Technische Universität München

Institute of Flight System Dynamics

Design, Implementation and Evaluation of an Incremental Nonlinear Dynamic Inversion Controller for a Nano-Quadrotor

–

Entwurf, Implementierung und Eveluierung eines Inkrementellen Nichtlinearen Dynamischen Inversionsreglers für einen Nano-Quadrotor

Semesterarbeit

Author:

Matriculation number:

Supervisor:

April 2020

Statutory Declaration

I, — —, declare on oath towards the Institute of Flight System Dynamics of Technische Universität München, that I have prepared the present Semester Thesis independently and with the aid of nothing but the resources listed in the bibliography.

This thesis has neither as-is nor similarly been submitted to any other university.

Garching, April 22, 2020

— —

Kurzfassung

Inkrementelle Nichtlineare Dynamische Inversion (INDI) is ein vielversprechender Reglertyp, welcher häufig für die Regelung von nichtlinearen Flugsystemen unterschiedlicher Art eingesetzt wird. Ein Vorteil der INDI ist, dass kein detalliertes Modell der geregelten Strecke benötigt wird. Außerdem, wirkt INDI sehr effektiv gegen Störungen. Diese Semesterarbeit befasst sich mit der Auslegung eines INDI-Reglers für die Lage- und Positionsregelung eines Nano-Quadrotors. Die Arbeit beginnt mit der Herleitung des Regelalgorithmus. Das hergeleitete Regelgesetz wird dann erst in Simulink entwickelt und anschließend auf der Quadrotor-Hardware implementiert. Dazu werden solche Implementierungsaspekte wie die Ermittlung der Eingangsmatrix, Messung und Berechnung der Aktuatorzeitkonstante sowie Ermittlung des Schub-Mapping-Parameters besprochen. Abschließend, wird der implementierte Regler auf die Fähigkeit Störungen zu unterdrücken getestet. Der finale Regler ist über das offizielle Open-Source-Firmware des Crazyflie-Quadrotors verfügbar.

Abstract

Incremental Nonlinear Dynamic Inversion (INDI) is a promising control technique, widely used for control of different types of aircraft systems. Besides providing high-performance nonlinear control, this controller type does not require a detailed model of the controlled aircraft and is effective against disturbances. This semester thesis describes the development of an INDI controller to control the attitude and the position of a nano-quadrotor. It begins with the derivation of the control algorithm. The controller is then firstly developed in the Simulink environment and afterwards implemented on the embedded hardware of the quadrotor. Subsequently, the implementation aspects of the INDI controller such as estimation of the control effectiveness, measurement of the actuator time constant and estimation of the thrust mapping parameter are discussed. Finally, the implemented controller is tested on the ability to cope with disturbance. The final version of the implemented control algorithm is available via official open source firmware of the Crazyflie quadrotor.

Table of Contents

List of Figures

Table of Acronyms

Acronym	Description
EOM	Equations of Motion
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
IR	Infrared
MAV	Micro Air Vehicle
MCU	Microcontroller
NDI	Nonlinear Dynamic Inversion
NED	North East Down
ToF	Time-of-Flight

Table of Symbols

Latin Letters

Symbol	Unit	Description
$A(s),\ A(z)$	–	Transfer function of the actuator dynamics
$A'(s),\ A'(z)$	–	Transfer function of the actuator dynamics model
$a(t)$	–	Function of the first order lag element in time domain
b	m	Lever arm of the rotors to the C.G. along the body y-axis
d	m/s^2	Disturbance
d_0, d_1, d_2	–	Denominator coefficients of the discrete second order filter
$E(t)$	–	Objective function
$\boldsymbol{F}$	N	Force vector
$\boldsymbol{F}_A$	N	Aerodynamic force vector
$\boldsymbol{F}_G$	N	Gravitational force vector
$\boldsymbol{F}_P$	N	Propulsive force vector
$\boldsymbol{f}(\boldsymbol{x})$	–	Nonlinear vector field
$\boldsymbol{G}$	–	Control effectiveness matrix of the outer loop
$\boldsymbol{G}_1$	–	Control effectiveness matrix of the inner loop
$\boldsymbol{G}_2$	–	Control effectiveness matrix of the inner loop
$\boldsymbol{G}_{inner}$	–	Dynamics matrix of the inner loop
$\boldsymbol{G}_{outer}$	–	Dynamics matrix of the outer loop
$\boldsymbol{G}(x)$	–	Input matrix
g	m/s^2	Gravitational acceleration
$H(s),\ H(z)$	–	Transfer function of the second order filter
$\boldsymbol{h}(\boldsymbol{x})$	–	Nonlinear vector field
$\boldsymbol{I}$	Nm^2	Inertia tensor of the quadrotor
$\boldsymbol{I_{rzz}}$	Nm^2	zz-element of the inertia tensor of the rotor
i	–	Relative degree of the dynamic system
K	–	Compensation gain
K_{cmd}	–	Thrust mapping parameter
K_η	–	Gain of the inner loop

K_ω	–	Gain of the inner loop
K_ξ	–	Gain of the outer loop
$K_{\dot{v}}$	–	Gain of the outer loop
k	kg/s	Drag coefficient
k	–	Gain of the actuator dynamics
k_F	$\frac{N}{rad/s}$	Force constant of the rotors
k_M	$\frac{Nm}{rad/s}$	Moment constant of the rotors
L	–	Lie derivative operator
l	m	Lever arm of the rotors to the C.G. along the body y-axis
$\boldsymbol{M}$	Nm	Moment vector
$\boldsymbol{M}_A$	Nm	Aerodynamic Moment vector
$\boldsymbol{M}_c$	Nm	Control moment vector
$\boldsymbol{M}_G$	Nm	Gravitational Moment vector
$\boldsymbol{M}_{gyro}$	Nm	Moment vector due to the gyroscopic effects of the rotor
$\boldsymbol{M}_{OB}$	–	Transformation matrix from the body-fixed into NED frame
$\boldsymbol{M}_P$	Nm	Propulsion moment vector
m	kg	Mass of the quadrotor
m_{cf}	kg	Mass of the Crazyflie quadrotor
m_r	kg	Additional mass (disturbance
n	–	Number of samples
n_0, n_1, n_2	–	Numerator coefficients of the discrete second order filter
P	–	Parameter
p	rad/s	Roll rate
q	rad/s	Pitch rate
r	rad/s	Yaw rate
s	–	Laplace variable
T	N	Thrust
T	s	Time constant of the actuator dynamics
T_{cmd}	–	Thrust in motor units
T_s	s	Sample time of the controller
$\tilde{T}$	N	Thrust increment

t	s	Time
u	–	Input vector
$\boldsymbol{u}_c$	–	Control vector
u	m/s	x-component of the velocity vector
V	m/s	Component of the velocity vector
v	m/s	y-component of the velocity vector
$\boldsymbol{v}$	m/s	Velocity vector
W	N	Weight of the quadrotor
w	m/s	z-component of the velocity vector
$\boldsymbol{x}$	–	State vector
x	m	Position coordinate in x-direction
$\boldsymbol{y}$	–	Output vector
y	m	Position coordinate in y-direction
$\boldsymbol{z}$	–	Transformed state vector
z	–	z-Transformation variable
z	m	Position coordinate in z-direction

Greek Letters

Symbol	**Unit**	**Description**
α	–	Parameter of the discrete first order lag element
ζ	–	Relative damping coefficient of the second order filter
$\boldsymbol{\eta}$	rad	Attitude vector
Θ	rad	Pitch angle
$\tilde{\Theta}$	rad	Pitch angle increment
Θ_c	rad	Control pitch angle
$\boldsymbol{\nu}$	–	Virtual control input vector
$\boldsymbol{\nu}_{\dot{\boldsymbol{v}}}$	m/s^2	Virtual control input vector of the outer loop
$\boldsymbol{\nu}_{\dot{\boldsymbol{\omega}}}$	rad/s^2	Virtual control input vector of the inner loop
$\boldsymbol{\xi}$	m	Position vector
Φ	rad	Roll/Bank angle

$\tilde{\Phi}$	rad	Roll angle increment
Φ_c	rad	Control roll angle
$\boldsymbol{\phi}(\boldsymbol{x})$	–	State transformation function

Indices

Symbol	Description
0	Initial point in time
a	Passed through the actuator dynamics
B	Body-fixed coordinate frame
D	Down
E	East
f	Filtered
G	Center of gravity
G	Gravitational
K	Kinematic coordinate frame
N	North
O	North-East-Down coordinate frame
R	Reference point
x	x-component of a variable
y	y-component of a variable
z	z-component of a variable

Other Symbols

Symbol	Description
L	Lie derivative operator
∇	Nabla-Operator

1 Introduction

1.1 Motivation

The popularity of Micro Air Vehicles (MAVs) in research and industry has grown over the past years. The large interest in MAVs can be explained with their potential of performing a wide range of civilian and military tasks, reaching from the investigation of hazardous regions up to agriculture mapping [1]. One of the most popular representatives of MAVs are quadrotors, belonging to the rotary-wing MAVs. Their dynamic behaviour is usually of a nonlinear character. Thus, to control such an aircraft, robust control strategies have to be applied. One such strategy is called Incremental Nonlinear Dynamic Inversion (INDI), which is the incremental form of the nonlinearity cancelling technique Nonlinear Dynamic Inversion (NDI). The INDI technique makes it possible to control an aircraft without fully modelling its dynamics. Instead, INDI uses a primitive aircraft model, which eases the development and implementation of the control algorithm. In addition, INDI has good properties of coping with unmodelled gust and disturbances.

1.2 Contribution of the Thesis

The main goal of this thesis was to implement an INDI controller to control the linear acceleration of a nano-quadrotor. Furthermore, to have the ability to control the quadrotors position, the INDI loop had to be augmented with a velocity as well as a position controller. For the role of the research quadrotor, a small quadrotor named Crazyflie 2.1 was employed. Besides the derivation of the control algorithm itself, this thesis covers the process of estimation of the control effectiveness terms, as well as the estimation of other parameters such as actuator dynamics time constant and thrust mapping parameter, which are necessary for proper functioning of the controller. Additionally to the hardware implementation of the control algorithm, this work describes the development of the controller as well as the quadrotor model in Simulink. Finally, it discusses the capability of the successfully implemented INDI controller to cope with disturbances. The hardware version of the control algorithm was then made available to general public via official open source firmware.

1.3 Structure of the Thesis

This semester thesis is organized into five chapters. The introductory chapter is followed by the second chapter, containing the theoretical background. Here, after introducing dynamic equations of motion of an aircraft (2.1), NDI (2.2) and INDI (2.3) control algorithms are discussed. The INDI technique is then used to derive the attitude (2.3.2) and position (2.3.3) controllers for the Crazyflie quadrotor. Chapter three focuses on the implementation aspects of the INDI algorithm derived in the previous chapter. It starts with a short description of the quadrotor hardware (3.1). In the second part (3.2) of the third chapter, simplified model of the Crazyflie quadrotor as well as the attitude and position controllers are developed in the Simulink environment. The third section (3.3) of the same chapter covers aspects that are relevant for the hardware implementation of the INDI control algorithm, such as the estimation of the control effectiveness terms and thrust mapping parameter. After the controller was implemented, it was tested and validated. The test results are summarized in the fourth chapter. The final chapter contains a short conclusion of the performed work.

2 Theoretical Background

The aim of this chapter is to provide the theoretical background which serves as a basis for some of the methods which are presented and applied in the course of this thesis. At the beginning, in section 2.1 general Equations of Motion (EOM) of an aircraft are presented. These are later used to derive the control algorithm and build the Simulink model of the Crazyflie quadrotor. Sections 2.2 and 2.3 describe the Nonlinear Dynamic Inversion (NDI) and Incremental Nonlinear Dynamic Inversion (INDI) methods in general. Later in this chapter (subsections 2.3.2 and 2.3.3) these methods are used to derive the inner and the outer loop of the INDI flight controller for the Crazyflie quadrotor.

2.1 Dynamic Equations of Motion of an Aircraft

The dynamic equations which describe the general motion of an aircraft are usually coupled first order implicit nonlinear differential equations. However, considering a quadrotor control problem, a lot of plausible assumptions (e.g. neglecting the Coriolis and centrifugal acceleration due to the earth rotation) can be made to obtain more simplified versions of those equations. This section presents simplified general EOM and describes the assumptions which were considered to derive them.

For the derivation of the equations of motion the aircraft system will be assumed to be a rigid body. Such a rigid body system can be described uniquely by 12 states. Note that by taking in account additional effects, such as propulsion system dynamics, multi-body dynamics etc., the number of required state variables will increase.

Additionally to the rigid body assumption, it is also assumed that the earth is flat and non-rotating and the reference point of the sum of all external forces acting on the body corresponds to the center of gravity of the body. Thus, the linear momentum equation is written as

$$\sum(\boldsymbol{F}^G)_B = m \cdot \left[(\dot{\boldsymbol{v}}^G)_B + (\boldsymbol{\omega}^{OB}) \times (\boldsymbol{v}^R)_B\right] \tag{2.1}$$

where $\sum(\boldsymbol{F}^G)_B \in \mathbb{R}^{3\times1}$ is the sum of the external forces acting on the system and applied to the center of gravity G, m the mass of the body, $(\dot{\boldsymbol{v}}^G)_B \in \mathbb{R}^{3\times1}$ the linear acceleration of the point G, $(\boldsymbol{v}^R)_B \in \mathbb{R}^{3\times1}$ the velocity of any reference point R, $(\boldsymbol{\omega}^{OB}) \in \mathbb{R}^{3\times1}$ the angular velocity of the body-fixed frame (B) with respect to the North East Down (NED) (O) coordinate frame. Subscript B denotes that all variables are specified in the body-fixed coordinate frame.

The rotational motion of a body is described with the angular momentum equation. To derive it, additionally to the assumptions made above it is also considered that the mass and the mass distribution are quasistationary, meaning $\frac{d}{dt}m = 0$ and $\frac{d}{dt}(\boldsymbol{I})_B = 0$ with $(\boldsymbol{I})_B$ being the inertia tensor of the body defined in the body-fixed frame. Thus, the angular momentum is

$$\sum(\boldsymbol{M}^G)_B = (\boldsymbol{I}^G)_B \cdot (\dot{\boldsymbol{\omega}}^{OB}) + (\boldsymbol{\omega}^{OB}) \times \left[(\boldsymbol{I}^G)_B \cdot (\boldsymbol{\omega}^{OB})\right] \tag{2.2}$$

where $\sum(\boldsymbol{M}^G)_B \in \mathbb{R}^{3\times1}$ is the sum of the external moments acting on the system around the center of gravity G.

The remaining two equations, that are necessary to fully describe the motion of a rigid body in space are the attitude differential equation and the position differential equation. The attitude differential equation describes the relationship between angular rates p, q, r and derivatives of the Euler angles $\dot{\Phi}, \dot{\Theta}, \dot{\Psi}$ leading to

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \frac{\sin\Phi}{\cos\Theta} & \frac{\cos\Phi}{\cos\Theta} \end{bmatrix}_B \begin{bmatrix} p \\ q \\ r \end{bmatrix}_B \tag{2.3}$$

where the angular rates are related to the derivatives of the Euler angles through the *strapdown matrix*.

There are different options to represent the position differential equation. Here, for completeness only, it is written as a simple relationship between the change of the position coordinates in the local NED frame and the velocity coordinates of the same frame

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_O = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_O \tag{2.4}$$

The Equations (2.1)-(2.4) presented above can be used to represent a motion of a general aircraft in a three-dimensional space. Note that terms $\sum(\boldsymbol{F}^G)_B$ and $\sum(\boldsymbol{M}^G)_B$ contain all external forces and moments acting on the rigid body. Considering a general aircraft system, these could be the aerodynamic forces and moments caused by the air flow, propulsion forces and moments, forces caused by the gravitation etc.. The detailed modelling of external forces and moments is presented in sections 2.3.2, 2.3.3 and 3.2.

2.2 Nonlinear Dynamic Inversion

In this subsection, the Nonlinear Dynamic Inversion method is explained. The NDI approach is based on feedback linearization and is also called *Input-Output Linearization*. Often, such type of controllers is involved in tracking control tasks, where the objective is to track some desired trajectory [2]. To derive it, consider the following nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \tag{2.5a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{2.5b}$$

where $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^{m\times 1}$ the input vector, $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ the output vector, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^{m\times 1}$ nonlinear vector fields and $\boldsymbol{G} \in \mathbb{R}^{m\times n}$ an input matrix. Note that the system presented in Equations (2.5) is affine in the input, which is not allways fulfilled. Using a state transformation $\boldsymbol{z} = \boldsymbol{\phi}(\boldsymbol{x})$, the affine system from Equation 2.5a can be transformed into a *normal* (*canonical*) representation.

The core idea behind the input-output linearization method is to find a direct relationship between the desired system output and the control input. After the relationship is found it is inverted to generate the control law. To derive this relationship the output $\boldsymbol{y}$ is differentiated until the input $\boldsymbol{u}$ appears

$$\begin{aligned}\dot{\boldsymbol{y}} = \frac{\partial \boldsymbol{y}}{\partial t} = \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial x}\frac{\partial \boldsymbol{x}}{\partial t} = \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial x}\dot{\boldsymbol{x}} = \nabla \boldsymbol{h}(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}] \\ = \nabla \boldsymbol{h}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) + \nabla \boldsymbol{h}(\boldsymbol{x})\boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} = L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) + L_{\boldsymbol{G}}\boldsymbol{h}(\boldsymbol{x})\boldsymbol{u}\end{aligned} \tag{2.6}$$

In Equation (2.6) $L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})$ is called Lie derivative of $\boldsymbol{h}(\boldsymbol{x})$ with respect to $\boldsymbol{f}(\boldsymbol{x})$. The Lie derivative is defined as $L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) = \nabla \boldsymbol{h}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$ with ∇ being the Nabla operator. Thus, it represents a directional derivative of $\boldsymbol{h}(\boldsymbol{x})$ along the direction of the vector field $\boldsymbol{f}(\boldsymbol{x})$. If the term $L_{\boldsymbol{G}}\boldsymbol{h}(\boldsymbol{x})$ is nonzero, the relationship between input and output is

$$\dot{\boldsymbol{y}} = L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) + L_{\boldsymbol{G}}\boldsymbol{h}(\boldsymbol{x})\boldsymbol{u} \tag{2.7}$$

Now Equation (2.7) can be used to formulate the control law by solving it for $\boldsymbol{u}$ and substituting $\dot{\boldsymbol{y}}$ with $\boldsymbol{\nu}$

$$\boldsymbol{u} = (L_{\boldsymbol{G}}\boldsymbol{h}(\boldsymbol{x}))^{-1}(\boldsymbol{\nu} - L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})) \tag{2.8}$$

The variable $\boldsymbol{\nu}$ is called a *virtual control input* and represents the desired output of the system.

In the example provided above the input-output relationsip was found after the first differentiation of the output $\boldsymbol{y}$. But if after the first differentiation the term $L_{\boldsymbol{G}}\boldsymbol{h}(\boldsymbol{x})$ is zero, output $\boldsymbol{y}$ has to be differentiated until the Lie derivative with respect to $\boldsymbol{G}$ is nonzero. The i-th derivative of the output is then

$$\frac{\partial^i \boldsymbol{y}}{\partial t} = \frac{\partial^i \boldsymbol{h}(\boldsymbol{x})}{\partial t} = L^i_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) + L_{\boldsymbol{G}}L^{i-1}_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})\boldsymbol{u} \tag{2.9}$$

with i being the *relative degree* of the system. Using Equation (2.9) to formulate the control law leads to the following expression for the control input $\boldsymbol{u}$

$$\boldsymbol{u} = L_{\boldsymbol{G}}L^{i-1}_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})^{-1}(\boldsymbol{\nu} - L^i_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})) \tag{2.10}$$

Thus, Equation (2.10) applied to Equation (2.9) yields the simple linear relation

$$\boldsymbol{y}^i = \boldsymbol{\nu} \tag{2.11}$$

The NDI method was widely adopted for civil and military aircrafts and has numerous of extensions [3]. Nevertheless, it has some drawbacks. The major one is that the control law derived using the NDI approach is dependent on the full system dynamics model [4]. The equations of motion of an aircraft usually have a complicated nonlinear character. Thus, describing complex physical phenomena often leads to the inconsistency between the real aircraft and its mathematical representation used in the model. As some of those inconsistencies are inevitable there is a need to build a control law which performance is less dependent on the uncertainties of the model. A method called Incremental Nonlinear Dynamic Inversion can be used to achieve this, it is discussed in the next subsection.

2.3 Incremental Nonlinear Dynamic Inversion

INDI is an incremental form of NDI for which the lack of the accurate system dynamics model does not critically affect the performance of the control algorithm [5]. At first the general form of the INDI is presented in 2.3.1, then the inner and outer control loops for the Crazyflie quadrotor are derived in 2.3.2 and 2.3.3.

2.3.1 General INDI

Assuming that the output variable corresponds to the state varible ($\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{x}$), the incremental form of the system can be obtained by taking a Taylor series expansion of the Equation 2.5a

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}_0) + \boldsymbol{G}(\boldsymbol{x}_0)\boldsymbol{u}_0 \\ &+ \frac{\partial}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\Big|_{\boldsymbol{x}=\boldsymbol{x}_0,\boldsymbol{u}=\boldsymbol{u}_0}(\boldsymbol{x}-\boldsymbol{x}_0) \\ &+ \frac{\partial}{\partial \boldsymbol{u}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\Big|_{\boldsymbol{x}=\boldsymbol{x}_0,\boldsymbol{u}=\boldsymbol{u}_0}(\boldsymbol{u}-\boldsymbol{u}_0)\end{aligned} \tag{2.12}$$

The first term on the right side of the Equation 2.12 is $\dot{\boldsymbol{x}}_{\boldsymbol{0}}$. Also, evaluating the differentiation of the third term leads to

$$\begin{aligned}\dot{\boldsymbol{x}} &= \dot{\boldsymbol{x}}_0 \\ &+ \frac{\partial}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\Big|_{\boldsymbol{x}=\boldsymbol{x}_0,\boldsymbol{u}=\boldsymbol{u}_0}(\boldsymbol{x}-\boldsymbol{x}_0) \\ &+ \boldsymbol{G}(\boldsymbol{x}_0)(\boldsymbol{u}-\boldsymbol{u}_0)\end{aligned} \tag{2.13}$$

The second term of the Equation 2.13 contains partial derivative with respect to the state vector. Considering very small time increments of the controller loop and applying the *principle of time scale separation* the second term vanishes. This is a valid assumption if the dynamics of the actuators are fast compared with the dynamics of the system [5]. Thus, Equation 2.13 is further simplified to

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_0 + \boldsymbol{G}(\boldsymbol{x}_0)(\boldsymbol{u}-\boldsymbol{u}_0) \tag{2.14}$$

By solving Equation 2.14 for $\boldsymbol{u}$ and substituting $\dot{\boldsymbol{y}}$ with $\boldsymbol{\nu}$, the INDI control law is obtained

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{G}(\boldsymbol{x}_0)^{-1}(\boldsymbol{\nu} - \dot{\boldsymbol{x}}_0) \tag{2.15}$$

where $\dot{\boldsymbol{x}}_0$ is a measurable value from the previous step, $\boldsymbol{u}_{\boldsymbol{0}}$ the control input from the previous step, $\boldsymbol{\nu}$ the reference value and $\boldsymbol{G}(\boldsymbol{x}_0)$ the control effectiveness matrix. With $\boldsymbol{\Delta u} = \boldsymbol{u} - \boldsymbol{u}_0$ the control law from Equation 2.15 represents an incremental version of Equation 2.8. Instead of computing the complete control input command $\boldsymbol{u}$, this control law results in computing the increment of the control input $\boldsymbol{\Delta u}$ and adding it to the previous value $\boldsymbol{u}_0$. As it is less dependent on the model of the system dynamics, it is able to increase the robustness of the system [4].

2.3.2 Inner loop INDI for a Quadrotor

In this subsection, using theory from 2.3.1 the inner loop INDI controller is derived to control angular acceleration of the Crazyflie quadrotor. The derivation of the inner loop, as well as the outer loop controllers is based on the INDI controller architecture introduced by Smeur et al. [6], [7]. Equation 2.2 from section 2.1 serves as a basis for this derivation. The desired variable to be controlled by the inner loop INDI is the angular acceleration of the quadrotor in the body-fixed coordinate frame. As in the case of a real quadrotor control problem the value of the thrust can be seen as an output of the dynamic system, it makes sense to incorporate thrust as a control variable into the control law as well. Thus, the angular momentum Equation 2.2 is augmented with the total thrust T of all four rotors [7]. As only the body-fixed frame is used in the following control law derivation, the subscripts B are not used in this subsection. Furthermore, all external forces and moments apply to the center of gravity of the quadrotor, thus the superscripts G are omitted. Solving Equation 2.2 for the angular acceleration results in

$$\begin{bmatrix} \dot{\boldsymbol{\omega}} \\ T \end{bmatrix} = \underbrace{\begin{bmatrix} -\boldsymbol{I}^{-1}(\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}) \\ 0 \end{bmatrix}}_{\boldsymbol{F}(\boldsymbol{\omega})} + \underbrace{\begin{bmatrix} \boldsymbol{I}^{-1}(\boldsymbol{M}_G + \boldsymbol{M}_A + \boldsymbol{M}_P) \\ T \end{bmatrix}}_{\boldsymbol{G}(\boldsymbol{\omega}, \boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}})} \tag{2.16}$$

where $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are angular velocity and acceleration of the body-fixed frame (B) with respect to the NED (O) coordinate frame, $\boldsymbol{M}_G$ the gravitational moment, $\boldsymbol{M}_A$ the aerodynamic moment and $\boldsymbol{M}_P$ the propulsion moment. The vector $\boldsymbol{\Omega}$ contains angular velocities of all four rotors and serves as an input variable of the system. It is assumed that the gravitational force is applied to the center of gravity of the quadrotor and does not cause any moment around it. Due to the absence of the aerodynamic moment, this term is also omitted and can be seen as a disturbance [6]. The remaining propulsion moment is written as $\boldsymbol{M}_P = \boldsymbol{M}_C - \boldsymbol{M}_{gyro}$ where $\boldsymbol{M}_C$ is the control moment generated by the rotors and $\boldsymbol{M}_{gyro}$ the moment containing the gyroscopic effect of the rotors. This two moments can be explicitly written as

$$\boldsymbol{M}_C = \begin{bmatrix} -bk_F & bk_F & bk_F & -bk_F \\ lk_F & lk_F & -lk_F & -lk_F \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \boldsymbol{\Omega} \tag{2.17}$$

$$\boldsymbol{M}_{gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \end{bmatrix} \dot{\boldsymbol{\Omega}} + \begin{bmatrix} \boldsymbol{\omega}_y & 0 & 0 \\ 0 & \boldsymbol{\omega}_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \\ -I_{rzz} & I_{rzz} & -I_{rzz} & I_{rzz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega} \tag{2.18}$$

where I_{rzz} is an element of the inertia matrix I_r of the rotor, l and b lever arms as denoted in the Figure 2-1, k_F and k_M force and moment constants of the rotors.

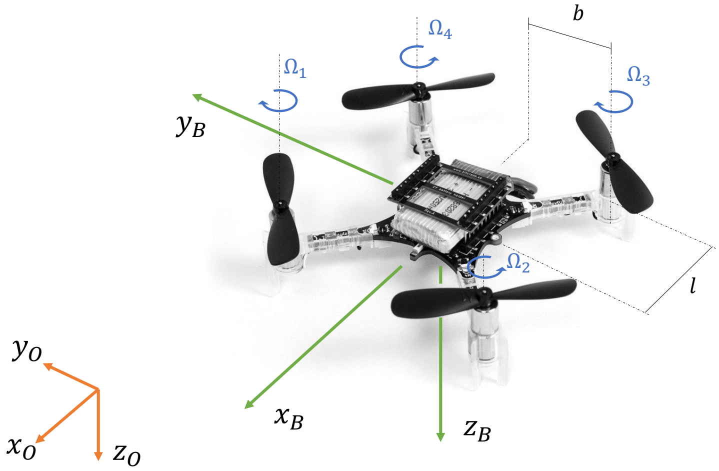

Figure 2-1: Image of the Crazyflie 2.1 quadrotor adopted from [8] with indicated lever arms of the motors.

As already explained in the previous subsection, to derive the incremental control law the Taylor series expansion is perfromed on Equation 2.16

$$\begin{aligned}\begin{bmatrix}\dot{\boldsymbol{\omega}}\\ T\end{bmatrix} &= \boldsymbol{F}(\boldsymbol{\omega}_0) + \boldsymbol{G}(\boldsymbol{\omega}_0, \boldsymbol{\Omega}_0, \dot{\boldsymbol{\Omega}}_0)\\ &+ \frac{\partial}{\partial \boldsymbol{\omega}}\left[\boldsymbol{F}(\boldsymbol{\omega}) + \boldsymbol{G}(\boldsymbol{\omega}_0, \boldsymbol{\Omega}_0, \dot{\boldsymbol{\Omega}}_0)\right]\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}(\boldsymbol{\omega} - \boldsymbol{\omega}_0)\\ &+ \frac{\partial}{\partial \boldsymbol{\Omega}}\left[\boldsymbol{G}(\boldsymbol{\omega}_0, \boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}}_0)\right]\Big|_{\boldsymbol{\Omega}=\boldsymbol{\Omega}_0}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0)\\ &+ \frac{\partial}{\partial \dot{\boldsymbol{\Omega}}}\left[\boldsymbol{G}(\boldsymbol{\omega}_0, \boldsymbol{\Omega}_0, \dot{\boldsymbol{\Omega}})\right]\Big|_{\dot{\boldsymbol{\Omega}}=\dot{\boldsymbol{\Omega}}_0}(\dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_0)\end{aligned} \tag{2.19}$$

By applying differentiation and rewriting some of the terms, the following equation is obtained:

$$\begin{bmatrix}\dot{\boldsymbol{\omega}}\\ T\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{\omega}}_0\\ T_0\end{bmatrix} + \boldsymbol{G}_1(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0) + T_s\boldsymbol{G}_2(\dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_0) \tag{2.20}$$

The first term of the Equation 2.20 is the angular acceleration based on the current angular rates ω_0 and inputs $\boldsymbol{\Omega}_0$. T_0 is the current thrust value. The last term on the right side of the Equation 2.20 is scaled with the sample time T_s which is introduced only to simplify further mathematical transformations. The expressions of the control moment $\boldsymbol{M}_C$ and the gyroscopic moment of the rotors $\boldsymbol{M}_{gyro}$ have been summarized to control effectiveness matrices $\boldsymbol{G}_1$ and

$\boldsymbol{G}_2$ which are defined as

$$\boldsymbol{G}_1(\boldsymbol{\omega}) = \begin{bmatrix} \boldsymbol{I}^{-1} \begin{bmatrix} -bk_F & bk_F & bk_F & -bk_F \\ lk_F & lk_F & -lk_F & -lk_F \\ k_M & -k_M & k_M & -k_M \end{bmatrix} - \boldsymbol{I}^{-1} \begin{bmatrix} \omega_y & 0 & 0 \\ 0 & \omega_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \\ -I_{rzz} & I_{rzz} & -I_{rzz} & I_{rzz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ k_F \cdot \mathbf{1}_{1\times4} \end{bmatrix} \tag{2.21}$$

$$\boldsymbol{G}_2 = \begin{bmatrix} -\boldsymbol{I}^{-1}T_s^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \end{bmatrix} \\ \mathbf{0}_{1\times4} \end{bmatrix} \tag{2.22}$$

with the terms $\mathbf{1}_{1\times4}$ and $\mathbf{0}_{1\times4}$ being 1×4 vectors of ones and zeros, respectively.

To prepare the linearized dynamic Equation 2.20 for discrete implementation on a computing system, the discrete approximation (z domain) of the derivative is used: $\dot{\boldsymbol{\Omega}} = (\boldsymbol{\Omega} - \boldsymbol{\Omega}z^{-1})T_s^{-1}$. Furthermore, angular acceleration $\dot{\boldsymbol{\omega}}_0$ which is obtained from the differentiated gyroscope measurements is usually noisy. Using a second order filtering can help to reduce measurement nose. At the same time such a filter introduces time delay, which has to be considered in the derivation, as it is important to have a unique delay for all variables which are used in the Taylor expansion [6]. Thus, the same second order filter $H(z)$ is applied to all variables with a subscript 0. By applying the filtering (subscript f) and the finite differences method to the Equation 2.20, its discrete version results in

$$\begin{bmatrix} \dot{\boldsymbol{\omega}} \\ T \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_f \\ T_f \end{bmatrix} + (\boldsymbol{G}_1 + \boldsymbol{G}_2)(\boldsymbol{\Omega} - \boldsymbol{\Omega}_f) - \boldsymbol{G}_2 z^{-1}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_f) \tag{2.23}$$

By solving Equation 2.23 for $\boldsymbol{\Omega}$ and substituting $\dot{\boldsymbol{\omega}}$ with the virtual control input $\boldsymbol{\nu}_{\dot{\omega}}$, the control law of the inner loop is obtained

$$\boldsymbol{\Omega}_c = \boldsymbol{\Omega}_f + (\boldsymbol{G}_1 + \boldsymbol{G}_2)^{-1} \left(\begin{bmatrix} \boldsymbol{\nu}_{\dot{\omega}} - \dot{\boldsymbol{\omega}}_f \\ \tilde{T} \end{bmatrix} + \boldsymbol{G}_2 z^{-1}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_f) \right) \tag{2.24}$$

where $\boldsymbol{\Omega}_c$ is a vector of commanded rotational rates for every rotor and $\tilde{T} = T - T_f$ and $\tilde{\boldsymbol{\Omega}} = \boldsymbol{\Omega}_c - \boldsymbol{\Omega}_f$ being thrust and propeller rates increments respectively, provided by the inner loop INDI. The block diagram of the inner loop INDI controller is presented in Figure 2-2.

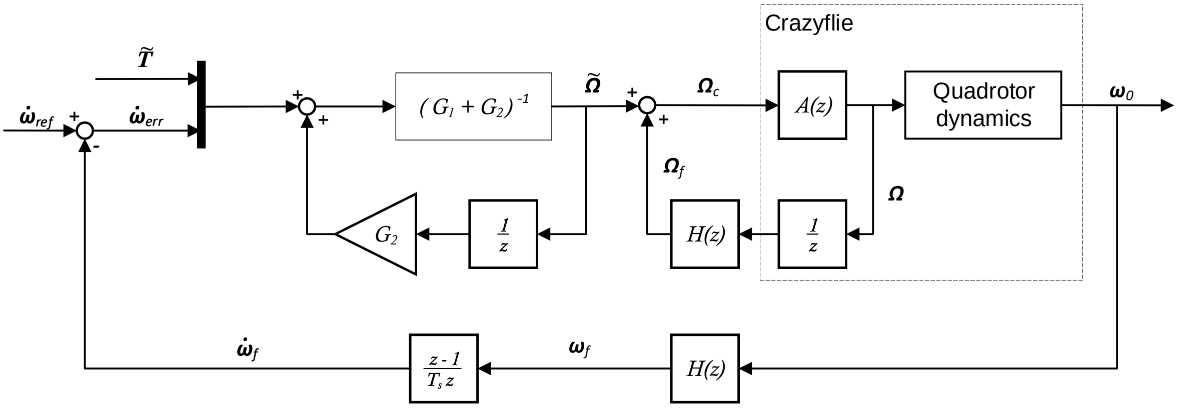

Figure 2-2: Block diagram of the inner loop INDI controller. $A(z)$ is the actuator dynamics (see 3.2.2) and $H(z)$ is the second-order filter (see 3.2.3).

The inner INDI controller which controls the angular acceleration is then augmented with two simple controllers which control the angular velocity ω and the attitude η of the quadrotor (see Figure 2-3). Each of them consists of a single gain. Thus, the reference values of the INDI part of the inner loop are provided by these two controllers.

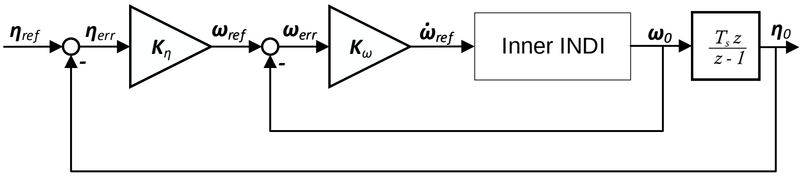

Figure 2-3: Attitude controller as an augmentation of the inner loop INDI controller.

2.3.3 Outer loop INDI for a Quadrotor

In this subsection the derivation of the outer loop INDI controller is introduced. The outer loop INDI controls translational acceleration of the quadrotor. Linear momentum Equation 2.1 serves as a basis for this derivation. For the simplicity of the further transformations the derivation is performed in the NED frame (subscript O). Thus, the Equation 2.1 becomes

$$\sum(\boldsymbol{F}^G)_O = m \cdot (\dot{\boldsymbol{v}}^G)_O \tag{2.25}$$

As it is assumed that all forces apply to the center of gravity of the quadrotor, the superscript G is omitted in the future. Thus, solving Equation 2.25 for $\dot{\boldsymbol{v}}$ and assuming that the sum of all forces acting on the quadrotor consists only of gravitational, propulsive and aerodynamic forces, the following equation is obtained

$$\dot{\boldsymbol{v}} = m^{-1}((\boldsymbol{F}_G)_O + (\boldsymbol{F}_P)_O + (\boldsymbol{F}_A)_O) \tag{2.26}$$

The aerodynamic force $(\boldsymbol{F}_A)_O$ is modelled as an unknown function of the velocity $\boldsymbol{v}$ and wind vector $\boldsymbol{\chi}$. For the gravitational force $(\boldsymbol{F}_G)_O$ the simplest one-dimensional gravity model is assumed.

$$(\boldsymbol{F}_A)_O = \boldsymbol{f}(\boldsymbol{v}, \boldsymbol{\chi}) \tag{2.27}$$

$$(\boldsymbol{F}_G)_O = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}_O \tag{2.28}$$

The propulsive force $(\boldsymbol{F}_P)_O$ (produced by the rotors) has only one nonzero component if defined in the body-fixed frame. To transform it from the body-fixed to the NED frame the rotation matrix $\boldsymbol{M}_{OB}$ is used. Here, the transformation from the B-frame to the O-frame is defined by the following rotation sequence: Φ, Θ, Ψ. Depending on the rotation sequence, a different orientation coordinate causes gimbal lock. The introduced rotation sequence is suitable for horizontally oriented aircrafts as the gimbal lock is caused for $\Theta = \pm 90°$. This results in the following expression for the propulsive force

$$\begin{aligned}
(\boldsymbol{F}_P)_O &= \boldsymbol{M}_{OB} \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix}_B \\
&= \begin{bmatrix} c\Theta c\Psi & c\Psi s\Theta s\Phi - s\Psi c\Phi & c\Psi s\Theta c\Phi + s\Psi s\Phi \\ c\Theta s\Psi & s\Psi s\Theta s\Phi + c\Psi c\Phi & s\Psi s\Theta c\Phi - c\Psi s\Phi \\ -s\Theta & c\Theta s\Phi & c\Theta c\Phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix}_B \\
&= \begin{bmatrix} (c\Psi s\Theta c\Phi + s\Psi s\Phi)(-T) \\ (s\Psi s\Theta c\Phi - c\Psi s\Phi)(-T) \\ (c\Theta c\Phi)(-T) \end{bmatrix}_O
\end{aligned} \tag{2.29}$$

where cx and sx denote $\cos(x)$ and $\sin(x)$ respectively.

To obtain the incremental form of the linear momentum equation, the same strategy as in the case of the inner loop derivation is applied: the Equation 2.26 is linearized using Taylor series expansion and neglecting higher order terms

$$\begin{aligned}
\dot{\boldsymbol{v}} = m^{-1}&[\boldsymbol{F}_G + \boldsymbol{F}_P(\Phi_0, \Theta_0, \Psi_0, T_0) + \boldsymbol{F}_A(\boldsymbol{v}_0, \boldsymbol{\chi}_0)] \\
&+ m^{-1} \frac{\partial}{\partial \boldsymbol{v}} [\boldsymbol{F}_A(\boldsymbol{v}, \boldsymbol{\chi}_0)] \bigg|_{\boldsymbol{v}=\boldsymbol{v}_0} (\boldsymbol{v} - \boldsymbol{v}_0) \\
&+ m^{-1} \frac{\partial}{\partial \boldsymbol{\chi}} [\boldsymbol{F}_A(\boldsymbol{v}_0, \boldsymbol{\chi})] \bigg|_{\boldsymbol{\chi}=\boldsymbol{\chi}_0} (\boldsymbol{\chi} - \boldsymbol{\chi}_0) \\
&+ m^{-1} \frac{\partial}{\partial \Phi} [\boldsymbol{F}_P(\Phi, \Theta_0, \Psi_0, T_0)] \bigg|_{\Phi=\Phi_0} (\Phi - \Phi_0) \\
&+ m^{-1} \frac{\partial}{\partial \Theta} [\boldsymbol{F}_P(\Phi_0, \Theta, \Psi_0, T_0)] \bigg|_{\Theta=\Theta_0} (\Theta - \Theta_0) \\
&+ m^{-1} \frac{\partial}{\partial \Psi} [\boldsymbol{F}_P(\Phi_0, \Theta_0, \Psi, T_0)] \bigg|_{\Psi=\Psi_0} (\Psi - \Psi_0) \\
&+ m^{-1} \frac{\partial}{\partial T} [\boldsymbol{F}_P(\Phi_0, \Theta_0, \Psi_0, T)] \bigg|_{T=T_0} (T - T_0)
\end{aligned} \tag{2.30}$$

The first term on the right side of the Equation 2.30 is the acceleration at the previous time step $\dot{\boldsymbol{v}}_0$. It can be obtained from the onboard accelerometer and then tranformed to the NED frame. The next two terms represent partial derivatives of the aerodynamic forces. Due to the absence of an accurate model of the aerodynamic effects, these terms are omitted and considered as disturbances. It is assumed that the change of the yaw angle Ψ is small and can be neglected as well. By applying considered assumptions Equation 2.30 is written as

$$\dot{\boldsymbol{v}} = \dot{\boldsymbol{v}}_0 + m^{-1}\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0)(\boldsymbol{u} - \boldsymbol{u}_0) \tag{2.31}$$

where

$$\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0) = \begin{bmatrix} (-c\Psi_0 s\Theta_0 s\Phi_0 + s\Psi_0 c\Phi_0)T_0 & c\Psi_0 c\Theta_0 c\Phi_0 T_0 & c\Psi_0 s\Theta_0 c\Phi_0 + s\Psi_0 s\Phi_0 \\ (-s\Psi_0 s\Theta_0 s\Phi_0 - c\Psi_0 c\Phi_0)T_0 & s\Psi_0 c\Theta_0 c\Phi_0 T_0 & s\Psi_0 s\Theta_0 c\Phi_0 - c\Psi_0 s\Phi_0 \\ -c\Theta_0 s\Phi_0 T_0 & -s\Theta_0 c\Phi_0 T_0 & c\Theta_0 c\Phi_0 \end{bmatrix} \tag{2.32}$$

and

$$\boldsymbol{u} = \begin{bmatrix} \Phi \\ \Theta \\ T \end{bmatrix} \tag{2.33}$$

$\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0)$ is the control effectiveness matrix which is computed based on the attitude and thrust values from the previous step. $\boldsymbol{u}$ is a vector of control variables which are passed to the inner loop.

The linear acceleration $\dot{\boldsymbol{v}}_0$ which is obtained from accelerometer measurements is usually noisy. By applying the same filter $H(z)$ as in the case of the inner loop derivation, the Equation 2.31 is given by:

$$\dot{\boldsymbol{v}} = \dot{\boldsymbol{v}}_f + m^{-1}\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0)(\boldsymbol{u} - \boldsymbol{u}_f) \tag{2.34}$$

Solving Equation 2.34 for $\boldsymbol{u}$ and substituting $\dot{\boldsymbol{v}}$ with the virtual control variable $\boldsymbol{\nu}_{\dot{\boldsymbol{v}}}$, the control law of the outer loop is obtained

$$\boldsymbol{u_c} = \boldsymbol{u}_f + m\boldsymbol{G}^{-1}(\Phi_0, \Theta_0, \Psi_0, T_0)(\boldsymbol{\nu}_{\dot{\boldsymbol{v}}} - \dot{\boldsymbol{v}}_f) \tag{2.35}$$

Figure 2-4 shows the block diagram of the outer loop INDI. In Figure 2-4 $\tilde{\Phi}$ and $\tilde{\Theta}$ are increments of the roll and pitch angles which are added to the current roll and pitch angle values and then passed to the inner loop as reference values. The increments are defined as

$$\tilde{\Phi} = \Phi_c - \Phi_f \tag{2.36a}$$
$$\tilde{\Theta} = \Theta_c - \Theta_f \tag{2.36b}$$

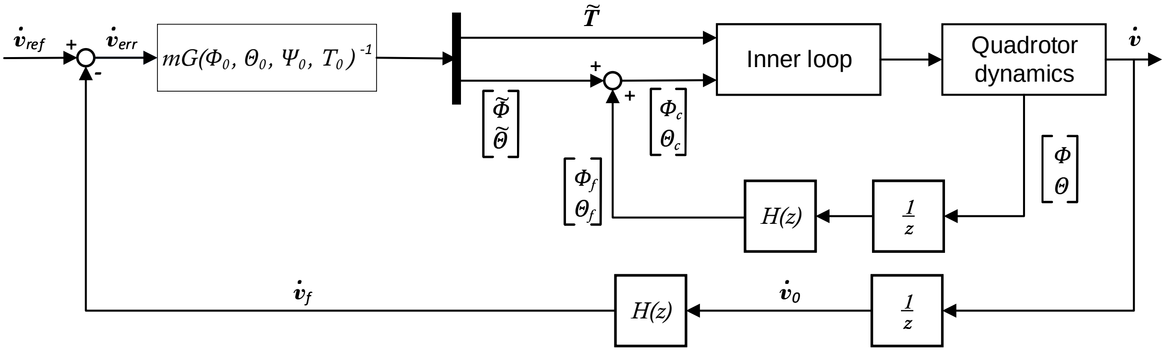
$\dot{v}_{ref}$
+
-
$\dot{v}_{err}$
$mG(\Phi_0, \Theta_0, \Psi_0, T_0)^{-1}$
$\tilde{T}$
$\begin{bmatrix}\tilde{\Phi}\\ \tilde{\Theta}\end{bmatrix}$
+
+
$\begin{bmatrix}\Phi_c\\ \Theta_c\end{bmatrix}$
$\begin{bmatrix}\Phi_f\\ \Theta_f\end{bmatrix}$
Inner loop
Quadrotor dynamics
$\dot{v}$
$\begin{bmatrix}\Phi\\ \Theta\end{bmatrix}$
H(z)
$\frac{1}{z}$
$\dot{v}_f$
H(z)
$\dot{v}_0$
$\frac{1}{z}$

Figure 2-4: Block diagram of the outer loop INDI controller.

Using the same approach as with the inner loop, the outer loop which controls linear acceleration is augmented with two gains (see Figure 2-5). These gains represent two controllers which control linear velocity v and position ξ.

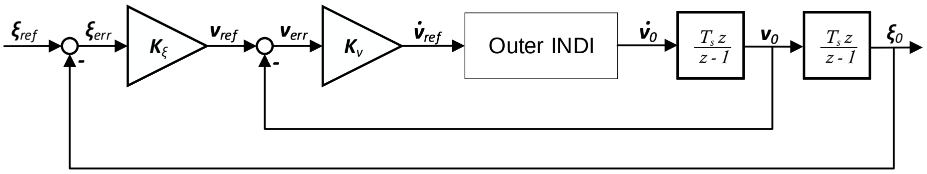

Figure 2-5: Position controller as an augmentation of the outer loop INDI controller.

3 Implementation

This chapter describes different aspects of the practical realization of the control architecture introduced in the previous chapter. Section 3.1 gives an overview about the quadrotor system on which the INDI outer loop controller was implemented and tested. In section 3.2 the simulation model of the Crazyflie quadrotor with implemented INDI inner and outer loops is presented. Finally, in section 3.3 the process of the controller implementation on the hardware system is described. Here, additionally to the complete outer loop implementation, it is described how the already existing inner loop was modified to achieve a better performance.

3.1 Research Quadrotor

With its arm length of approx. $4\ \text{mm}$, the Crazyflie 2.1 quadrotor developed by Swedish company Bitcraze belongs to the Micro Air Vehicle (MAV) class (see Figure 3-1) [9]. Its total weight is approx. $27\ \text{g}$ with a capability to increase it up to $42\ \text{g}$, which makes it possible to extend the default hardware with a pair of additional extension decks of various functionality. As it is an open source project, the main firmware, development tools and other software packages are freely available. This is one of the reasons why this small quadrotor gained a lot of popularity in different academia fields, being used for tasks reaching from the classical control theory all the way down to the obstacle avoidance problems [10].

The Crazyflie's main application Microcontroller (MCU) is STM32F405. It runs the firmware with frequency of $1000\ \text{Hz}$ and communicates with a large number of peripheral devices such as Bluetooth and long range radio, power management system MCU, Inertial Measurement Unit (IMU) etc.. The Inertial Measurement Unit consists of the 3-axis accelerometer, the 3-axis gyroscope and the high precision pressure sensor.

The quadrotor used in this project was extended with a flow deck sensor. The flow deck itself is equipped with two sensors: the optical flow sensor which measures the visual motion and the Time-of-Flight (ToF) laser-ranging sensor which measures the range to the next object along the body-fixed z-axis of the quadrotor. By employing these sensors, the flow deck can provide estimates of the current position and linear velocity of the aircraft.

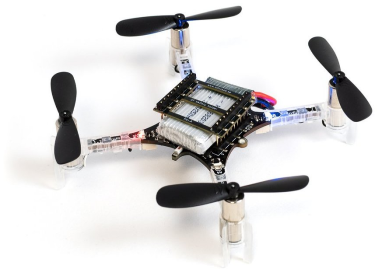

Figure 3-1: Crazyflie 2.1 nano-quadrotor [8].

3.2 Simulink Model

Before implementing the outer loop of the INDI controller derived in 2.3.3 on the real Crazyflie quadrotor it was decided firstly to build a simulation model using the Matlab/Simulink software. Such a simulation has a number of advantages compared to the direct implementation on the hardware. The main one is, it allows one to conveniently test different model components (e.g. filtering algorithms, actuator dynamics models etc.) on their dedicated functions, facilitating the debugging process. It also can ease the process of parameter tuning of the controller.

In the framework of this thesis the Simulink model was also used to better understand the controller structure and its characteristics. Based on the simulation it was determined which of the control effectiveness terms have a greater influence on the system response and which ones can be neglected. Additionally, the simulation model provided a good overview of potential computational problems and mistakes which could lead to drastically growing values and unstable behaviour of the system. To solve some of them, saturation blocks have been used in different places.

3.2.1 Quadrotor Dynamics

The core of the Crazyflie simulation is the model of the quadrotor dynamics. It is based on dynamic equations of motion presented in Chapter 2.1. Similarly to section 2.3.3, it was assumed that the sum of the forces in the linear momentum equation is composed of three components, namely gravitational, propulsive and aerodynamic. The expressions of the gravity and propulsive forces have not been changed, so the Equations 2.28 and 2.29 were adopted for the simulation model. The thrust of the quadrotor was modelled as a linear function of propeller rates $\boldsymbol{\Omega}$. Instead of using quadratic relationship between the thrust and propeller rates (which can be often found in the literature), the choice fell on the linear relationship to avoid mixing of linear relations with quadratic ones. This would make the control allocation problem more difficult, since the term which contains the effect of the propeller inertia $\boldsymbol{G_2}$ is linear. Thus, the thrust T is given as:

$$T = -k_F(\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4) \tag{3.1}$$

where k_F is the force constant of the rotors. In subsection 3.3.3 it is shown how to estimate the thrust coefficient directly from the flight data. However, for the simulation model k_F was approximated based on the assumption that the influence of the force constant of the rotors k_F (considering the lever arm) on the total moment is greater than the one of the moment constant of the rotors k_M, and that the *thrust to weight ratio* expression is $\frac{T}{W} \approx 2$ (as the real Crazyflie quadrotor is able to hover at half throttle).

To account for aerodynamic effects such as drag the aerodynamic force was approximated with the following expression

$$(\boldsymbol{F}_A)_B = \begin{bmatrix} -k_x(u_K)_B \\ -k_y(v_K)_B \\ -k_z(w_K)_B \end{bmatrix} \tag{3.2}$$

where $(u_K)_B$, $(v_K)_B$ and $(w_K)_B$ are kinematic velocities of the quadrotor along the axes x_B, y_B and z_B respectively, given in the body-fixed coordinate frame. k_x, k_y and k_z are experimentally estimated drag coefficients adopted from [11].

The angular momentum equation used in the simulation model has the same form as defined in Equation 2.2. The numerical values of the inertia tensor $(\boldsymbol{I}^G)_B$ were estimated using the pendulum method in [11]. The total external moment acting on the Crazyflie contains only the propulsive moment $\boldsymbol{M}_P$ as it is defined in 2.3.2. It consist of the control moment generated by the rotors $\boldsymbol{M}_C$ and moment containing the gyroscopic effect of the rotors $\boldsymbol{M}_{gyro}$ (see Equations 2.17 and 2.18). The moment constant of the rotor was computed to fulfil the requirements on k_F and k_M which are described above. To obtain the numerical estimate of the propeller inertia matrix I_{rzz}, the propeller was approximated with a bar element. This has been shown (see 3.2.4) to be a valid assumption since the influence of the propeller inertia (due to its negligible weight) is very small.

3.2.2 Actuator Dynamics

The inner loop INDI controller relies on the measurement of the rotor angular rates. If the actuator feedback is not available, it is possible to use its model instead. The model of the actuator dynamics was assumed to be a first order lag filter with the following transfer function defined in a continuous Laplace domain

$$A(s) = \frac{k}{1+Ts} \tag{3.3}$$

where k is the gain and T the time constant. Section 3.3 shortly describes the process of using flight data to estimate the time constant T. In the Simulink model it was assumed that the actuator feedback is available, hovewer for the implementation on hardware it is necessary to have a discrete version of the transfer function in Equation 3.3 (see subsection 3.3.2 for details). Transforming it from continuous domain into discrete by applying *Zero-Order Hold method* results in

$$A(z) = \frac{k\alpha}{z+(\alpha-1)} \tag{3.4}$$

with $\alpha = 1 - e^{\frac{-T_s}{T}}$ and sample time T_s. The default frequency with which the Crazyflie controller updates its state is 500 Hz. Thus, sampling time T_s was set to the reciprocal value of the default frequency.

3.2.3 Filtering

As it was shown in Chapter 2 (see Figures 2-2 and 2-4), the inner and outer loops employ several second order filters. The main task of two of them is to reduce noise in the measurements of the angular and linear acceleration. Other filters are applied to remaining feedback quantities. By using the same parameters for every filter, this technique accounts for the time delay which is introduced by the filters and ensures that all quantities that were fed back are from the same point in time. The general transfer function of a second order filter in continuous domain is defined by the following formula

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.5}$$

with the relative damping coefficient ζ and the eigenfrequency ω_n. To implement transfer function from Equation 3.6 on a digital computer it must be transformed to the discrete domain. To perform this transformation *bilinear transformation* (also called *Tustin-Transformation*) was used. Using the bilinear transformation (with frequency warping technique), the mapping relation between s and z results in

$$s \mathrel{\widehat{=}} \frac{\omega_n}{\tan(\omega_n \frac{T_s}{2})} \frac{1 - z^{-1}}{1 + z^{-1}} \tag{3.6}$$

Thus, the discrete version of the Equation 3.6 is:

$$H(z) = \frac{n_0 + n_1 z^{-1} + n_2 z^{-2}}{d_0 + d_1 z^{-1} + d_2 z^{-2}} \tag{3.7}$$

where

$$n_0 = 1 \tag{3.8a}$$

$$n_1 = 2 \tag{3.8b}$$

$$n_2 = 1 \tag{3.8c}$$

$$d_0 = \frac{1}{\tan^2(\omega_n \frac{T_s}{2})} + \frac{2\zeta}{\tan(\omega_n \frac{T_s}{2})} + 1 \tag{3.8d}$$

$$d_1 = -\frac{2}{\tan^2(\omega_n \frac{T_s}{2})} + \frac{4\zeta}{\tan(\omega_n \frac{T_s}{2})} \tag{3.8e}$$

$$d_2 = \frac{1}{\tan^2(\omega_n \frac{T_s}{2})} - \frac{2\zeta}{\tan(\omega_n \frac{T_s}{2})} + 1 \tag{3.8f}$$

3.2.4 Simulation Results

This subsection shortly presents simulation results of the closed loop INDI controller (consisting of the inner and outer loops). Both control loops were implemented (as depicted in Figures 2-2 and 2-4) in discrete domain, the model of the Cryzyflie dynamics was implemented in continuous domain. This is usually done, because a real physical system such as a quadrotor operates in continuous time, however a control algorithm representing a cyber component always runs on a digital system with a discrete sampling rate.

The input of the overall system is a position vector. Thus, the closed loop model was tested by applying a sequence of position steps of different intervals. Figure 3-2 shows three plots containing responses of the outer loop controllers for the x-component of the position, linear velocity and linear acceleration states.

The input of the inner loop is the attitude, which is computed and provided by the outer loop. Figure 3-3 shows the response of inner loop (for the same position step sequence as above) which controls the attitude, the angular velocity and the angular acceleration of the quadrotor.

As mentioned in the beginning of this chapter, a simulation can be used to determine terms of the model which have a greater influence on the overall controller performance. Matrix $\boldsymbol{G}_2$ of the control effectiveness term in Equation 2.24 contains only the term of the inertia tensor of the propeller I_{rzz}. Since I_{rzz} is small, it was expected that G_2 will not have a big impact on the model response. Figure 3-4 confirms this expectation, by plotting the step response of the Crazyflie model with and without considering the gyroscopic effects of the quadrotor propellers.

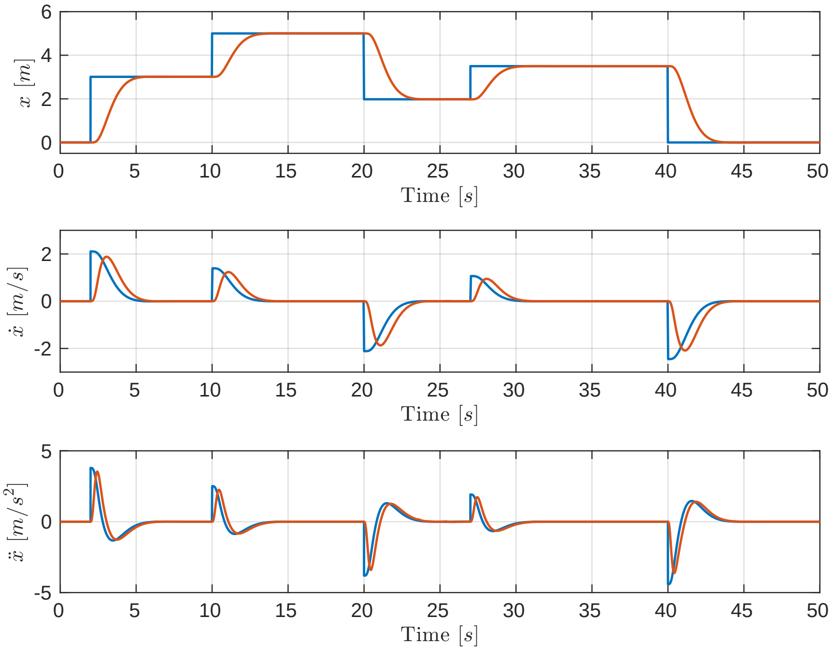

Figure 3-2: Response of the outer loop simulation model to a step input. Blue lines represent reference values, red lines denote measured values.

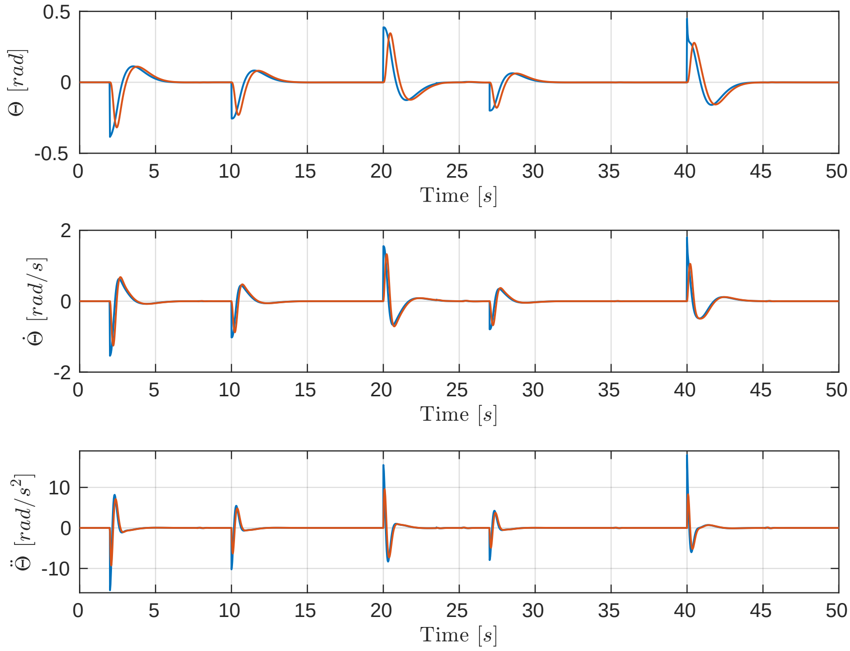

Figure 3-3: Response of the inner loop simulation model to the input provided by the outer loop. Blue lines represent reference values, red lines denote measured values.

It can be seen that both responses are almost the same. In the next section, (see 3.3.1) through the estimation of the control effectiveness terms using the flight data, this observation will be practically confirmed. In practice, considering only important terms, can reduce the implementation effort of a particular control algorithm.

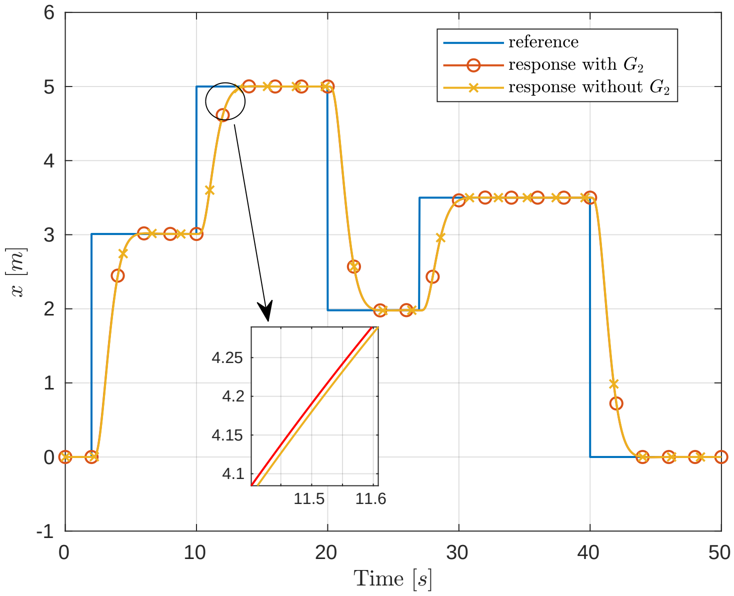

Figure 3-4: Responses of the closed loop system using simulation of the Crazyflie dynamics with and without considering the gyroscopic effects of the propellers.

3.3 Implementation on the Hardware

This section describes some important aspects of the controller implementation on the Crazyflie hardware. As it was mentioned at the beginning of this chapter, the inner loop INDI controller was already implemented on the provided hardware. Thus, this section is mainly dedicated to the implementation of the outer loop.

3.3.1 Parameter Estimation of the Inner Loop

To ensure proper functioning of the inner loop, some of the mistakes in the already existing implementation were corrected. Additionally, the control effectiveness parameters $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ were reestimated with a new flight data. For the matrix $\boldsymbol{G_1}$ it was assumed that it does not change over the operational domain and can be considered as static.

To estimate model parameters of a quadrotor two strategies can be applied: estimation through measurements of each parameter that is part of the control effectiveness terms and estimation with flight data. Using the first strategy, requires additional measuring equipment to achieve accurate results. By contrast, having access only to IMU data and pursuing the second approach, can reduce the amount of effort and also lead to plausible results [12]. To collect the flight data (gyroscope measurements $\boldsymbol{\omega}$ and actuator inputs $\boldsymbol{\Omega}$), the only requirement is that the quadrotor has a functioning controller (e.g. PID controller) to perform flights. After the first estimates of the control effectiveness parameters are obtained, further flight can be performed with the INDI controller.

Equation 2.23 serves as a basis for the parameter estimation. Since the existing implementation of the inner loop does not include the increment term of the thrust, Equation 2.23 reduces to

$$\dot{\boldsymbol{\omega}}_f = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}_{f,a} \\ \dot{\boldsymbol{\Omega}}_{f,a} \end{bmatrix} \tag{3.9}$$

Note that the terms $\boldsymbol{\Omega}_{f,a}$ and $\dot{\boldsymbol{\Omega}}_{f,a}$ are passed through the actuator dynamics $A(s)$ (subscript “a”). Furthermore, the Equation 3.9 is differentiated to amplify high frequencies of the signal for a better parameter estimation performance. Differentiation works like a high-pass filter, passing through only fast changing portion of the signal (e.g. fast and sharp maneuvers). It also eliminates bias of the signal. Considering these changes, Equation 3.9 results in

$$\ddot{\boldsymbol{\omega}}_f = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\Omega}}_{f,a} \\ \ddot{\boldsymbol{\Omega}}_{f,a} \end{bmatrix} \tag{3.10}$$

When the log data is available, the *least squares* approach can be applied to find solution of the Equation 3.10.

To confirm the observation that $\boldsymbol{G_2}$ term has a very small influence on the overall control effectiveness, made in 3.2.4, contributions of both terms ($\boldsymbol{G_1}$ and $\boldsymbol{G_2}$) were compared. Figure 3-5 shows the contribution of the estimated $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ terms to the yaw component of the differentiated angular acceleration. Clearly, the contribution of $\Delta\ddot{\Omega}_z \cdot \boldsymbol{G_2}$ is noticeably smaller than the one of $\Delta\dot{\Omega}_z \cdot \boldsymbol{G_1}$.

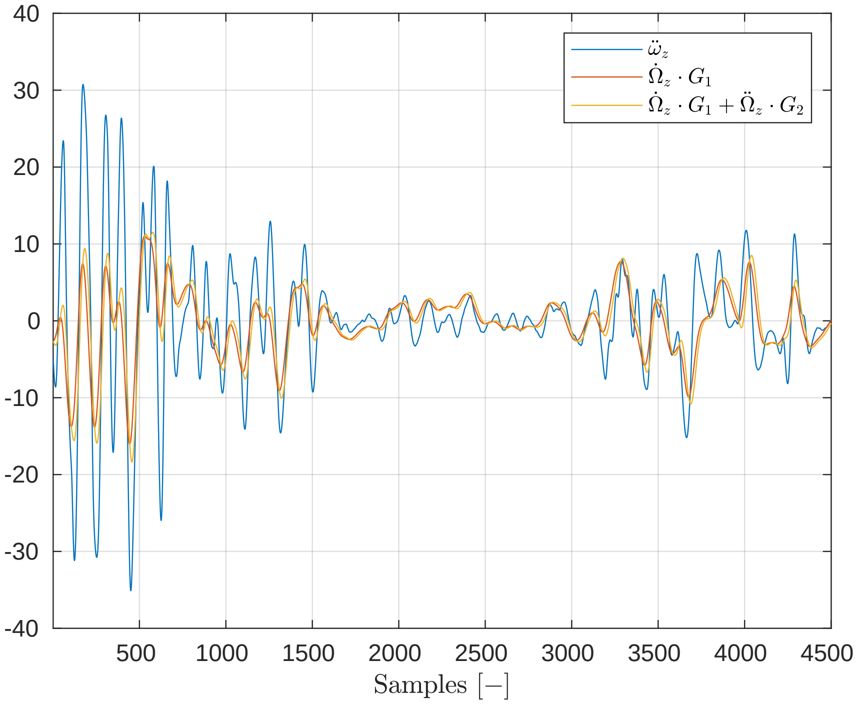

Figure 3-5: Contribution of the estimated G_1 and G_2 terms to the yaw component of the angular acceleration. ω_z and Ω_z were obtained from the flight data.

3.3.2 Estimation of the Actuator Dynamics Time Constant

The INDI controller derived in Chapter 2 controls the Crazyflie quadrotor by adding increments of the propeller rates to the current propeller rates. Unfortunately, the hardware of the quadrotor does not support the measurement of the propeller rates. Thus, to be able to feedback the propeller rates they are passed through the actuator dynamics model. The feedback of the propeller rates is then obtained as shown in Figure 3-6. Here, $A(z)$ is the real and unknown actuator dynamics and $A'(z)$ is the model of the actuator dynamics approximated with a first order lag element (as shown in 3.2.2).

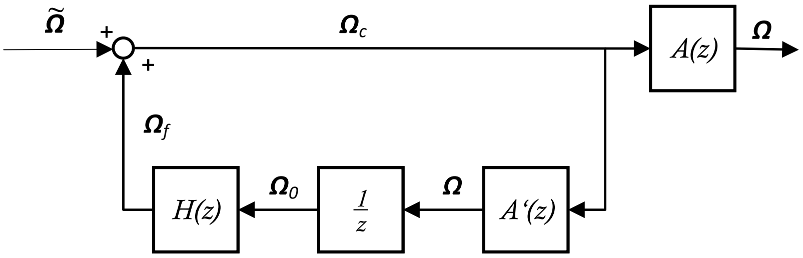

Figure 3-6: Estimation of the actuator state in case of the missing actuator feedback.

If the transfer function of the actuator $A'(z)$ is assumed to be of the first order, the only unknown parameter is the time constant T (as it is assumed that the commanded propeller rates are not amplified by the actuator, the gain k is simply 1). To estimate the time constant T, propeller rates were measured for different step inputs. Finally, the general form of the first order lag element was fitted to the measured data.

To perform the measurements a tachometer setup was used. The working principle of the setup is as follows: an Infrared (IR) transmitter emits light which is continuously detected by an IR receiver. The propeller of the quadrotor is then placed in such a way that when it rotates, every pass of its blades prevents the infrared light passing from the emitter to the receiver. This triggers an interrupt on the microcontroller which counts the passes of the propeller.

The transfer function of the first order lag element (Equation 3.3) in time domain has the following form

$$a(t) = k(1 - e^{\frac{-t}{T}}) \tag{3.11}$$

The fitting is then done by minimizing the squared error between measured $a_{measured}$ and predicted a data. For multiple measurements in discrete time steps the objective function $E(T)$ is defined as

$$E(T) = \sum_t (a(T) - a_{measured}(T))^2 \tag{3.12}$$

T is then computed using the least squares approach such that Equation 3.12 is minimal. Note that in Equation 3.12 $a(T)$ is a function of the time constant T, since it is an optimization parameter. Figure 3-7 shows the obtained result of the optimization. Here, the red curve was fitted to the measured data (blue line). In the figure the signal contains $n = 155$ samples recorded with a frequency of 200 Hz. The estimated value for the time constant is $T = 0.061658$ s.

A simpler alternative to estimate the time constant would be reading off the time point at which the response reaches approx. 63.2% of its final value. Using this approach, the value of the time constant is approximately 0.0675 s, which is very close to the value estimated through solving the optimization problem.

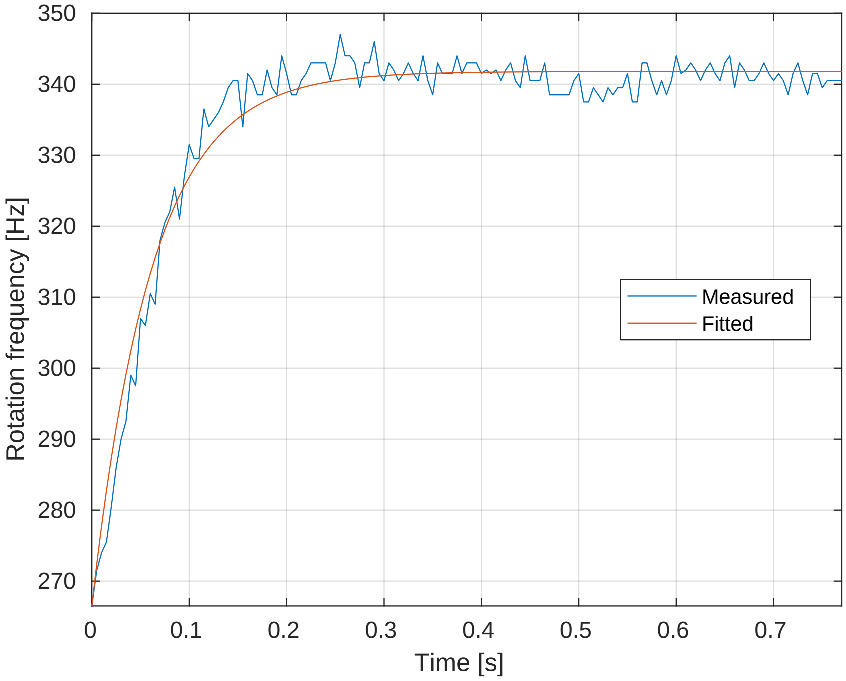

Figure 3-7: The blue line represents propeller rates obtained from a measurement. The red line is the fitted curve to the measured data.

3.3.3 Estimation of the Thrust Mapping

The thrust increments provided by the outer loop INDI are values in Newton. Later, in the inner loop INDI, commanded thrust values are computed by adding thrust increments to the current thrust values, the result is then passed to the motor mixing algorithm. Depending on the commanded roll, pitch and yaw values, the motor mixing algorithm computes values for each rotor and sends these commands to each motor. Unfortunately, the values which are sent to the motors are not in the same units as the thrust. These values are defined in a specific range (e.g motors of the Crazyflie accept integer numbers from 0 to 60000). Mapping a thrust value in physical units to a motor command value is called *thrust mapping* [13].

It was assumed that the relation between the physical thrust value T and the thrust value represented in motor units T_{cmd} is linear. Thus, this relation can be defined as

$$T = T_{cmd}K_{cmd} \tag{3.13}$$

with K_{cmd} being the *thrust mapping parameter*. This parameter can be estimated similarly to the control effectiveness parameters from the previous section, by using the flight data of a vertical flight. Next several paragraphs describe this approach in detail.

The third component of the Equation 2.35 serves as a basis for the estimation of the mapping parameter K_{cmd}. If a vertical flight is assumed, the angle states Φ and Θ have to be zero, which leads to the vanishing of the terms with x- and y-component of the linear acceleration $\dot{\boldsymbol{v}}$. Furthermore, instead of the thrust T_0 in the G matrix, the specific thrust $\frac{T_0}{m}$ is used. This transformation will simplify the expression for the thrust increment $\tilde{T}$, by eliminating the mass m from it. With the above assumptions Equation 2.35 simplifies to

$$\tilde{T} = \dot{v}_{err,z} \tag{3.14}$$

where $\dot{v}_{err,z}$ is the z-component of the linear acceleration error $\dot{v}_{err}$.

Replacing thrust with a thrust increment in Equation 3.13 and inserting it into Equation 3.14, results in

$$\tilde{T}_{cmd} = \dot{v}_{err,z} K_{cmd}^{-1} \tag{3.15}$$

By making use of the finite difference approximation, Equation 3.15 can be rewritten as

$$\dot{T}_{cmd} = \ddot{v}_z K_{cmd}^{-1} \tag{3.16}$$

where T_{cmd} is the commanded thrust, and $\dot{v}_z$ the linear acceleration along the z-axis. Both values can be obtained from the flight data. Similarly to subsection 3.3.1, differentiating will amplify the high-frequency portion of the signal and contribute to the elimination of the bias. Equation 3.16 can also be obtained by differentiating Equation 2.26 and applying the same assumptions to it which were used earlier in this subsection. Note that to obtain realistic trend of the commanded thrust values, just before the differentiation, these values have to be passed through the actuator dynamics model $A'(s)$. Equation 3.16 then has the following form

$$\dot{T}_{cmd,a} = \ddot{v}_z K_{cmd}^{-1} \tag{3.17}$$

where $T_{cmd,a}$ is the thrust value filtered with the actuator dynamics model. Additionally, $\dot{T}_{cmd,a}$ was filtered with a second order filter to remove the noise for better estimation results. The least squares approach is then applied to obtain K_{cmd}.

Figure 3-8 shows the results of the thrust mapping parameter estimation. The first subplot contains the trends of the quantities from Equation 3.17. The second subplot shows that the trend of the mapped commanded thrust $T_{cmd,a} \cdot K_{cmd}$ matches the course of the linear acceleration $\dot{v}_z$ for the estimated parameter K_{cmd}.

3.3.4 Implementation of the Outer Loop on the Crazyflie's Hardware

This subsection provides several notes regarding the implementation of the outer loop on the real quadrotor. As the already existing implementation of the inner loop on the quadrotor was not including thrust increments as control inputs, the interface between the outer and the inner loops had to be adjusted. Thus, at the place, where thrust increments $\tilde{T}$ are passed to the inner loop, a complete thrust value T_c ("c" stands for "commanded") has to be provided.

To obtain T_c, current thrust value T_0 is filtered with a second order filter $H(z)$ and passed through the model of the actuator dynamics $A'(z)$. To ensure a unique time delay in both loops, the filtering parameters ($\omega_n = 8$ rad/s, $\zeta = 0.707$) were adopted from the inner loop

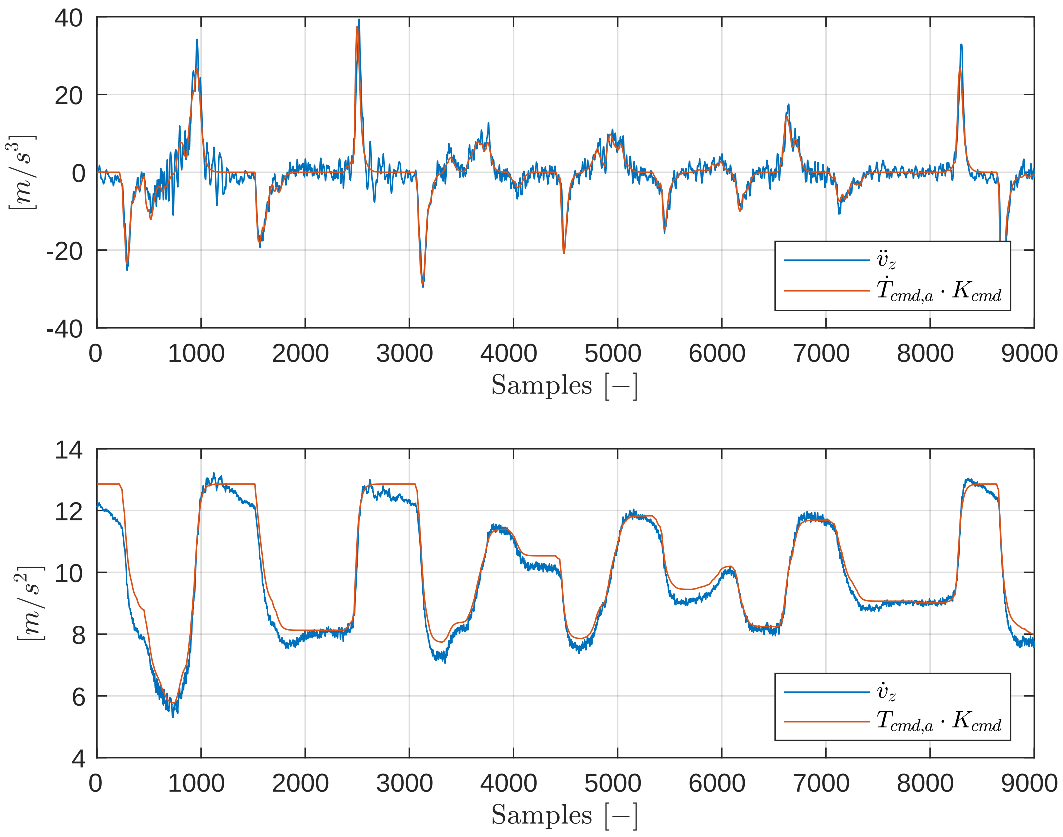

Figure 3-8: Fist subplot: mapped derivative of the commanded thrust $\dot{T}_{cmd,a}$. Second subplot: mapped commanded thrust $\dot{T}_{cmd,a}$. Values were obtained from the flight data.

implementation. The result is then added to the thrust increment $\tilde{T}$, provided by the outer loop INDI. This computation is graphically shown in Figure 3-9. The part of the outer loop which is different from the one used in the Simulink model (see Figure 2-4) is presented in blue color.

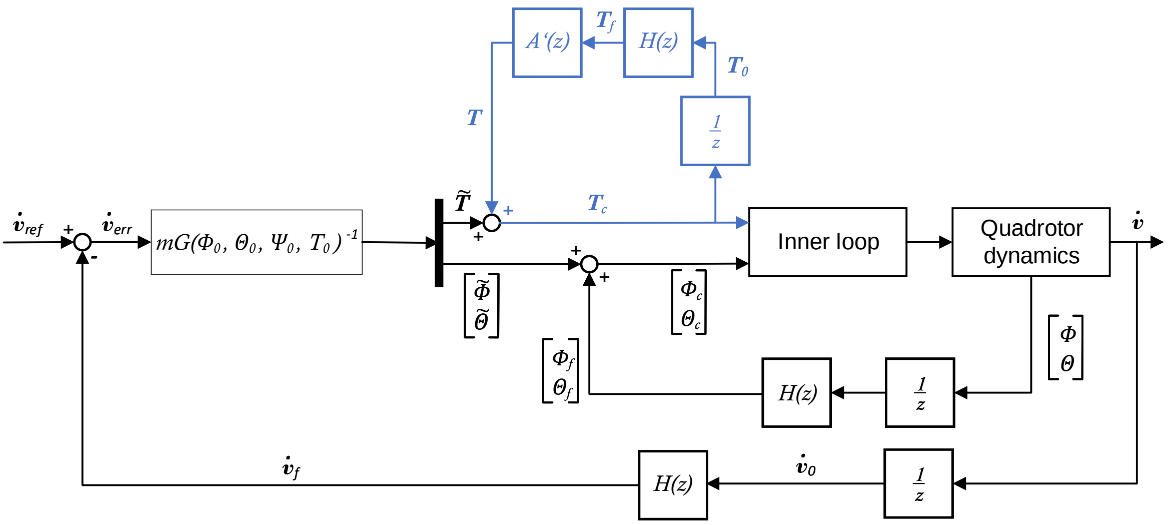
$A'(z)$
T_f
$H(z)$
T_0
T
$\frac{1}{z}$
$\dot{v}_{ref}$
+
-
$\dot{v}_{err}$
$mG(\Phi_0, \Theta_0, \Psi_0, T_0)^{-1}$
$\tilde{T}$
+
+
T_c
Inner loop
Quadrotor dynamics
$\dot{v}$
$\begin{bmatrix}\tilde{\Phi}\\ \tilde{\Theta}\end{bmatrix}$
+
+
$\begin{bmatrix}\Phi_c\\ \Theta_c\end{bmatrix}$
$\begin{bmatrix}\Phi_f\\ \Theta_f\end{bmatrix}$
$H(z)$
$\frac{1}{z}$
$\begin{bmatrix}\Phi\\ \Theta\end{bmatrix}$
$\dot{v}_f$
$H(z)$
$\dot{v}_0$
$\frac{1}{z}$

Figure 3-9: Block diagram of the outer loop from 2.3.3 with adjustments (blue) to account for the inner loop implementation with no thrust increment as input.

An additional aspect of the hardware implementation, which is worth to mention is the computation of the inverse of the $\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0)$ matrix. As the $\boldsymbol{G}(\Phi_0, \Theta_0, \Psi_0, T_0)$ is a square matrix, its inverse can be computed with a regular formula for the inverse of a 3×3 matrix. However, to account for possible numerical errors and vanishing of matrix terms with very small values, the inverse was implemented as a pseudoinverse (*Moore-Penrose Inverse*).

4 Results

This chapter describes the performance of the implemented outer loop. In section 4.1 the disturbance rejection is experimentally analyzed. Section 4.2 presents the general response of the position controller for different step inputs.

4.1 Disturbance Rejection

In [6] was shown, that the transfer function of the inner loop INDI can be simplified to the actuator dynamics $A(s)$. This section aims to show that the transfer function of the outer loop INDI of a purely vertical motion in z_O-direction does not depend from the quadrotor dynamics terms as well and can be described solely with the actuator dynamics $A(s)$ and filtering $H(s)$ terms. To account for a general case, a disturbance rejection term d was added to the examined transfer function, which made it possible to demonstrate the disturbance rejection ability of the INDI in the same experiment.

Figure 4-1 shows the block diagram of the combined INDI in z_O-direction. The assumptions applied to derive the block diagram are described below.

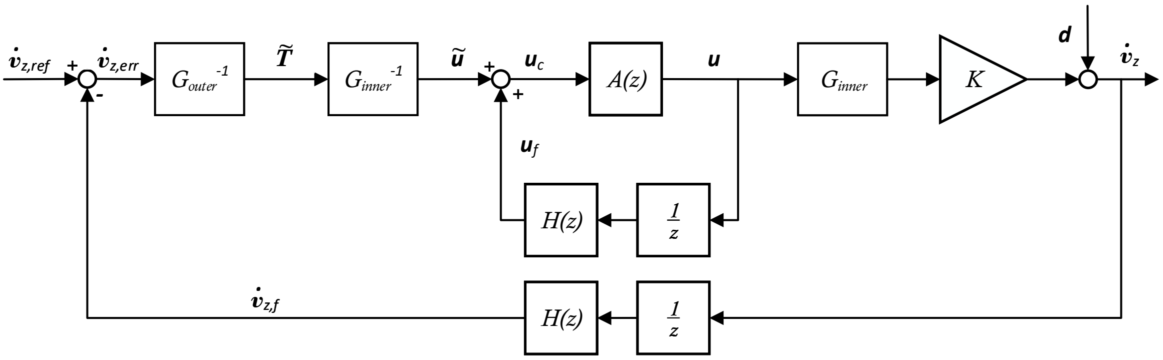

Figure 4-1: Block diagram of the combined INDI.

Here, $\boldsymbol{G}_{inner}$ and $\boldsymbol{G}_{outer}$ represent the quadrotor dynamics terms of the inner and outer loop respectively. If no roll and pitch motion ($\Phi = 0$, $\Theta = 0$) is assumed, the outer loop dynamics is simply $\boldsymbol{G}_{outer} = 1$. The term d represents disturbance (here due to an additional weight $m_r g$). Thus, the disturbance is $d = \frac{m_r g}{m_{cf}}$. By introducing additional mass m_r to the quadrotor system, the estimated terms of $\boldsymbol{G}_{inner}$ change and slightly deviate from those derived in 3.3.1. Thus, to account for this change, the gain $K = \frac{m_{cf}}{m_{cf}+m_r}$ is introduced, where $m_{cf} = 40$ g is the mass of the Crazyflie quadrotor without the additional mass.

The transfer function of the inner loop from the block diagram in Figure 4-1 is defined as:

$$\frac{u}{\tilde{u}} = \frac{A(z)}{1 - z^{-1}A(z)H(z)} \tag{4.1}$$

Using Equation 4.1 to compute the transfer function of the outer loop and applying assumptions mentioned above, the vertical acceleration $\dot{v}_z$ is then given by:

$$\begin{aligned}\dot{v}_z &= \frac{1 - z^{-1}A(z)H(z)}{1 + (K-1)z^{-1}A(z)H(z)}d + \frac{KA(z)}{1 + (K-1)z^{-1}A(z)H(z)}\dot{v}_{z,ref} \\ &= P^{-1}[1 - z^{-1}A(z)H(z)]d + P^{-1}KA(z)\dot{v}_{z,ref}\end{aligned} \tag{4.2}$$

with $P = 1 + (K-1)z^{-1}A(z)H(z)$.

The goal of the experiment was to compare the vertical acceleration of the quadrotor $\dot{v}_{z,measured}$, measured by the accelerometer during the test flight, with the analytically computed term $\dot{v}_z$ from Equation 4.2, given the disturbance d and the reference acceleration $\dot{v}_{z,ref}$. Therefore, the same reference value $\dot{v}_{z,ref}$ was provided to the INDI controller implemented on the real Crazyflie as well as to the analytical transfer function from Equation 4.2. Ten experiments were performed (see Figure 4-2), in which the quadrotor was disturbed twice during its flight with the additional mass $m_r = 5\text{g}$. The disturbance weight $m_r g$ was applied in the form of a step function, by placing it on the top of the Crazyflie above its center of gravity. After aprox. 10 s after placing the mass, it was removed from the quadrotor. The controller was running at 500 Hz.

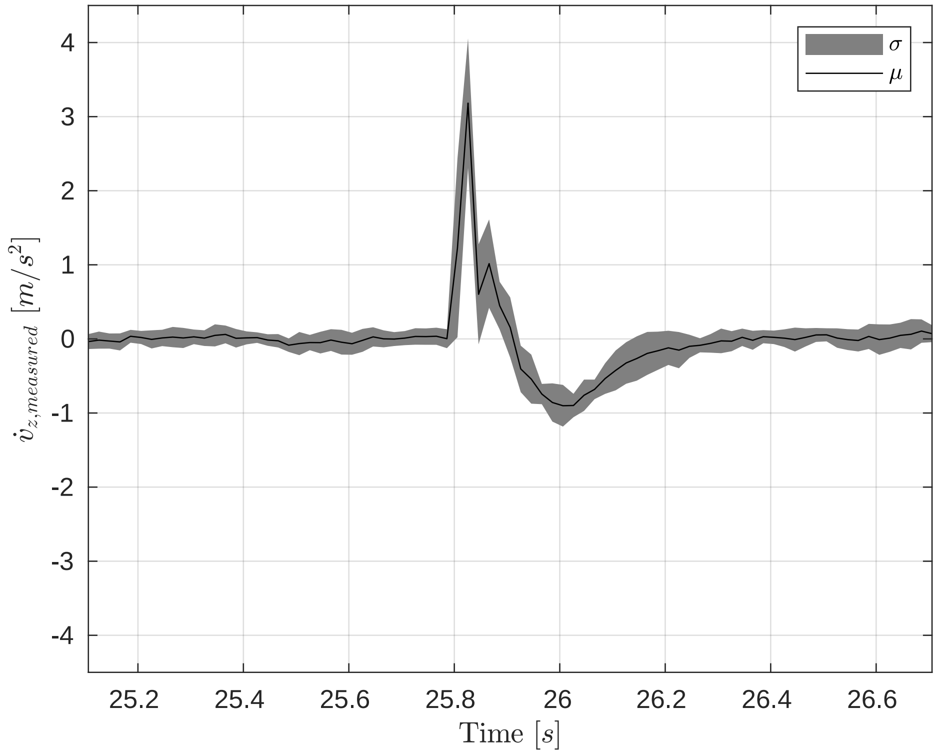

Figure 4-2: Mean μ and standard deviation σ from ten experiments is shown for the measured acceleration $\dot{v}_{z,measured}$.

Figure 4-3 shows the result of this experiment for one of the performed repetitions.

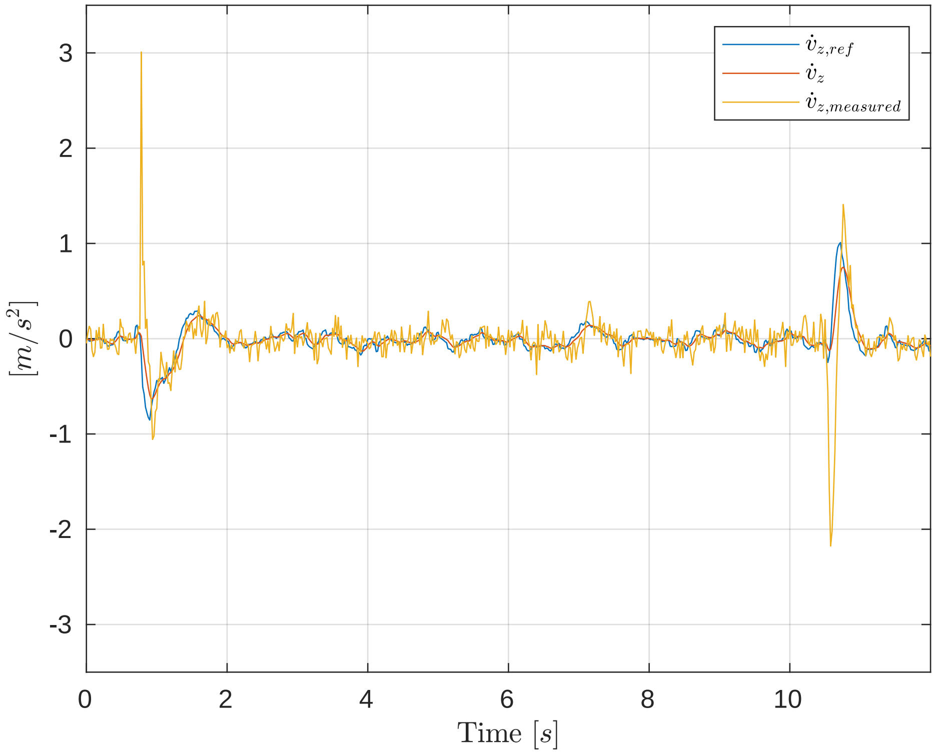

Figure 4-3: Response of the vertical acceleration of the outer loop INDI in an experimental flight with disturbance

In the figure above two cases can be distinguished: the moment in which the additional weight is applied and the moment when the weight is removed. The moment when the disturbance d occurs is characterized by the large difference (yellow spike in positive direction) between the measured acceleration and the reference acceleration. Similarly, the moment when the weight is removed is characterized by the large spike in the negative direction. In both cases, after approx. half a second the measured acceleration $\dot{v}_{z,measured}$ tracks the analytically computed acceleration $\dot{v}_z$ from Equation 4.2. This result confirms the hypothesis made at the beginning of this section, the outer loop behaviour with no roll and pitch motion indeed corresponds to the analytically derived transfer function in 4.2. It can be described solely with the actuator dynamics $A(s)$ and filtering $H(s)$ terms. The performed experiment also demonstrates that the INDI controller is quite effective in dealing with disturbances.

4.2 Position Controller Response

As on the top level the control of the Crazyflie quadrotor (with the implemented outer loop) works by providing the reference position to the position controller, this section briefly shows the response of the position controller for different step inputs. The position and velocity controllers which augment the outer loop INDI consist solely of proportional gains, which were manually tuned to achieve stable responses.

Figure 4-4 contains two subplots with responses of the position and velocity controllers respectively. In the first subplot the response to position steps in x-direction of different length is depicted. Second subplot shows the corresponding response of the velocity controller.

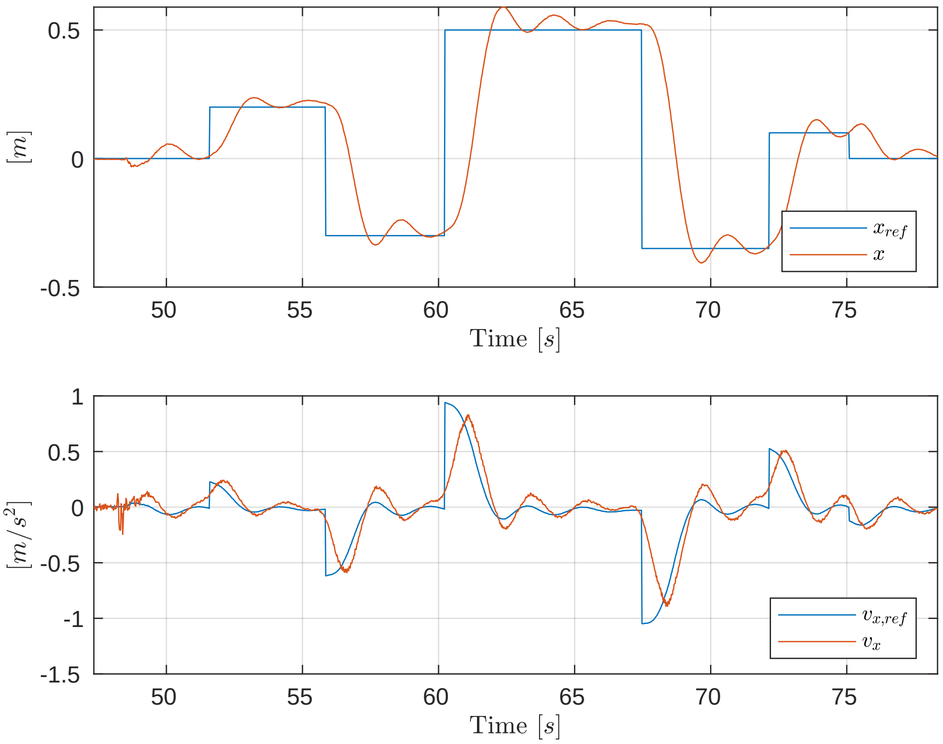

Figure 4-4: Responses of the position controller (first subplot) and the velocity controller (second subplot) obtained from a test flight.

An interesting aspect is that the dynamics in x- and $y-$direction is expected to be different compared to the one in $z-$direction. This is due to the fact, that in the first case, the attitude controller serves as an “actuator”, however in the second case, the propellers are actuators. Both actuator types have different dynamics.

5 Discussion

In this semester thesis, the Incremental Nonlinear Dynamic Inversion (INDI) control algorithm was discussed on the example of the Crazyflie quadrotor. In the first part, a simplified model of the Crazyflie quadrotor was developed in the Simulink environment. Both, the inner loop (attitude controller) and the outer loop (position controller) were then derived and subsequently simulated on this simplified model. In the second part, the outer loop was implemented on the embedded hardware of the Crazyflie. To implement the algorithm, different parameters such as: actuator dynamics time constant, control effectiveness terms, etc. were estimated using the flight data from test flights. Finally, the implemented outer loop was successfully tested on the ability to cope with disturbances. Working control algorithm was then made available to general public via official open source firmware of the Crazyflie quadrotor.

The INDI approach has shown itself to be straightforward in the implementation, since most parameters (e.g. control effectiveness, thrust mapping) can be estimated from the flight data using the onboard sensors and do not require complex methods based on experimental measurements.

Another advantage of INDI is the ability to reject disturbances. Unmodelled dynamics and disturbances are measured with the angular (inner loop) and linear (outer loop) acceleration. This means that in comparison to the Nonlinear Dynamic Inversion (NDI) method, there is no need for complex modelling for INDI. As an example, the most part of the outer loop dynamics was modelled knowing just the thrust model and the relations between different coordinate frames. In the previous chapter it was experimentally shown, that even this knowledge can be unnecessary in some flight modes and the outer loop behaviour can be described only by the actuator and filtering dynamics, as it was shown for the case of the purely vertical motion.

These characteristics make INDI a flexible and promising control algorithm for Micro Air Vehicles (MAV).

References

[1] T. A. Ward, C. J. Fearday, E. Salami, and N. B. Soin. A bibliometric review of progress in micro air vehicle research. *International Journal of Micro Air Vehicles*, 9(2):146–165, 2017.

[2] J.J.E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, Inc, Upper Saddle River, New Jersey 07458, 1991.

[3] J.F. Horn. Non-linear dynamic inversion control design for rotorcraft. *https://www.mdpi.com/journal/aerospace*, 2019.

[4] S. Sieberling, J.A. Mulder, and Q.P. Chu. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance Control and Dynamics, 33.2010 no.6*, 2010.

[5] Eduardo Simões Silva. Incremental nonlinear dynamic inversion for quadrotor control. 2015.

[6] E.J.J Smeur, Q.P. Chu, and G.C.H.E. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance Control and Dynamics, Vol. 39 no.3*, 2015.

[7] E.J.J Smeur, Q.P. Chu, and G.C.H.E. de Croon. Cascaded incremental nonlinear dynamic inversion control for mav disturbance rejection. *Control Engineering Practice, Vol. 73, p. 79-90*, 2018.

[8] Bitcraze, Crazyflie 2.1. `https://www.bitcraze.io/products/crazyflie-2-1/`. Accessed: 2020-04-22.

[9] L. Petricca, P. Ohlckers, and C. Grinde. Micro- and nano-air vehicles: State of the art. *International Journal of Aerospace Engineering*, 2011, 03 2011.

[10] Bitcraze, Crazyflie 2.1 in Research. `https://www.bitcraze.io/portals/research/`. Accessed: 2020-04-22.

[11] J. Förster. Bachelor's thesis: System identification of the crazyflie 2.0 nano quadrocopter. 2015.

[12] P. G. Hamel and J. Kaletka. Advances in rotorcraft system identification. *Prog. Aerospace Sci*, 33:259–284, 06 1997.

[13] M. Faessler, D. Falanga, and D. Scaramuzza. Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robotics and Automation Letters*, PP:1–1, 12 2016.