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
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

1.3 Structure of the Thesis

2 Theoretical Background

...

2.1 Dynamic Equations of Motion of an Aircraft

In this section general equations of motion of an aircraft are presented.

For the derivation of the equation 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 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 force vector 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, $(\boldsymbol{\omega}^{OB}) \in \mathbb{R}^{3\times1}$ 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. 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 $\frac{d}{dt}(I)_B = 0$ respectively?. $(I)_B$ is 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 rigid body 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

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

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 above presented Equations (2.1)-(2.4) can be used to represent a motion of a general aircraft. $\sum(\boldsymbol{F}^G)_B$ and $\sum(\boldsymbol{M}^G)_B$ are sums of the external forces and moments acting on the aircraft. 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 (...Simulink).

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 system vector, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{1}}$ 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 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 (2.11) applied to Equation (2.9) yields the simple linear relation

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

Despite wide usage (especially in the past of the flight control) of the NDI method and its numerrous extensions (cite-tag Horn), it also has drawbacks.

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]. As shown in the previous chapter (cite-tag prev-chapter), 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. This method is called Incremental Nonlinear Dynamic Inversion (INDI) and will be discussed in the next section.

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

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x_0}) + \boldsymbol{G}(\boldsymbol{x_0})\boldsymbol{u_0} \\ &+ \frac{\partial}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\bigg|_{\boldsymbol{x}=\boldsymbol{x_0},\boldsymbol{u}=\boldsymbol{u_0}}(\boldsymbol{x}-\boldsymbol{x_0}) \\ &+ \frac{\partial}{\partial \boldsymbol{u}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\bigg|_{\boldsymbol{x}=\boldsymbol{x_0},\boldsymbol{u}=\boldsymbol{u_0}}(\boldsymbol{u}-\boldsymbol{u_0})\end{aligned} \tag{2.12}$$

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

$$\begin{aligned}\dot{\boldsymbol{x}} &= \dot{\boldsymbol{x}}_\mathbf{0} \\ &+ \frac{\partial}{\partial \boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}]\bigg|_{\boldsymbol{x}=\boldsymbol{x_0},\boldsymbol{u}=\boldsymbol{u_0}}(\boldsymbol{x}-\boldsymbol{x_0}) \\ &+ \boldsymbol{G}(\boldsymbol{x_0})(\boldsymbol{u}-\boldsymbol{u_0})\end{aligned} \tag{2.13}$$

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. Thus, the Equation 2.13 is further simplified to

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_\mathbf{0} + \boldsymbol{G}(\boldsymbol{x_0})(\boldsymbol{u}-\boldsymbol{u_0}) \tag{2.14}$$

Solving Equation 2.14 for $\boldsymbol{u}$ and substituting $\dot{\boldsymbol{y}}$ with $\boldsymbol{\nu}$, the INDI control law can be obtained

$$\boldsymbol{u} = \boldsymbol{u_0} + \boldsymbol{G}(\boldsymbol{x_0})^{-1}(\boldsymbol{\nu} - \dot{\boldsymbol{x}}_\mathbf{0}) \tag{2.15}$$

where $\dot{\boldsymbol{x}}_\mathbf{0}$ 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. This control law results in computing the increment $\boldsymbol{\Delta u}$ of the control input and adding it to the previous value $\boldsymbol{u_0}$, instead if computing complete control input command $\boldsymbol{u}$. As it is less dependent on the model of the system dynamics, it increases the robustness of the system [3].

2.3.2 Inner INDI loop for Quadrotor

- Derivation of the inner INDI loop (what we want to control, detailed equations).

2.3.3 Outer INDI loop for Quadrotor

- Derivation of the outer INDI loop (what we want to control, detailed equations).

3 Implementation

3.1 Research Quadrotor

- Some facts about Crazyflie hardware (foto, uC frequency, weight, length)

3.2 Implementation in Simulink (Simulink Model)

3.2.1 Purpose

- Estimation of relevant components (Matrices...)
- Testing of the PD-gains
- Testing the filter

3.2.2 Structure

- Force frame from B to O for an easy representation
- Show each equation of forces with explaining the made assumptions
- Parameters
- Actuator dynamics
- Filter
- Images of the Simulink model

3.2.3 Simulation Results

3.3 Implementation on the Hardware

3.3.1 Structure of the Code

3.3.2 Testing with contact Forces and Moments

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.

Appendix