UNIVERSITY OF AGDER

Tilt Rotor Tricopter

Control system for the holonomic multirotor platform

by

Daniel Salem

Sindre Gjertsen

Supervisors:

Morten Ottestad

Øyvind Magnussen

University of Agder, 2013

Faculty of Technology and Science

Department of Engineering

Abstract

Development of a new approach to the multicopter segment of the Unmanned Areal Vehicle (UAV) family is presented. The system is designed on a T-shaped tricopter platform with ability to tilt all three motors, hereby defined as Tilt Rotor Tricopter (TRT). The highly coupled nonlinear system is investigated through the mathematical model, and verified by simulations. Linearization of the system has been achieved around hovering conditions, and represented in the state-space environment. The Linear Quadratic Regulator (LQR) has been implemented on the physical system, achieving controllable flight. As the prestudy discovered no identical platforms to the TRT, we are proud to present a working new segment to the diversity of multicopter platforms.

Acknowledgment

We would like to thank the University for providing us with the opportunity to conduct this thesis and supplying us with the necessary components. Our supervisors have been available for questions and help throughout the semester, which is a crucial part of this thesis outcome. Øyvind Magnussen has been very helpful with coding the flight controller and aiding us in the testing phase of the thesis. A special thanks to Geir Hovland for investing his time and knowledge in assisting us with the control design and regulation. Morten Kjeld Ebbesen has spent hours in our office explaining and helping us understand the world of 3D kinematics. A last thanks to our friends and loved ones for understanding and supporting us all throughout these 5 years of education.

Contents

Nomenclature

Notation	Denotation
ω	Motor angular velocity
F_{1-3}	Force produced by a motor
α_{1-3}	Servo angle
m_{body}	Total body mass
J_{body}	Inertia of body
X_b, Y_b, Z_b	Body coordinate system
X_n, Y_n, Z_n	Navigation coordinate system
ϕ, θ, ψ	Roll, Pitch, Yaw
R_x, R_y, R_z	Rotational matrix
C_b^n, C_n^b	Coordinate transformation matrix
p, q, r	Angular velocity of the body
g	Gravity
M_{thrust}	Torque induced by thrust
M_{motor}	Torque induced by rotation of all motors
M_{gyro}	Torque induced by gyroscopic effect
M_{tot}	Total torque acting on the body
L_{ix}, L_{iy}, L_{iz}	Length from motor to center of gravity
$h(x, u)$	Nonlinear function of the system
A, B, C	System, input, output state-space matrices
x, u, y	State, input, output vector
K_1, K_2	Feedback gain matrices
SP_x, SP_u	State setpoint, input setpoint

Table 1: Variables and Symbols

Notation	Denotation
TRT	Tilt Rotor Tricopter
UAV	Unmanned Areal Vehicle
rpm	Revolutions per minute
PPM	Pulse-position modulation
ESC	Electronic Speed Controller
IMU	Inertial Measurement Unit
Rx and Tx	Radio receiver and transmitter
COG	Center Of Gravity
DCM	Directional Cosine Matrix
CAD	Computer-Aided Design
DOF	Degree Of Freedom
LQR	Linear Quadratic Regulator
MIMO	Multiple Input Multiple Output

Table 2: Abbreviations and Acronyms

Chapter 1

Introduction

Unmanned Areal Vehicles (UAV) are establishing a foothold as the next step in the direction of computer controlled flight. With continuous inputs, such as a directional vector given by user through a radio controller, or a fixed point in space, defined as GPS coordinates, the UAV is capable of semi-autonomous and autonomous flight. Maintaining stability while airborne requires quick and appropriate adjustments to the stochastic disturbances. Such adjustments, if made by an operator, would be subject to delays and to the operators skill level. By introducing the advanced control capabilities of a microprocessor with a range of sensory inputs, maintaining stability is no longer a concern of the operator.

Recent development in microchip technology, improving size and processing capabilities along with massive reductions in production and assembly costs, has fueled a new generation of vehicles. Technology previously only available to heavily funded research departments and often developed for military purposes are now commonly found in appliances like toys and mobile phones.

Pushing further into the field of unmanned flight, focusing especially on agility and stability at low speeds, leads to the development of advanced UAV's. In situations such as a difficult search and rescue, monitoring the welded structure of an oil platform or the conducting cables of a power grid, having eyes on target is an essential aid. Being able to record from this perspective appeals to film makers and photographers alike, and the demand for compact and easy to use units are increasing.

The multicopter, often described as a tricopter or quadcopter depending on its rotor count, was developed to meet this demand. A multicopter commonly uses 3, 4, 6 or 8 rotors to achieve flight, and while increasing the number of rotors increases lifting capabilities and stability, it also negatively effects power consumption and thereby require expensive high capacity batteries. Developing a multicopter with fewer rotors, with increased agility and maintaining the stability of its bigger brethren, is a new field of rapid progression.

The tricopter, commonly resembling a Y or T shape, uses three motors to achieve flight. To maintain agility and stability while minimizing attitude alterations, requires introduction of additional degrees of freedom. By introducing tilt to each motor, the control systems accuracy in disturbance rejection is improved because the number of solutions to the differential equations controlling flight is increased. In comparison, the addition of tilt allows for an increased agility as that of a multicopter with fixed rotors.

Prestudies revealed no previous attempts at creating a tilt rotor multicopter accessible to the general public, although the technology has been used in military aircraft such as the Bell Boeing V-22 Osprey [8] and the IAI Panther UAV [9]. There are several studies and projects focusing on different aspects of multicopter flight, and the scientific background is well documented.

Previous research conducted at the university by Ø.Magnussen and K.E.Skjønhaug [13] about the dynamics of the quadrocopter, has led to the desire of gaining additional knowledge in the field of multicopters. The source of inspiration behind this thesis is the Orca tricopter from the computer game Command & Conquer and the bicopter Scorpion from the movie Avatar. The main goal of this thesis is to develop a functional tricopter with capability of tilting all three motors. This platform shall be manufactured using standard RC-model components and have comparable cost to other multicopters, to be accessible by the general public. The mathematical equations shall be derived, simulated and verified. A suitable controller shall be designed and tested on the simulation model, implementing it on the physical system for experimental testing. The final testing shall contain experimental flight recording and be discussed based on the performance of the control system.

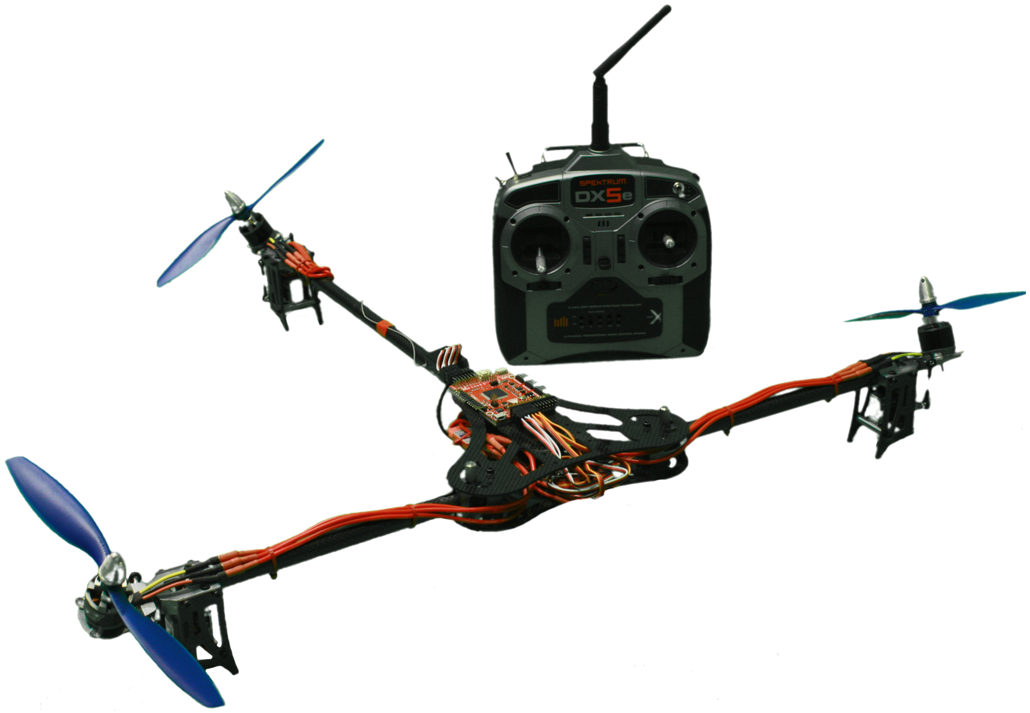

Figure 1.1: View of produced prototype

Figure 1.1 shows the final setup of the functional TRT model. The development of a tricopter with tilt capability on all three rotors is presented in this thesis from scratch to flight test. In Chapter 2 the system overview will be discussed. This contains the different components chosen for the physical TRT model and results after component testing. Chapter 3 will explain the reference geometry and rotational matrixes, then the differential equations based on the Newton-Euler approach will be derived. Chapter 4 discusses the construction of the simulation model and the process where the differential equations gets verified by a SimulationX model. The development of the control system is presented in Chapter 5, both the linearizing process and parameter estimation for the Linear Quadratic Regulator. Chapter 6 presents the implementation of the control system to the physical model and the results from flight tests.

Chapter 2

System overview

To develop the low cost platform, available to the general public, it was decided that TRT will be constructed using only prefabricated components. This chapter addresses choice of the components, mechanical structure, component properties and sensors necessary to complete the project. A complete part list can be found in appendix D.

2.1 Mechanical structure

The mechanical structure of the TRT consist of a frame with 6 actuators. There are 3 brushless DC motors driving the propellers, that can be independently tilted by 3 servomotors. Figure 2.1 shows the mechanical overview, together with a detailed view of the servo tilt mechanism.

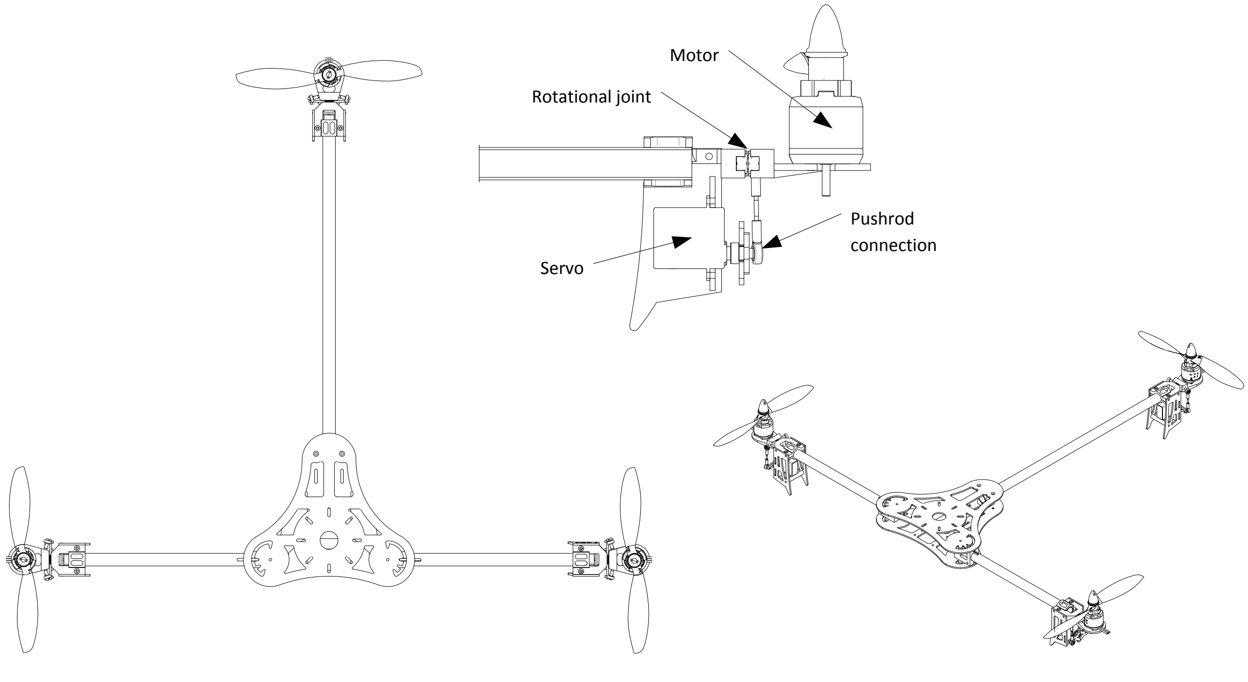

Figure 2.1: Mechanical overview

Platform

The platform is a T-shaped frame, with a motor on the tip of each arm. This platform is based on the Turnigy Talon Tricopter V1.0 frame, which is a carbonfiber tricopter frame with capability of tilting only its rear motor. The two front arms are standard 220mm tubes while the rear one has been extended to 380mm. Two rotational joint has been added to the two front arms, thereby allowing tilting of all three motors. The rotational joint consists of a servo connected to the motor by a pushrod connection, allowing for ±50° tilt. Detailed view of tilt mechanism is shown in figure 2.1.

Propulsion

The power source of the system is a 3-cell Li-Po battery providing 11.1$[V]$. The brushless DC motors that has been selected for the project are the NTM Prop Drive 28-26, with a $K_e = 1E^{-3}[\frac{rpm}{V}]$, capable of theoretically reaching 11 100$[rpm]$ with no load. Since the motors will be driving 8x4.5$[in]$ propellers that are experiencing air resistance, the resulting maximal rotational speed will be reduced. Equation 2.1 describes correlation between the rotational speed ω and applied torque T_m. Parameters K_t, K_e, R, L are constants dependent on the motor type, while u_{sup} and i are supplied voltage and circuit current correspondingly.

$$\begin{aligned} u_{sup} &= R \cdot i + L \cdot \tfrac{di}{dt} + K_e \cdot \omega \\ T_m &= K_t \cdot i \\ \omega &= (u_{sup} - \tfrac{T_m}{K_t} \cdot R - L \cdot \tfrac{di}{dt})\tfrac{1}{K_e} \end{aligned} \tag{2.1}$$

Figure 2.2 show the experimental data from the motor test. The angular velocity of the motor was increased at a very low rate during the test, reducing the motor inductance to the minimum. It can be seen that motor reaches maximum speed of around 8 600$[rpm]$, producing approximately 650[g] of thrust.

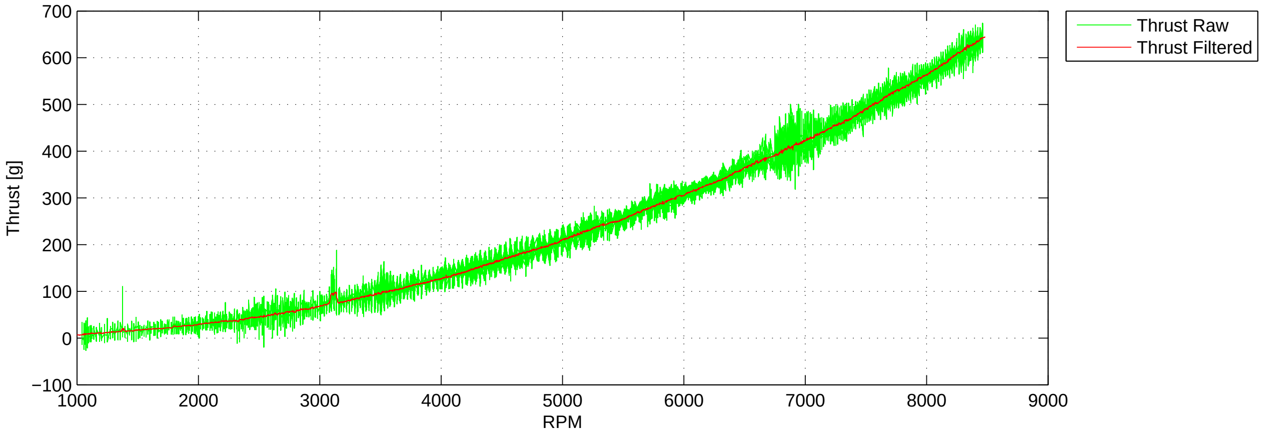

Figure 2.2: rpm to Thrust

The lift force is produced only by the propellers and it is therefore important to have enough thrust to overcome the gravity. The total weight and COG placement of the TRT determines the amount of lift each motor need to produce at the hovering state. The effective lift force of each motor is the function of the motor tilt angle, which in the worst case results in factor of 0.64[-]. As the TRT is estimated to be 1.2$[kg]$, the effective lift force is still larger than the total weight.

To test the step response of the motors for both falling and rising step, the motor was running at a constant thrust level then given a step change of 135[g]. As figure 2.3 shows, both rising and falling response behave close to equal to each other. This validates that the same model can be used for acceleration and deceleration of the motors.

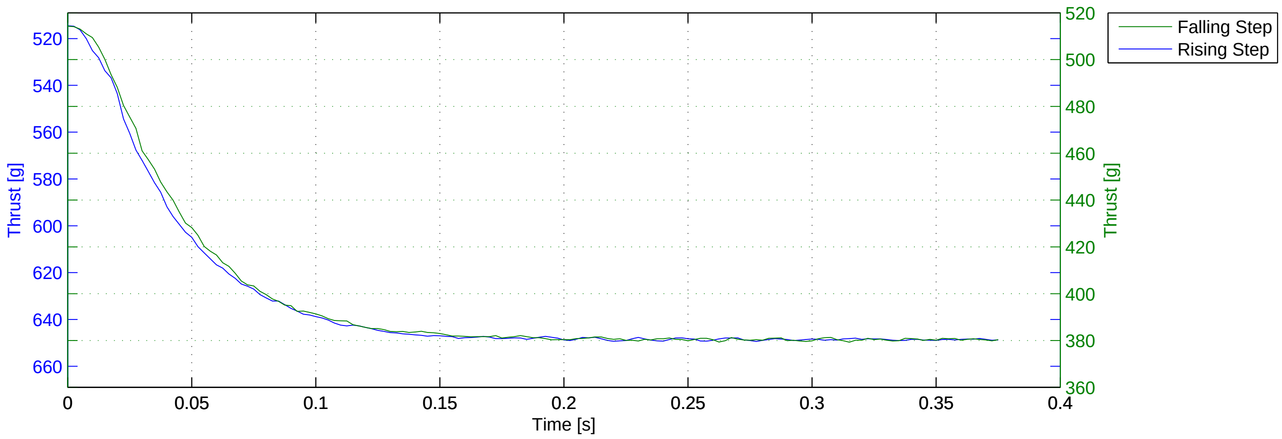

Figure 2.3: Stepresponse on falling and rising step

The motors are controlled by the Electronic Speed Controller (ESC), which converts the input Pulse-Position Modulated (PPM) signal into the output 3-phase motor signal. The input PPM signal has to be created by the microcontroller. The created signal is designed for the 16[MHz] microcontroller, where 16000 cycles is equal to 1[ms]. Each PPM signal period starts with a 1[ms] pulse, which initializes the period. Pulse length past 1 [ms] is the control signal, which varies in length from 0 to 1 [ms]. The maximum combined pulse length is 2 [ms]. The total period of the PPM signal is 20[ms], where the last 18[ms] is called a synchronization period, as shown in figure 2.4.

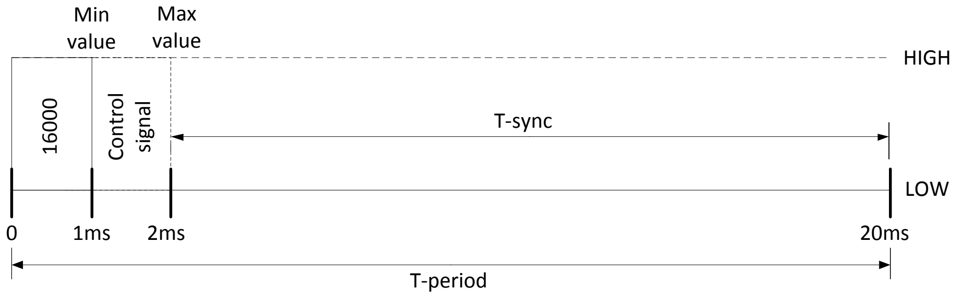

Figure 2.4: PPM signal structure

The control signal for the motor is described by the PE function and is build on the rpm to thrust relationship shown in the figure 2.2. The maximum value of the function is limited to 16000, which corresponds to the maximal rotational velocity of the motor. The polynomial equation is calculated by the *polyfit()* function in Matlab and presented in the equation 2.2.

$$PE = p_0 F_i^3 + p_1 F_i^2 + p_2 F_i + p_3; \qquad i = 1..3$$

where:

$$\begin{aligned} p_0 &= 39.3005 \\ p_1 &= -603.7002 \\ p_2 &= 4758.5445 \\ p_3 &= 34.4823 \end{aligned} \tag{2.2}$$

The figure 2.5 illustrates the estimated polynomial function in operational range of the motors. It can be seen that the 3rd order polynomial function is capable of recreating the filtered thrust curve quite accurately.

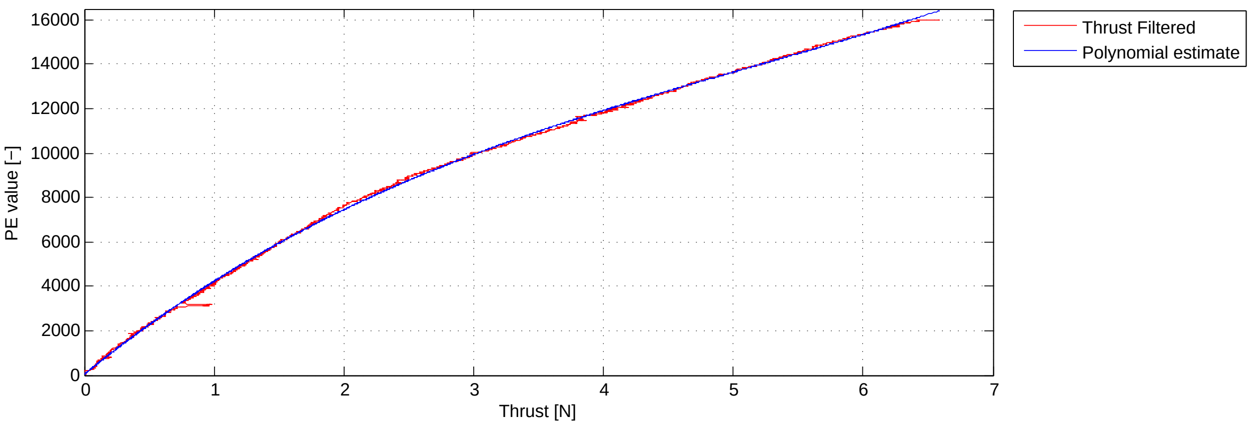

Figure 2.5: Polynomial thrust estimate

The servomotors are controlled using the PPM signal as well. The control signal for servomotors is described by the function SA in the equation 2.3, where α is the desired angle in radians and sr is the operational range of the servomotor in degrees.

$$SA = 8000 + \alpha \cdot \tfrac{16000}{sr \cdot \pi / 180} \tag{2.3}$$

When the servomotors are supplied with a voltage of $4.8[V]$, the turn rate is $60°$ per $0.07[s]$ with no load. These are powerful digital servos with a stall torque equal to $2.4[\frac{kg}{cm}]$. The high specifications of the servomotors led to that no practical tests was conducted, due to low anticipated angle adjustment of the motors during flight.

2.2 Signal flow

On the TRT there are several components that interact with each other. The ATmega 2560 microcontroller is the centerpiece for the signal flow, as shown in figure 2.6. The ATmega receives accelerations and angle rates from the IMU and pilot commands from the receiver (Rx), then calculates signals needed to perform the given commands and sends it to the servos and ESC's. The ESC's then sends the converted signal to the motors.

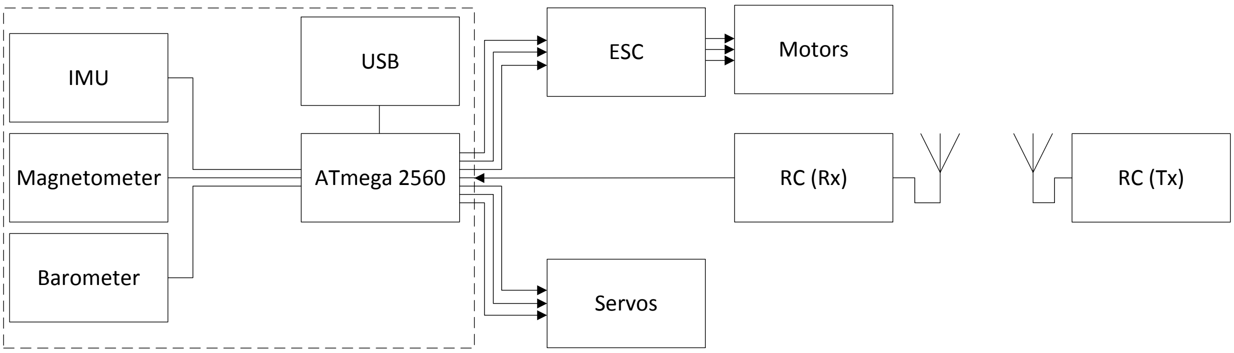

Figure 2.6: Signal flow overview

Flight controller

The TRT is equipped with the MultiWii Mega flight controller. It contains an Inertial Measurement Unit (IMU) with 3 axes accelerometer, 3 axes gyroscope and additionally it equipped with 3 axes magnetometer and 1 barometer. The microcontroller used on the Multiwii Mega flight controller is the ATmega 2560. This is a simple microcontroller that has 16MHz clock frequency and are badly suited to handle floating point arithmetics. It has limited computational resources which means that the algorithms has to be effetely coded and as limited as possible.

Sensors

The IMU used on the MultiWii Mega flight controller is a 6 DOF chip of type MPU6050. The accelerometer measures accelerations which the IMU is exposed to. Accelerometers are known to be affected by white Gaussian noise, which is a drawback when it comes to sensitive control systems. The Gyroscope measures the angular velocities of the IMU. It is possible to integrate the gyroscope data to get angle, however the angle tend to drift due to bias error in the signal [3]. Gyroscope are less affected by noise compared to the accelerometer.

The magnetometer is of type HMC5883L, that can measure the direction and the strength of the earth magnetic field. This gives the relative heading in relation to the earth magnetic field. As all other compasses, surrounding and the self produced magnetic fields will influence the accuracy of the measurements. This effect can be compensated for by utilizing sensor fusion technics.

The barometer of type MS5611 measures the atmospheric pressure and can therefore be used to keep track of the altitude. Since atmospheric pressure is influenced by temperature, wind, air humidity etc, its not an accurate sensor without being calibrated for the conditions the day of the flight. The optimal use of this sensor is the altitude hold function, where it only has to detect changes in the atmospheric pressure, rather then give out an numeric altitude.

Communication

A radio controller (RC) is used to send commands to the flight controller. The transmitter (Tx) and receiver (Rx) uses the 2.4 GHz frequency band, which is the most commonly used hobby RC frequency. The layout of the controller is shown in figure 2.7. It sends the pilot commands wireless to the onboard receiver, which is connected to the flight controller.

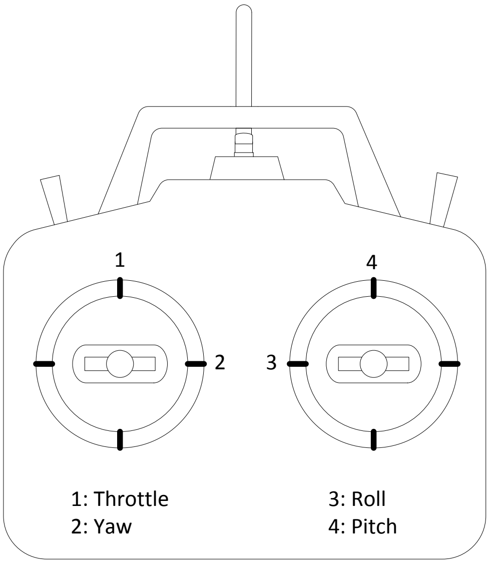

Figure 2.7: Radio controller layout

The pilot commands will consist of roll and pitch angles, yaw rate and a throttle gain. As shown in figure 2.7 the left stick controls the throttle and yaw, where the right stick controls the pitch and roll angles. By choosing to control pitch and roll angle instead of angle rate, allows that zero position on the right stick equal to a zero degree pitch/roll angle. It is not desirable that TRT will return to the zero yaw angle when releasing the controller, therefore the angular yaw rate has been selected as proper yaw control.

Chapter 3

Mathematical model

The mathematical model describing the physics of the tilt rotor tricopter was derived with the objective of creating both a simulation model, and an advanced control system based on its dynamics. For an object in a 3D environment, there are at least two different reference systems, local and global reference system. With forces acting on the body in different reference system, these need to be transformed to coexist in the same equations. The transformations matrices and reference systems will be explained in chapter 3.1. The mathematical model has been build up using the Newton-Euler approach, which describes the translational and rotational dynamics of a body in a 3D environment. It combines the forces and torques acting on the COG of the TRT to a nonlinear set of equations.

A perfect replication of a physical system can never be achieved by a mathematical model [5]. There will always be some simplifications and/or different phenomenon that can not be modeled accurately. By including what has the most dominant effects on the system, and excluding the peripherals, the mathematical model can be used to estimate the TRTs behaviour under certain circumstances. In order to reach the objective, several assumptions were made to avoid making the system unnecessarily complex. An evaluation of the tricopters behaviour and its physical properties led to the following assumptions:

- The structure of the propellers and body is considered rigid.
- Air is considered static and without turbulence.
- Attitude angle does not exceed 90°
- Thrust model based on force vectors.
- Ground is used as the inertial reference frame.

3.1 Reference frames

In order to derive the mathematical model for the TRT, reference frames have to be established. This system have several moving parts, which affect behavior of the TRT in the air. The figure 3.1 shows the simplified structure of the tilt rotor tricopter with actuated parts of the system. The main coordinate system for the TRT is placed in the COG, also referred as body frame and axis are denoted X_b, Y_b and Z_b. The body frame describes the forces and moments acting on the TRT. To track the attitude and position of the TRT in the room, an external coordinate system is used which is referred to as the navigation frame with axes X_n, Y_n and Z_n. A local coordinate system is established for each of the motors. Forces F_{1-3} which are induced due to the rotating propellers are illustrated with single arrows and corresponding induced torques with double arrows. Each motor can be actuated separately and tilted in respect to the main body structure with an angle α_{1-3} as denoted in the figure.

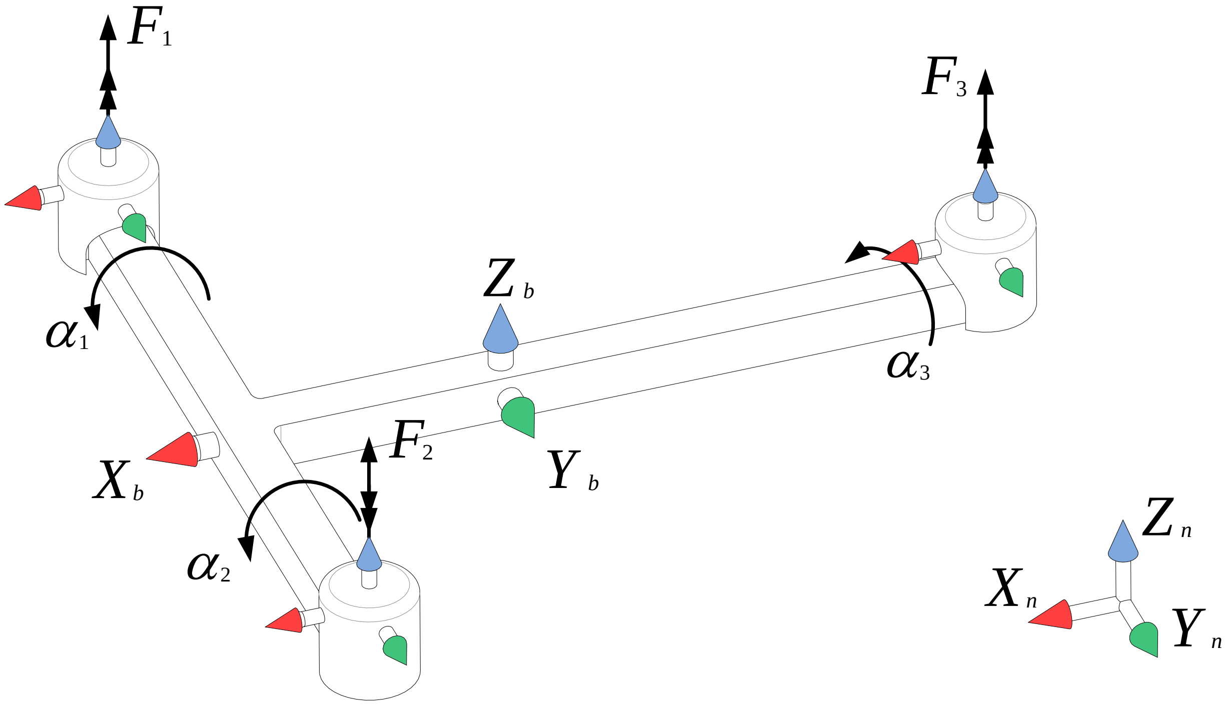

Figure 3.1: Reference frames of the TRT

The origin for the body frame in placed in the COG. This gives the opportunity to utilize the Newton-Euler approach to describe the dynamics of the TRT. Due to the variable orientation of the forces acting on the COG, transformation of those is needed in order to represent them in the body coordinate system.

Coordinate system transformation

To describe the orientation of a coordinate system in relation to the other, rotational angles around each axis have to be known. Given the initial alignment of coordinate systems show in the figure 3.1, transformation of a vector from the body frame to the navigation frame is shown in the figure 3.2.

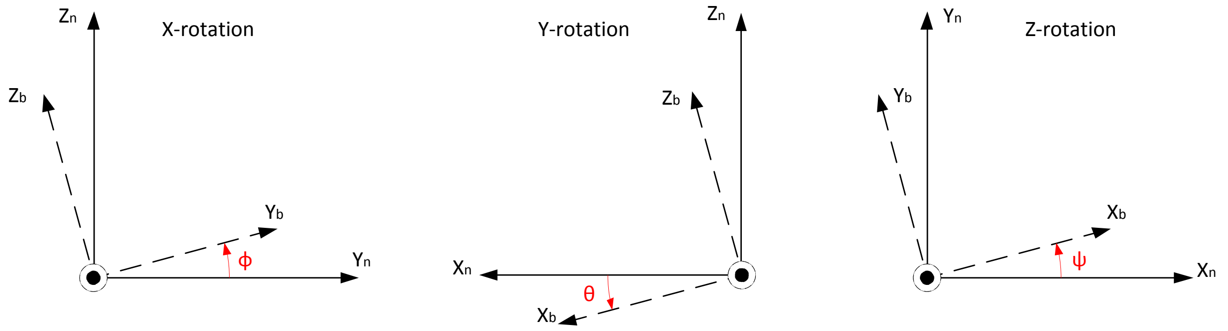

Figure 3.2: Rotating transformation

The applied method of describing the orientation of the rigid body in space is developed by Peter G. Tait and George H. Bryan [10] and is a variation of the Euler's rotation theorem. By consequently rotating around the updated axis in the sequence of X-Y-Z, orientation of the rigid body can be described. The rotational matrices for each axis are given by equation 3.1. This equation describes transformation of the vector from the body frame to the navigation frame one axes at the time with angles denoted to be: ϕ - roll, θ - pitch, ψ- yaw, which are also referred to as Euler angles. Due to similar orientation of other coordinate frames in the system, same formula can be applied for transformation between the motor and the body frame.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}; R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}; R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Total deviation in the attitude between two coordinate systems can be described by equation 3.2. The matrix C_b^n is also referred to as Directional Cosine Matrix (DCM) and in the particular case of this system, transforms any vector from body frame to navigation frame. Inverse of the DCM matrix, denoted as C_n^b performs the transformation from the navigation frame to the body frame [10].

$$C_b^n = R_x(\phi)R_y(\theta)R_z(\psi) = (C_n^b)^{-1}$$

$$C_b^n = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\theta)\sin(\psi) & \sin(\theta) \\ \sin(\theta)\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\psi) & -\sin(\theta)\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi) & -\cos(\theta)\sin(\phi) \\ -\sin(\theta)\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\psi) & \sin(\theta)\cos(\phi)\sin(\psi) + \sin(\phi)\cos(\psi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{3.2}$$

Angular rates

Angular rates of the rigid body in the navigation coordinate system can be described using the Euler angles, which are defined previously. Due to properties of 3D kinematics, the angular rates of the body coordinate system can not be directly integrated to gain the values of the attitude angles [10]. Equation 3.3 describes transformation of the angular rates in body frame (p, q, r) to angular rates in the navigation frame $(\dot{\phi}, \dot{\theta}, \dot{\psi})$, also called Euler rates.

$$\begin{gathered}\begin{bmatrix} p \\ q \\ r \end{bmatrix} = (R_x(\phi)R_y(\theta)R_z(\psi))^T \cdot \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + (R_y(\theta)R_z(\psi))^T \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\psi) & 0 \\ -\cos(\theta)\sin(\psi) & \cos(\psi) & 0 \\ \sin(\theta) & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}\end{gathered} \tag{3.3}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T^{-1} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.4}$$

The Euler rates can be integrated to obtain the attitude of the TRT, however some specific orientation of the TRT will result in the singular matrix T. The Tait-Bryan formulation dictates the limitation of the workspace so that the range of the angles are: $\phi = \pm 180°$, $\theta = \pm 90°$ and $\psi = \pm 180°$. This limitation is considered to be acceptable since TRT will not be able to achieve flight in such orientations due to constructional and mechanical properties.

3.2 Newton-Euler dynamics

The mathematical model of the TRT have been derived using Newton-Euler approach, combining the translational and rotational dynamics. This approach describes the motion of the COG of the TRT and is based on the force and moments equilibrium. The equations are presented in the matrix form, which simplifies transformation of the forces and moments to the same reference frame. The main challenge for deriving of the mathematical model is to identify which parameters influence the dynamics of the TRT.

Equation 3.5 describes the force equilibrium for the COG of the TRT. The system is affected by both internal and external forces, which all are transformed to the body frame. The motor thrust can be directly described in the body frame with a simple rotational matrix, while gravity is defined in the navigational frame and have to be multiplied by DCM matrix.

$$\begin{gathered} m_{body} \cdot \ddot{X} = F_{gravity} + F_{thrust} = F_{tot} \\ \begin{bmatrix} \ddot{X}_b \\ \ddot{Y}_b \\ \ddot{Z}_b \end{bmatrix} = C_n^b \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m_{body}} \left(R_y(\alpha_1) \cdot \begin{bmatrix} 0 \\ 0 \\ F_1 \end{bmatrix} + R_y(\alpha_2) \cdot \begin{bmatrix} 0 \\ 0 \\ F_2 \end{bmatrix} + R_x(\alpha_3) \cdot \begin{bmatrix} 0 \\ 0 \\ F_3 \end{bmatrix} \right) \end{gathered} \tag{3.5}$$

A fixed-motor multicopter will always have thrust along the Z-axis of the body frame, where on the TRT motors can be tilted. The tilt angles are accounted for by multiplying the thrust vector with a rotational matrix, resulting in a force vector acting along multiple axis in the body coordinate system. The force vector F1 and F2 can rotate around the Y-axis, as shown in figure 3.1, and therefore multiplied with the rotational matrix R_y, where F3 is multiplied with R_x due to rotation around the X-axis.

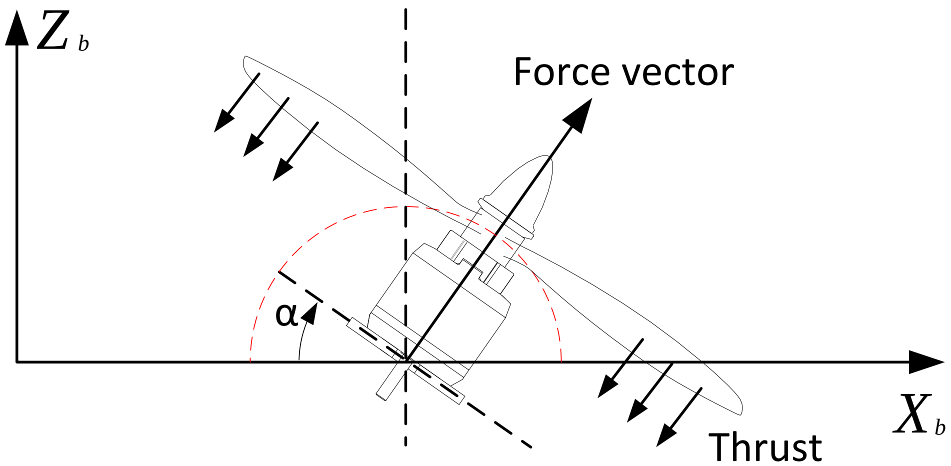

Figure 3.3: Force vector illustrated for front motors

Some simplifications has been done regarding the propeller thrust in this model. In the physical model, the thrust produced by the propeller rotates with an arc around the rotational joint of the motor, with a radius equal to the height of the motor. In the mathematical model, the forces vector rotates around a rotational joint of the motor without the radius as show in the figure 3.3. This has been considered to reduce the accuracy of the mathematical model to a minimum due to small displacement of the force origin.

The moment equilibrium is described in the equation 3.6 and is derived in the body coordinate system. The TRT is affected by several moments, originated by the different sources.

$$J_{body} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = M_{thrust} + M_{motor} - M_{gyro} = M_{tot} \tag{3.6}$$

Each thrust vector have a displacement from COG, therefore inducing a torque around the different axis of the body frame. The displacement vectors are measured from the center of the propeller to the COG of the TRT. The axis, around which M_{thrust} creates torque, varies as a result of the motor tilt. To be able to calculate the torque in the body frame, the rotational matrixes have been included to account for this effect. The torque contribution is shown in equation 3.7.

$$M_{thrust} = \begin{bmatrix} L_{1x} \\ L_{1y} \\ L_{1z} \end{bmatrix} \times \left[R_y(\alpha_1) \cdot \begin{bmatrix} 0 \\ 0 \\ F_1 \end{bmatrix} \right] + \begin{bmatrix} L_{2x} \\ L_{2y} \\ L_{2z} \end{bmatrix} \times \left[R_y(\alpha_2) \cdot \begin{bmatrix} 0 \\ 0 \\ F_2 \end{bmatrix} \right] + \begin{bmatrix} L_{3x} \\ L_{3y} \\ L_{3z} \end{bmatrix} \times \left[R_x(\alpha_3) \cdot \begin{bmatrix} 0 \\ 0 \\ F_3 \end{bmatrix} \right] \tag{3.7}$$

The thrust is produced by the propellers which are mounted to brushless DC motors. Spinning motors induce torque that acts in the opposite direction of the motor rotation, as described in the equation 3.8. The direction of the torque M_{motor} will also be affected by motor tilt, therefore the rotation matrixes has been included to adjust for it. The torque constant K_t is specific for the given type of motors with propellers.

$$M_{motor} = K_t \left(R_y(\alpha_1) \cdot \begin{bmatrix} 0 \\ 0 \\ F_1 \end{bmatrix} + R_y(\alpha_2) \cdot \begin{bmatrix} 0 \\ 0 \\ -F_2 \end{bmatrix} + R_x(\alpha_3) \cdot \begin{bmatrix} 0 \\ 0 \\ F_3 \end{bmatrix} \right) \tag{3.8}$$

The TRT will change its direction and attitude in the air, which will cause the angular velocity vector to change its direction, inducing the torque referred to as M_{gyro}. This therm is a part of the general form of moment equilibrium for the rigid body motion. Equation 3.9 describes the induced torque as a function of the angular rate and the inertial matrix of the body.

$$M_{gyro} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times J_{body} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.9}$$

Dynamics of the TRT can be defined by the nonlinear function h(x,u) shown in equation 3.10. Parameters describing the pose of the TRT are the attitude angles and position, measured in respect to the navigational coordinate system. Since the navigational frame is assumed to be the inertial reference frame, the force equilibrium can be represented in the navigational frame by simple transformation using the C_b^n. The attitude angles can be estimated by using the Euler rates.

$$h(x,u) = \begin{bmatrix} \ddot{X}_n \\ \ddot{Y}_n \\ \ddot{Z}_n \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} C_b^n \cdot \begin{bmatrix} \ddot{X}_b \\ \ddot{Y}_b \\ \ddot{Z}_b \end{bmatrix} \\ T^{-1} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ J_{body}^{-1} \cdot M_{tot} \end{bmatrix} \tag{3.10}$$

3.3 Excluded influences

Drag

When an object moves through the air, the relative velocity between the object and the air creates a force in the opposite direction. The force depends on the air density ρ, the velocity relative to the air v, a drag coefficient C and the reference area A. This has not been implemented in the mathematical model due to advanced aerodynamics with the turbulence created by the propeller [11].

$$F_{drag} = \frac{1}{2} \cdot \rho \cdot \begin{bmatrix} A_x C_x & 0 & 0 \\ 0 & A_y C_y & 0 \\ 0 & 0 & A_z C_z \end{bmatrix} \cdot \begin{bmatrix} v_x^2 \\ v_y^2 \\ v_z^2 \end{bmatrix} \tag{3.11}$$

Atmospheric parameters

Atmospheric influences has not been included in the calculations due to the complexity of the governing equations required to calculate these data accurately. Parameters such as air temperature, pressure and humidity actively influence the lift produced by the propeller, but has been considered static for simplicity. Wind conditions will cause the TRT to drift, but as the TRT has no position feedback, there is no way to correct this properly. Because the TRT will be tested in an indoor environment, these atmospheric parameters will have a minimal effect on the system, and are therefore negligible.

Gyroscope effect of propellers and motor tilt

When an object rotates, it induces a gyroscopic effect. Since the TRT can tilt its rotors, two different cases of changing the rotor axis angle are investigated. The equations 3.12 and 3.13 describe the gyroscopic effect of the rotation of the rotor, where M_1 the attitude change of the TRT and M_2 is tilt of the motor assembly. Note that Jm and Jms describes two different inertia moments due to different axis of rotation.

The torque M_1 is induced, when TRT rotates with locked motor angle.

$$M_1 = \begin{bmatrix} Jm_x & 0 & 0 \\ 0 & Jm_y & 0 \\ 0 & 0 & Jm_z \end{bmatrix} \times \left(R_y(\alpha_1) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix} \right) \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} Jm_x & 0 & 0 \\ 0 & Jm_y & 0 \\ 0 & 0 & Jm_z \end{bmatrix} \times \left(R_y(\alpha_2) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_2 \end{bmatrix} \right) \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} Jm_x & 0 & 0 \\ 0 & Jm_y & 0 \\ 0 & 0 & Jm_z \end{bmatrix} \times \left(R_x(\alpha_3) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_3 \end{bmatrix} \right) \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.12}$$

The torque M_2 is induced when changing motor angles at the constant attitude of the TRT.

$$M_2 = \begin{bmatrix} Jms_x & 0 & 0 \\ 0 & Jms_y & 0 \\ 0 & 0 & Jms_z \end{bmatrix} \times \left(R_y(\alpha_1) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix} \right) \cdot \begin{bmatrix} 0 \\ \dot{\alpha_1} \\ 0 \end{bmatrix} + \begin{bmatrix} Jms_x & 0 & 0 \\ 0 & Jms_y & 0 \\ 0 & 0 & Jms_z \end{bmatrix} \times \left(R_y(\alpha_2) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_2 \end{bmatrix} \right) \cdot \begin{bmatrix} 0 \\ \dot{\alpha_2} \\ 0 \end{bmatrix} + \begin{bmatrix} Jms_x & 0 & 0 \\ 0 & Jms_y & 0 \\ 0 & 0 & Jms_z \end{bmatrix} \times \left(R_x(\alpha_3) \cdot \begin{bmatrix} 0 \\ 0 \\ \omega_3 \end{bmatrix} \right) \cdot \begin{bmatrix} \dot{\alpha_3} \\ 0 \\ 0 \end{bmatrix} \tag{3.13}$$

The change of the motor angle is actuated by the servo, accelerating a mass and inducing the torque M_3.

$$M_3 = \begin{bmatrix} Jms_x & 0 & 0 \\ 0 & Jms_y & 0 \\ 0 & 0 & Jms_z \end{bmatrix} \cdot \begin{bmatrix} \ddot{\alpha_1} \\ \ddot{\alpha_2} \\ \ddot{\alpha_3} \end{bmatrix} \tag{3.14}$$

As the TRT is not designed for acrobatic flying, it will not be exposed to quick changes both of servo angles and attitude. Since M_1, M_2 and M_3 requires high rates of change to noticeable influence the system, these have not been included in the simulations and mathematical equations.

Chapter 4

Modeling and simulation

4.1 SolidWorks model

One crucial part of simulating a physical system is to include an accurate model containing the same geometrical parameters as the physical system. Being a commercial product the Talon Tricopter V1.0 has a 3D CAD model available to public [2]. Since the TRT has several different component and slightly modified construction, the CAD model had to be revised. The weight of all parts was measured and added into the CAD model, including the battery, wires and peripheral equipment. From this 3D model the inertia matrix and the COG of the system has been extracted, thereby allowing the simulations to be as accurate as possible.

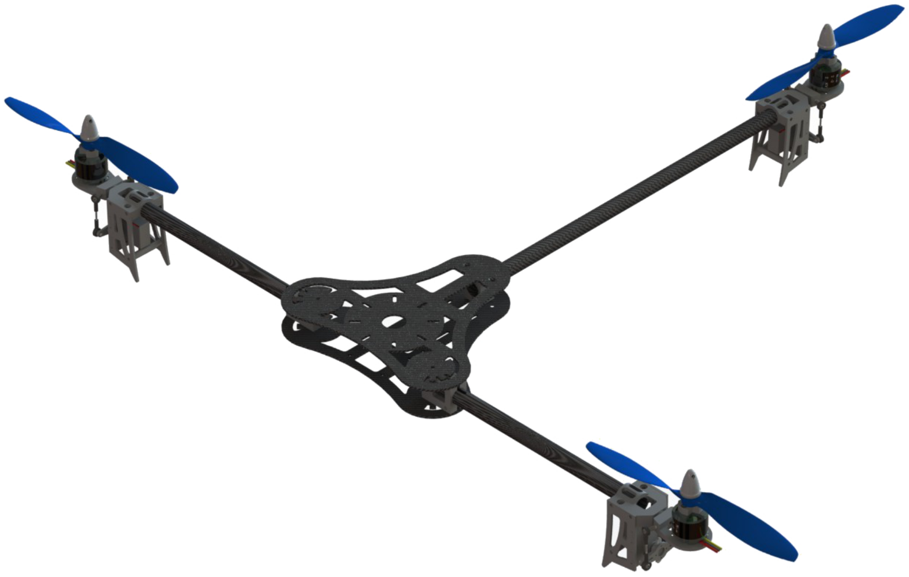

Figure 4.1: SolidWorks 3D CAD model

Figure 4.1 shows the CAD model of the TRT. The peripheral equipment was removed from the figure, due to a cleaner illustration of the model.

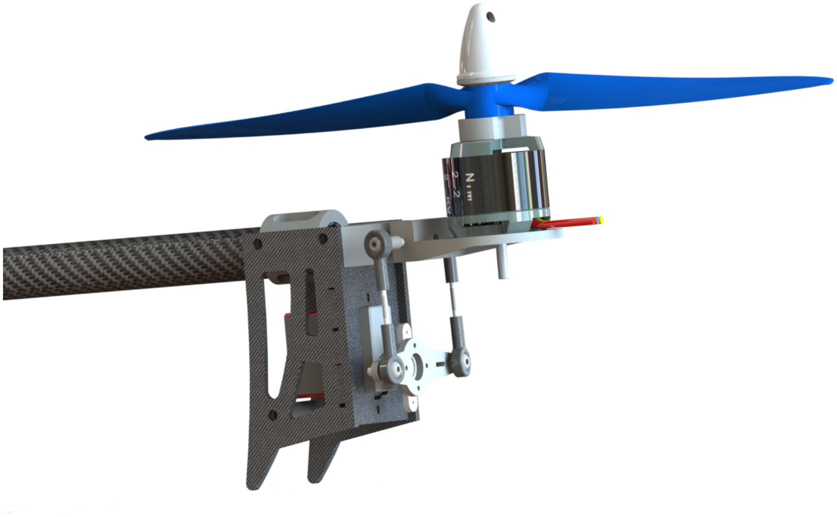

Figure 4.2: SolidWorks 3D CAD model

Figure 4.2 shows a CAD model of the mechanical design for the motor tilt function. The servo is attached to the motor mount with a pushrod connection. The orientation of the upper connection point limits the tilt arc to the 50° to each side. Such mechanism adds a lot of weight to the tip of each arm, increasing the inertia of the model, which is a downside of this prefabricated frame.

Table 4.1 shows the physical parameters extracted from the CAD model. The inertial matrix of the system has been simplified to a diagonal matrix. These parameters are used in the mathematical equations and the simulation model.

Parameter	Value	Unit
Total mass		
m_{body}	1.22	[kg]
Inertia		
J_{xx}	0.0310	$[kgm^2]$
J_{yy}	0.0325	$[kgm^2]$
J_{zz}	0.0626	$[kgm^2]$
COG to Motor1		
L_{1x}	0.1063	$[m]$
L_{1y}	-0.3403	$[m]$
L_{1z}	-0.0070	$[m]$
COG to Motor2		
L_{2x}	0.1063	$[m]$
L_{2y}	0.3403	$[m]$
L_{2z}	-0.0070	$[m]$
COG to Motor3		
L_{3x}	-0.4342	$[m]$
L_{3y}	0.0000	$[m]$
L_{3z}	-0.0070	$[m]$

Table 4.1: Physical parameters extracted from CAD model

4.2 SimulationX model

The SimulationX model has been build using Multi Body System (MBS) components and shown in the figure 4.3. All of the components are positioned using the dimensions from the SolidWorks model, giving the simulation model correct geometrical properties. To make the response of the simulation model as close to the physical model as possible, the inertia and weight of the CAD model was applied. This means that all components have neglectable mass ($1 \cdot 10^{-11}$ kg), except the small sphere placed at COG containing the inertia matrix and weight of the system.

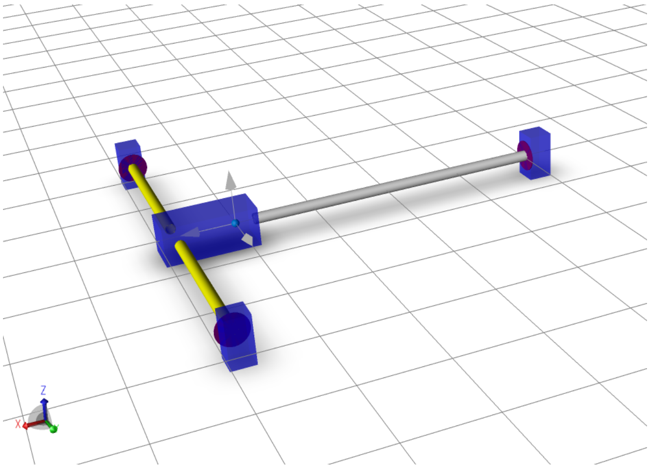

Figure 4.3: 3D view of the SimulationX model

The main goal of the simulation model is the verify the Newton-Euler approach, which has been build up using force and angles. Presets has been used as servos to drive the rotational joints which have infinite torque and power. The servo and motor dynamics can be implemented at a later stage if a more complete mathematical model is desired. The lift is simulated by a force vector and the torque caused by the rotating propeller is included with a torque vector. The simulation model is based on the same assumptions as the mathematical model described in Chapter 3.

The challenge of using SimulationX for this method, is that SimulationX does not give out the equations it uses for simulations. This means that is it difficult to identify all the forces and influences SimulationX take into consideration under simulations. In the early phase of co-simulation, the difference between the two models where a major concern. This led to a process where the SimulationX model and the mathematical model was step by step increased in complexity. By performing this stepwise process, it was possible to confirm that the mathematical model, contained the same forces and effects as the SimulationX model. By exposing the SimulationX model to high rates of roll, pitch and yaw, it could be concluded that the gyroscopic effect of the body plays a major role in the model. It is often seen that in 2D dynamics the gyroscopic effect is neglected, where in 3D it is a crucial part of the body dynamics.

4.3 Verification of the equations

To verify the mathematical equations, the model built in SimulationX has been exported to Matlab Simulink. By comparing the behavior of each model with the same parameters, it is possible to verify that the equations are correct. The tests are performed with no control system in place and the input signals are tuned to provoke motion in all 6 DOF. The flight test begins in hovering state, then increases velocity in the all three directions. To track the behavior of both models, a 3D visualization has been made, which is shown in the figure 4.4.

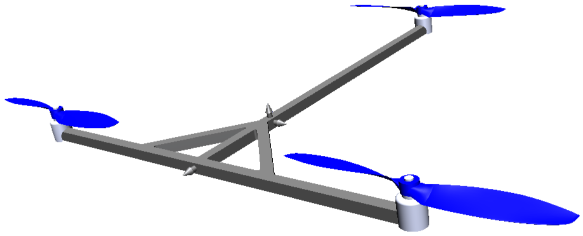

Figure 4.4: 3D view of the Simulink model

The input parameters was constant throughout the test of 5[s], resulting in the following flight path shown in figure 4.5. The resulting pose are discussed in more details by reviewing the attitude and position separatly.

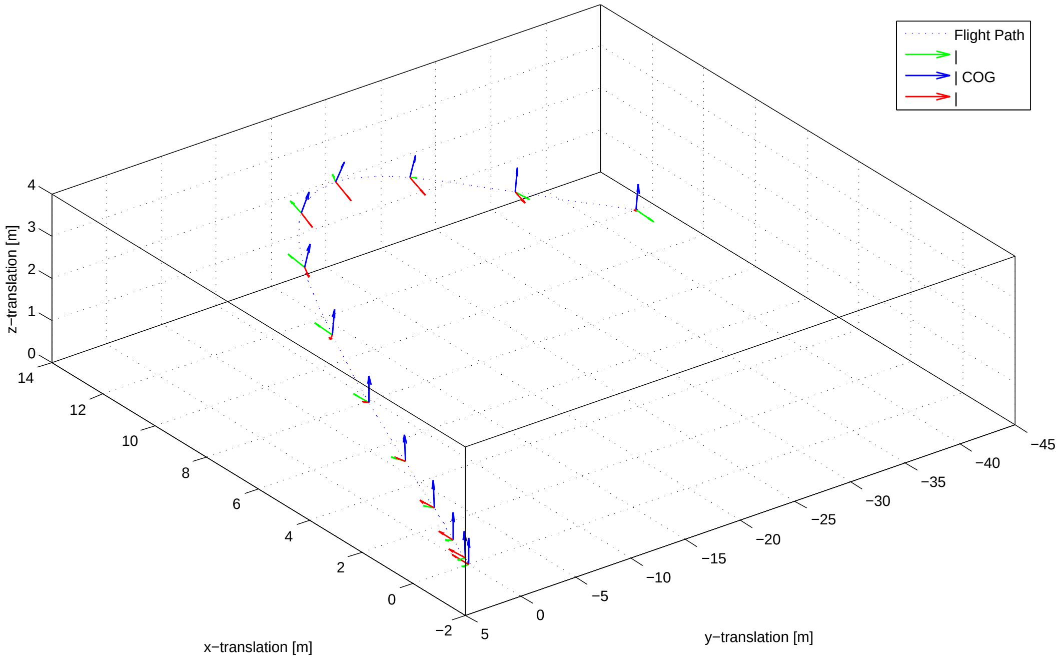

Figure 4.5: 3D plot of the flight path

Attitude

The figure 4.6 shows the attitude response of both models from the verification test. It can be seen that the mathematical equations almost identically reproduce the SimulationX model results. Such similar results indicates that the mathematical equation correctly describes the torque equilibrium, as derived in the equation 3.6.

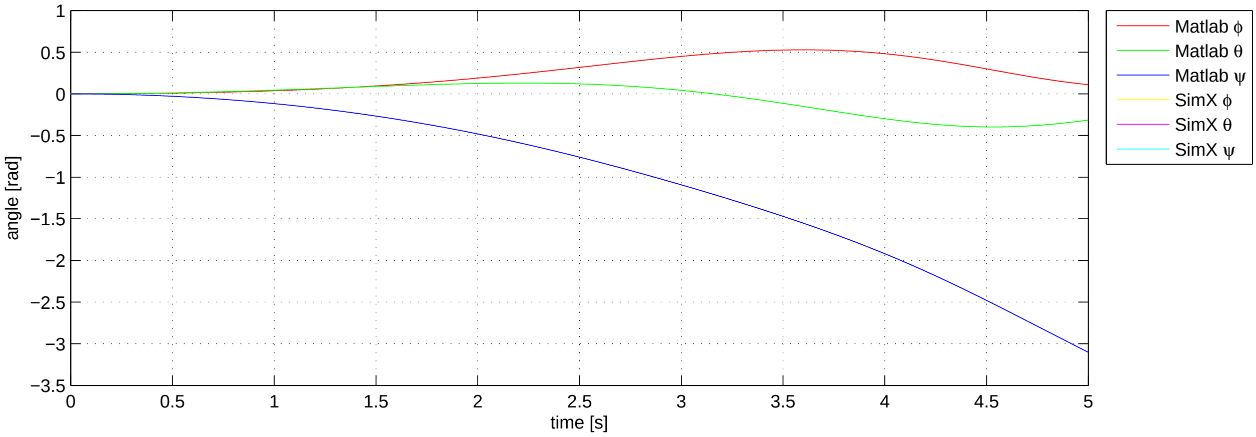

Figure 4.6: Attitude from the verification test

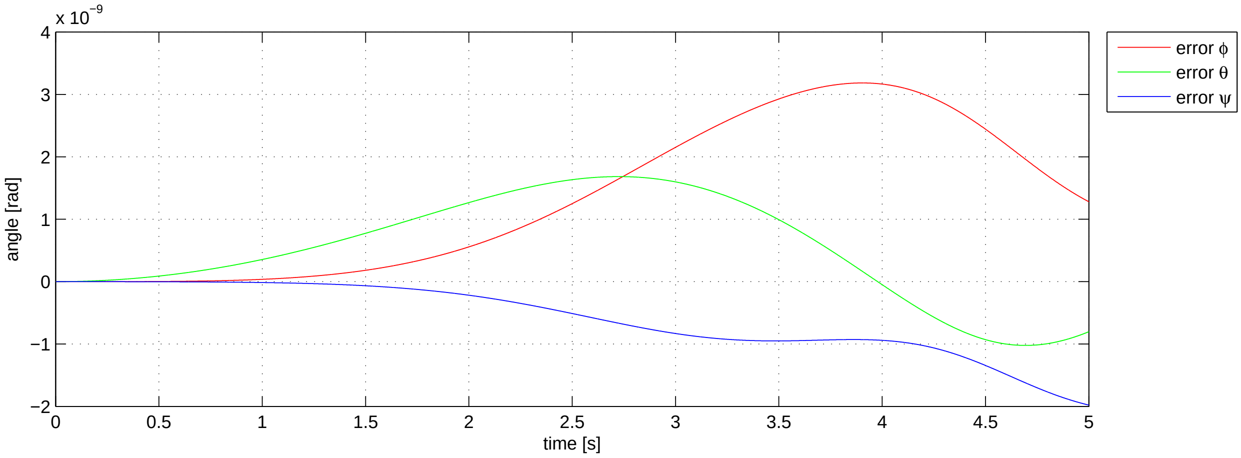

Figure 4.7: Attitude error between the models

The attitude error between the models, which is shown in the figure 4.7, reaches the maximum value of $3.2E^{-9}$[rad]. This deviation in the results can be caused by numerical approximation. Due to the small order of the error value, it can safely be neglected.

Position

As figure 4.8 shows, the mathematical equations and the model from SimulationX are behaving near identical. It is worth noticing that monitoring the position is only possible in the simulation, but it can be used to verify the force equilibrium derived in the equation 3.5.

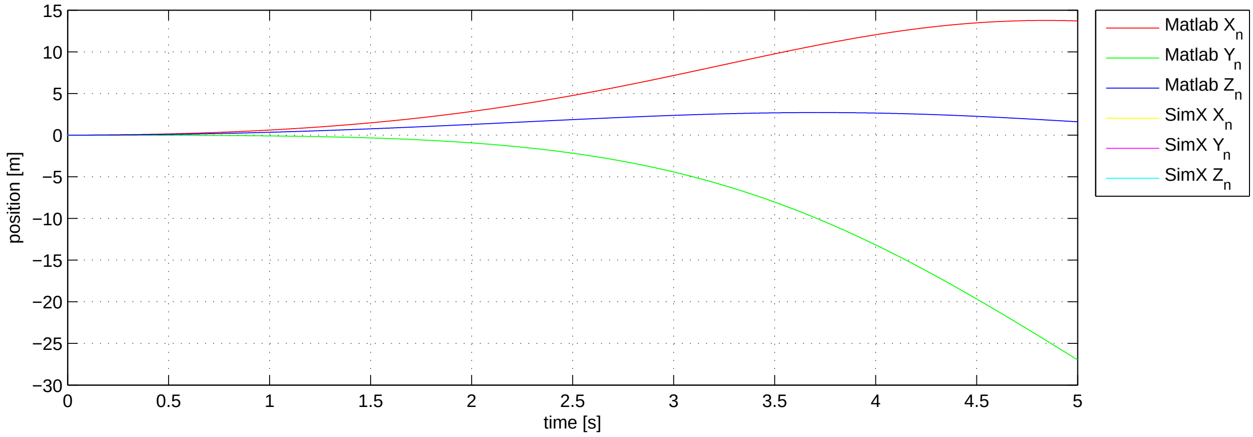

Figure 4.8: Position from the verification test

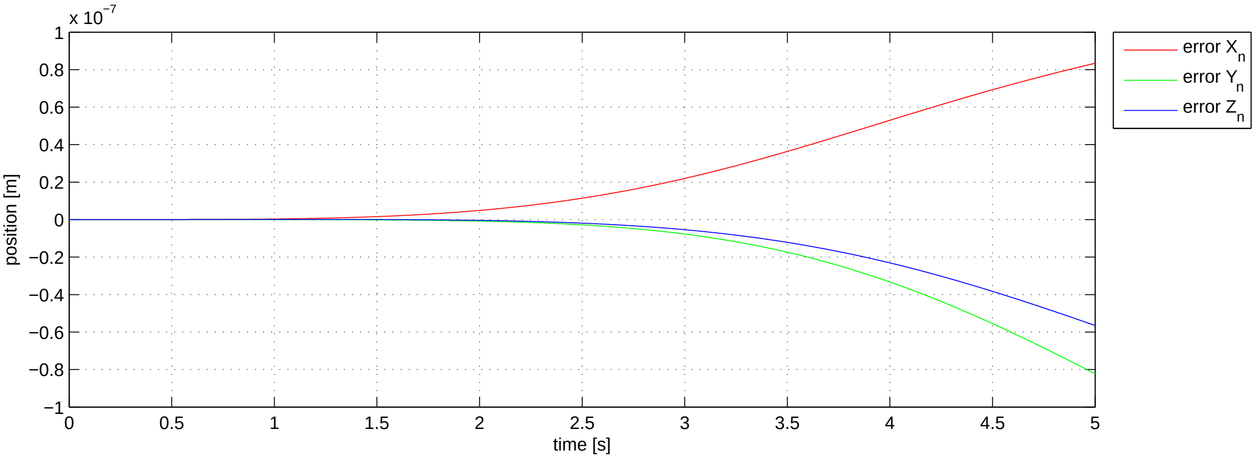

Figure 4.9: Position error between the models

The positional error between the models is greater than for the attitude, but considering the distance traveled of 30[m] during the simulation of 5[s], the maximum error of $0.8E^{-7}$[m] is insignificant. Due to the lack of positional feedback on the physical system, the attitude angles are the parameters which control system will be based on.

Chapter 5

Control system

The goal of this thesis is to design a control system capable of maintaining a hovering state and performing controllable low speed flight in an indoor environment. The design strategy chosen for the control system is based on utilizing the dynamic model in a state-space environment. The structure and components of the system are described in the Chapter 2 and gives necessary information about the mechanical aspect of the system as well as overview over available sensors. The dynamics of the tilt rotor tricopter are described in the Chapter 3 provides information about the in air behavior of the system.

However, due to the highly nonlinear nature of the mathematical model, the control system will be designed based on a linearized model [6] around the hovering state of the TRT. A Linear Quadratic Regulator (LQR)[12] is well suited for a Multi Input Multi Output (MIMO) system, which is the case with the TRT model. Usage of the LQR guarantees stability of the system, which is achieved by minimizing the quadratic optimization problem and feedback from the sensors. The control system is developed using Matlab Simulink software package. The complete control system for the linearized model is also verified by applying it to the nonlinear system, which ensures adequate behavior of the TRT.

5.1 Linearized physical system

Since the TRT will not be used for hight speed aerial maneuvers and limited only to the hovering and low speed flight, the mathematical model can be simplified accordingly. The linear model of the TRT is modeled as the state-space structure with multiple input and output signals. At the hovering state, the system have to maintain the predefined position and orientation in the room. This implies that the model has to be linearized around the equilibrium point (x_0, u_0), so that the first derivative of the state variables are equal to zero.

In order to linearize the general nonlinear function, it is typical to apply the first order Taylor Series expansion [4], as shown in the equation 5.1. To implement this method on the current system, the initial step is to identify the input and state variables of the linear system, which are describing the TRT's behavior around the equilibrium point.

$$h(x,u) \approx h(x_0,u_0) + \left.\frac{\partial h(x,u)}{\partial x}\right|_{(x_0,u_0)} (x - x_0) + \left.\frac{\partial h(x,u)}{\partial u}\right|_{(x_0,u_0)} (u - u_0) \tag{5.1}$$

The input to the system is composed of the three forces $F_{1..3}$ induced by the rotating propeller, followed by three servo angles $\alpha_{1..3}$ which control the direction of the force vector. The outcome of the input vector becomes $u = [F_1, F_2, F_3, \alpha_1, \alpha_2, \alpha_3]^T$. The state variables of the system are selected to consist of Euler angles (ϕ, θ, ψ) and angular rotational rates of the body coordinate system (p, q, r). Such set of output parameters should be sufficient to stabilize the attitude of the TRT, leaving the positional control to the operator. The resulting state vector is given in form of : $x = [\phi, \theta, \psi, p, q, r]^T$, which have available feedback signal.

Linearized physical model of TRT in state-space structure will result in A,B and C matrices, where elements of the A and B matrices are given by $a_{i,j}, b_{i,j}$ correspondingly, as shown by equation 5.2.

$$a_{i,j} = \left.\frac{\partial h_i(x,u)}{\partial x_j}\right|_{(x_0,u_0)}, b_{i,j} = \left.\frac{\partial h_i(x,u)}{\partial u_j}\right|_{(x_0,u_0)} \tag{5.2}$$

The A,B and C matrices are derived for the chosen variables in the state vector. The detailed description of nonzero elements of matrices are available in the Appendix A.

$$A = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & 0 \\ 0 & 0 & a_{2,3} & a_{2,4} & a_{2,5} & 0 \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} \\ 0 & 0 & 0 & 0 & a_{4,5} & a_{4,6} \\ 0 & 0 & 0 & a_{5,4} & 0 & a_{5,6} \\ 0 & 0 & 0 & a_{6,4} & a_{6,5} & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} \\ b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} \end{bmatrix} \tag{5.3}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally, the structure of the linearized state-space system can be described by the equation 5.4

$$\begin{aligned} \dot{x}(t) &= A(x(t) - x_0) + B(u(t) - u_0) \\ y(t) &= C(x(t) - x_0) \end{aligned} \tag{5.4}$$

where:

$$\begin{aligned} x &= [\phi, \theta, \psi, p, q, r]^T \\ u &= [F_1, F_2, F_3, \alpha_1, \alpha_2, \alpha_3]^T \end{aligned}$$

The derived state-space model of the TRT is in a general form and can be applied to any equilibrium point of choice. The physical system is capable of hovering in the air with different attitude parameters. This means that there are infinite number of possible equilibrium states within the limitations of DCM and physical constrains, which are described in Chapter 3.

5.2 Initial parameter estimation

To get the numerical values for the A,B and C matrices, the system equations have to be evaluated at the chosen equilibrium point with input parameter such that first derivative of the state variables are zero. The initial input vector u_0 has to be computed for the given linearization point x_0 to fulfill the force and moment equilibrium. Due to nonlinearity of the system, the values for the input vector are calculated by using the Matlab *fsolve()* function. This function allows to calculate the value of u_0 by using the optimization algorithm, so that $h(x_0, u_0) \to 0$.

One of the most convenient equilibrium point that can be chosen for hovering, is the zero-attitude point. This point can be described by the state variable vector as $x_0 = [0, 0, 0, 0, 0, 0]^T$, where the Euler angles as well as angular velocities around the body frame axis are zero. It is worth noticing that in such hovering state the TRT will remain level at any given position $[X_n, Y_n, Z_n]^T$ in the room, since it is not bounded by the state vector. By accurately identifying the input parameters for the equilibrium point, it is possible to increase the precision and performance of the control system. The correct parameters may also decrease the positional drifting of the TRT.

Two configurations of the input vector u_0 are reviewed to achieve the zero-attitude point. Since both parameter sets satisfies the criterion for an equilibrium point, the input vector is selected based on the ability of holding the predefined pose and suitability for the TRT construction. The simulation is conducted in Matlab Simulink without implemented control system, which also indicates how unstable the TRT model is.

The first one is the classical tricopter configuration, where the two propellers in front are aligned perpendicularly to the body while the rear one is angled to compensate for yaw rotation. The estimated values of the input vector is $u_1 = [4.8063, 4.8054, 2.3527, 0, 0, -0.018]^T$. As it can be seen from the figure 5.1, the TRT is able to hold its position fairly well, deviating only by 0.53[m] from the predefined point over the period of 5 seconds.

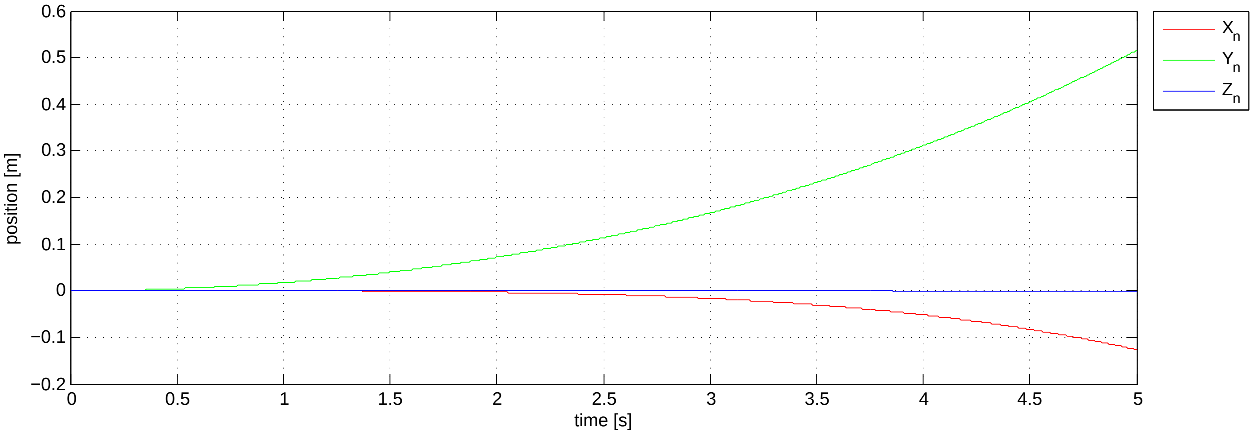

Figure 5.1: Translation with classical parameters

The second configuration is adapted to the TRT design, exploiting the additional DOF of the actuators. By holding the rear motor perpendicular to the body and tilting the front motors, the yaw rotation can be stabilized. The estimated value of the input vector is $u_2 = [4.8050, 4.8063, 2.3530, -0.0056, 0.0056, 0]^T$. Using this set of input parameters, TRT holds predefined position more accurate with deviation of 0.2[m] as shown in figure 5.2.

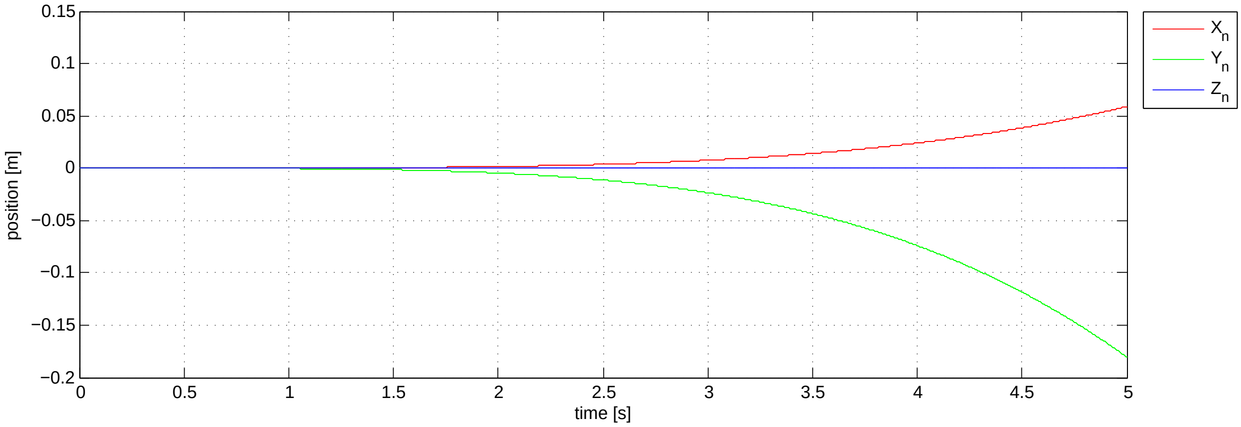

Figure 5.2: Translation with TRT parameters

An additional advantage of using input parameters u_2 is in transition from hovering to forward flight without tilting the body of TRT. Such transition can be done simply by tilting the front motors simultaneously forward, while with the classical setup the angle of all three motors have to be adjusted. Considering the advantages of the parameter set u_2, it was chosen to be used in the linearized model of the physical system. The matrices A,B and C can now be numerically estimated, with given values of x_0 and u_0. The detailed description of numerical values for matrices are available in the Appendix B.

5.3 Linear Quadratic Regulator

The controller implemented in this thesis is based on modern control theory, which uses a dynamical model of the systems to compute the optimal gain. The type of controller is a Linear Quadratic Regulator, and derived from the linearized state-space model of the physical system. This type of a regulator is well suited for MIMO systems and guaranties stability, while the performance is dependent on the choice of the controller parameters. To recap the derived state-space model of the TRT, linearized for hovering, parameters are as follows:

$$\begin{aligned}\dot{x}(t) &= A(x(t) - x_0) + B(u(t) - u_0)\\ y(t) &= C(x(t) - x_0)\end{aligned}$$

where:

$$\begin{aligned}x &= [\phi, \theta, \psi, p, q, r]^T & x_0 &= [0, 0, 0, 0, 0, 0]^T\\ u &= [F_1, F_2, F_3, \alpha_1, \alpha_2, \alpha_3]^T & u_0 &= [4.8050, 4.8063, 2.3530, -0.0056, 0.0056, 0]^T\end{aligned}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & a_{1,4} & 0 & 0\\ 0 & 0 & 0 & 0 & a_{2,5} & 0\\ 0 & 0 & 0 & 0 & 0 & a_{3,6}\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6}\\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6}\\ b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} \end{bmatrix};$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

At this point the linear model is identified and numerically estimated(see Apendix A and B). This gives the possibility to investigate the dynamical characteristics of the system matrix A, as well as observability and controllability of the plant. Controllability is a critical property of the plant, which enables the internal state of the physical system to be changed in its entirety by applying the the external input to the system. Equation 5.5 describes the criteria for a system to be classified as fully controllable.

$$\begin{gathered}\text{Rank}(C_m) = \text{dim}(x);\\ C_m = [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B]\end{gathered} \tag{5.5}$$

The observability of the system is an indication on the determinability of the state vector, based on the external input to the system. A fully observable plant enables to measure all states, giving the control system the full state feedback. Equation 5.6 describes the criteria for a system to be classified as fully observable.

$$\begin{gathered}\text{Rank}(O_m) = \text{dim}(x);\\ O_m = [C \quad CA \quad CA^2 \quad CA^3 \quad CA^4 \quad CA^5]^T\end{gathered} \tag{5.6}$$

Applied for the current model of the TRT the Rank(C_m)=Rank(O_m)=6, which is equivalent to the dimension of the state vector x. This means that the linear model of the TRT is fully controllable and fully observable.

After the process of linearizing the TRT model around the (x_0, u_0) point, system matrix A has lost a lot of the elements describing the dynamics at high rotational rate. Considering the boundaries of the intended flight mode, such limitation is acceptable. However, at high speed maneuvers the control system will no longer be optimal, resulting in deviation from predicted behavior. The overview of the control strategy is presented in figure 5.3, where the gain K_1 is the contribution from the LQR.

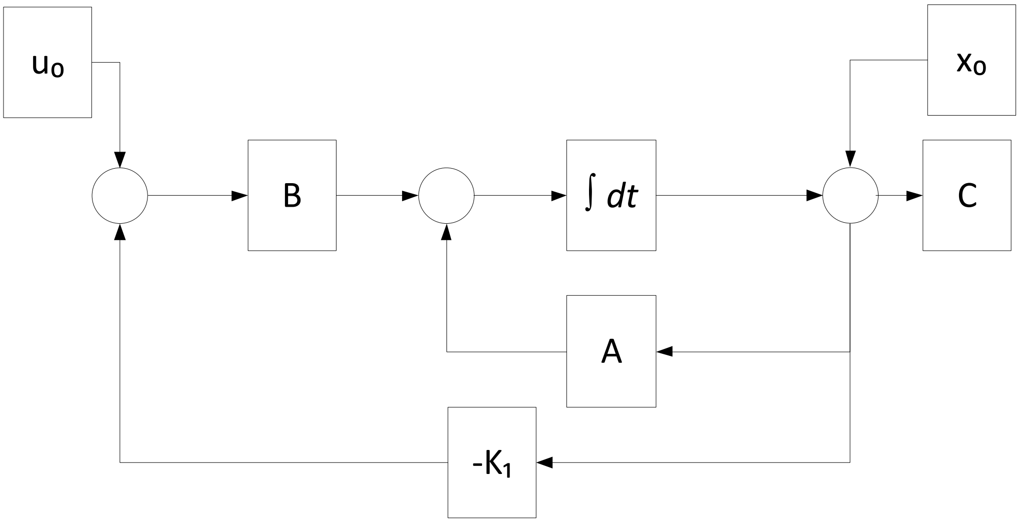

Figure 5.3: LQR control loop

As it was mentioned earlier, LQR is based on the optimization of the cost function, balancing the change of the input and state vector. Governing equations for estimating the gain, assuming the infinite-horizon continuous-time system, K_1 are given as such:

$$\begin{gathered} \tilde{x} = x - x_0 \qquad \tilde{u} = u - u_0 \\ J_{LQR} = \int_0^\infty (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u}) dt \\ u = -K_1 \tilde{x} \\ K_1 = R^{-1} B^T S \\ SA + A^T S - SBR^{-1} B^T S + Q = 0 \end{gathered} \tag{5.7}$$

The J_{LQR} is the quadratic cost function based on the total energy of the system. The matrix Q penalizes the change of the energy state of the controlled output, while the matrix R corresponds to the change in the energy state of the control signal. Minimizing total energy of the system, gain K_1 can be obtained resulting in the optimal input vector u. The constant gain K_1 can be found by solving the algebraic Riccati equation and calculating the positive-semidefinite matrix S.

The first choice of values for the cost function parameters are calculated by using the Bryson's rule [7]. For the diagonal matrices Q and R weight of the elements are given by equation 5.8.

$$\begin{gathered} Q_{ii} = \frac{1}{\text{maximum acceptable value of } x_i^2} \\ R_{ii} = \frac{1}{\text{maximum acceptable value of } u_j^2} \end{gathered} \tag{5.8}$$

Although Bryson's rule do not provide the optimal value of the parameters, it is a good starting point for the iterative process of tuning the parameters to gain the best performance. One of the effects of the diagonal layout of Q and R matrices, is a discrete penalization of the state and input vector. Non-diagonal setup will increase the performance of the system, but is very complicate to estimate.

5.4 Nullspace of the system matrix

Earlier in this chapter it was stated that the TRT is able to hover in air with different attitude orientations. This is caused by the introduction of additional DOF for the system, making it possible to independently move and rotate in all six DOF of the navigational frame. It was also stated that it is desirable to maintain the zero-attitude orientation while hovering, which is given by the state vector $x_0 = [0, 0, 0, 0, 0, 0]^T$. The linear model of TRT does not have an unique input-output relation due to $Ax = 0$ containing a non-trivial solution. By investigating the system matrix A it is possible to determined the nullspace of the system.

$$Ax = \phi \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \theta \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \psi \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + p \cdot \begin{bmatrix} a_{1,4} \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + q \cdot \begin{bmatrix} 0 \\ a_{2,5} \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + r \cdot \begin{bmatrix} 0 \\ 0 \\ a_{3,6} \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.9}$$

The equation 5.9 shows how the value of the three first states (ϕ, θ, ψ) do not effect the output of the system. To be able to maintain the desired orientation of the TRT, the control system has to be modified accordingly and give the necessary constrain to the physical system.

5.5 LQR with integral action

To constrain the additional DOF of the system and be able to control the attitude of the TRT, existing LQR is supplied with integral action feedback. The figure 5.4 shows setup for the modified control loop.

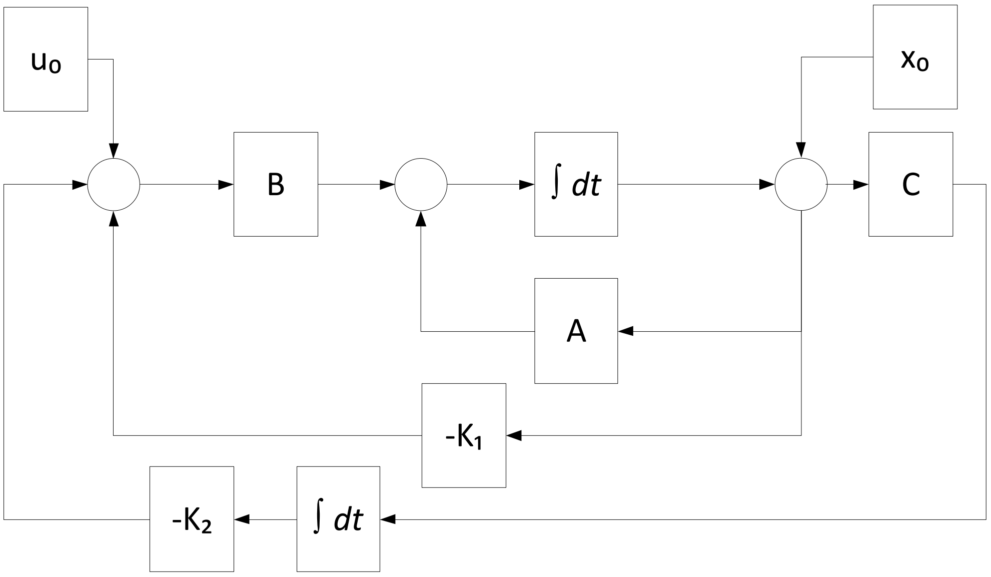

Figure 5.4: LQR with integrating action

Purpose of the integral action is to track the total deviation of the attitude states (ϕ, θ, ψ) and force it to zero. The augmented system [12] becomes:

$$\begin{aligned} \dot{x}'(t) &= A'x'(t) + B'u(t) \\ y(t) &= C'x'(t) \\ \text{where:} & \\ x' &= [z^T, x^T]^T \\ u &= [K_2, K_1]x' \end{aligned} \tag{5.10}$$

$$A' = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} \qquad B' = \begin{bmatrix} 0 \\ B \end{bmatrix} \qquad C' = \begin{bmatrix} 0 & C \end{bmatrix}$$

The method for calculating the controller gain remains the same, as it was previously described. Since C is an identity matrix, this will create integral action on all states in the vector x. The only two state that require the integral action are ϕ and θ, which means that the matrix C can be replaced with one that selects only the required states.

5.6 Setpoint control

Once stability of the TRT is ensured, the next step is to provide the pilot with means of controlling the TRT in the air. Figure 5.5 describes integration of the transmitter signal into the control loop.

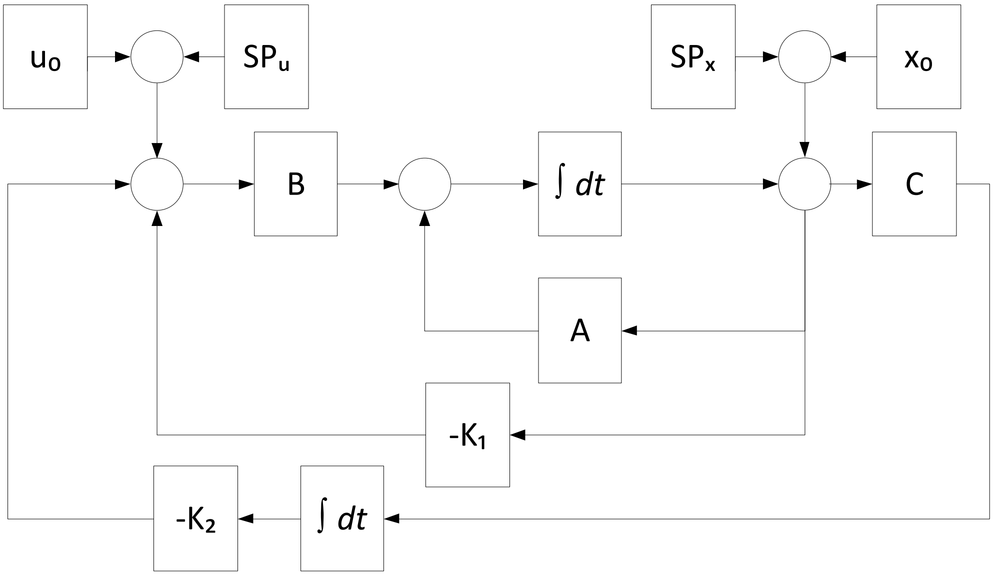

Figure 5.5: LQR with integrating action and setpoint control

The value of the SP_x modifies the attitude offset, resulting in motion of the TRT by tilting the body. The SP_u controls the total thrust of the system and feedforwards the angle of the motors to achieve motion of the TRT without tilting it.

The figure 5.6 shows how the signal from the transmitter is interacting with the control system parameters to achieve the desired behavior. The pilot is provided with 6 controls, one for each degree of freedom. The SP_x signal consists of 3 channels; *Ctrl.Roll*, *Ctrl.Pitch* and *Ctrl.Yaw rate*. By subtracting the value of SP_x from the original vector x_0, offset from the initial value will be created, provoking the system to react. The positive value of the SP_x will result in the positive angle change.

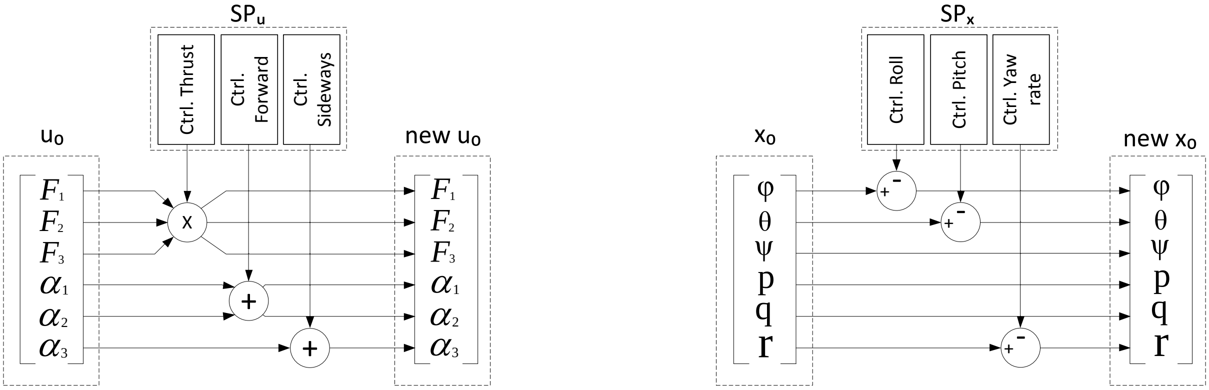

Figure 5.6: Setpoint control

The SP_u signal consists of 3 channels as well; *Ctrl.Thrust*, *Ctrl.Forward* and *Ctrl.Sideways*. The *Ctrl.Thrust* controls the total trust produced by the TRT, accelerating in up and down. The *Ctrl.Forward* and *Ctrl.Sideways* manipulates the motor angle, resulting in the longitudinal and transverse translation without change in the attitude. These two signals rely on the LQR to maintain the attitude values. Alternatively, the inverse kinematics can be used to improve the feedforward signal.

5.7 Altitude control

The control system is designed to work around the zero-attitude. By flying around and performing maneuvers in the air, the TRT will deviate from the zero-attitude point and will start to lose altitude. It is typical that the pilot control the throttle manually, but this can also be done with a simple altitude controller shown in the figure 5.7. This controller was tested in the simulation model, but is not implemented in the physical system.

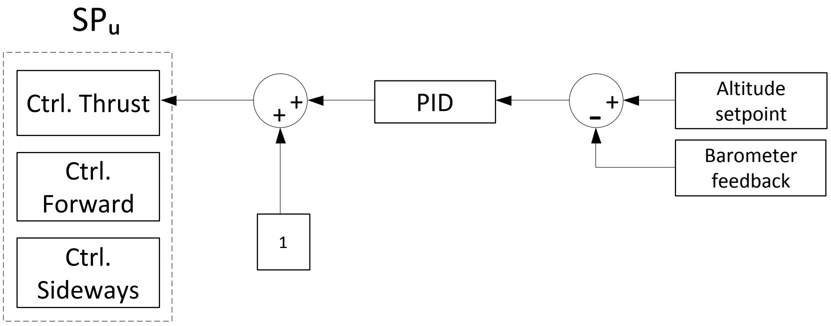

Figure 5.7: Altitude control loop

5.8 Controller parameter tuning

The cost function parameters was estimated by means of the simulation model. As a starting point the Bryson's rule was applied. In order to improve response of the system, parameters were manually adjusted to follow a predefined reference signal. Figure 5.8 shows the pitch response of the TRT to the reference signal, using the final controller parameters. Calibration of the roll response is done in the same manner, producing very similar results, therefore not included in this paper.

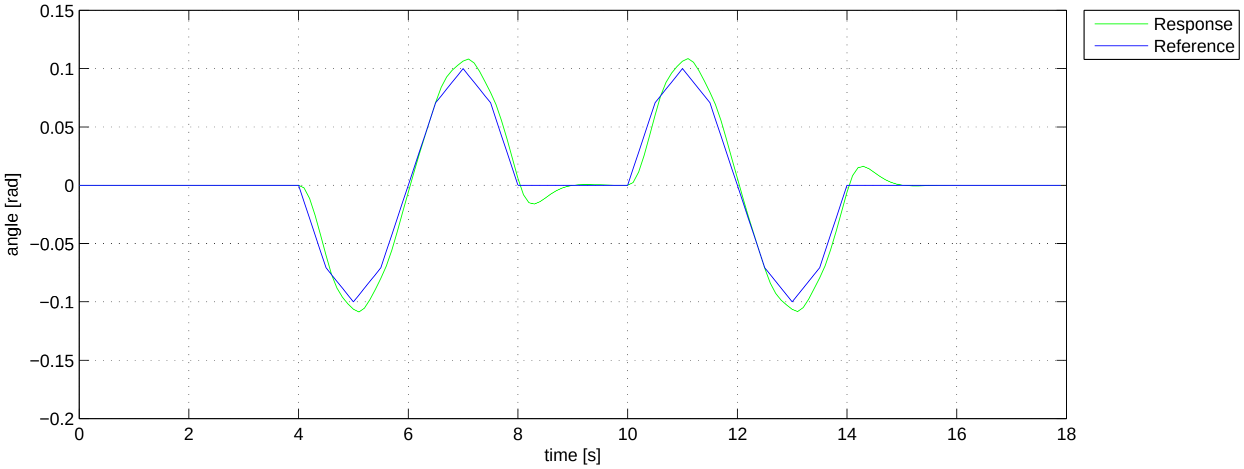

Figure 5.8: TRT pitch response to sinusoidal input

The reference signal consists of two periods of the 9-point sampled sinus wave, split by 2 second pause. The reference signal has an amplitude of 0.1 radians and a period of 4 seconds, which was assumed to be close to the pilot input in real life situation. For the augmented system with state vector $x = [\int\phi, \int\theta, \phi, \theta, \psi, p, q, r]^T$ and input vector $u = [F_1, F_2, F_3, \alpha_1, \alpha_2, \alpha_3]^T$, the cost function parameters are estimated to be as follows :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.08 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.08 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.08 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 \end{bmatrix} \quad R = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \tag{5.11}$$

Chapter 6

Results

All of the theoretical aspects of this thesis have been reviewed, from the system identification in Chapter 3, simulation model in Chapter 4 to the control system in Chapter 5. The closing part of this thesis will put the theoretical research and control system to the test by simulations and implementing it to the physical system.

To verify that the control system is ready for implementation on the physical model, simulation experiments are conducted on the nonlinear model. Section 6.1 and 6.2 presents the results of simulated translative flight path, using either change of attitude or motor tilt to realize the motion. The results show that the TRT is capable of moving in all 6 DOF, thereby the holonomic term can be accurately used to describe the TRT. The altitude controller is simulated separately, to verify that an altitude can be maintained in both scenarios.

Validation of the controller is conducted in section 6.5, by implementation on the assembled TRT platform. Because the controller verification is done by means of simulation, validation by the physical model also confirms the dynamic equations within the simulation environment. The assumptions made during development were taken into account, and a deliberate choice of flight test grounds was made to minimize the effects of the external factors. Two separate flight tests are conducted, testing roll and pitch response.

6.1 Motion by changing the attitude

Most multicopters control their translative motion by changing the attitude. The first simulation, shown in the figure 6.1, provokes motion of the TRT by the classical method. The setpoint for the attitude is manipulated by the SP_x to achieve the sinusoidal change in the attitude. The simulation is split up in the three sections. Part 1 is from 5[s] to 15[s], contains motion along the Y_n axis. Part 2 is from 20[s] to 30[s] and contains motion along the X_n axis. Part 3 is from 32[s] to 35[s] and contains yaw rotation.

Part 1

The TRT receives a sinusoidal signal, which initializes rotation in the negative roll direction. The roll angle reaches magnitude of the 0.8°, which leads to the movement of 1.8[m] by the TRT in positive y-direction. The control system initialize the roll maneuver by shifting the force balance between F_1 and F_2, leaving the servomotors unaffected. F_1 and F_2 are modulated by the control system to make sure that the TRT holds the given roll setpoint.

Part 2

The pitch signal, received by TRT, is similar to the roll signal but with higher amplitude. This signal controls the pitch angle to maximum value of 1.4°. The TRT moves 3.7[m] during this pitch maneuver, before it settles. To initialize the pitch angle, the control system gives the F_3 a boost and at the same time reducing both F_1 and F_2. All three motors work to keep the TRT stable with the given signal.

Part 3

The TRT receives a signal to change the heading of the body. The TRT rotates nearly 7° around the Z_n axis with a small overshoot. The control system tilt all three motors to control the yaw rotation, leaving the forces nearly unchanged.

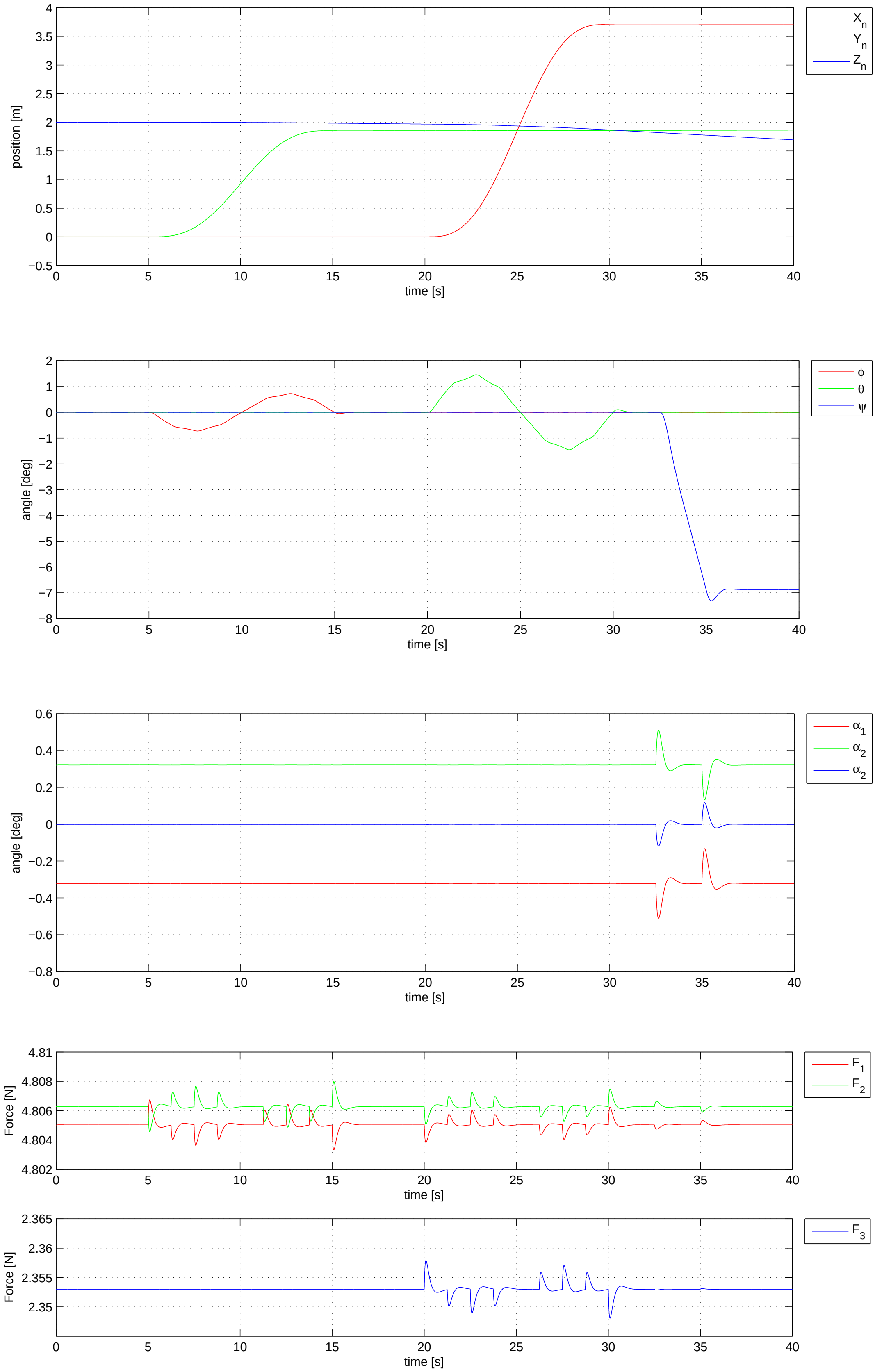

Figure 6.1: from top to the bottom: 1) Position 2) Attitude 3) Motor angle 4) Motor thrust

6.2 Motion by motor tilt

The TRT performs translative motion in the same manner as in section 6.1, but tilting the motors instead of the body. The figure 6.2 shows the simulation where the additional DOF are utilized. The signal is feedforwarded to the input setpoint SP_u, to control the motor angle instead of attitude angle in SP_x. The simulation is splited into two sections, where part 1 from 5[s] to 15[s], contains motion along the Y_n axis. Part 2 is from 20[s] to 30[s] and contains motion along the X_n axis.

Part 1

The sinusoidal signal is sent to the rear servomotor, which initializes motion of the TRT. This leads to movement of 1.2[m] in the positive Y_n direction, without the need to roll first. The control system prevent change in the attitude by compensating with both the motors and the motor angles. The two frontal motors hold the attitude orientation nearly constant, continuously adjusting for the shifting torque, and deviating only by 0.15° in yaw angle.

Part 2

The two frontal servos receives a sinusoidal signal input of the same magnitude. By tilting both motors forward, the TRT starts moving in the positive X_n direction, covering the distance of 3[m]. As in the first part of the simulation, the control system prevents attitude change during the movement. While performing the tilt action, the front motors adjusts the produced thrust, maintaining the constant roll angle. The yaw angle remains unchanged during the movement. The rear motor adjust only the thrust produced by the motor, keeping the angle at zero degrees tilt during the forward flight.

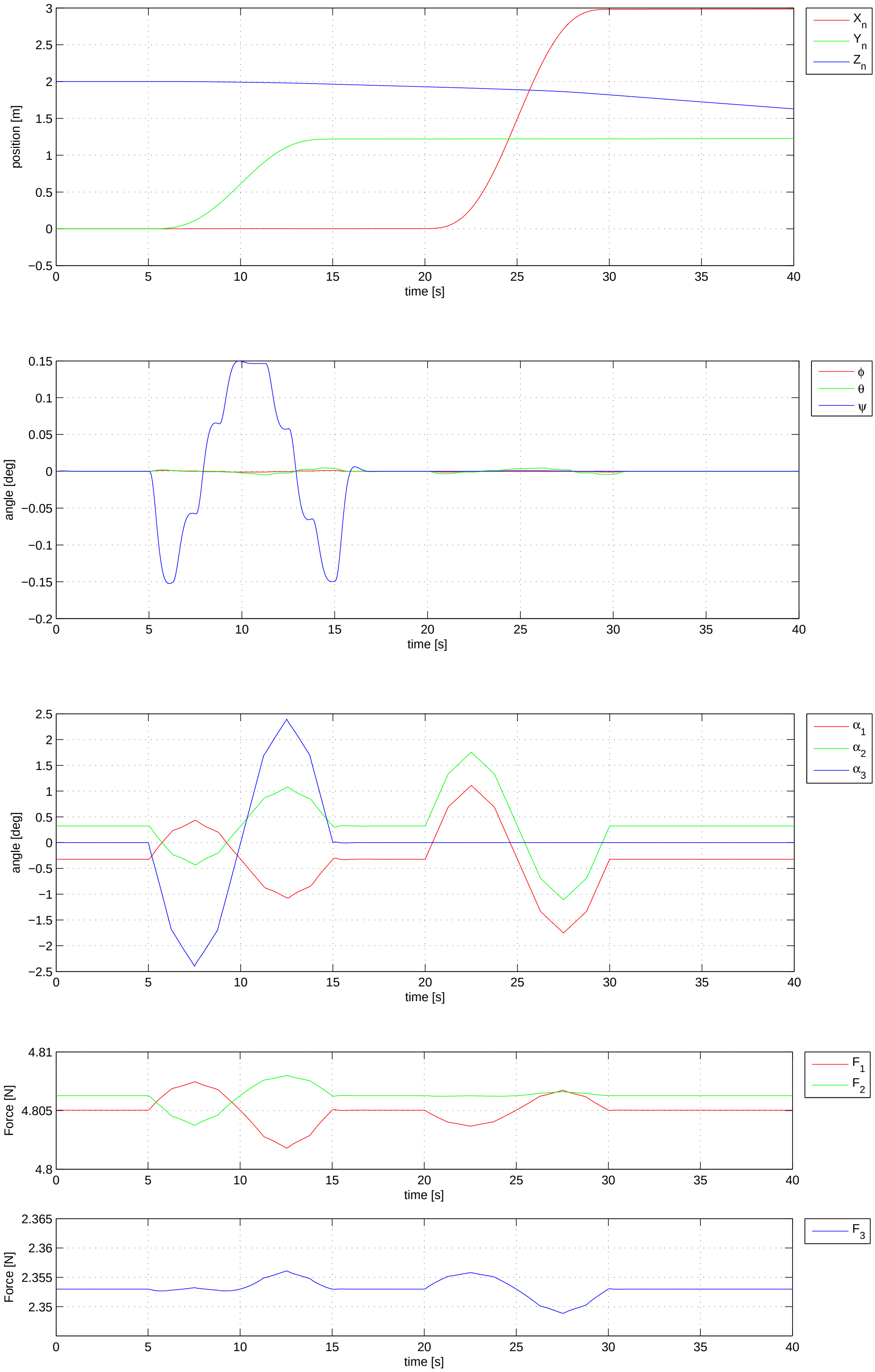

Figure 6.2: from top to the bottom: 1) Position 2) Attitude 3) Motor angle 4) Motor thrust

6.3 Altitude controller

The altitude controller was implemented in the simulation model to test the ability of keeping the altitude stable during motion of the TRT. The simulation is based on the motion due to motor tilt, which was previously described in the section 6.2, with added altitude control loop. The figure 6.3 shows the positional response of the model from the simulation. The altitude setpoint was initialized in the beginning of the simulation, which enables the the controller to hold the TRT at the stable height of 2[m].

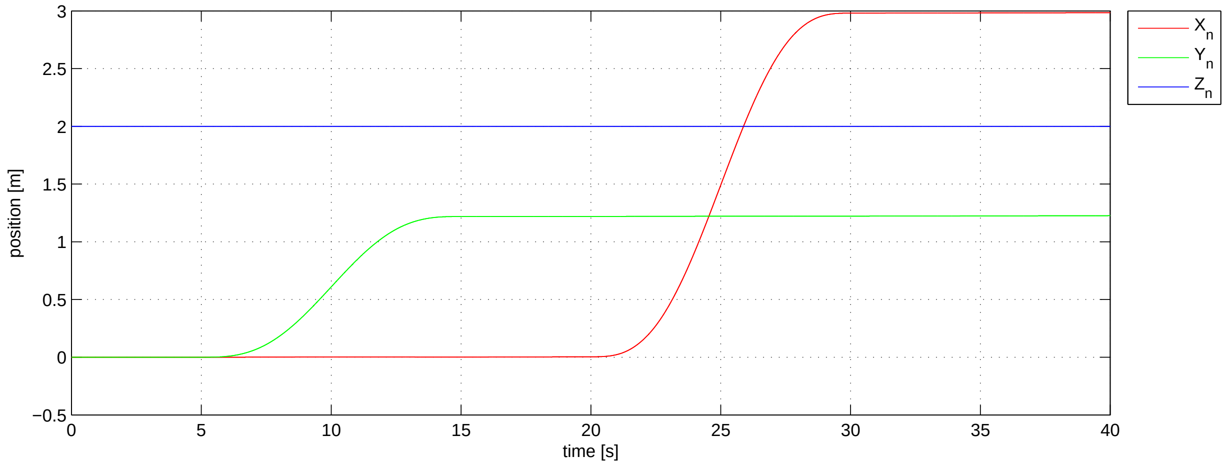

Figure 6.3: The altitude controller simulation results

6.4 Practical implementation

The flight controller board is equipped with an AVR ATmega2560 microcontroller which requires to be programmed with C-code. The basic program for initialization and processing sensor data was provided by Ø.Magnussen [1], leaving the implementation of the control system and signal output. The microcontroller have a single processing thread containing the whole code from data acquisition to signal output. The computational speed is highly dependent on the amount of code and number of the floating point operations. The first part of the program, containing the sensor management and providing the values for the attitude and angular rate of the TRT, is frequency dependent. This means that the rest of the code for control system, signal scaling, signal output and data logging have to be computationally undemanding, enabling the whole program to run at 300[Hz]. This was achieved by keeping the code compact and reducing the number of variables used in the program. Data logging was performed by wired communication using RS-232 protocol and was transmitted to the PC and processed in LabView software package at 150Hz. Both the wired communication with PC and limited processing power of the microcontroller made the data logging a challenging task, making it possible only for flights in an indoor environment. The code for main loop of the flight controller program is provided in the Appendix C.

The control system that is used on the physical model is implemented strait from the simulation model without modifications or further controller parameter tuning. Compared to the simulation model, the nonlinear function describing the dynamics of the TRT, is substituted with the real physical system. The figure 6.4 shows the overview of the control system implemented on the physical system.

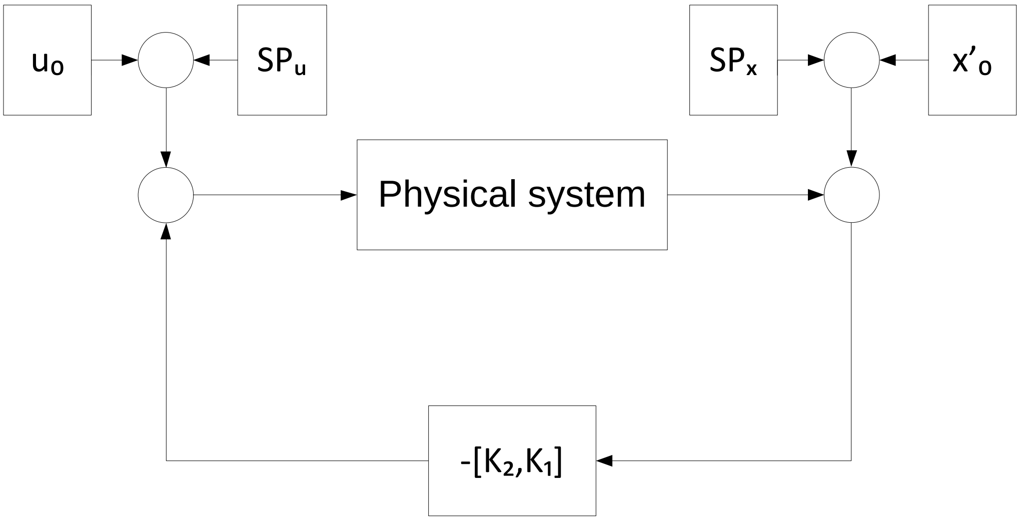

Figure 6.4: Implementation of control loop on the physical system

The program implemented on the flight controller consists of the same principle blocks as the simulation model. The input is based on the initial vector u_0 and the pilot commands, which provokes the movement of the TRT. The response of the system, caused by the input, is measured and deviation from attitude setpoint value is calculated. The deviation, also called the error signal, is multiplied with the constant gain producing the controller correction signal. This signal the added to the system input at the next time step and will result in smaller deviation from the attitude setpoint.

6.5 Flight test

The TRT was tested under several different conditions, both in narrow indoors environment and outdoors where the TRT was affected by wind. Under all test flights the TRT showed good stability while hovering and had an anticipated response to the input signals. The behavior of the TRT during the indoor flights were measured and logged to be analyzed later on.

Two separate tests were conducted to verify the stability and response of the TRT to pilot inputs. To minimize disturbance to the system caused by wind and propeller reflected turbulence, both tests were done in an indoor environment at the altitude of approximately 1 meter above the ground. Due to the wired logging of data, the testing area was limited, requiring the pilot to navigate to a predefined position before performing the test maneuver. Each test is divided into two parts, where the first part consists of the pilot controlling the TRT to hold a predefined position and the second part consists of applying the large input in periodical manner. The collected data from the tests compares the pilot input to the response of the TRT, measured by internal IMU.

The first test is presented in the figure 6.5, describing the roll response. The first 4 seconds in the figure 6.5 shows position holding of the TRT, where after the 4th second the pilot initialized positive and negative roll input.

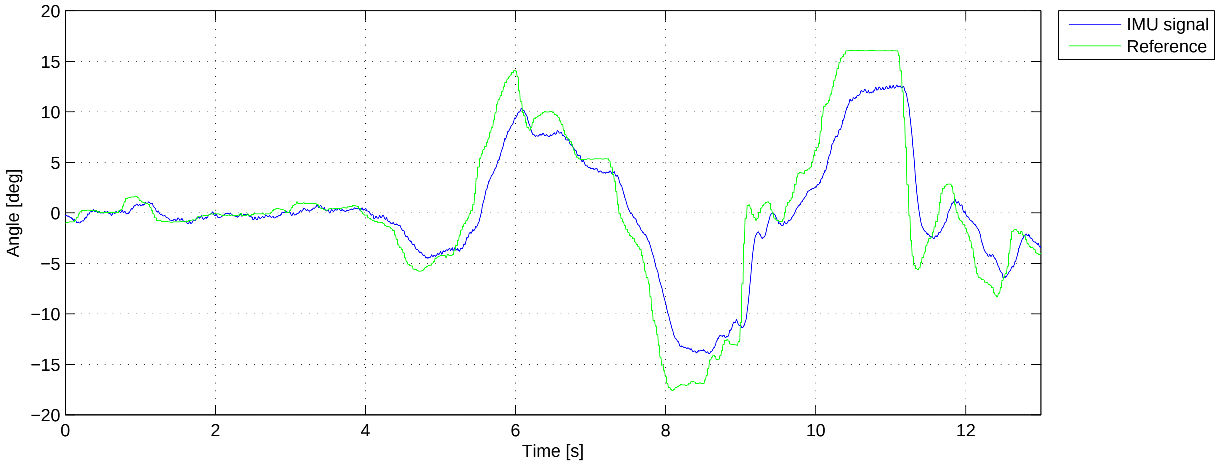

Figure 6.5: Free flight roll response

The second test is presented in the figure 6.6, describing the pitch response. The first 6 seconds corresponds to holding the predefined position, where after the 6th second, pilot initialized the positive and negative pitch input.

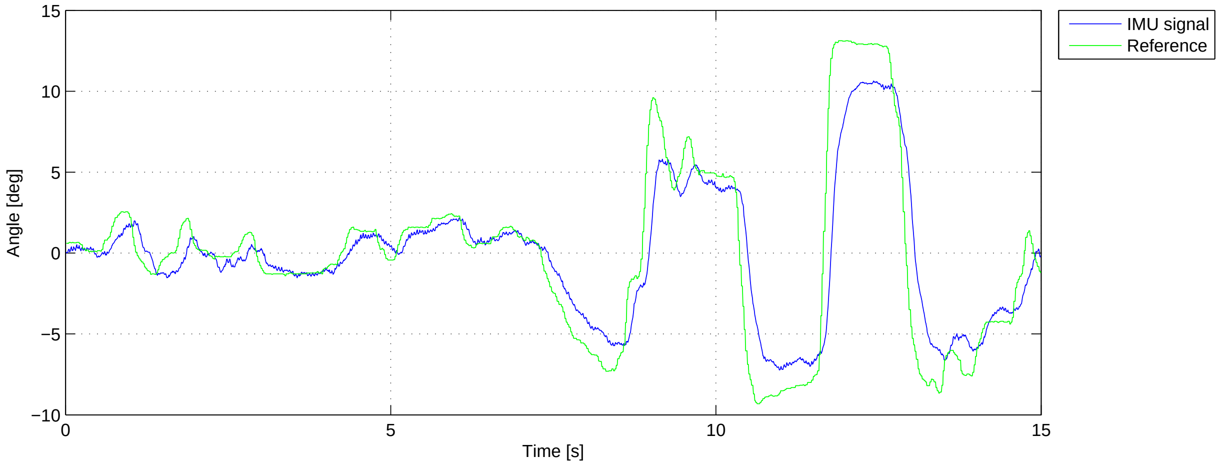

Figure 6.6: Free flight pitch response

By reviewing figure 6.5 and 6.6, it is possible to see that the TRT has a good and stable behavior in the air. As all flying object without positional feedback it experience drifting, which has to be corrected for by the pilot. Tests shows that in an indoor environment, drift in the longitudinal direction is greater than in the transversal direction. Tracking of the reference signal by the TRT is quite good considering the magnitude of the reference signal. The general shape of the reference signal is reproduced good, but some phase delay is visible. The control system was tuned to follow the sinusoidal signal of amplitude 0.1 and period of 4 seconds, but it is obvious that in real life TRT have to be able to follow signals with much greater gradient. To achieve a better performance of the system, the control system parameters should be revised.

Chapter 7

Conclusion

A dynamic model of the tilt rotor tricopter has been derived using the Newton-Euler approach. The TRT consisted of several reference frames, which needed different rotation matrixes to convert the forces to the main body. The dynamic model was verified using a SimulationX model. A complete 3D CAD model has been used to get geometrical and inertial data to the simulations. The simulations has been conducted in Matlab Simulink, where a 3D model was implemented to visualize the behavior of the simulation model.

A state-space representation of the dynamic equations has been presented. This has been linearized around hovering position to design a control system. The control system that has been developed is based on the linear quadratic regulator. An integral action has been implemented due to nullspace problems and steady state error.

By conducting the motor tests, the relationship between the thrust and rpm/PPM signals were recorded. This relationship was recreated by a third order polynomial equation, which has shown to be an accurate estimation. The motors and propellers generated enough thrust to hold the TRT in the air and be able to maneuver around.

To verify that the control system works and keeps the TRT stable, it has been implemented on a real model. Note that the parameters used on the real model are implemented straight from the simulation without changes. This shows that by thoroughly identifying the dynamic of the model, one can achieve a valid controller for a highly nonlinear cupeled system. Longitudinal and transverse translation has not been implemented on the real model but has been successfully tested in the simulations.

In the current configuration the TRT is a pilot operated UAV. The control system keeps the TRT stable but it needs a pilot to send attitude commands using a 2.4GHz RC controller. There is a GPS module available for the selected flight controller, which can allow for outdoor waypoint guidance.

The low cost flight controller Multiwii Mega has proven itself as a descent controller for this project, though it has some computational limitations. By keeping the online code short and effective, 300Hz of operating frequency was accomplished.

It is worth mentioning that the entire cost of the equipment used on the TRT included shipping and taxes, has a total of 3500NOK. This proves that further research in the field of multicopter and their control systems, are no longer only for the heavily founded research departments.

The TRT has conducted several free flight test verifying that the control system designed during this thesis is functional and is able to handle a highly nonlinear coupled system as the TRT is. As a result of the successful flights, it can be concluded that the simplifications regarding the dynamic model does not affect the stability noticeable under the given conditions. As the prestudy discovered no identical platforms to the TRT, we are proud to present a working new segment to the diversity that is the multicopter platform.

Bibliography

[1] *Øyvind Magnussen.* Phd student at University of Agder.

[2] Brad Adlam. http://grabcad.com/brad.adlam.

[3] Thomas Bräunl. *Embedded Robotics. Mobile Robot Design and Applications with Embedded Systems.* Springer, 2006.

[4] Zoran Gajic. *Linear dynamic systems and signals.* Prentice Hall, 2003.

[5] Finn Haugen. *Dynamiske System, model, analyse og simulering.* Tapir akademiske forlag, 2007.

[6] Finn Haugen. *Advanced Dynamics and Control.* TechTeach, 2010.

[7] João P. Hespanha. *Linear System Theory.* Princeton Univeristy, 2009.

[8] Bell Helicopter Textron Inc. *V-22 Osprey guidebook.* 2011/2012.

[9] Israel Aerospace Industries. *Panther, Fixed wing VTOL UAS.* Israel Aerospace Industries, 2012.

[10] Parviz E. Nikravesh. *Computer-aided Analysis of Mechanical Systems.* Prentic-Hall Inc., 1988.

[11] Gareth D. Padfield. *Helicopter Flight Dynamics. The Theory and application of Flying Qualities and Simulation Modelling.* Wiley, 2008.

[12] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control.* JOHN WILEY & SONS, 2005.

[13] Kjell Eivind Skjønhaug Øyvind Magnussen. *Modeling, Design and Experimental Study for a Quadcopter System Construction.* PhD thesis, University of Agder, 2011.

List of Figures

List of Tables

Appendix A

Linearization

Linearized physical model of TRT in state-space structure will result in A,B and C matrices, where elements of those matrices are given by $a_{i,j}, b_{i,j}, c_{i,j}$ correspondingly.

$$a_{i,j} = \left.\frac{\partial h_i(x,u)}{\partial x_j}\right|_{(x_0,u_0)}, b_{i,j} = \left.\frac{\partial h_i(x,u)}{\partial u_j}\right|_{(x_0,u_0)}, c_{i,j} = \left.\frac{\partial g_i(x)}{\partial x_j}\right|_{(x_0)}$$

$$A = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & 0 \\ 0 & 0 & a_{2,3} & a_{2,4} & a_{2,5} & 0 \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} \\ 0 & 0 & 0 & 0 & a_{4,5} & a_{4,6} \\ 0 & 0 & 0 & a_{5,4} & 0 & a_{5,6} \\ 0 & 0 & 0 & a_{6,4} & a_{6,5} & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} \\ b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The nonzero elements in matrices A and B are calculated below. The C matrix is an identity matrix due to direct readout from sensors.

$$\begin{aligned}
a_{1,1} &= \frac{\partial\dot{\phi}}{\partial\phi} = 0\\
a_{1,2} &= \frac{\partial\dot{\phi}}{\partial\theta} = (sin(\theta) * (p * (2 * sin(\psi/2)^2 - 1) + q * sin(\psi)))/(sin(\theta)^2 - 1)\\
a_{1,3} &= \frac{\partial\dot{\phi}}{\partial\psi} = -(q * cos(\psi) + p * sin(\psi))/cos(\theta)\\
a_{1,4} &= \frac{\partial\dot{\phi}}{\partial p} = cos(\psi)/cos(\theta)\\
a_{1,5} &= \frac{\partial\dot{\phi}}{\partial q} = -sin(\psi)/cos(\theta)\\
a_{1,6} &= \frac{\partial\dot{\phi}}{\partial r} = 0
\end{aligned}$$

$$\begin{aligned}
a_{2,1} &= \frac{\partial\dot{\theta}}{\partial\phi} = 0\\
a_{2,2} &= \frac{\partial\dot{\theta}}{\partial\theta} = 0\\
a_{2,3} &= \frac{\partial\dot{\theta}}{\partial\psi} = p * cos(\psi) - q * sin(\psi)\\
a_{2,4} &= \frac{\partial\dot{\theta}}{\partial p} = sin(\psi)\\
a_{2,5} &= \frac{\partial\dot{\theta}}{\partial q} = cos(\psi)\\
a_{2,6} &= \frac{\partial\dot{\theta}}{\partial r} = 0
\end{aligned}$$

$$\begin{aligned}
a_{3,1} &= \frac{\partial\dot{\psi}}{\partial\phi} = 0\\
a_{3,2} &= \frac{\partial\dot{\psi}}{\partial\theta} = -(p * cos(\psi) - q * sin(\psi))/cos(\theta)^2\\
a_{3,3} &= \frac{\partial\dot{\psi}}{\partial\psi} = (sin(\theta) * (q * cos(\psi) + p * sin(\psi)))/cos(\theta)\\
a_{3,4} &= \frac{\partial\dot{\psi}}{\partial p} = -cos(\psi) * tan(\theta)\\
a_{3,5} &= \frac{\partial\dot{\psi}}{\partial q} = sin(\psi) * tan(\theta)\\
a_{3,6} &= \frac{\partial\dot{\psi}}{\partial r} = 1
\end{aligned}$$

$$\begin{aligned}
a_{4,1} &= \frac{\partial\dot{p}}{\partial\phi} = 0\\
a_{4,2} &= \frac{\partial\dot{p}}{\partial\theta} = 0\\
a_{4,3} &= \frac{\partial\dot{p}}{\partial\psi} = 0\\
a_{4,4} &= \frac{\partial\dot{p}}{\partial p} = 0\\
a_{4,5} &= \frac{\partial\dot{p}}{\partial q} = -(r * (J_{yy} - J_{zz}))/J_{xx}\\
a_{4,6} &= \frac{\partial\dot{p}}{\partial r} = -(q * (J_{yy} - J_{zz}))/J_{xx}
\end{aligned}$$

$$\begin{aligned}
a_{5,1} &= \frac{\partial\dot{q}}{\partial\phi} = 0\\
a_{5,2} &= \frac{\partial\dot{q}}{\partial\theta} = 0\\
a_{5,3} &= \frac{\partial\dot{q}}{\partial\psi} = 0\\
a_{5,4} &= \frac{\partial\dot{q}}{\partial p} = (r * (J_{xx} - J_{zz}))/J_{yy}\\
a_{5,5} &= \frac{\partial\dot{q}}{\partial q} = 0\\
a_{5,6} &= \frac{\partial\dot{q}}{\partial r} = (p * (J_{xx} - J_{zz}))/J_{yy}
\end{aligned}$$

$$\begin{aligned}
a_{6,1} &= \frac{\partial\dot{r}}{\partial\phi} = 0\\
a_{6,2} &= \frac{\partial\dot{r}}{\partial\theta} = 0\\
a_{6,3} &= \frac{\partial\dot{r}}{\partial\psi} = 0\\
a_{6,4} &= \frac{\partial\dot{r}}{\partial p} = -(q * (J_{xx} - J_{yy}))/J_{zz}\\
a_{6,5} &= \frac{\partial\dot{r}}{\partial q} = -(p * (J_{xx} - J_{yy}))/J_{zz}\\
a_{6,6} &= \frac{\partial\dot{r}}{\partial r} = 0
\end{aligned}$$

$$
\begin{aligned}
b_{4,1} &= \frac{\partial \dot{p}}{\partial F_1} = (L_{1y} * cos(\alpha_1) + K_t * sin(\alpha_1))/J_{xx} \\
b_{4,2} &= \frac{\partial \dot{p}}{\partial F_2} = (L_{2y} * cos(\alpha_2) - K_t * sin(\alpha_2))/J_{xx} \\
b_{4,3} &= \frac{\partial \dot{p}}{\partial F_3} = (L_{3y} * cos(\alpha_3) + L_{3z} * sin(\alpha_3))/J_{xx} \\
b_{4,4} &= \frac{\partial \dot{p}}{\partial \alpha_1} = (F_1 * (K_t * cos(\alpha_1) - L_{1y} * sin(\alpha_1)))/J_{xx} \\
b_{4,5} &= \frac{\partial \dot{p}}{\partial \alpha_2} = -(F_2 * (K_t * cos(\alpha_2) + L_{2y} * sin(\alpha_2)))/J_{xx} \\
b_{4,6} &= \frac{\partial \dot{p}}{\partial \alpha_3} = (F_3 * (L_{3z} * cos(\alpha_3) - L_{3y} * sin(\alpha_3)))/J_{xx}
\end{aligned}
$$

$$
\begin{aligned}
b_{5,1} &= \frac{\partial \dot{q}}{\partial F_1} = -(L_{1x} * cos(\alpha_1) - L_{1z} * sin(\alpha_1))/J_{yy} \\
b_{5,2} &= \frac{\partial \dot{q}}{\partial F_2} = -(L_{2x} * cos(\alpha_2) - L_{2z} * sin(\alpha_2))/J_{yy} \\
b_{5,3} &= \frac{\partial \dot{q}}{\partial F_3} = -(L_{3x} * cos(\alpha_3) + K_t * sin(\alpha_3))/J_{yy} \\
b_{5,4} &= \frac{\partial \dot{q}}{\partial \alpha_1} = (F_1 * (L_{1z} * cos(\alpha_1) + L_{1x} * sin(\alpha_1)))/J_{yy} \\
b_{5,5} &= \frac{\partial \dot{q}}{\partial \alpha_2} = (F_2 * (L_{2z} * cos(\alpha_2) + L_{2x} * sin(\alpha_2)))/J_{yy} \\
b_{5,6} &= \frac{\partial \dot{q}}{\partial \alpha_3} = -(F_3 * (K_t * cos(\alpha_3) - L_{3x} * sin(\alpha_3)))/J_{yy}
\end{aligned}
$$

$$
\begin{aligned}
b_{6,1} &= \frac{\partial \dot{r}}{\partial F_1} = (K_t * cos(\alpha_1) - L_{1y} * sin(\alpha_1))/J_{zz} \\
b_{6,2} &= \frac{\partial \dot{r}}{\partial F_2} = -(K_t * cos(\alpha_2) + L_{2y} * sin(\alpha_2))/J_{zz} \\
b_{6,3} &= \frac{\partial \dot{r}}{\partial F_3} = (K_t * cos(\alpha_3) - L_{3x} * sin(\alpha_3))/J_{zz} \\
b_{6,4} &= \frac{\partial \dot{r}}{\partial \alpha_1} = -(F_1 * (L_{1y} * cos(\alpha_1) + K_t * sin(\alpha_1)))/J_{zz} \\
b_{6,5} &= \frac{\partial \dot{r}}{\partial \alpha_2} = -(F_2 * (L_{2y} * cos(\alpha_2) - K_t * sin(\alpha_2)))/J_{zz} \\
b_{6,6} &= \frac{\partial \dot{r}}{\partial \alpha_3} = -(F_3 * (L_{3x} * cos(\alpha_3) + K_t * sin(\alpha_3)))/J_{zz}
\end{aligned}
$$

Appendix B

Numeric Linearized model

Numeric values for A,B and C matrices, linearized around the following state variable vector $x_0 = [0, 0, 0, 0, 0, 0]^T$ and estimated input vector $u_0 = [4.8050, 4.8063, 2.3530, -0.0056, 0.0056, 0]^T$.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10.9787 & 10.9758 & 0 & 0.9155 & -1.5067 & -0.5313 \\ -3.2695 & -3.2719 & 13.3600 & -1.1229 & -0.9472 & -0.5656 \\ 0.0944 & -0.1552 & 0.1248 & 26.1234 & -26.1238 & 16.3206 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix C

The main loop used on the TRT is presented below, containing the implemented control system.

```c
#include "param.h"
#include "uart.h"
#include "twi.h"
#include "MPU6050.h"
#include "Timer.h"
#include "MotorPWM.h"
#include "MS5611.h"
#include "Receiver.h"
#include "AngleCalc.h"
#include <stdio.h>
#include <avr/interrupt.h>

char i;
char *Ptr;
char temp;
char TempBuffer[40];
double k1[] = {-22.3474,  -7.3037,   0.4259,  -7.9113,  -2.6128,   0.1605,  -1.3110,  -0.4381,   0.0284};
double k2[] = { 22.3450,  -7.3162,  -0.5557,   7.9106,  -2.6175,  -0.2067,   1.3109,  -0.4390,  -0.0361};
double k3[] = { -0.0068,  29.8593,   0.2662,  -0.0027,  10.6822,   0.0947,  -0.0005,   1.7913,   0.0158};
double k4[] = {  0.1692,  -0.3162,   7.2294,   0.0573,  -0.1132,   2.5835,   0.0090,  -0.0190,   0.4327};
double k5[] = { -0.3195,  -0.2621,  -7.2279,  -0.1105,  -0.0936,  -2.5829,  -0.0178,  -0.0157,  -0.4326};
double k6[] = { -0.1750,  -0.1595,   4.5193,  -0.0636,  -0.0571,   1.6151,  -0.0109,  -0.0096,   0.2705};
double p[] = {39.3005, -603.7002, 4758.5445, 34.4823};
double M1, M2, M3 , a1, a2, a3;
double Er_int_roll;
double Er_int_pitch;
double Er_int_yaw;
double Er_roll;
double Er_pitch;
double Er_yaw;
double Er_p;
double Er_q;
double Er_r;
int     Tempdd;

#define DebugHz 50
int Debug;
int blink;

/*******************************************************************************************************
                MAIN
*******************************************************************************************************/
int main(void)
{
        // Initialize
        I2C_PULLUPS_ENABLE;
        //SpektrumBinding();
        //DDRK = 0xFF;
        ALARMPIN_PINMODE
        STABLEPIN_PINMODE;

        IMU.Gyro_OffsetX = 0;
        IMU.Gyro_OffsetY = 0;
        IMU.Gyro_OffsetZ = 0;
        Rx.QuatAccGain = 0.005;
        Rx._GyroCalibrateSamples = 100;
        Rx._LoopTime = 156;
        // Transmitter settings
        Rx._RecRollGain = 0.3;
        Rx._RecPitchGain = 0.3;
        Rx._RecYawGain = 0.5;
        Rx._RecThrustGain = 1.2;

        sei();          // Enable interrupt for UART and Timer

        MotorPWM_Init();
        Servo1_pwm = 0;
        Servo2_pwm = 0;
        Servo3_pwm = 0;
        Servo_Update();
        ESC_Calibration();
        twiInit(10);                                    _delay_ms(10);
        IMU_init();                                         _delay_ms(100);
        Baro_init();                                    _delay_ms(10);
        Timer_Init(TimerTop, TimerPreScale);
        uart_init(UART_BAUD_SELECT_DOUBLE_SPEED(UART_BAUD_RATE,F_CPU));
        uart1_init(UART_BAUD_SELECT_DOUBLE_SPEED(UART_BAUD_RATE,F_CPU));
        _delay_ms(5);
        Reciver_Init(1,0);
        // Error signal  initialization ----------------------------
        IMU_Get();
        QuatCalc();
        Quat2Euler();
        Er_int_roll=Euler.roll;
        Er_int_pitch=Euler.pitch;
        Er_int_yaw=0;
        //------------------------------------------------
        TCNT0 = 0;
    while(1){
    // Read Sensors
        IMU_Get();                // Get IMU Data
    // Attitude
        QuatCalc();
        Quat2Euler();
    // State error  collect ------------------------------------
        Er_int_roll=Er_int_roll+Euler.roll*(1/LoopFreq);
        Er_int_pitch=Er_int_pitch+Euler.pitch*(1/LoopFreq);
        //Er_int_yaw=Er_int_yaw+Er_yaw*(1/LoopFreq); //zeroed signal
        Er_roll=Euler.roll-Ctrl.Roll;
        Er_pitch=Euler.pitch-Ctrl.Pitch;
        Er_yaw=Er_yaw + IMU.d_GyroZ*(1/LoopFreq);
        Er_p=IMU.d_GyroX;
        Er_q=IMU.d_GyroY;
        Er_r=IMU.d_GyroZ-Ctrl.Yaw;
    // Control system--------------------------------------------
        M1= 4.8056 - (Er_int_roll*k1[0]+Er_int_pitch*k1[1]+Er_int_yaw*k1[2]+Er_roll*k1[3]+Er_pitch*k1[4]
            +Er_yaw*k1[5]+Er_p*k1[6]+Er_q*k1[7]+Er_r*k1[8]);
        M2= 4.8056 - (Er_int_roll*k2[0]+Er_int_pitch*k2[1]+Er_int_yaw*k2[2]+Er_roll*k2[3]+Er_pitch*k2[4]
            +Er_yaw*k2[5]+Er_p*k2[6]+Er_q*k2[7]+Er_r*k2[8]);
        M3= 3.3000 - (Er_int_roll*k3[0]+Er_int_pitch*k3[1]+Er_int_yaw*k3[2]+Er_roll*k3[3]+Er_pitch*k3[4]
            +Er_yaw*k3[5]+Er_p*k3[6]+Er_q*k3[7]+Er_r*k3[8]);
        a1= -0.0056 - (Er_int_roll*k4[0]+Er_int_pitch*k4[1]+Er_int_yaw*k4[2]+Er_roll*k4[3]+Er_pitch*k4[4]
            +Er_yaw*k4[5]+Er_p*k4[6]+Er_q*k4[7]+Er_r*k4[8]);
        a2= 0.0056 - (Er_int_roll*k5[0]+Er_int_pitch*k5[1]+Er_int_yaw*k5[2]+Er_roll*k5[3]+Er_pitch*k5[4]
            +Er_yaw*k5[5]+Er_p*k5[6]+Er_q*k5[7]+Er_r*k5[8]);
        a3= 0 - (Er_int_roll*k6[0]+Er_int_pitch*k6[1]+Er_int_yaw*k6[2]+Er_roll*k6[3]+Er_pitch*k6[4]
            +Er_yaw*k6[5]+Er_p*k6[6]+Er_q*k6[7]+Er_r*k6[8]);

    //Motor [N] to [puls]  scaling----------------------------
    M1= M1*Ctrl.Thrust;
    M2= M2*Ctrl.Thrust;
    M3= M3*Ctrl.Thrust;

    Motor1_pwm= p[0]*(M1*M1*M1) + p[1]*(M1*M1) + p[2]*(M1) + p[3] ;
    Motor2_pwm= p[0]*(M2*M2*M2) + p[1]*(M2*M2) + p[2]*(M2) + p[3] ;
    Motor3_pwm= p[0]*(M3*M3*M3) + p[1]*(M3*M3) + p[2]*(M3) + p[3] ;

    //Servo [rad]  to  [puls]  scaling------------------------------
        a1 = a1*9167.324722;
        a2 = a2*9167.324722;
        a3 = a3*9167.324722;
        Servo1_pwm = -a1;
        Servo2_pwm = a2;
        Servo3_pwm = -a3;
    //------------------------------------------------
    // Update Info Screen
        if  (blink == 2)
        {
                blink = 0;          // Prepare new round
                STABLEPIN_TOGGLE;
        }
    blink++;
    // Loop Timer
        if    (TCNT0 > LoopTimeCnt) {ALARMPIN_ON;} //else {ALARMPIN_OFF;} // Alarm, loop is too slow for the given frequency
        while(TCNT0 < LoopTimeCnt){}                                                              // Wait for looptime see param.h
        TCNT0 = 0;

    // Motor set after timer to get the update synchronous
        ESC();
        Servo_Update();
 }
} // main
```

Appendix D

Part list

All the parts except the controller have been bought at http://www.Hobbyking.com, where the flight controller has been bought at http://www.diymulticopter.com/

Item	Type	Quantity
Frame	Turnigy Talon Tricopter (V1.0)	1
Tail Servo Mount	Turnigy Talon Tricopter (V1.0)	2
Carbon Fiber Tube (3pcs)	Turnigy Talon Tricopter (V1.0)(380mm)	1
Motor	NTM Prop Drive 28-26 1000KV / 315W	3
Prop Adapter	NTM Prop Drive 28 Series Accessory Pack	3
ESC	Turnigy Plush 25 Amp	3
Servo	TGY-306G	3
Propeller	8x4.5 SF (2+2pcs)	2
Controller	MultiwWii Atmega 2560 (no-GPS)	1
Battery	Turnigy nano-tech 3300mah 3S 45 90C Lipo Pack	1
ESC Power Breakout Cable	4mm to 3 x 3.5 bullet	1
Servo Extender Cable	45cm Servo lead extension (pack)	1
Bullet plug	3.5mm 10pack	3
Wire Red	16AWG ,1m	3
Wire Black	16AWG ,1m	1
Shrink Tube	1m Red	1
Nylon Spacer	5.6x14mm (10pcs) M3	1
Nylon Screws	M3x8 (10pcs)	1
Vibration Padding	25x25x3mm(10pcs)	1

Table 7.1: Part list

Appendix E