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 Evaluierung eines Inkrementellen Nichtlinearen Dynamischen Inversionsreglers für einen Nano-Quadrotor

Semesterarbeit

Author: Evghenii Volodscoi

Matriculation number: 03663176

Supervisor: Dr. Ewoud Smeur

April 2020

Statutory Declaration

I, Evghenii Volodscoi, declare on oath towards the Institute of Flight System Dynamics of Technische Universitt Mnchen, 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,

Kurzfassung

Deutsche Kurzfassung der Arbeit.

Abstract

English abstract of the thesis.

Table of Contents

List of Figures

List of Tables

Table of Acronyms

Acronym	Description
EOM	Equations of Motion
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
NDI	Nonlinear Dynamic Inversion
NED	North East Down

Table of Symbols

Latin Letters

Symbol	Unit	Description
F	N	Force
g	m/s^2	Gravitational acceleration

Greek Letters

Symbol	Unit	Description
α	rad	Angle of attack
ζ	–	Damping of a linear second order system

Indices

Symbol	Unit	Description
m		Variable related to pitch moment
W		Wind

1 Introduction

1.1 Motivation

1.2 Contribution of the Thesis

- 3. Implementation in Simulink for testing (both loops)
- 3. Implementation on HW (outer loop)

1.3 Structure of the Thesis

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 Matlab/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 $\sum (\boldsymbol{F}^G)_B \in \mathbb{R}^{3\times 1}$ 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, $(\dot{\boldsymbol{v}}^G)_B \in \mathbb{R}^{3\times 1}$ the linear acceleration of the point G, $(\boldsymbol{\omega}^{OB}) \in \mathbb{R}^{3\times 1}$ 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 $\sum (\boldsymbol{M}^G)_B \in \mathbb{R}^{3\times 1}$ 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

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

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 chapter 3.2.

2.2 Nonlinear Dynamic Inversion

In this subsection, the Nonlinear Dynamic Inversion (NDI) 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 objective is to track some desired trajectory [1]. 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

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

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 an *equivalent 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 [2]. 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 [3]. 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 (INDI) can be used to achieve this, it is discussed in the next subsection.

2.3 Incremental Nonlinear Dynamic Inversion

The INDI is an incremental form of the NDI for which the lack of the accurate system dynamics model does not critically affect the performance of the control algorithm [4]. 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

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 is fast compared with the dynamics of the system [4]. Thus, the 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, $\boldsymbol{u_0}$ 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 the 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 [3].

2.3.2 Inner loop INDI for 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 [5], [6]. 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 [6]. As only the body-fixed frame is used in the following control law derivation, the subscripts B are not used in this subsection. Also all external forces and moments apply to the center of gravity of the quadrotor, thus the superscripts G are also omitted. Solving Equation 2.2 for the angular acceleration results in

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

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 [5]. 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 the 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.

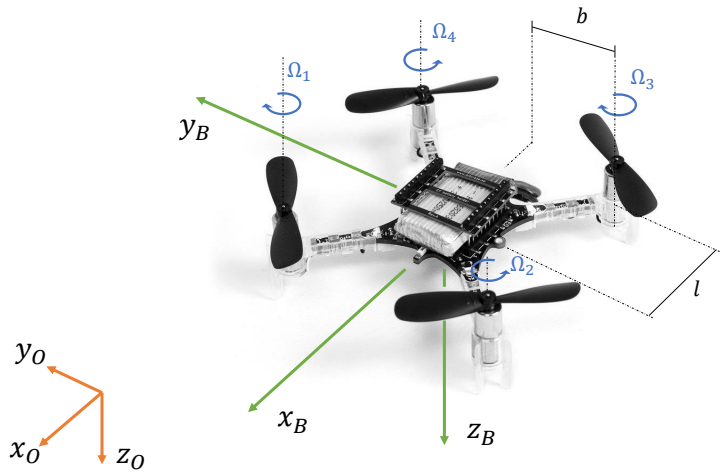

Figure 2-1: Image of the Crazyflie 2.1 quadrotor adopted from [7] with indicated lever arms of the motors.

As already explained in the previous subsection to derive the incremental control law the Taylor series expansion is perfromed on the 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, 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 $\boldsymbol{\omega}_0$ and inputs $\boldsymbol{\Omega}_0$ and can be denoted as $\dot{\boldsymbol{\omega}}_0$. 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 $\boldsymbol{G}_2$

$$\boldsymbol{G}_1 = \begin{bmatrix} \boldsymbol{I}^{-1} \begin{bmatrix} -bk_F & bk_F & bk_F & -bk_F \\ lk_F & lk_F & -lk_F & -lk_F \\ k_M & -k_M & k_M & -k_M \end{bmatrix} - \boldsymbol{I}^{-1} \begin{bmatrix} \boldsymbol{\omega}_y & 0 & 0 \\ 0 & \boldsymbol{\omega}_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \\ -I_{rzz} & I_{rzz} & -I_{rzz} & I_{rzz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ k_F \cdot \mathbf{1}_{1\times 4} \end{bmatrix} \tag{2.21}$$

$$\boldsymbol{G}_2 = \begin{bmatrix} -\boldsymbol{I}^{-1} T_s^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{rzz} & -I_{rzz} & I_{rzz} & -I_{rzz} \end{bmatrix} \\ \mathbf{0}_{1\times 4} \end{bmatrix} \tag{2.22}$$

with the terms $\mathbf{1}_{1\times 4}$ and $\mathbf{0}_{1\times 4}$ 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. It has been shown, that 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 [5]. Thus, the same second order filter is applied to all variables with a subscript 0. By applying filtering (subscript f) and 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 $\dot{\boldsymbol{\Omega}}$ and substituting $\dot{\boldsymbol{\omega}}$ with the equivalent input $\boldsymbol{\nu}_{ang}$, the control law of the inner loop is obtained

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

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

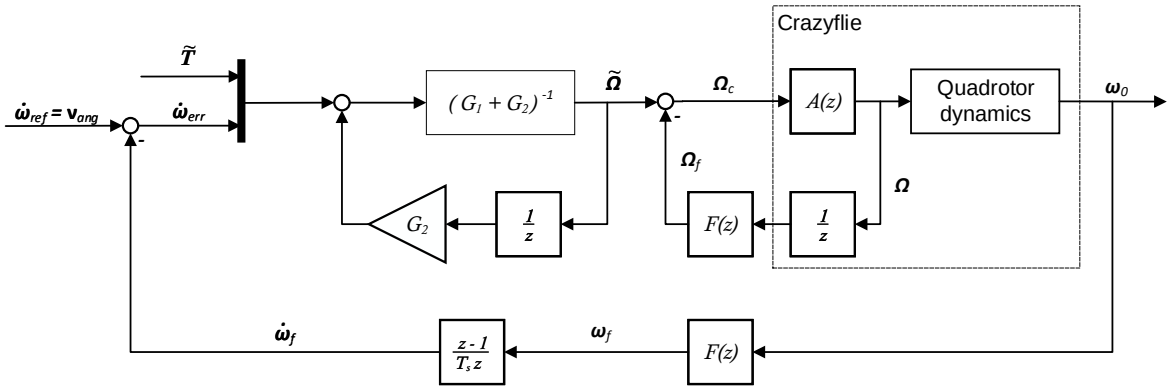

Figure 2-2: Block diagram of the inner loop INDI controller. $A(z)$ is the actuator dynamics (see 3.2.2) and $F(z)$ is the second-order filter (see 3.2.3).

The inner INDI controller which controls the angular acceleration is then augmented with two simple controllers which control the angular velocity $\boldsymbol{\omega}$ and the attitude $\boldsymbol{\eta}$ 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.

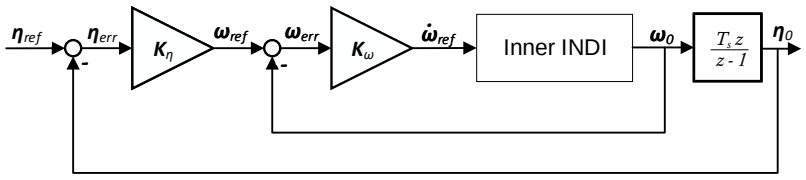

Figure 2-3: Attitude controller as an augmentation of the inner loop INDI controller.

2.3.3 Outer loop INDI for 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, 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 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 to the NED frame the rotation matrix $\boldsymbol{M}_{OB}$ is used. This results in

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

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

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

The first term on the right side of the Equation 2.30 is the acceleration at 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 on the right are assumed to be zero since it is very complicated to develop an accurate model of the aerodynamic force as the nature of the wind influence can vary and remains unknown. It is also 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

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

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. Applying the same filter 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 equivalent control variable $\boldsymbol{\nu}_{lin}$, the control law of the outer loop is obtained

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

Figure 2-4 shows the block diagram of the outer loop INDI.

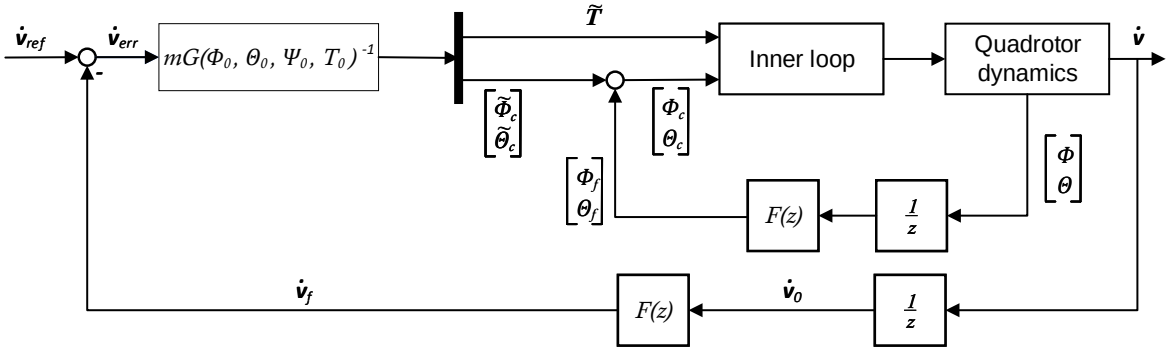

Figure 2-4: Block diagram of the outer loop INDI controller.

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

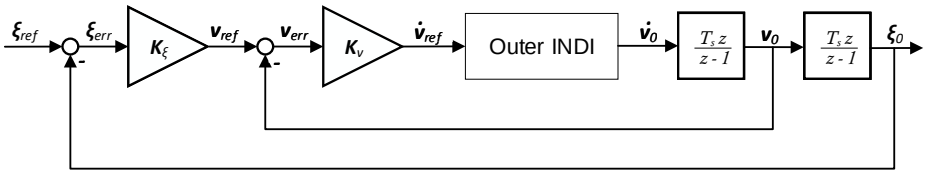

Figure 2-5: Position controller as an augmentation of the outer loop INDI controller.

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

The Crazyflie 2.1 quadrotor developed by Swedish company Bitcraze belongs to the nano-quadrotor class (see Figure 3-1). Its total weight ranges from 27g to 42g, which makes 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.

The Crazyflie's main application Microcontroller (MCU) is STM32F405. It runs the firmware with frequency of 1000 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 (IMU) 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) sensor which measures the range to the next object along the body-fixed z-axis of the quadcopter. 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 [7].

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 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 dynamic 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 applied 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 the 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 value T was modelled as a linear function of rotational rates of the propellers

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

where k_F is the force constant of the rotors. In subsection **??** 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) is greater than the influence of the moment constant of the rotors k_M, and that the *thrust to weight ratio* expression is $\frac{T}{W} \approx 2$.

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

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

where $(u_K)_B$, $(v_K)_B$ and $(w_K)_B$ are kinematic velocities of the quadrotor along the axes x_K, y_K and z_K respectively, given in the body-fixed coordinate frame. k_x, k_y and k_z are experimentally estimated drag coefficients adopted from [8].

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 pendulum method in [8]. 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. 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. 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

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

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 bilinear transformation, the mapping relation between s and z results in

$$s \widehat{=} \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \tag{3.6}$$

Thus, the discrete version of the Equation 3.6 is

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

where

$$n_0 = \omega_n^2 \tag{3.8a}$$
$$n_1 = 2\omega_n^2 \tag{3.8b}$$
$$n_2 = \omega_n^2 \tag{3.8c}$$
$$d_0 = \left[(\frac{2}{T_S})^2 + \frac{4\zeta\omega_n}{T_S} + \omega_n^2 \right] \tag{3.8d}$$
$$d_1 = \left[\frac{8}{T_S^2} + 2\omega_n^2 \right] \tag{3.8e}$$
$$d_2 = \left[(\frac{2}{T_S})^2 - \frac{4\zeta\omega_n}{T_S} + \omega_n^2 \right] \tag{3.8f}$$

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 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 3-5 confirms this expectation, by plotting the step response of the Crazyflie model with and without considering the gyroscopic effects of the quadrotor

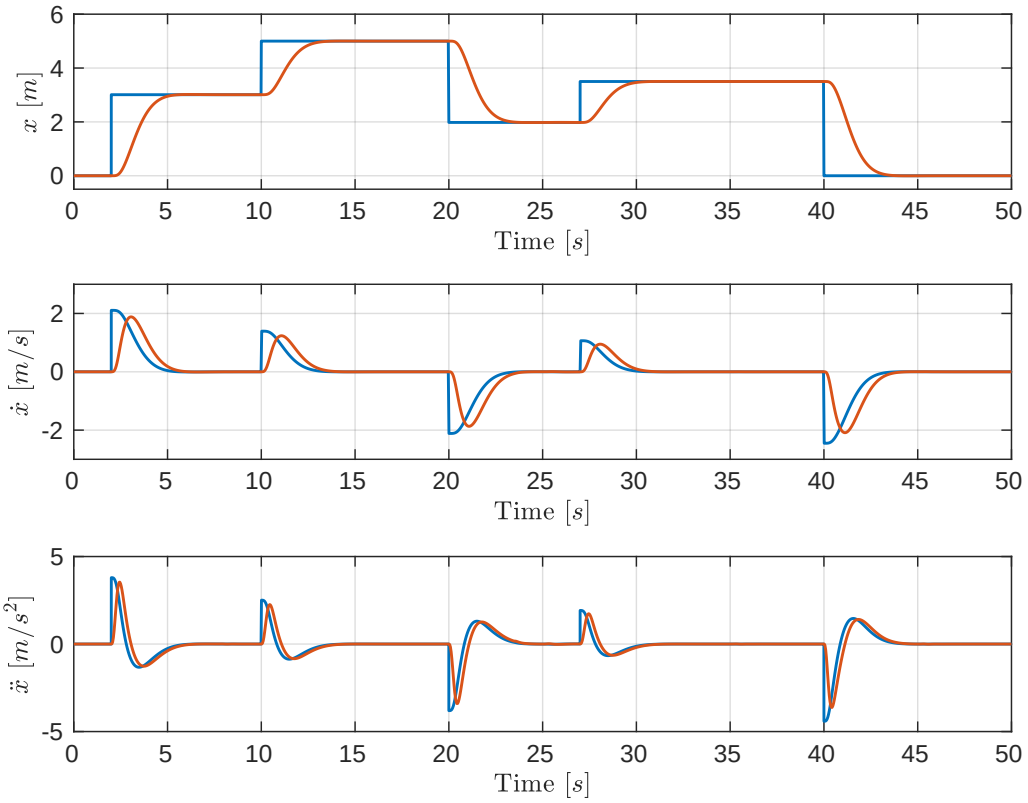

Figure 3-2: Response of the outer loop model to a step input. Blue lines represent reference values, red lines denote measured values.

propellers. It can be seen that both responses are almost similar. 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.

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 implementation errors were corrected. Additionally, the control effectiveness parameters G_1 and G_2 were reestimated with a new flight data.

To estimate model parameters of a quadrotor two strategies can be applied: estimation through measurements and estimation with flight data. Using the first strategy it is usually difficult to

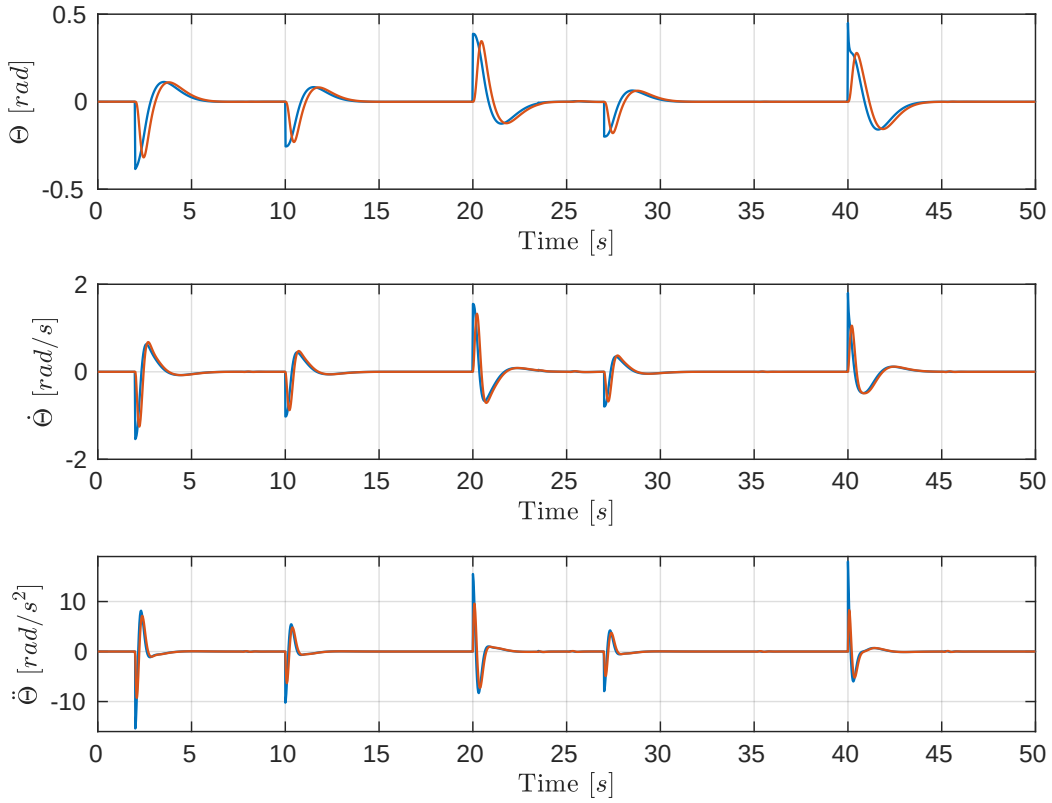

Figure 3-3: Response of the inner loop model to the input provided by the outer loop. Blue lines represent reference values, red lines denote measured values.

achieve accurate results [6]. By contrast, pursuing the second approach can reduce the amount of effort and also lead to plausible results. To collect the flight data, 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

$$\Delta\dot{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\Omega} \\ \Delta\dot{\boldsymbol{\Omega}} \end{bmatrix} \tag{3.9}$$

where $\Delta\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_f$ and $\Delta\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_f$. Note that the terms $\Delta\boldsymbol{\Omega}$ and $\Delta\dot{\boldsymbol{\Omega}}$ have to be passed through the actuator dynamics $A(s)$. 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,

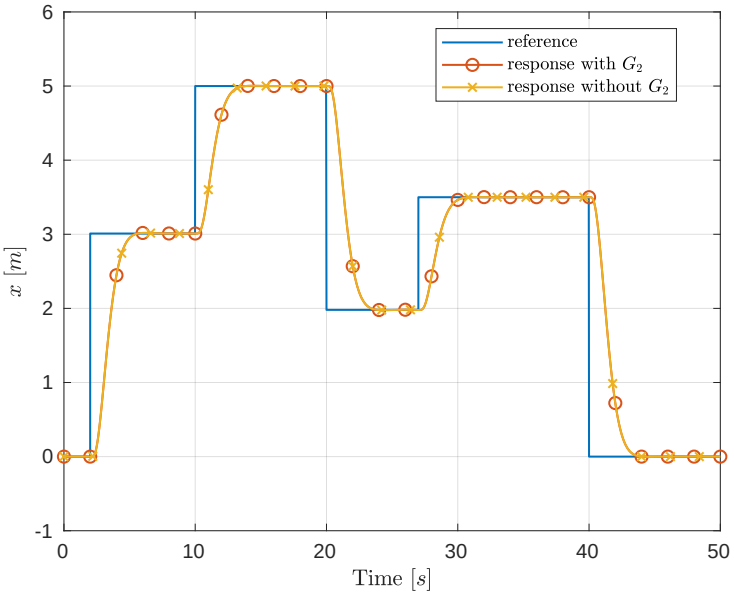

Figure 3-4: Responses of the closed loop using dynamics model with and without considering the gyroscopic effects of the propellers.

Equation 3.9 results in

$$\Delta\ddot{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{G_1} & \boldsymbol{G_2} \end{bmatrix} \begin{bmatrix} \Delta\dot{\boldsymbol{\Omega}} \\ \Delta\ddot{\boldsymbol{\Omega}} \end{bmatrix} \tag{3.10}$$

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, contribution 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 $\Delta\dot{\Omega}_z \cdot \boldsymbol{G_1}$.

3.3.2 Parameter Estimation of the Outer Loop

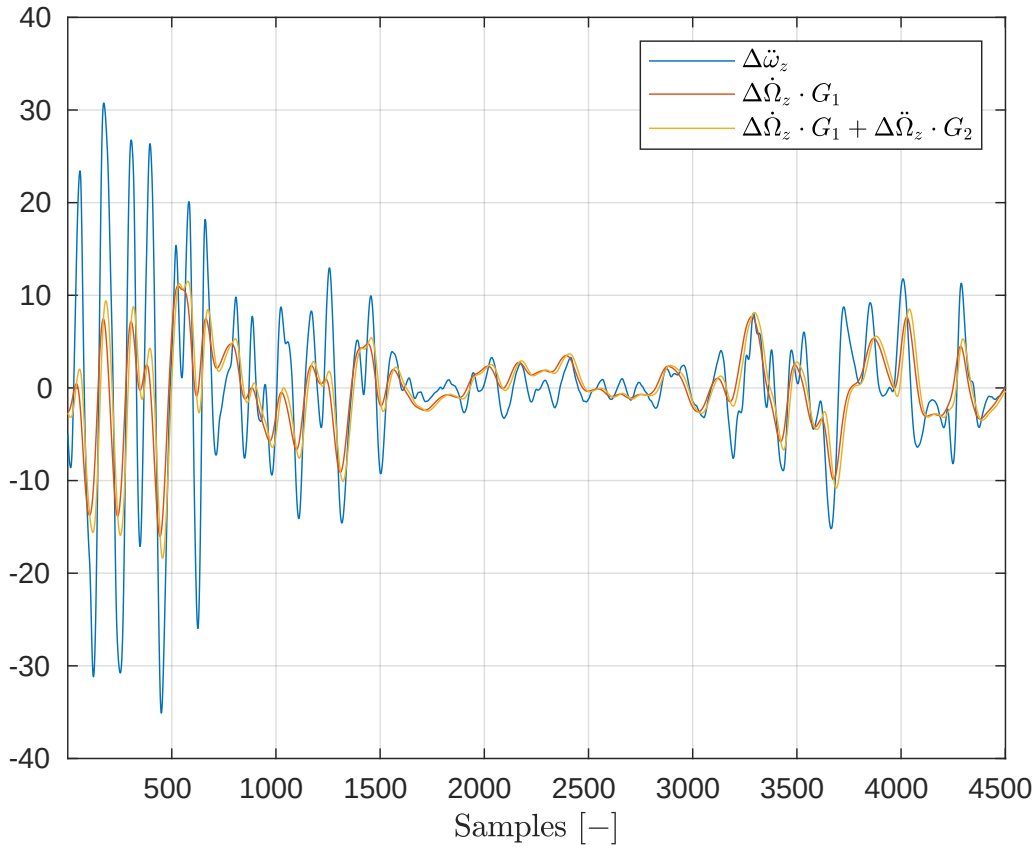

Figure 3-5: Contribution of the estimated G_1 and G_2 terms to the yaw component of the angular acceleration.

4 Results

5 Discussion

References

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

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

[3] 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.

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

[5] 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.

[6] 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.

[7] Bitcraze, Crazyflie 2.1. `https://www.bitcraze.io/crazyflie-2-1/`. Accessed: 2020-03-05.

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

Appendix