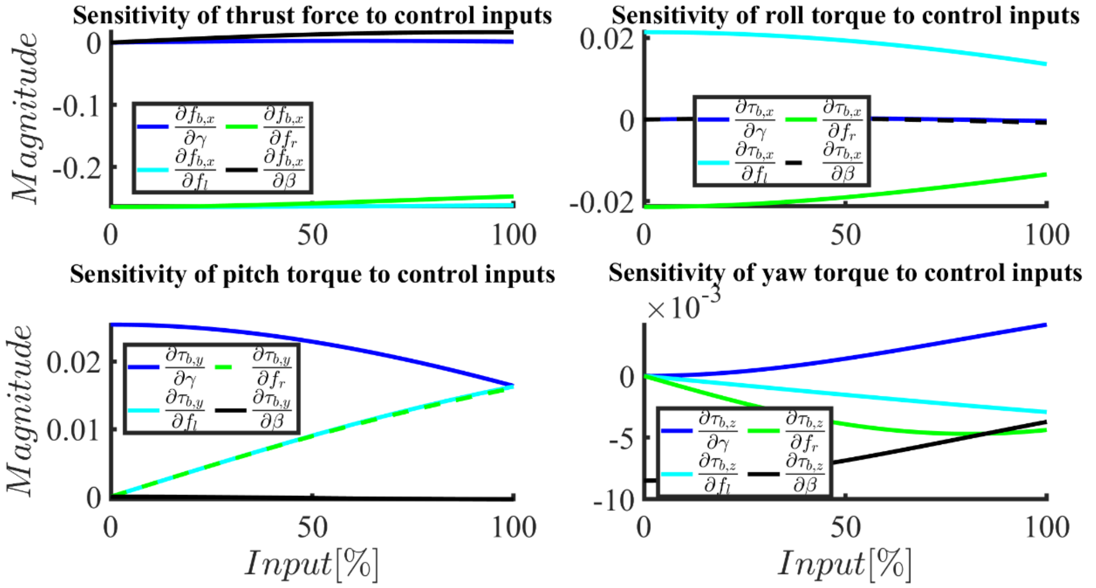

Figure III-3: Normalized Jacobian values increased from 0% ($\gamma = 0°, \beta = 0°$) to 100% ($\gamma = 25°, \beta = 50°$) simultaneously. The frequencies are held constant with a value of $15\ Hz$.

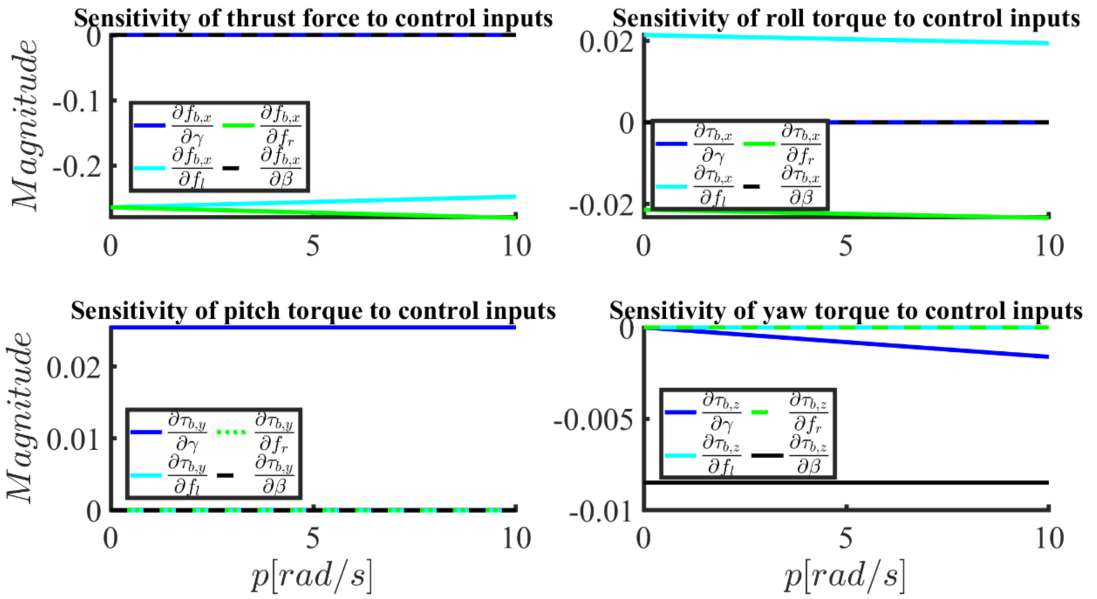

Figure III-4: Jacobian values with angular speed p in the range of $[0\ \ 10]\,rad/s$, while $\gamma = 0°$ $\beta = 0°$ and $f_L = f_R = 15\,Hz$.

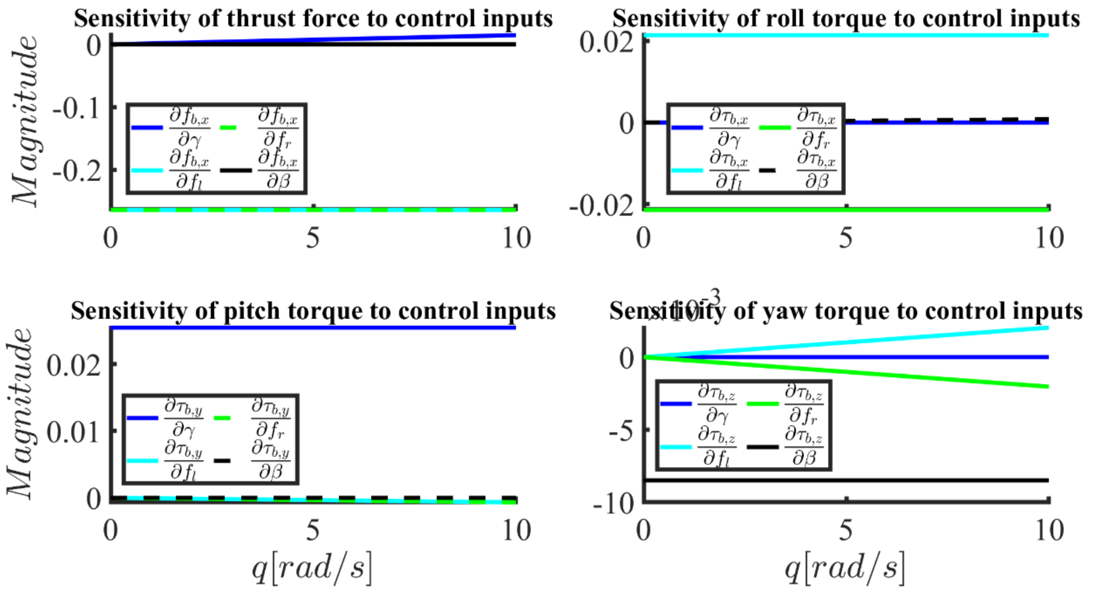

Figure III-5: Jacobian values with angular speed q in the range of $[0\ \ 10]\,rad/s$, while $\gamma = 0°$ $\beta = 0°$ and $f_L = f_R = 15\,Hz$.

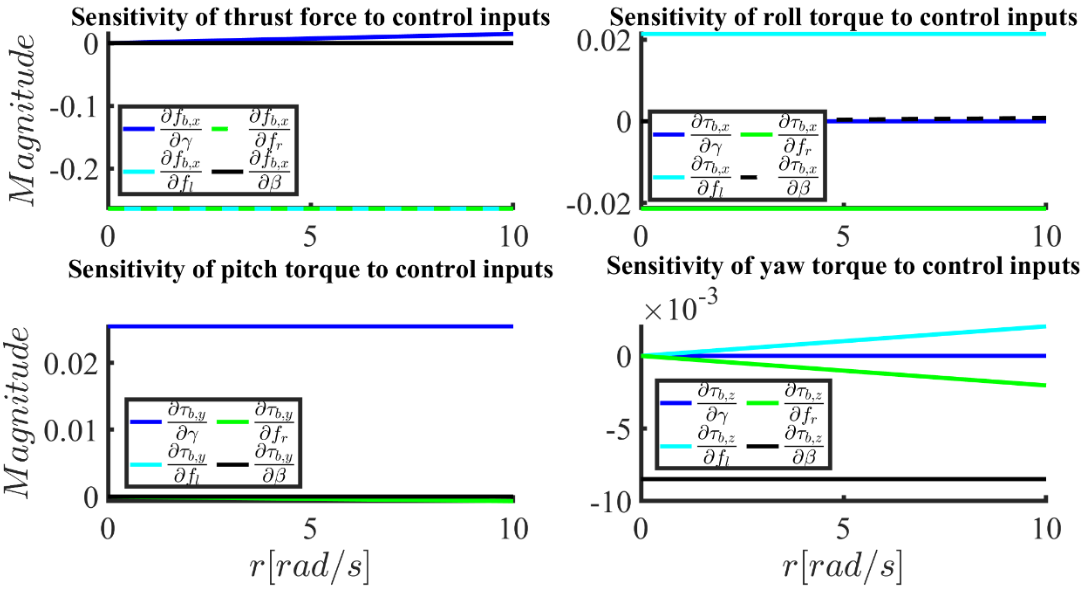

Figure III-6: Jacobian values with angular speed r in the range of $[0\ \ 10]\,rad/s$, while $\gamma = 0°$ $\beta = 0°$ and $f_L = f_R = 15\,Hz$.

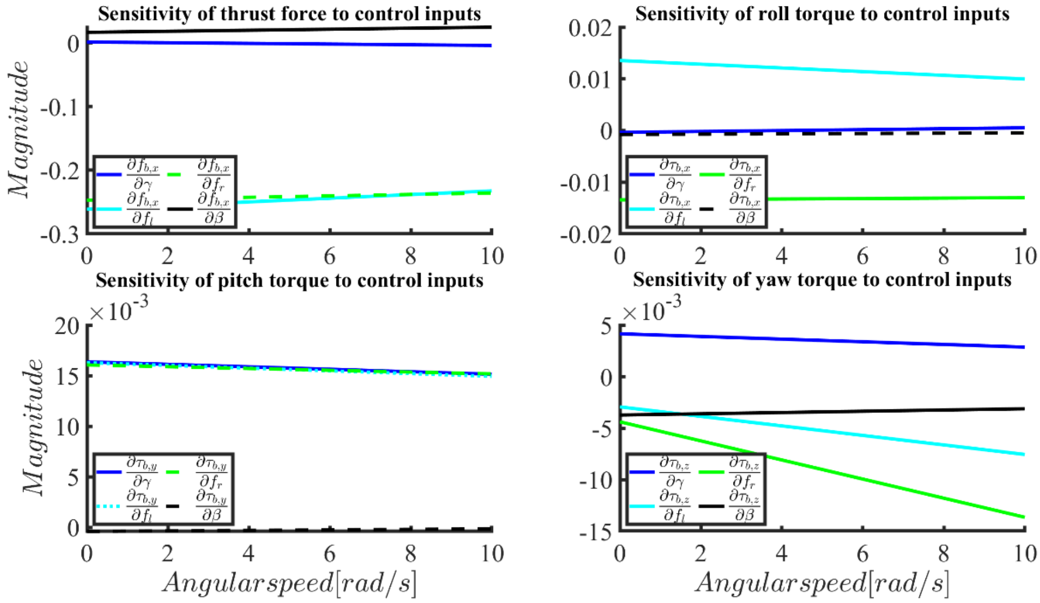

Figure III-7: Jacobian values with angular speeds in the range of $[0\ 10\ rad/s]$, while $\gamma = 25°$ $\beta = 50°$ and $f_L = f_R = 15\,Hz$.

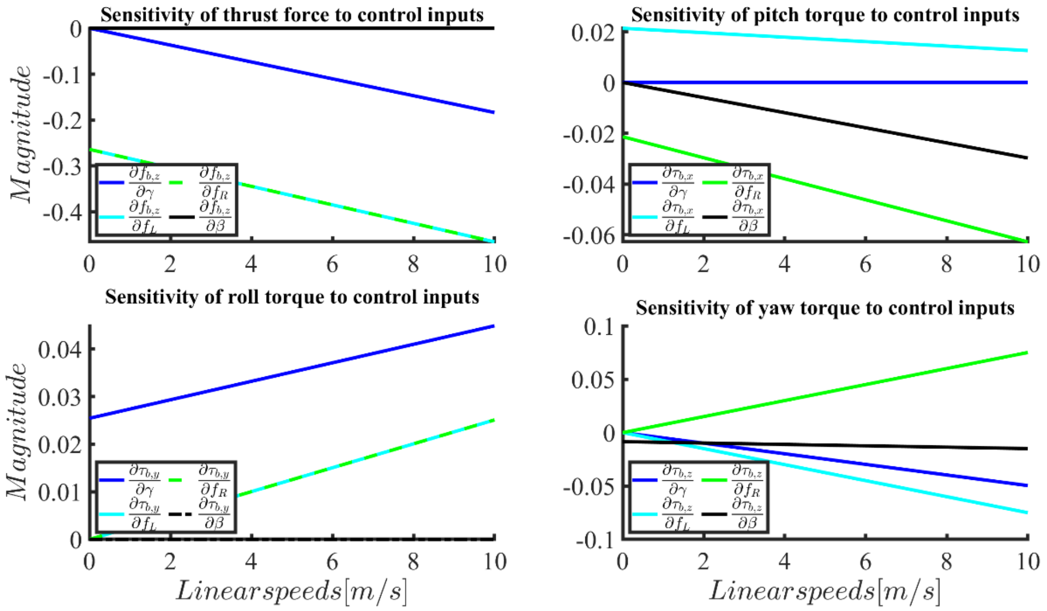

Figure III-8: Jacobian values with linear speeds in the range of $[0\ 10\ m/s]$*, while* $\gamma = 0°$ $\beta = 0°$ *and* $f_L = f_R = 15\ Hz$*.*

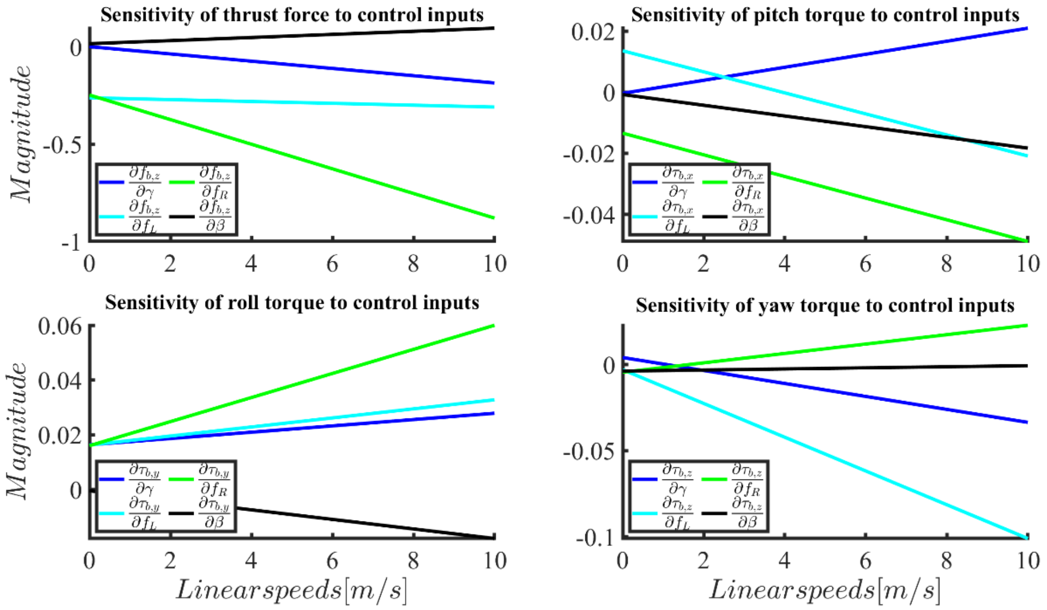

Figure III-9: Jacobian values with linear speeds in the range of $[0\ 10\ m/s]$*, while* $\gamma = 25°$ $\beta = 50°$ *and* $f_L = f_R = 15\ Hz$*.*

High coupling terms arise when linear and angular speeds increase. $f_{b,x}$ is coupled with γ and β as the angular speeds increase for both $\gamma = 0°$ $\beta = 0°$ and $\gamma = 25°$ $\beta = 50°$. $\tau_{b,x}$ is coupled with γ and β as the angular speeds increase for both $\gamma = 0°$ $\beta = 0°$ and $\gamma = 25°$ $\beta = 50°$. $\tau_{b,x}$ is coupled with γ and β as the linear speeds increase for both $\gamma = 0°$ $\beta = 0°$ and $\gamma = 25°$ $\beta = 50°$. $\tau_{b,y}$ is coupled with linear speeds for $\gamma = 25°$ $\beta = 50°$. $\tau_{b,z}$ is coupled with γ and β as the angular speeds increase for both $\gamma = 0°$ $\beta = 0°$ and $\gamma = 25°$ $\beta = 50°$.

b. Force and torque outputs by varying the control inputs

The following four plots are obtained by varying the control inputs independently. Thereafter visualizations of the concept are shown. These graphs give insight to what extent the nonlinearities take hold of the system. For every plot, unless stated otherwise, the velocities and angular velocities are zero. The figures supported by the visualization of the dynamic model. The green lines represent the variable length parameters (γ on top and β below), the black and cyan lines represent the damping and thrust forces of the CoPs, the dashed lines represent the constrained kinematics and the blue lines represent the constraint between the wing root and the CoPs.

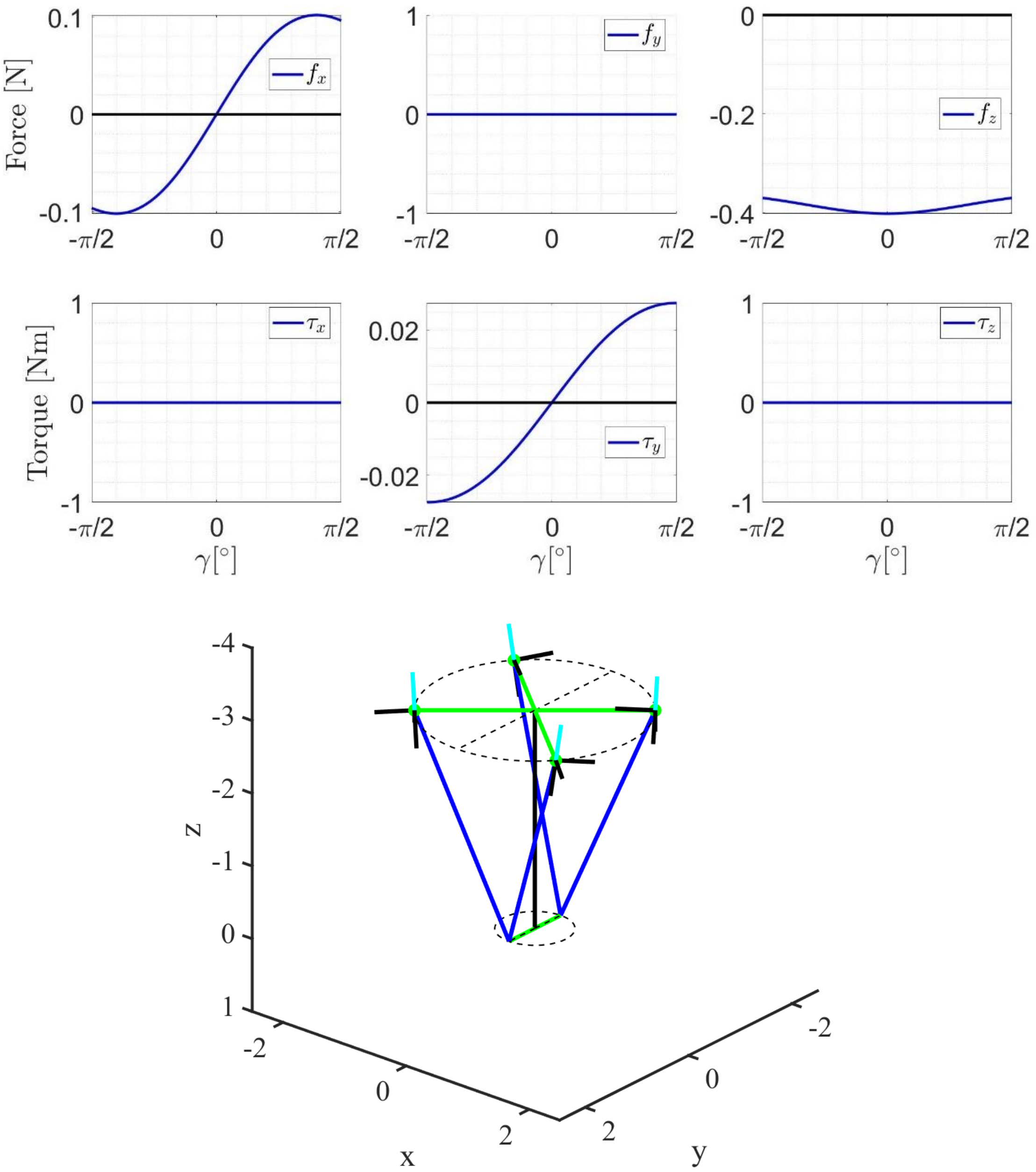

Figure III-10: Varying γ in the range of $[-90°\ 90°]$ in order to generate pitch motion. The dynamical visualization is shown for $\gamma = -50°$ and $\gamma = 50°$. Note that γ is also physically bounded by $50°$.

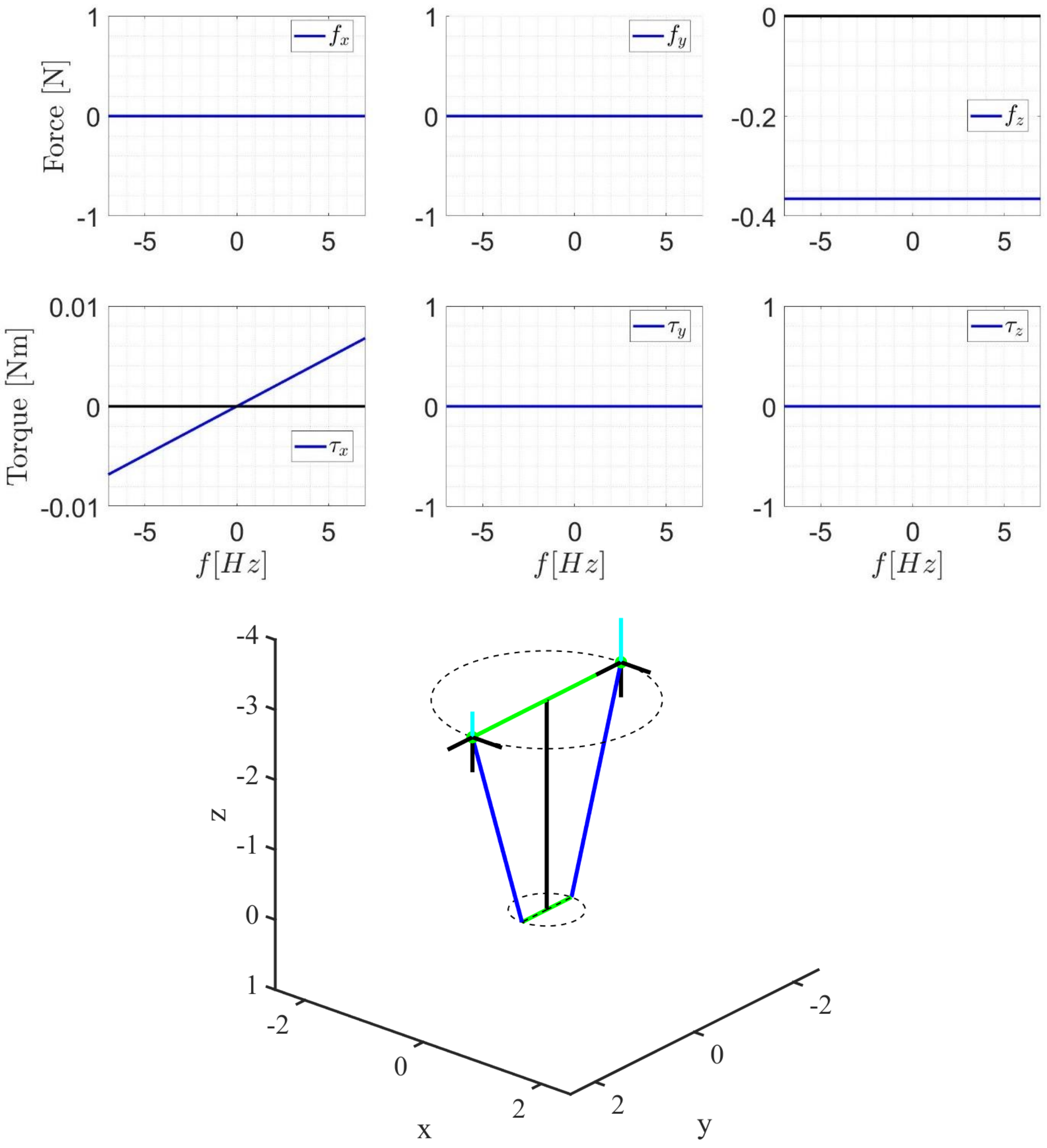

Figure III-11: Varying f_l in the range of $[15 - 22Hz]$ and f_r in the range of $[15 - 8Hz]$ with respect to each other to generate roll motion. The dynamical visualization is shown for $f_l > f_r$.

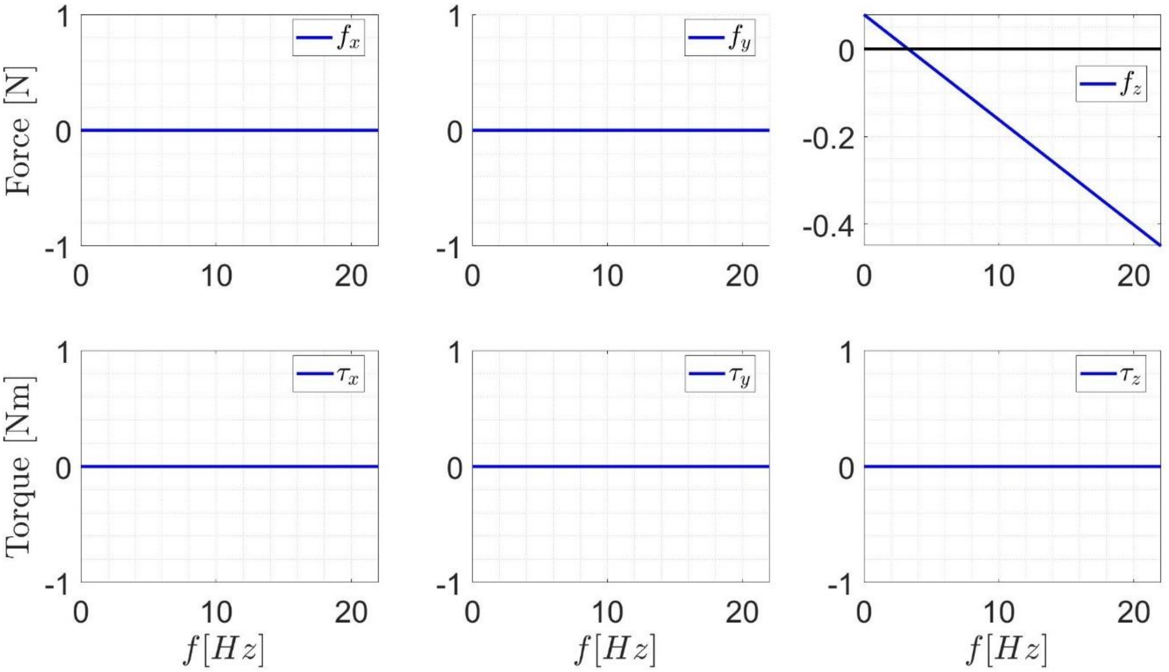

Figure III-12: Varying f_l and f_r in the range of $[0 - 22Hz]$ in order to generate thrust and vertical motion.

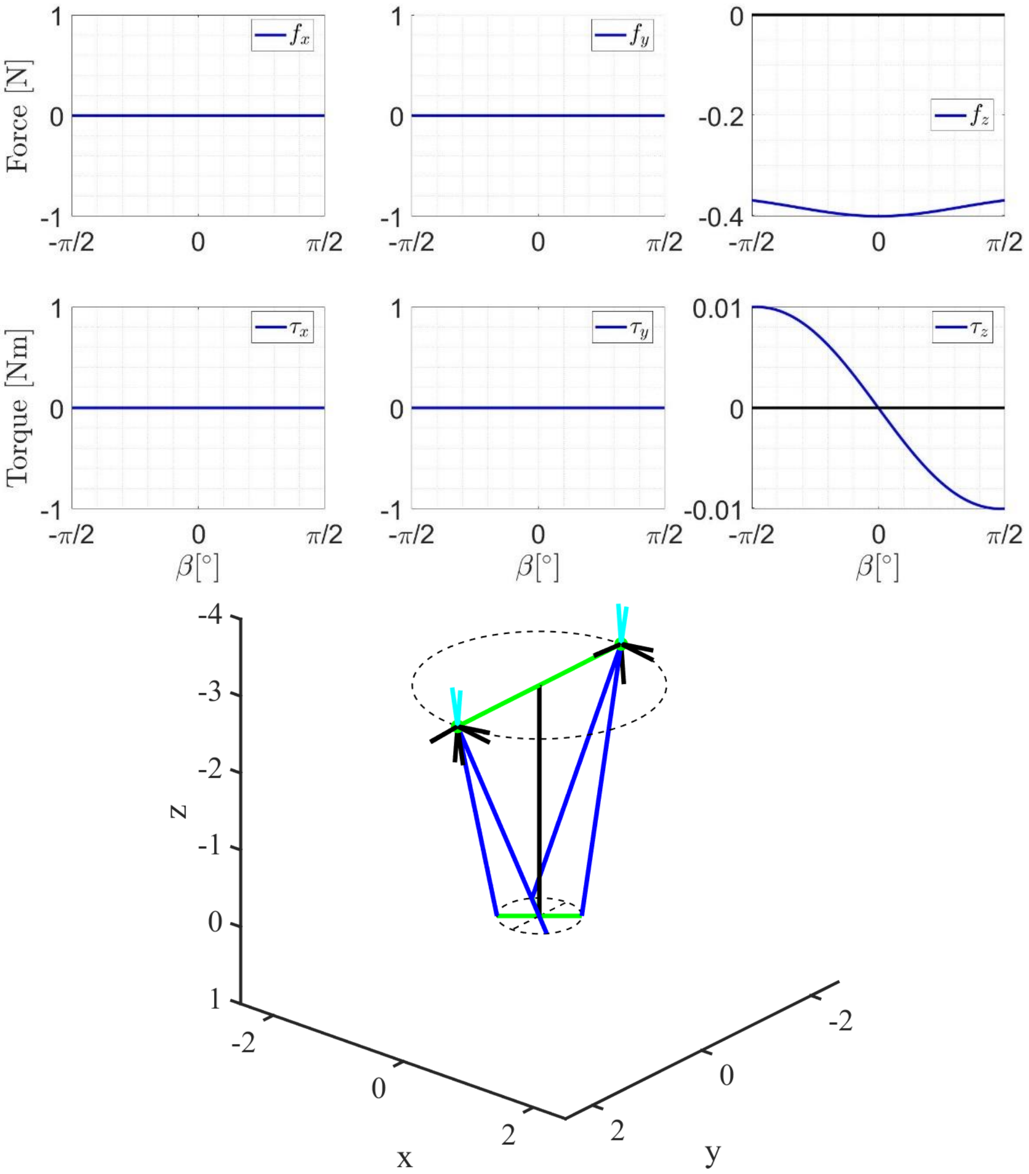

Figure III-13: Varying β in the range of $[0-90°]$ in order to generate yaw motion. The dynamical visualization is shown for $\beta = -50°$ and $\beta = 50°$. Note that β is physically bounded by $\beta = 25°$ and that $\beta = 50°$ is chosen for visual clarity.

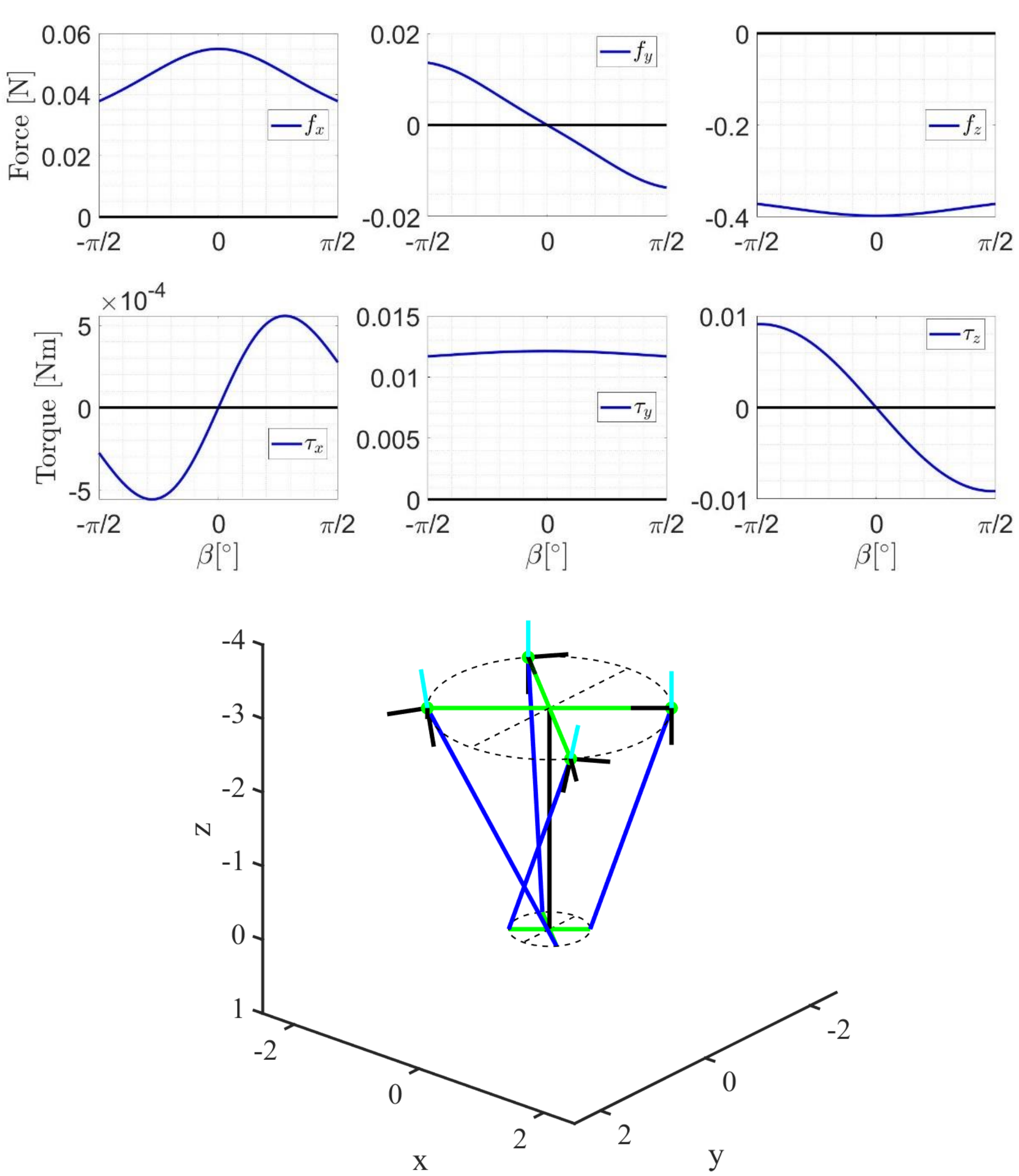

Figure III-14: Combined case by taking the maximum value of $\gamma = 25°$ and varying β in the range of $[0 - 90°]$. The dynamical visualization is shown for $\gamma = -50,\ \beta = -50°$ and $\gamma = 50,\ \beta = 50°$.

Table III-1: Numerical outcomes for different flapping counter force cases. The wings have a flapping frequency of 15.47 Hz.

Case	Value	Outputs $V = \left[f_{b,x}\ f_{b,y}\ f_{b,z}\ \tau_{b,x}\ \tau_{b,y}\ \tau_{b,z}\right]^T$
$Longitudinal$	$u = 1\,m/s$	$[-0.13, 0, -0.36, 0,0.0034,0]^T$
	$q = 1\,rad/s$	$[0.0034,0, -0.36,0, -0.0001,0]^T$
$Lateral$	$v = 1\,m/s$	$[0, -0.13, -0.36, -0.0034, 0,0]^T$
	$p = 1\,rad/s$	$[0, -0.0034, -0.36, -0.0003,0,0]^T$
$Vertical$	$w = 1\,m/s$	$[0,0, -0.39,0,0,0]^T$
	$r = 1\,rad/s$	$[0,0, -0.36,0,0, -0.0008]^T$

c. Dynamic model Matlab simulation

The behavior of the CoPs can be visualized in Matlab. Given the control inputs and length parameters of the Nimble, the visualization of the dynamic model is given as in

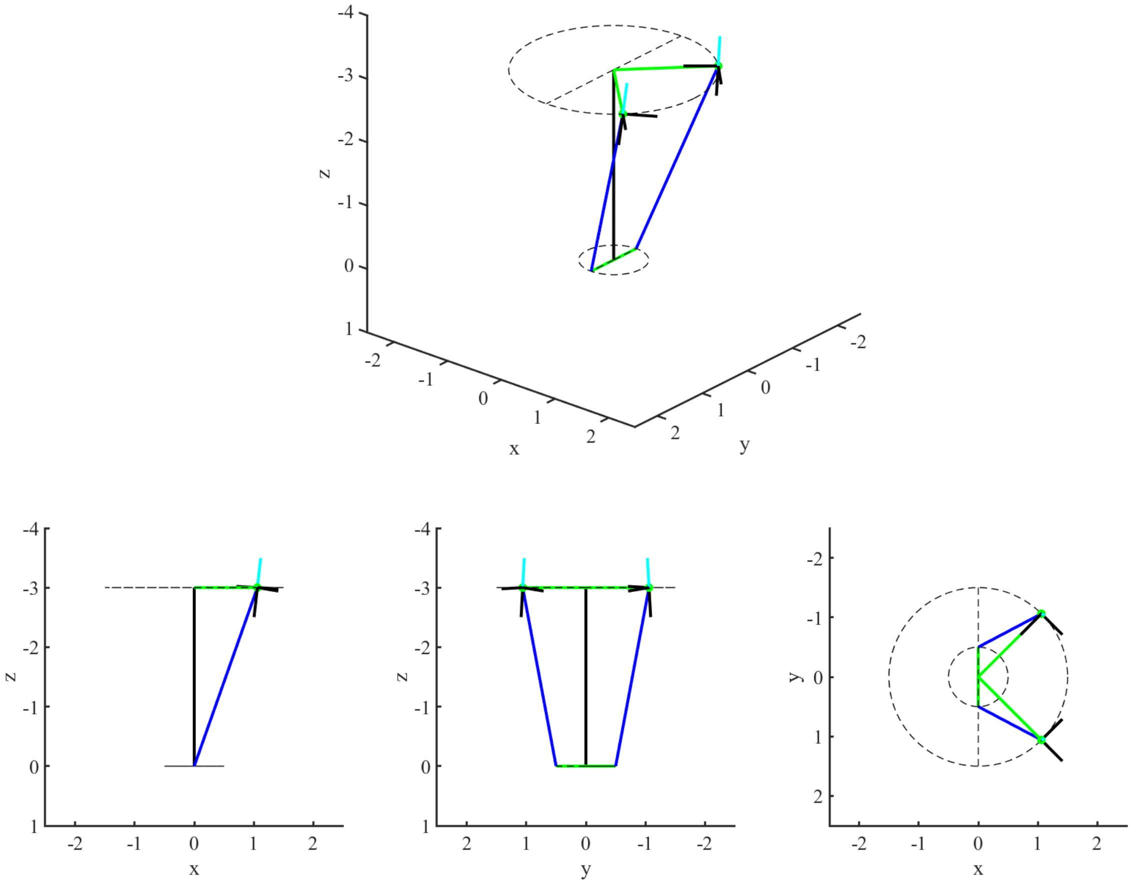

Figure III-15: Aerodynamic model visualization for $\gamma = 45^\circ$ and $\beta = 0^\circ$ *to support Figure 1-1.*

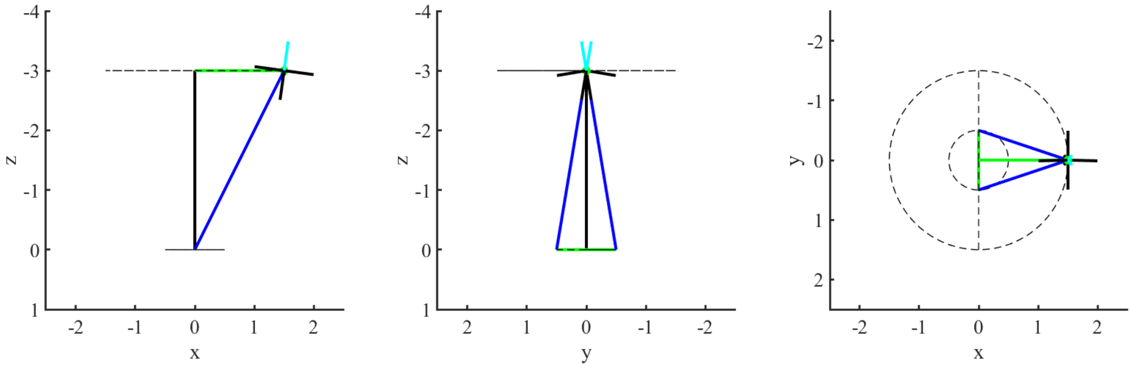

Figure III-16: Aerodynamic model visualization for $\gamma = 90^\circ$ *and* $\beta = 0^\circ$ *to support Figure 1-1.*

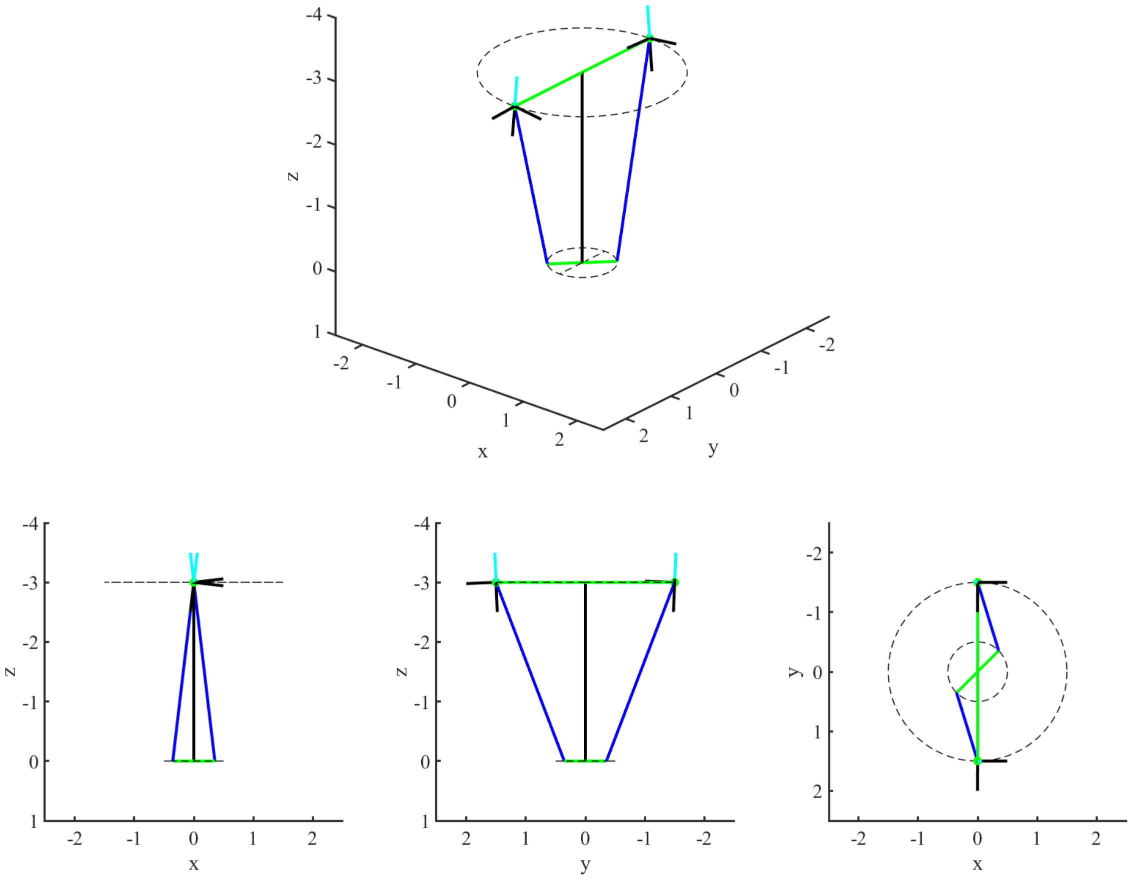

Figure III-17: Aerodynamic model visualization for $\beta = 45°$.

IV. Trajectory control simulations

a. Simple smooth longitudinal trajectories

Simple smooth longitudinal trajectories are shown below in Figure IV-1 and Figure IV-2. The trajectory tracking shows good performance and the tracking error is zero when the tracking target is achieved.

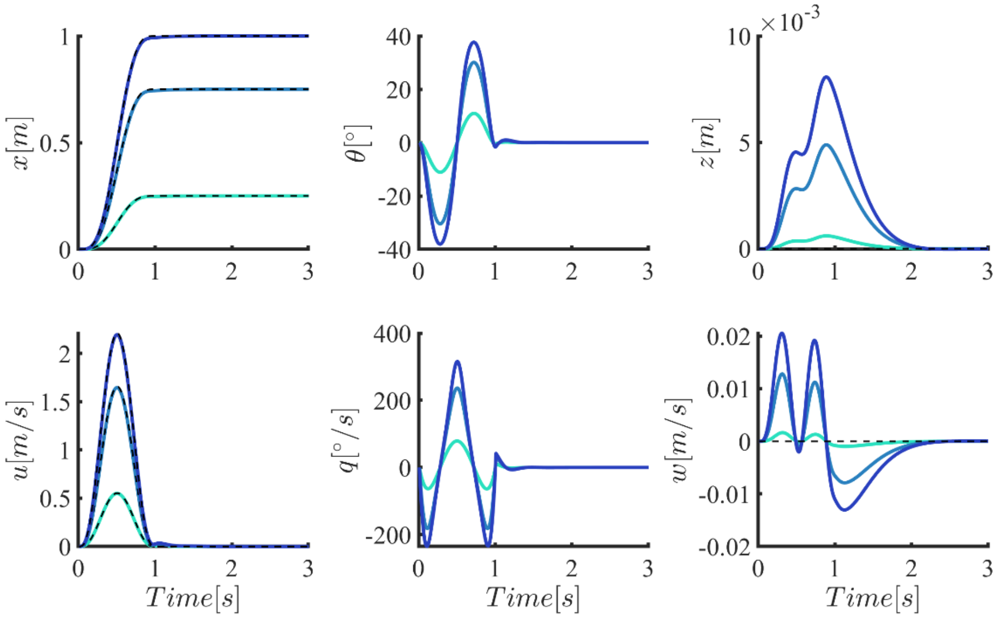

Figure IV-1: States of longitudinal trajectories with end positions $x_e = [0.25\,m\ 0.75\,m\ 1.00\,m]$. The black dotted graphs represent the desired trajectory and the blue graphs show the closed-loop behavior when $F_{b,x} = 0$.

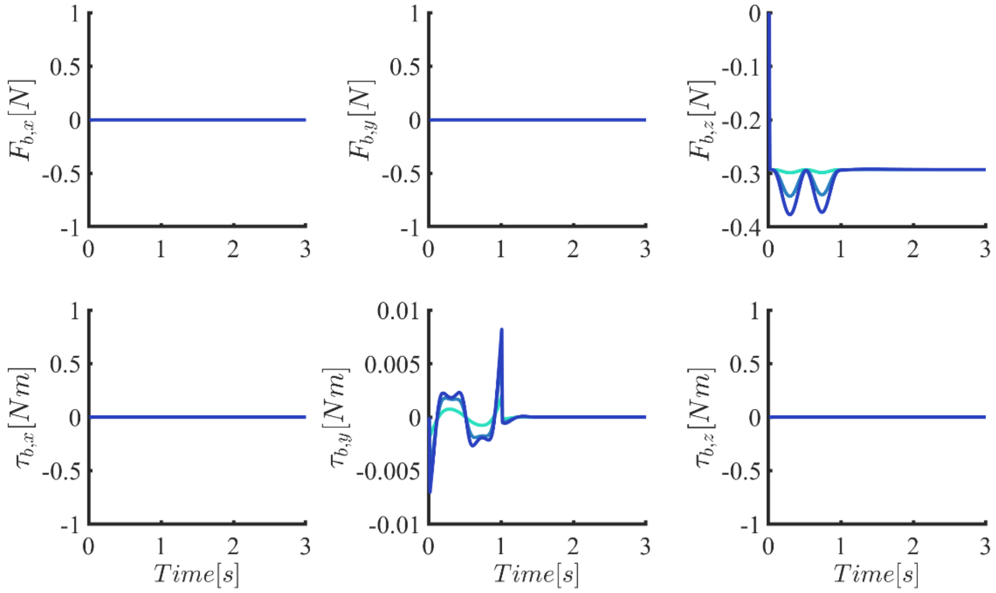

Figure IV-2: Forces and torques of longitudinal trajectories with end positions $x_e = [0.25\,m\ 0.75\,m\ 1.00\,m]$. The blue graphs show the closed-loop behavior when $F_{b,x} = 0$.

Another simulation is shown for different initial conditions in Figure IV-3 and Figure IV-4. The error between the actual and desired states smoothly converge. The position tracking error converges to zero exponentially according to proposition 2.

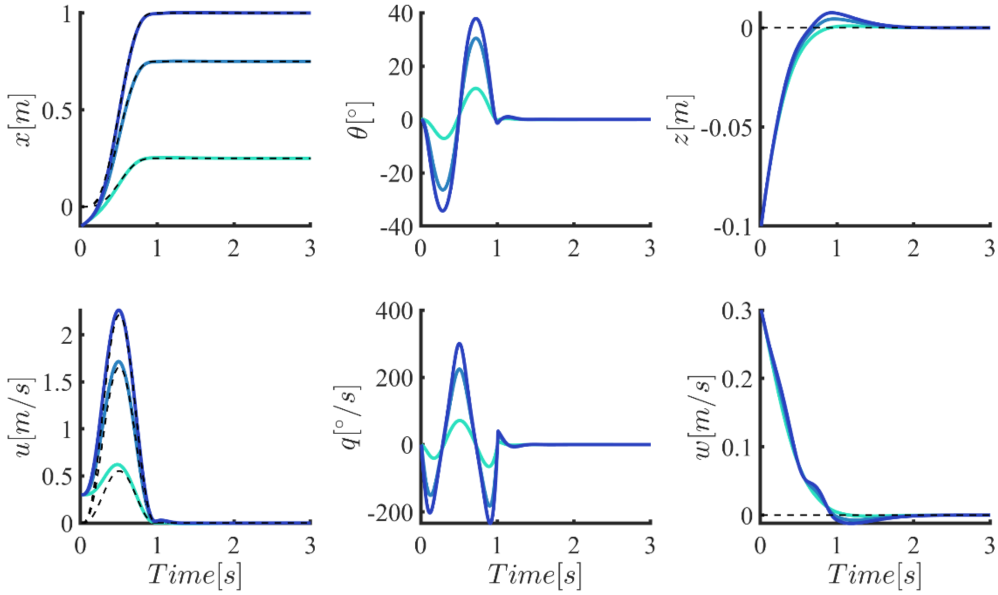

Figure IV-3: States of longitudinal trajectories with end positions $x_e = [0.25\,m\;0.75\,m\;1.00\,m]$. The black dotted graphs represent the desired trajectory and the blue graphs show the closed-loop behavior when $F_{b,x} = 0$. The initial states are $p = [-0.1\;0 - 0.1]$ and $v = [0.3,0,0,3]$.

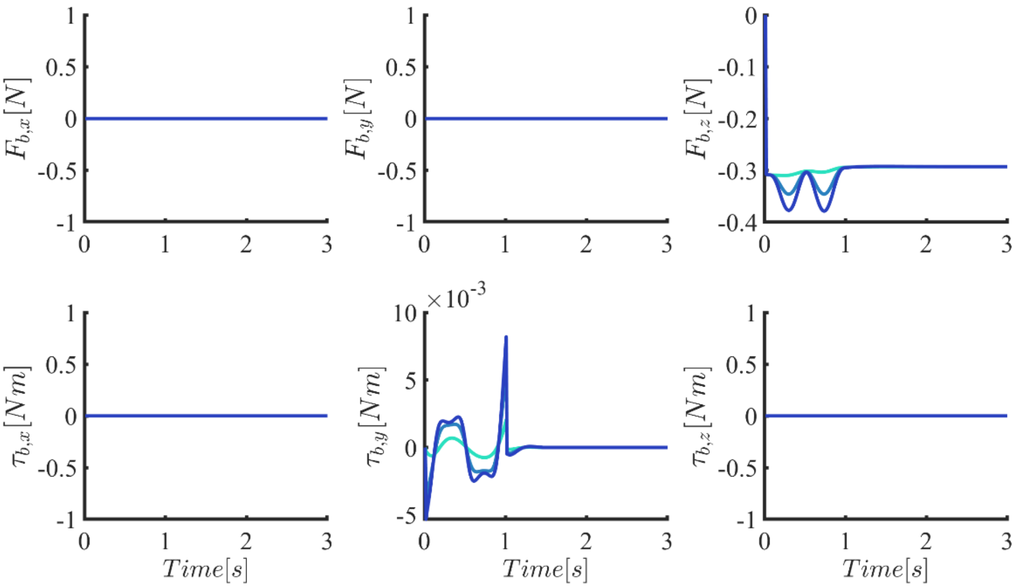

Figure IV-4: Forces and torques of longitudinal trajectories with end positions $x_e = [0.25\,m\;0.75\,m\;1.00\,m]$. The blue graphs show the closed-loop behavior when $F_{b,x} = 0$.The initial states are $p = [-0.1\;0 - 0.1]$ and $v = [0.3,0,0,3]$.

b. Simple smooth lateral trajectories

Simple smooth lateral trajectories are shown below in and Figure IV-5 and Figure IV-6. The trajectory tracking shows good performance and the tracking error is zero when the tracking target is achieved.

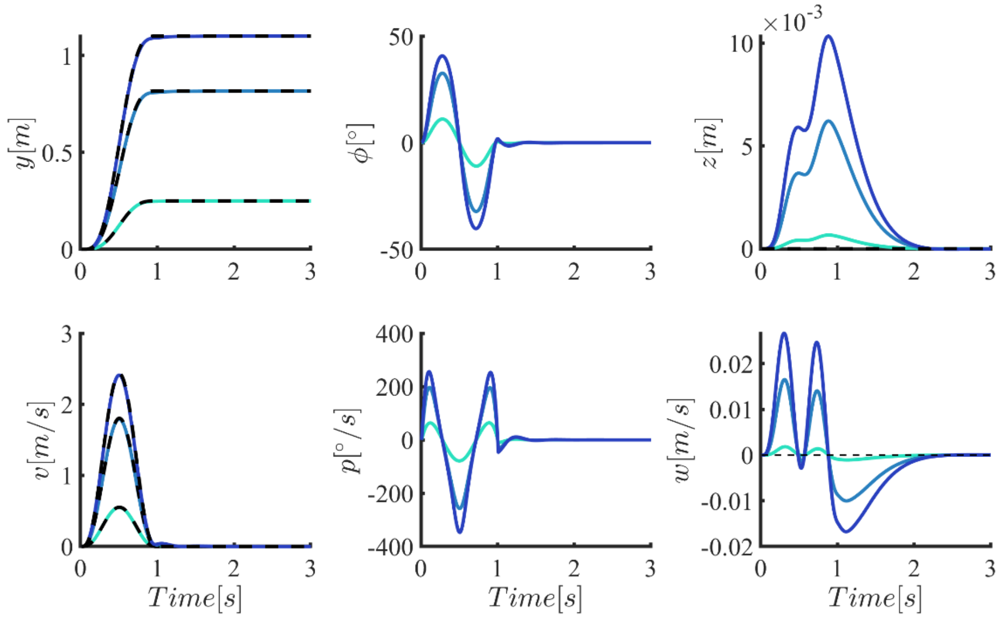

Figure IV-5: States of lateral trajectories with end positions $x_e = [0.25\ m\ 0.75\ m\ 1.00\ m]$. *The black dotted graphs represent the desired trajectory and the blue graphs show the closed-loop behavior when* $F_{b,y} = 0$.

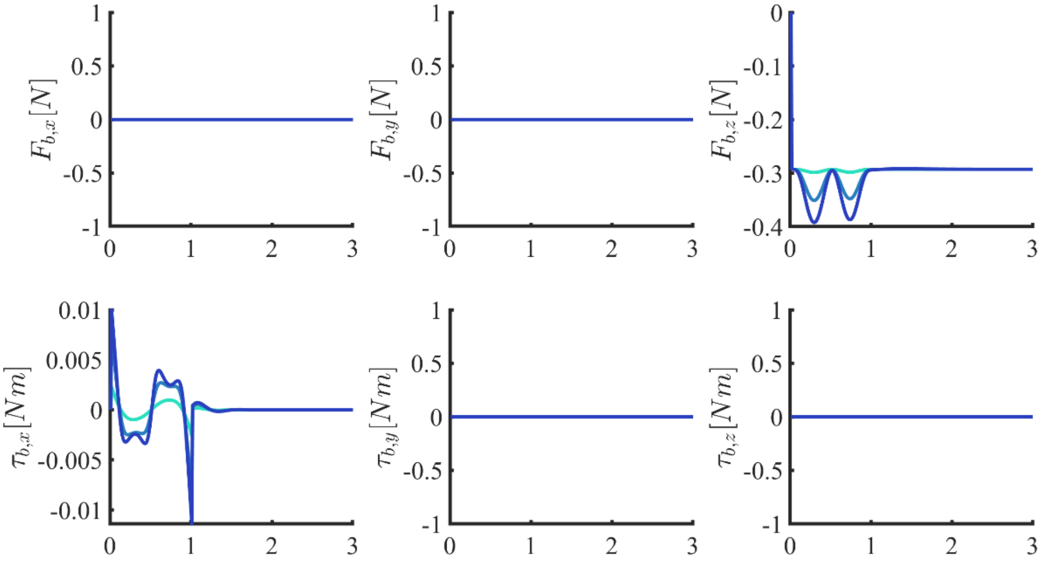

Figure IV-6: Forces and torques of lateral trajectories with end positions $x_e = [0.25\ m\ 0.75\ m\ 1.00\ m]$. *The blue graphs show the closed-loop behavior when* $F_{b,y} = 0$.

Another simulation is shown for different initial conditions in Figure IV-7 and Figure IV-8. The error between the actual and desired states smoothly converge. The position tracking error converges to zero exponentially according to proposition 2.

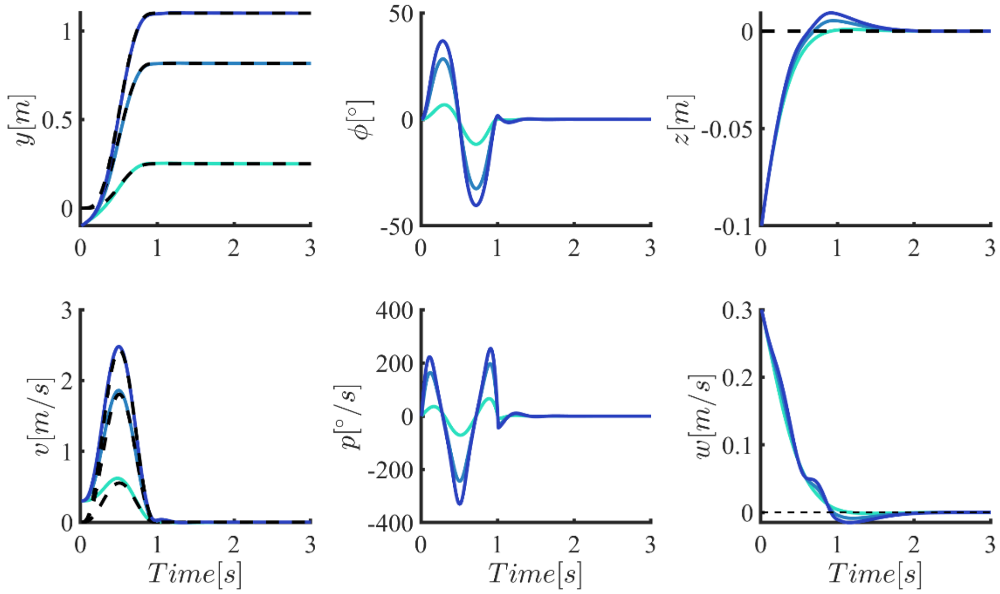

Figure IV-7: States of longitudinal trajectories with end positions $x_e = [0.25\,m\;0.75\,m\;1.00\,m]$. The black dotted graphs represent the desired trajectory, the green graphs represent the closed-loop behavior when $F_{b,z} = 0$ and the blue graphs show the closed-loop behavior when $F_{b,x} = 0$. The initial states are $p = [-0.1\;0 - 0.1]$ and $v = [0.3,0,0,3]$.

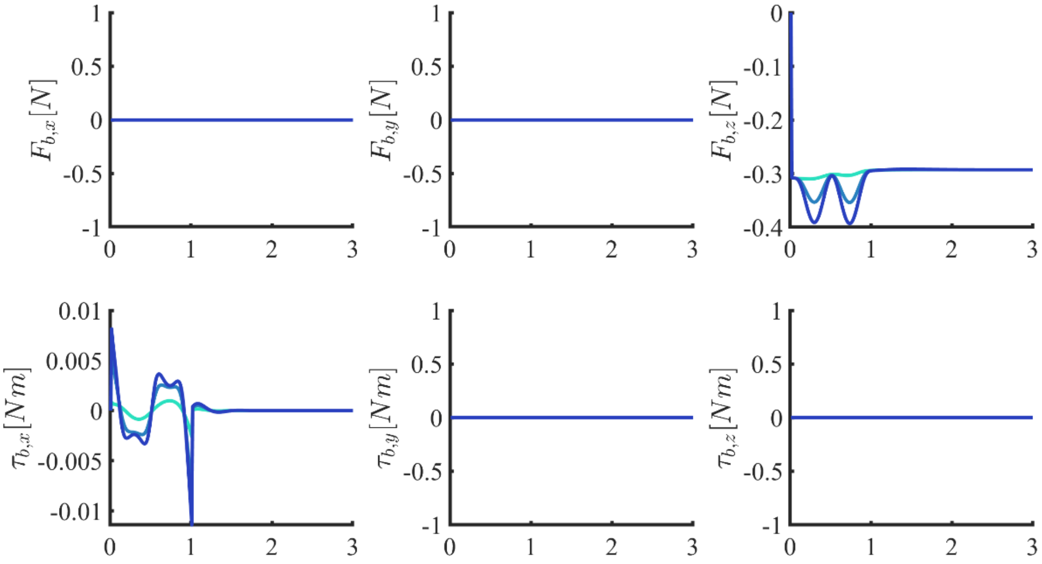

Figure IV-8: Forces and torques of longitudinal trajectories with end positions $x_e = [0.25\,m\;0.75\,m\;1.00\,m]$. The blue graphs show the closed-loop behavior when $F_{b,x} = 0$.The initial states are $p = [-0.1\;0 - 0.1]$ and $v = [0.3,0,0,3]$.

V. Parameter estimation

Next to the longitudinal parameter estimation in [7], the lateral dynamics are investigated in this section. System identification deals with building a mathematical model of a dynamic system from measured input-output responses from certain experiments. It is beneficial to use a relatively simple model with the least unknown parameters, but still an accurate enough model to use for stability analyses, control design and simulation. In other words, the model should preserve prediction capacity, yet adequately represent physical phenomena. The model has to match data sufficiently, facilitate the successful estimation of unknown parameters whose existence can be substantiated, and have adequate prediction capability. Parameter estimation plays a crucial role in accurately describing system behavior as parametric dynamic models. Parameter estimation has been a crucial task due to challenges in precisely estimating aerodynamic parameter and inertia tensors. The Nimble is inherently unstable and therefore system identification has to be performed in closed loop. The maneuvers, and thereby the input-output data, should excite the relevant dynamics, only the natural dynamics are typically damped by the controller [20]. Nonlinear optimization techniques are required to solve this nonlinear estimation problem. Therefore, the parameter estimation is tackled by employing the Nonlinear Least Squares. The unknown parameters are estimated by minimizing the sum of squared errors between measured and modelled output data.

a. In-Flight Data acquisition and data preparation

The in-flight data acquisition is an important aspect of the system identification cycle. The experiment should be carefully carried out and includes the instrumentation system, the flight conditions, and in particular the maneuvers. The set of inputs is an important experiment choice and influences the accuracy of the system identification from flight measurements. Moreover, it is time consuming to perform experiments with the DelFly Nimble. Therefore, data is used from [10]. Due to the inherent instability of the attitude system and lack of space simple maneuvers are used for data acquisition. From [10], the in-flight data is obtained with the OptiTrack-Motion Capture System (OMCS). The OMCS is a set-up in the $10m \times 10m \times 7m$ flight testing facility, the CyberZoo, at the Delft University of Technology. The 12 Prime 17W OptiTrack cameras in the CyberZoo accurately measure the position and orientation of the Nimble, equipped with reflective markers, up to 360 Hz. For more information about the Data can be found in [10]. The measured variables are shown in Table V-1.

Table V-1: Measured data during flight.

Type sensor	Measured variables	Unit
OptiTrack	Position	(x, y, z)
	Attitude quaternions	(q_0, q_1, q_2, q_3)
	Dihedral angle	γ
IMU AHRS	Angular velocity	(p, q, r)
	Linear accelerations	(a_x, a_y, a_z)
On-board extra	Flapping frequency	(f_L, f_R)
	Set-points	$(\phi_{cmd}, \theta_{cmd}, \psi_{cmd})$
	Dihedral command	γ_{cmd}

To reduce noise and to preserve the body dynamics, a third order Butterworth filter with a cutoff frequency of 5 Hz was used. This falls under the assumption that the time varying effects are filtered out and that body and flapping dynamics were uncoupled. The OptiTrack system was selected as the primary data acquisition system to derive all the states required for system identification.

b. Parameters

The equations of motion have the unknown parameters defined in Table V-2. The mass of the Nimble is measured by a precision scale and the moments of inertia were estimated from the locations and masses of the individual components [7], [10]. The parameter estimation solver uses these parameters as initial guess initial estimate.

Table V-2: Parameters of the dynamic model of the DelFly Nimble. (with tracking markers)*

Parameter	Value	Unit	Description
$\boldsymbol{m}$	29.4	g	Body mass of the Nimble
$\boldsymbol{I_{xx}}$	$9.20e-5$ $1.02e-4^*$	kgm^2	Principle moment of inertia around the body its x-axis
$\boldsymbol{I_{yy}}$	$6.73e-5$ $7.87e-5^*$	kgm^2	Principle moment of inertia around the body its y-axis
$\boldsymbol{I_{zz}}$	$3.1e-58$ $3.54e-5^*$	kgm^2	Principle moment of inertia around the body its z-axis
$\boldsymbol{I_{xy}}$	$2.11e-10$ $-2.78\ \mathrm{e}-7^*$	kgm^2	Off-diagonal term around the body its x-axis while rotating about the body its y-axis (and vice versa)
$\boldsymbol{I_{xz}}$	$-9.92e-06$ $-9.15\ \mathrm{e}-6^*$	kgm^2	Off-diagonal term around the body its x-axis while rotating about the body its z-axis (and vice versa)
$\boldsymbol{I_{yz}}$	$1.50e-08$ $-3.97\ \mathrm{e}-7^*$	kgm^2	Off-diagonal term around the body its y-axis while rotating about the body its z-axis (and vice versa)
$\boldsymbol{x_{CoM}}$	$0.56e-03$ $-0.03e-3^*$	m	Longitudinal position of the CoM with respect to the intersection of the leading-edge plane and the yaw axis
$\boldsymbol{y_{CoM}}$	$-0.01e-03$ $0.15e-03^*$	m	Lateral position of the CoM with respect to the intersection of the leading-edge plane and the yaw axis
$\boldsymbol{z_{CoM}}$	$55.0e-03$ $51.2e-03^*$	m	Vertical position of the CoM with respect to the intersection of the leading-edge plane and the yaw axis
$\boldsymbol{l_\gamma}$	0.0810	m	The length from the CoM straight to the CoP in the horizontal xy-plane
$\boldsymbol{l_z}$	0.0271	m	The length from the CoM straight to the CoP in the z-direction
$\boldsymbol{l_\beta}$	$0.5l_\gamma$	m	Horizontal length from the CoM straight to the wing root rod attachment
$\boldsymbol{h}$	$1.5l_\gamma$	m	Vertical length from the CoP straight to the wing root rod attachment
$\boldsymbol{d_x}$	d_y	[-]	The damping coefficients of the CoP along the x-axis
$\boldsymbol{d_y}$	0.0042	[-]	The damping coefficients of the CoP along the y-axis
$\boldsymbol{d_z}$	9.1600e-04	[-]	The damping coefficients of the CoP along the z-axis
$\boldsymbol{c_1}$	0.0240	[-]	Coefficient used to map the flapping frequencies to the thrust force of the CoPs
$\boldsymbol{c_2}$	0.0785	[-]	Coefficient used to map the flapping frequencies to the thrust force of the CoPs
$\boldsymbol{d_{b,x}}$	0	[-]	Damping coefficients of the body along the x-axis
$\boldsymbol{d_{b,y}}$	0	[-]	Damping coefficients of the body along the y-axis
$\boldsymbol{d_{b,z}}$	0	[-]	Damping coefficients of the body along the z-axis

c. Matlab Model

The dynamic model derived in Chapter 3 is nonlinear and non-convex. In addition, the model is a multiple-input multiple-output (MIMO) closed-loop system and parameter identification is applied in Matlab with input and output (I/O) data.

Table V-3: System model for identification and parameter estimation.

Components of the Nimble	Description
Variables	$g, m, I_{xx}, I_{yy}, I_{zz}$
Parameters	$d_{b,x}, d_{b,y}, d_{b,z}$ $d_x, d_y, d_z, l_\gamma, l_z$
Equations	(3-26)
Input	$f_l, f_r, \gamma, \beta, F_l, F_R, \dot{\omega}, \dot{\omega}_b$
Output	ω, ω_b

MATLAB software uses optimization techniques to estimate model parameters. The used variables, parameters, equations, inputs and outputs are summarized in Table V-3 and the system identification model is shown in Figure V-1. Variables such as gravity mass and inertia can be identified by simple measurements. In each optimization iteration, the described model is simulated with the current parameter values. Subsequently, the squared error between the simulated and measured output is computed and minimized. The estimation is complete when the optimization method has found a local minimum, which is hopefully a global minimum. The parameters of Table V-2 are used for initial guesses. Moreover, constraints of the parameters are taken into account to avoid big deviations from the initial guesses.

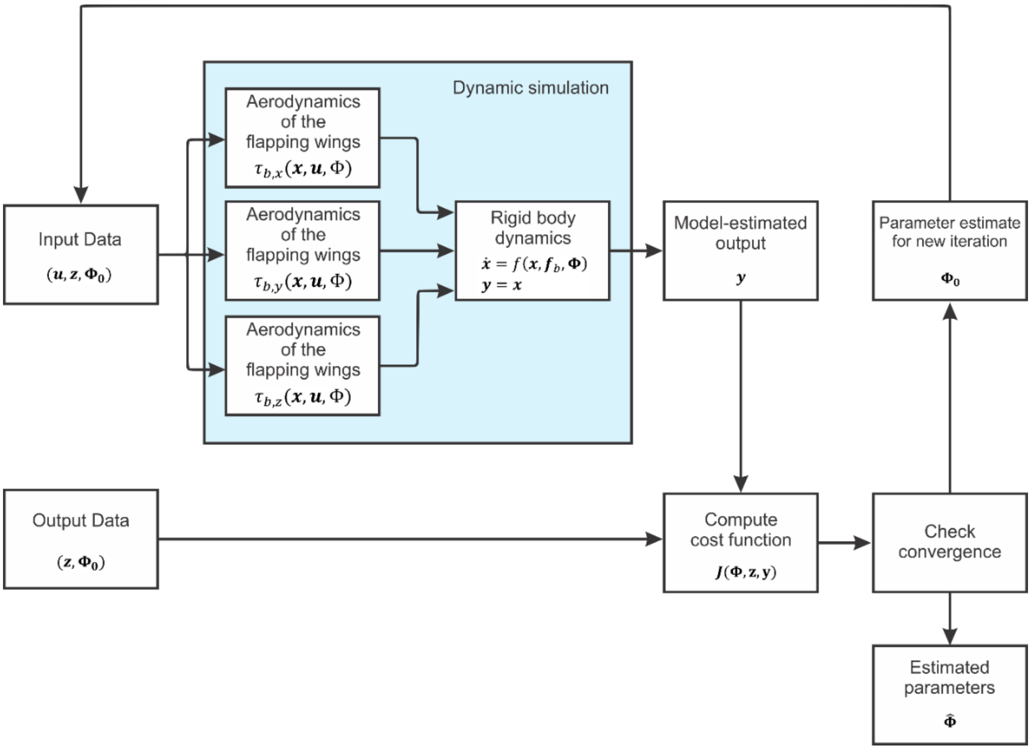
Dynamic simulation
Aerodynamics of the flapping wings $\tau_{b,x}(\boldsymbol{x},\boldsymbol{u},\Phi)$
Aerodynamics of the flapping wings $\tau_{b,y}(\boldsymbol{x},\boldsymbol{u},\Phi)$
Aerodynamics of the flapping wings $\tau_{b,z}(\boldsymbol{x},\boldsymbol{u},\Phi)$
Rigid body dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{f}_b, \boldsymbol{\Phi})$ $\boldsymbol{y} = \boldsymbol{x}$
Input Data $(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\Phi}_0)$
Model-estimated output $\boldsymbol{y}$
Parameter estimate for new iteration $\boldsymbol{\Phi}_0$
Output Data $(\boldsymbol{z}, \boldsymbol{\Phi}_0)$
Compute cost function $J(\boldsymbol{\Phi}, \boldsymbol{z}, \boldsymbol{y})$
Check convergence
Estimated parameters $\hat{\boldsymbol{\Phi}}$

Figure V-1: System identification model.

d. Parameter estimation algorithm

For the class of nonlinear non-convex unconstrained problems, the nonlinear least squares (NLS) can be employed to estimate the parameters. Generally, the NLS problem definition is described as finding a vector that minimizes the sum of squares, possibly subject to constraints. A general NLS model formulation is given by

$$\min_{\Phi} V(\Phi) = \sum_{k=1}^{N} \big(z(k) - f(\Phi)\big)^2, \tag{V-1}$$

The cost function $V(\Phi)$ minimizes the sum of the squared errors, the difference between the model output and the measurement output. In Equation (V-1), N represents the number of data samples, and $z(k)$ and $f(\Phi)$ are the measured and model-predicted outputs, respectively, at time step k.

Table V-4: Parameter estimation options.

Estimation options	Description
Optimization method	Nonlinear Least-Squares
Parameter tolerance	$1e-6$
Maximum iterations	$1e+4$
Function tolerance	$1e-6$

e. Estimated parameters

The model demonstrates good prediction capabilities for comparable maneuvers. The parameters are physically well defined and have an acceptable accuracy. The ideal case would be that the differences between predicted values from the model and measured values are caused by measurement noise and most deterministic elements in the measured output are represented by the identified model. In Figure V-2, the lateral dynamics of the DelFly Nimble are shown. Although some behaviors are not captured, probably due to that the excitation is not rich enough, the overall dynamics are well captured. Both the force along the y-direction and the roll torque have high correlations of determination values and low RMSEs. The force along the z-direction has a low correlation of determination, nevertheless a low RMSE. In Table V-6 the estimated parameters are shown. Next to the obtained knowledge of the longitudinal dynamics in [7], this validates the lateral dynamics.

Table V-5: Parameter estimation metrics for the lateral dynamics.

	$\boldsymbol{F_{by}}$	$\boldsymbol{F_{bz}}$	$\boldsymbol{\tau_{bx}}$
Correlation of determination ($\boldsymbol{R^2}$)	0.9326	-0.2723	0.8633
RMSE	0.0119	0.0089	$5.58e-4$

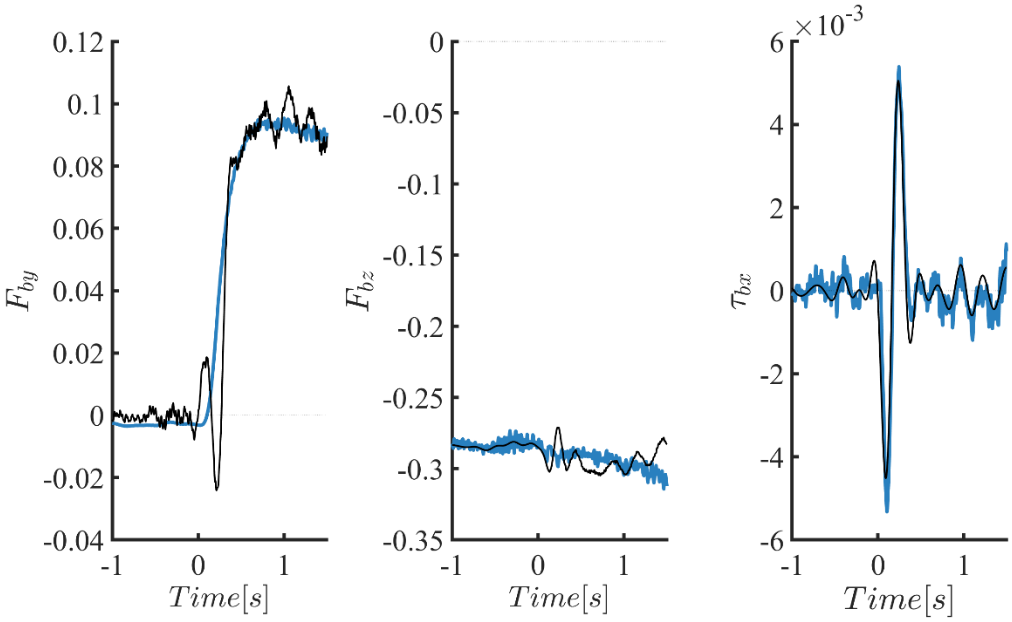

Figure V-2: Measured (black) and simulated lateral forces and moments (blue).

Table V-6: Estimated parameters. (In [7] $d_x = 4.21e-03 = 3.4e-03 \cdot 1.22.$)*

Parameter	l_γ	d_y	d_z	d_{by}	d_{bz}
Estimated Value	0.081	$3.4e-03^*$	$9.16e-04$	0.0202	0.1071
Estimated Value from [7]	0.081	–	$9.16e-04$	–	–

I. The derivatives of A and C

The derivatives of A and C are given by

$$\begin{aligned}
&A = -k_p e_p - k_v e_v - mgz^{\mathcal{I}} + m\ddot{p}_d,\\
&\dot{A} = -k_p e_v - k_v e_a + m\dddot{p}_d,\\
&\dot{A} = -k_p e_v - k_v(\ddot{p} - \ddot{p}_d) + m\dddot{p}_d,\\
&\ddot{A} = -k_p e_a - k_v e_j + m\ddddot{p}_d,\\
&\ddot{A} = -k_p(\ddot{p} - \ddot{p}_d) - k_v \ddot{e}_{\dot{p}} + m\ddddot{p}_d,\\
&C = z_c^{\mathcal{B}} \times x_d^{\mathcal{B}}\\
&\dot{C} = \dot{z}_c^{\mathcal{B}} \times x_d^{\mathcal{B}} + z_c^{\mathcal{B}} \times \dot{x}_d^{\mathcal{B}}\\
&\ddot{C} = \ddot{z}_c^{\mathcal{B}} \times x_d^{\mathcal{B}} + \dot{z}_c^{\mathcal{B}} \times \dot{x}_d^{\mathcal{B}} + \dot{z}_c^{\mathcal{B}} \times \dot{x}_d^{\mathcal{B}} + z_c^{\mathcal{B}} \times \ddot{x}_d^{\mathcal{B}}
\end{aligned}$$

The derivatives of the control body axes are given by

(I-1)

$$\begin{aligned}
&z_c^{\mathcal{B}} = -\frac{A}{\|A\|},\\
&y_c^{\mathcal{B}} = -\frac{C}{\|C\|},\\
&x_c^{\mathcal{B}} = y_c^{\mathcal{B}} \times z_c^{\mathcal{B}}\\
&\dot{z}_c^{\mathcal{B}} = -\frac{\dot{A}}{\|A\|} + \frac{A\cdot\dot{A}}{\|A\|^3}A,\\
&\dot{y}_c^{\mathcal{B}} = -\frac{\dot{C}}{\|C\|} + \frac{C\cdot\dot{C}}{\|C\|^3}C,\\
&\dot{x}_c^{\mathcal{B}} = \dot{y}_c^{\mathcal{B}} \times z_c^{\mathcal{B}} + y_c^{\mathcal{B}} \times \dot{z}_d^{\mathcal{B}},\\
&\ddot{z}_c^{\mathcal{B}} = -\frac{\ddot{A}}{\|A\|} + \frac{2A\cdot\dot{A}}{\|A\|^3}\dot{A} + \frac{\|\dot{A}\|^2 + A\cdot\ddot{A}}{\|A\|^3}A - \frac{3(A\cdot\dot{A})^2}{\|A\|^5}A,\\
&\ddot{y}_c^{\mathcal{B}} = -\frac{\ddot{C}}{\|C\|} + \frac{2C\cdot\dot{C}}{\|C\|^3}\dot{C} + \frac{\|\dot{C}\|^2 + C\cdot\ddot{C}}{\|C\|^3}A - \frac{3(C\cdot\dot{C})^2}{\|C\|^5}C,\\
&\ddot{x}_c^{\mathcal{B}} = \ddot{y}_c^{\mathcal{B}} \times z_c^{\mathcal{B}} + 2\dot{y}_c^{\mathcal{B}} \times \dot{z}_c^{\mathcal{B}} + y_c^{\mathcal{B}} \times \ddot{z}_d^{\mathcal{B}}.
\end{aligned}$$

II. Supplementary tables

List of components of the Nimble and their weights.

Component	Number	Unit Weight
Lisa/S autopilot	1	2.12g
DelTang Rx31 receiver	1	0.27g
MX-3A ESC	2	0.38 g
HK-5330 servo actuator (removed casing)	2	1.63g
micro SD card 2GB	1	0.24g
Hyperion CX G^3 25C 180 mAH LiPo battery	1	4.70g
BL DC motor	2	1.43g
Flapping mechanism	2	2.38g
Wing-pair (including leading and root edges)	2	1.19g
Control mechanisms (total)	1	1.35g
Robot structure (total)	1	2.50g
Wiring, glue & other (total)	1	1.61g
Total weight (no payload)	28.24g	
Total weight (with tracking markers)	29.85g	

III. Definitions

Group [23]**:**

The group operation is a set $\mathcal{M}$ with an operation $\cdot$ for which

Associativity

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) \; \forall a, b, c \in \mathcal{M}$$

Identity

$$\exists d, I \in \mathcal{M} \; s.t. \; d \cdot I = I \cdot d = d \; \forall d$$

Inverse

$$\exists a^{-1} \in \mathcal{M} \; s.t. a^{-1}a = I \quad \forall a \in \mathcal{M}$$

Lie Group [23]**:**

A Lie group is a differentiable manifold group which is smooth. The tangent space in the identity is an algebra, with a skew symmetric operation, called the Lie algebra.

Manifold [23]**:**

An n-dimensional manifold $\mathcal{M}$ is a topological space where every point is endowed with local Euclidean structure. Intuitively, an infinitely small vicinity of that point on the manifold is flat. Each point of this n-dimensional manifold has a neighborhood that is homeomorphic to the n-dimensional Euclidean space, i.e. there exists a continuous function between these spaces. This grants coordinate-free modeling of a system and avoids singularities. A differentiable manifold is a smooth and continuous in which tangent spaces can be defined to allow calculus. That tangents space allows for a connection between a position and a velocity at that position, providing a measure for error functions.