Design and Control of a Quadcopter

Michael Schwegel

January 17, 2015

Contents

1 Introduction

This paper treats the setup and mathematical description of a quadcopter. The equations of motion are derived in a rigurous way and are a linearization is presented to obtain a control law suitable for 8-bit microcontrollers. Recent technologies are investigated for their qualification to achieve an up to date, yet cheap hardware.

2 Mathematical Model

There are various ways of expressing the equations of motion for a quadcopter. The form and derivation of those depends strongly on the representation of rotations. Approaches based on quaternions, rotation matrices, such as the direct cosine method, and Euler/Cardan angles are the most common ways found in literature, see for example [1], [2–4] and [5, chap. 8.2] respectively. Methods using more than $p > 3$ parameters, like quaternions (4 parameters), don't have singularities, however $p - 3$ side conditions have to be met. This makes them numerically harder to handle. Also it takes some extra effort to use quaternions for control directly, because the equations and control laws have to be reformulated, like shown in [6] and others.

For all these possibilities control strategies based on linearizations were proposed, i.e. LQR [7] and model predictive control [5, chap. 9.4]. Other non linear control laws like the control with lyapunov functions, directly use the derived equations [7].

Here a model based on Cardan angles using the ZYX-scheme will be presented and examined. This approach is a compromise between the hard to implement, but 360° valid quaternions and the DCM, which is only valid for small angles.

2.1 Coordinate Transformations

The description of a vector in a rotated coordinate system is presented here before going into detailed equations of motion. The notation follows [8] and can be loosely followed with [9], where Cardan angles are called Euler angles. This is however not precise, since Euler angles are rotation matricies, where 2 axes are used to rotate a body (ie. ZYZ-scheme), whereas with Cardan angles all 3 axes are used (ie. XYZ-scheme). Additionally Euler and Cardan angles are means to describe the rotation of a vector. In the following however the representation of a (temporarily) fixed vector in a rotated coordinate system is needed. A rotation of a vector is just the inverse of the representation in a rotated coordinate system, which was generated by the rotation of the base by the same angle.

The rotation of a coordinate system B to C is done with a matrix $A_{CB} \in SO3$. Where $SO3$ is the set of symmetric, orthogonal 3×3 matricies with determinant 1. They are strictly positive definite by definition and have an eigenvalue of 1 and two complex conjugated eigenvalues on the complex unit circle $\mathrm{e}^{\pm i\varphi}$, where φ is the rotation angle. A vector ${}_B\boldsymbol{a}$, given in base B can be expressed in base I by representing it's unit vectors in terms of the target coordinate system

$$ {}_I\boldsymbol{a} = \underbrace{[{}_I\boldsymbol{e}_x^B \;\; {}_I\boldsymbol{e}_y^B \;\; {}_I\boldsymbol{e}_z^B]}_{A_{IB}} \cdot {}_B\boldsymbol{a} \tag{1} $$

Using the orthogonality property of $SO3$ with $A^T = A^{-1}, \forall A \in SO3$ a coordinate transform from base I to base B, A_{BI} becomes

$$ A_{BI} = A_{IB}^{-1} = R_{BI}^T = R_{BI}^{-1} \tag{2} $$

Using (1) to rotate the base of a right handed orthonormal coordinate system around the z axis by the mathematically positive angle ψ, yields

$$ A_{10} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3} $$

The matrix in (3) represents the elementary rotation of a coordinate system about the z axis by the angle ψ. With the z axis in vertical direction, this rotation is called yaw in vehicle dynamics (and aeronautics) with ψ being the yaw angle.

A quadcopter/rigid body can rotate about 3 arbitrary axes and their order makes a difference (the matricies don't commute). A ZYX-Rotation scheme is used here (the corresponding vector rotation is called Cardan angle). For a ZXZ-Euler angle representation see [10]. Both Euler and Cardan angles are successive rotations about the axes defined by the previous rotation. Euler angles however use the same (rotated) axis twice and another one in between. Cardan angles use rotations about all 3 axes.

The rotation about the initial y-axis by the pitch angle ϑ is corresponds to the matrix

$$A_{21} = \begin{bmatrix} \cos(\vartheta) & 0 & -\sin(\vartheta) \\ 0 & 1 & 0 \\ \sin(\vartheta) & 0 & \cos(\vartheta) \end{bmatrix} \tag{4}$$

and the rotation about the x axis and the roll angle φ can be expressed as

$$A_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \tag{5}$$

The representation of a vector in arbitrarily rotated coordinates can now be found by applying the 3 rotations consecutively

$$_3\boldsymbol{c} = \underbrace{A_{32}\, A_{21}\, A_{10}}_{=:\, A_{30}}\ {}_0\boldsymbol{c} \tag{6}$$

If the 0-system is interpreted as Earth-fixed and considered as inertial system in this scope and using $A_{30} = A_{BI} = A_{IB}^T$, a vector $_B\boldsymbol{c}$ in the body-fixed/3-coordinate system can be expressed as

$$_I\boldsymbol{c} = A_{IB}\ {}_B\boldsymbol{c}\,, \quad A_{IB} = \begin{bmatrix} c\vartheta\, c\psi & s\varphi\, s\vartheta\, c\psi - c\varphi\, s\psi & c\varphi\, s\vartheta\, c\psi + s\varphi\, s\psi \\ c\vartheta\, s\psi & s\varphi\, s\vartheta\, s\psi + c\varphi\, c\psi & c\varphi\, s\vartheta\, s\psi - s\varphi\, c\psi \\ -s\vartheta & s\varphi\, c\vartheta & c\varphi\, c\vartheta \end{bmatrix} \tag{7}$$

in terms of the inertial coordinate basis. Here the abbreviations sx and cy are used for $\sin(x)$ and $\cos(y)$ respectively. The matrix A_{IB} in equation (7) defines the rotation from the body fixed to the inertial system. This matrix transforms a given vector acting in the body system at an arbitrary attitude to be expressed in the inertial system. However for the full Newton-Euler equations the angular velocity is needed too. The angular velocity in the body-fixed system is defined as

$$
{}_B\boldsymbol{\omega} = {}_3\boldsymbol{\omega} = {}_3\begin{bmatrix}\dot\varphi\\0\\0\end{bmatrix} + A_{32}\;{}_2\begin{bmatrix}0\\\dot\vartheta\\0\end{bmatrix} + A_{32}\;A_{21}\;{}_1\begin{bmatrix}0\\0\\\dot\psi\end{bmatrix} \tag{8}
$$

Each angular velocity component has to be taken about the corresponding axis of rotation by transformation of the coordinate system in which they are given in. Since φ rotates about the body x axis, the $\dot\varphi$ component is just to be multiplied with ${}_B\boldsymbol{e}_x \equiv {}_3\boldsymbol{e}_x$. The $\dot\vartheta$ component rotates about the ${}_2\boldsymbol{e}_y$ axis and this axis has to be represented in the target system 3 with A_{32}. $\dot\psi$ rotates about the ${}_1\boldsymbol{e}_z$ axis and is therefore given in 3 by the transformation A_{31}. These coordinate transformations can be concatenated to a matrix for angular velocities

$$
N(\varphi,\vartheta)^{-1} := \begin{bmatrix}1 & 0 & -\sin(\vartheta)\\ 0 & \cos(\varphi) & \sin(\varphi)\cos(\vartheta)\\ 0 & -\sin(\varphi) & \cos(\varphi)\sin(\vartheta)\end{bmatrix} \tag{9}
$$

which represents the basis for the angular velocities in body coordinates. With these definitions, the angular velocities in the inertial system are given as

$$
{}_I\boldsymbol{\omega} = N(\varphi,\vartheta)\,{}_B\boldsymbol{\omega} \tag{10}
$$

and

$$
N(\varphi,\vartheta) := \begin{bmatrix}1 & \sin(\varphi)\tan(\vartheta) & \cos(\varphi)\tan(\vartheta)\\ 0 & \cos(\varphi) & -\sin(\varphi)\\ 0 & \sin(\varphi)\sec(\vartheta) & \cos(\varphi)\sec(\vartheta)\end{bmatrix} \tag{11}
$$

denoted in Euler angles and their derivatives.

2.1.1 Gimbal Locking

Before deriving the equations of motion of a quadcopter, an important property of the Euler, Cardan angles must be taken into consideration. If we consider the rotation matrix from equation (3) and set ψ, the yaw angle to 0, the rotation matrix becomes the identity matrix and thus has no effect on the final position. For the matrix A_{IB} from equation (7) this singularity occurs for a pitch angle of $\vartheta = \pm\frac{\pi}{2}$. This can be seen considering the simpler matrix from equation (4) and search for a ϑ, where A_{21} becomes a identity matrix. If one axis is represented only by an identity matrix it can no longer contribute to the rotation. In practice this means that the roll and yaw angles are defined on parallel axes, which causes them to express the same rotation. Upon running into this singular angle, the third axis can be forced to do a 180° rotation in zero time to maintain the sign of the angles. Furthermore the matrix N^{-1} is singular at this point and therefore the N matrix is not defined. This effect is referred to as "Gimbal-Lock" and leads to destabilisation of the system. There are ways to avoid a gimbal lock, either in a mechanically (restrictions of the gimbal) or mathematically (flipping signs). Neither of these workarounds are used here, because small attitude angles are allowed only. A 90° pitch is common only for stunt flights, since this position can not be maintained (loss of lift).

2.2 Equations of Motion

2.2.1 Rotational Equations

For the rotational variables, with ${}_B\boldsymbol{\omega} = [p, q, r]^T$ denoting the rate of rotation in the body frame, the mechanical spin of a rigid body about its center of gravity in a body fixed coordinate system B is given by

$$ {}_BS_{CG} = {}_BJ \; {}_B\boldsymbol{\omega} \tag{12} $$

Since the moments of inertia J are taken in the bodies principle axes, the matrix is diagonal. Because the spin is given in the rotating coordinate system B, the Coriolis equation has to be used to take the time derivative for the Newton-Euler equations

$$ {}_B(\dot{S}_{CG}) = {}_BJ \; {}_B\dot{\boldsymbol{\omega}} + {}_B\boldsymbol{\omega} \times {}_BJ \; {}_B\boldsymbol{\omega} = {}_B\boldsymbol{\tau} \tag{13} $$

with ${}_B\boldsymbol{\tau}$ as vector of torques in the body frame. To get the resulting inertial angles and rates, equation (10) is used and integrated

$$ {}_I\boldsymbol{\eta} = \int N \; {}_B\omega \; dt \tag{14} $$

with ${}_I\boldsymbol{\eta} = [\varphi, \vartheta, \psi]^T$ in the inertial frame.

2.2.2 Translational Equations

Denoting the vector of gravitational acceleration ${}_I\boldsymbol{g} = [0, 0, g]^T$ and ${}_B\boldsymbol{F}$ the forces acting in the body frame, as well as the mass matrix $m \cdot I_3$, the equations for the translational coordinates in the inertial frame ${}_I\boldsymbol{x}$ can be found to be

$$m\, {}_I\ddot{\boldsymbol{x}} = A_{IB}\ {}_B\boldsymbol{F} - m\ {}_I\boldsymbol{g} \tag{15}$$

In equation (15) the thrust ${}_B\boldsymbol{F}$ is transformed from the body frame into the inertial coordinates by the matrix A_{IB} from equation (7). It is derived by applying Newton's laws.

3 Control Design

Quadrotor helicopter dynamics are very complex in detail. Multiple interactions of aerodynamic and mechanical nature happen at very short timescales. Further investigations regarding blade flapping and other effects can be reviewed in [1]. One has to limit the mathematical model in order to retain a controllable and managable scale for the controller. For this paper a linear model disregarding aerodynamic effects, the motor dynamics and the mechanical stiffness of the structure was applied.

Equations (13) and (15) together represent a system of 6 equations for the 6 degrees of freedom of the quadcopter. In this form it is easily shown that with ${}_B\boldsymbol{F}$ being a vector pointing upwards, the system is under-actuated. Implicitly two translational degrees of freedom are determined by the attitude of the vehicle. By intuition it is clear that pitch and roll of the vehicle result in a change in the horizontal position. This fact can be used to control just the attitude and thereby affect the position. If the attitude could be set perfectly then a controller for the position could be implemented to track a given flight path perfectly. Building on this idea, a cascade control system can be used. The outer loop determines the required angle for a given path, whereas the second loop controls the corresponding attitude. Since the inner loop is more time-critical and guarantees the stability of the vehicle, it is the main target of the present work to introduce a well suited control loop for the attitude and vertical positioning of the vehicle.

3.1 PID Control

A PID (proportional-differential-integral) controller is chosen as inner control loop. This basic control algorithm can be implemented easily and still shows various effects of basic control theory. One example is higher frequency oscillations with high proportional gains compared to lower frequency oscillations with high integral gains. The systems two integrators make render it unstable. Also an integrating controller is therefore not mandatory and if used, has to be implemented carefully, to avoid integral windup. Herein the integral has been hardly limited to avoid this phenomenon. There are possible extensions to the basic controller currently in use, for example the integrator can act only close to the setpoint. In the following the implementation of the controller will be discussed.

The controller inputs to the system equations

$$
\begin{aligned}
{}_{B}J\,{}_{B}\dot{\boldsymbol{\omega}} + {}_{B}\boldsymbol{\omega} \times {}_{B}J\,{}_{B}\boldsymbol{\omega} &= {}_{B}\boldsymbol{\tau} && \text{(16a)}\\
m\,{}_{I}\ddot{\boldsymbol{x}} &= A_{IB}\,{}_{B}\boldsymbol{F} - m\,{}_{I}\boldsymbol{g} && \text{(16b)}
\end{aligned}
$$

are ${}_{B}\tau_x$, ${}_{B}\tau_y$, ${}_{B}\tau_z$ and ${}_{B}\boldsymbol{F}$, the torques and forces produced by the rotors displayed in the body system. These values are all defined through the rotational velocity of the four motors. Assuming the PWM signal γ_i is proportional to the rotation rate of the motor i and l is the distance of a motor to the center of gravity of the quadcopter, k_F and k_M are constant motor and propeller specific constants, a linear actuation model is

$$
\begin{aligned}
{}_{B}\tau_x &= l \cdot k_F(\gamma_1^2 - \gamma_3^2) && \text{(17a)}\\
{}_{B}\tau_y &= l \cdot k_F(\gamma_2^2 - \gamma_4^2) && \text{(17b)}\\
{}_{B}\tau_z &= k_M(\gamma_1^2 - \gamma_2^2 + \gamma_3^2 - \gamma_4^2) && \text{(17c)}\\
|{}_{B}\boldsymbol{F}| &= k_F(\gamma_1^2 + \gamma_2^2 + \gamma_3^2 + \gamma_4^2) && \text{(17d)}
\end{aligned}
$$

The motors 1 and 3 as well as 2 and 4 have to be on opposing sides of the vehicle, see Figure 1. The quadratic form of the relation can be explained by the conservation of momentum in fluid dynamics. Thereafter the momentum transported within a flow is proportional to the velocity squared (the change in

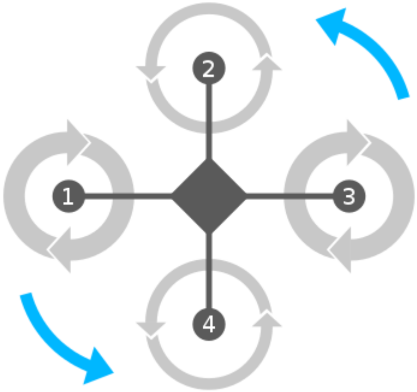


Figure 1: Top view schematics of the quadcopter showing relative rotation rates of the motors for yawing. Two co-rotating motors spin fast and the counter-rotating motors spin slower to generate a free torque. The thickness of the circles represents the spinning rate of the motors. The arrows on the outside show the direction of the resulting torque and the direction of the angular acceleration of the body.

momentum induces a force). Considering the mass balance, the angular velocity must be proportional to the downward velocity from the propellers. Thus the force is proportional to the square of the motor revolution and this fact is used in equations (17).

It is of value to write equations (17) in matrix notation. With the input vector $\boldsymbol{u} = \left[{}_B\tau_x, {}_B\tau_y, {}_B\tau_z, |{}_B\boldsymbol{F}|\right]^T$ we obtain

$$\boldsymbol{u} = A\,\boldsymbol{\gamma} \tag{18}$$

and, more importantly,

$$A^{-1} = \begin{bmatrix} \frac{1}{2k_F l} & 0 & \frac{1}{4k_M} & \frac{1}{4} \\ 0 & \frac{1}{2k_F l} & -\frac{1}{4k_M} & \frac{1}{4} \\ -\frac{1}{2k_F l} & 0 & \frac{1}{4k_M} & \frac{1}{4} \\ 0 & -\frac{1}{2k_F l} & -\frac{1}{4k_M} & \frac{1}{4} \end{bmatrix}. \tag{19}$$

From equation (19) one can see how inputs are translated into rotation rates and thereby how the quadcopter acts on the motors for specific moves. In particular the entry ${}_B\tau_x$ (first column) translates into one motor spinning faster and one slower (minus sign), while the other two remain untouched with two

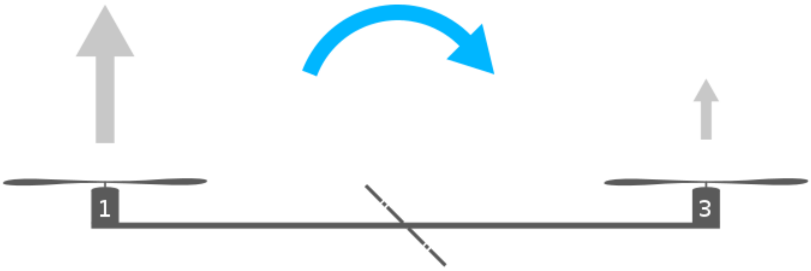


Figure 2: Side view schematics of the quadcopter showing relative thrust distribution on the motors while pitching or rolling. One motor spins up for more thrust, while the opposing one spins slower. This turns the vehicle within the vertical plane the motors lie in. Pitch and roll act alike due to symmetry. The thickness of the arrows represents the generated thrust, the arrow in the middle shows the direction of the resulting torque and the direction of the angular acceleration of the body.

zero elements. Equivalently the second column has the same effect acting upon ${}_B\tau_y$ and the pitch angle.

This can be seen in Figure 2. Column three of equation (19) distributes the desired yaw torque to co-rotating motors with positive signs for odd and negative signs for even numbers. Thus two co-rotating motors spin faster, the other two rotate slower to stop levelling the torque of the motors on the body and use this torque to yaw. This manoeuvre is shown in Figure 1. The fourth column distributes the desired thrust evenly to each motor. The presented engine configuration is called “plus” configuration and is characterized by the motors being mounted on the symmetry axes. A similar logic can be applied for different configurations, like the “X”-configuration, where the pitch and roll axes are rotated by 45° horizontally.

As mentioned earlier this project investigates the rotatory set of equations only, since no position control is desired. To apply a well known set of control tools to the system, the equations have to be linearised. The second term on the left side of equation (16a) is a term of second order and will be neglected. This can be done under the assumption that the angular rate vector $\boldsymbol{\omega}$ is small and the moments of inertia around x and y are equal, thus the vehicle is symmetrical. The remaining system consists of the two integrators

$$ {}_BJ \, {}_B\dot{\boldsymbol{\omega}} = {}_B\boldsymbol{\tau} \tag{20}$$

and ${}_I\dot{\boldsymbol{\eta}} = {}_B\boldsymbol{\omega}$. This is the result of equation (10) and a linearization of the matrix N, which results in an identity matrix. This is an inhomogeneous second order ordinary differential equation. To improve the intuition, the quadcopter can, in a first order approximation, be thought of as a solid body pushed on a frictionless ice-surface.

3.2 State Estimation

As an example how the state estimation problem for a multirotor can be solved, a discrete steady state Kalman filter approach is discussed in the following. This type of Kalman filter allows for precise estimation, while beeing computationally compareable to a simple complementary filter.

First, the usage of sensors for estimating the quadcopters attitude are discussed. These are then used to examine basic filtering methods to provide more precise estimates using those measurements.

3.2.1 Gyroscope

The gyroscope measures the angular velocity. To estimate the angle of inclination these measurements have to be integrated over time. Since measurements are never perfect and the integration has to be implemented numerically and discrete, errors are to be expected in the estimated angles. The biggest problem of using only gyroscopes is, that all errors in the measurements will be integrated (in discrete time: summed up). This leads to a significant drift in the estimated angle.

The gyroscopes mounted on the vehicle measure the angular rates of the gyroscope axes relative to the inertial system given in body fixed coordinates ${}_B\boldsymbol{\omega}_{IB}$. This is different from (14), because the derivatives of the Euler angles used for coordinate transformation each rotate about consecutively tilted axes, as described by equation (8). ${}_B\boldsymbol{\omega}_{IB}$ however is the vector of rates of rotation about the orthogonal and right-handed body coordinate frame. The formal way to achieve estimates of the inertial angles of inclination is

$$
{}_I\boldsymbol{\eta} = \int\limits_{\boldsymbol{\eta}_0}^{\infty} A_{IB}\ {}_B\boldsymbol{\omega}_{IB}\ dt \tag{21a}
$$

$$
{}_I\boldsymbol{\eta} = \sum A_{IB}\ {}_B\boldsymbol{\omega}_{IB}\ \Delta t \tag{21b}
$$

for continuous and discrete time respecitvely, using Δt as sample time. For the implementation, the angles of inclination are assumed to be small $\boldsymbol{\eta} \ll 1$ and the coordinate transformation becomes the identity matrix $A_{IB} \approx I_3$. The rates can be integrated from an arbitrary setpoint $\boldsymbol{\eta}_0$, which is later repeatedly corrected by the accelerometer. It will be shown, that the Kalman filter presented estimates the bias of the gyroscopes, which is inferred from the accelerometer. Since this measurement is always relative to the inertial frame, the effect of tilted axes are included in the bias estimates. With these two effects the assumption of small angles is corrected.

Based on the former observations the angles are estimated as

$$
{}_I\hat{\boldsymbol{\eta}} = \sum_{\boldsymbol{\eta}_0}^{\infty} {}_I\boldsymbol{\omega}_{IB}\ \Delta t\,, \qquad {}_I\boldsymbol{\omega}_{IB} \approx {}_B\boldsymbol{\omega}_{IB}\,. \tag{22}
$$

where the hat is used for variables which are estimated from measurements.

3.2.2 Accelerometer

Accelerometers provide means of estimating the pitch and roll angle in steady state. The gravitational acceleration ${}_I\boldsymbol{g} = [0, 0, g]^T$ is measured by the accelerometers. The measured accelerations ${}_B\boldsymbol{a}$ in steady state are

$$
{}_B\boldsymbol{a} \approx {}_B\boldsymbol{g} = A_{BI}\ {}_I\boldsymbol{g} \tag{23a}
$$

$$
{}_B\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \approx A_{BI}\ {}_I\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = g \cdot \begin{bmatrix} -sin(\vartheta) \\ sin(\varphi)\,cos(\vartheta) \\ cos(\varphi)\,cos(\vartheta) \end{bmatrix} \tag{23b}
$$

where the orthogonality relation $A_{BI} = A_{IB}^T$ is used, which leads to g times the third row of equation (7).

The pitch and roll angle can be estimated by comparing the direction of the measurements to the one of the inertial axes

$$tan(\hat{\varphi}) = \frac{a_y}{a_z} \qquad tan(\hat{\vartheta}) = \frac{-\, a_x}{\sqrt{a_y^2 + a_z^2}} \tag{24}$$

and like expected the magnitude g of the gravitational acceleration is canceled out. The trigonometric identities $sin(x)/cos(x) = tan(x)$ and $sin^2(x) + cos^2(x) = 1$ are used from equation (23b) to (24).

This method implies the assumption, that the gravitational acceleration is the only acceleration measured, as in equation (23a). The yaw angle, which turns around the inertial z-axis like $\boldsymbol{g}$ in steady state, can not be estimated in this way, due to their co-linearity.

When the vehicle is accelerated the estimates are distorted. Thus fast changes in the estimates are ignored, since they are assumed to be the effect of accelerations other than the gravitationl acceleration. Various low-pass filters can be used to filter out these misleading measurements. However a first order low-pass is sufficient and computationally very fast. This will be discussed further later.

3.2.3 Magnetometer

Magnetometers are used on inertial measurement units (IMU) to provide a steady means to estimate the yaw angle. The method used is analogous to the logic for the accelerometers. The earths 3-dimensional magentic field vector is ${}_I\boldsymbol{b} = [B_0\, cos\delta, 0, B_0\, sin\delta]^T$ in inertial coordinates. The magnetic field strength B_0 and the magnetic inclination δ, which represents the angle between the horizontal inertial plane and the maximal field measurement, depend on the distance to the magnetic poles. However both of these variables cancel out for the purpose of heading estimation, as will be shown.

Using the transformations of coordinates from equations (4), (5) and (3) respecively, the measurements of the magnetometer and the inertial magnetic field vector are transformed to the 1 system. 1 therefore represents a system, where the x-y plane is parallel to the one of the inertial reference frame. The 1 system can be viewed as horizontal, but rotated around the z-axis by the angle ψ, the heading/yaw angle. The measurements can be transformed to this system using

$$A_{12}\; A_{2B}\; {}_B\boldsymbol{b} = A_{1I}\; {}_I\boldsymbol{b} \tag{25}$$

giving an implicid equation, where everything but ψ in the transformation A_{1I} is known. For the matrices on the left hand side of equation (25) the orthogonality is used, giving

$$A_{21}^T A_{B2}^T \, {}_B\boldsymbol{b} = {}_1\boldsymbol{b} \tag{26a}$$

$$\begin{bmatrix} cos(\vartheta) & sin(\varphi)sin(\vartheta) & cos(\varphi)sin(\vartheta) \\ 0 & cos(\varphi) & -sin(\varphi) \\ -sin(\vartheta) & sin(\varphi)cos(\vartheta) & cos(\varphi)cos(\vartheta) \end{bmatrix} {}_B\boldsymbol{b} = B_0 \begin{bmatrix} cos(\psi)cos(\delta) \\ -sin(\psi)cos(\delta) \\ sin(\delta) \end{bmatrix} \tag{26b}$$

as the compass measurements in the 1 system. Taking minus the second row of equation (26b) and dividing it by the first row, the two magnetic field variables are cancelled out and the tangent of the heading angle is left as

$$tan(\hat{\psi}) = \frac{-{}_1b_y}{{}_1b_x} \tag{27}$$

as function of known variables only. Written out the tilt and pitch compensated measurements are

$${}_1b_x = cos(\vartheta)_B b_x + sin(\varphi)sin(\vartheta)_B b_y + cos(\varphi)sin(\vartheta)_B b_z \tag{28a}$$

$${}_1b_y = cos(\varphi)_B b_y - sin(\varphi)_B b_z \tag{28b}$$

and depend on the measurements in body coordinates and tilt and pitch angles estimated using the accelerometer measurements.

Because the signal to noise ratio of magnetometers are really low, the measurements have to be calibrated. This is necessary since most magnetic field sensors are sold without calibration, because nearby metallic components and cables or conducting parts induce another offset, which has to be dealt with any way. Calibration works by fitting the data to a unit ball manifold with radius B_0 and center at the origin. This approach uses 12 parameters, whereas a more efficient way is to correct for a bias and a gain on all 3 axes, which uses 6 parameters. Further details can be found in the sensor and nautical literature, for example [11].

3.2.4 Filtering Techniques

Since there are to be 3 types of sensors on a quadcopter the problem of using them together for a combined estimation of the state is often referred to as sensor fusion. A basic way of achieving this is to use the knowledge about the different sensor types and basic control logics. It is worthwhile to examine this approach to compare it to the Kalman filter.

As reasoned before, the gyroscope rate measurements $\boldsymbol{\omega}_{gyro}$ have to be integrated. Since a gyroscope typically has a constant offset, a high-pass filter will be used to improve robustness against this phenomenon. Furthermore the accelerometers angle estimate $\hat{\boldsymbol{\eta}}_{acc}$ will be filtered with a low-pass to reduce the effects of linear accelerations measured. The estimated angle $\hat{\boldsymbol{\eta}}$ can be expressed as a transfer function

$$\hat{\boldsymbol{\eta}} = \frac{1}{1+Ts}\hat{\boldsymbol{\eta}}_{acc} + \frac{Ts}{1+Ts}\frac{1}{s}\boldsymbol{\omega}_{gyro} = \frac{\hat{\boldsymbol{\eta}}_{acc} + T\boldsymbol{\omega}_{gyro}}{1+Ts}\,, \tag{29}$$

with s as the Laplace variable and T the cut-off frequency of the low-pass and high-pass filters. Equation (29) can be understood as 3 scalar equations, since all 3 rows are independent (for consistency it is further printed as vector).

Because we aim for a digital implementation, this continuous form has to be z-transformed. Various schemes are used to convert continuous into discrete transfer functions, each resulting in a different digital formulation of the algorithm. A first order backward difference scheme (backward Euler), with Δt the sample time,

$$s \mapsto \frac{z-1}{z\Delta t} \tag{30}$$

is used here to show a fundamental analogy between the presented method and a Kalman filter.Using the z-transformation in equation (30) to the transfer function (29) yields

$$\hat{\boldsymbol{\eta}}_k = \frac{\hat{\boldsymbol{\eta}}_{acc} + T\boldsymbol{\omega}_{gyro}}{1 + \frac{T}{\Delta t} - \frac{T}{\Delta t}z^{-1}} \tag{31}$$

which can be reformulated by applying the time shift operator z^{-1}, which shifts the index k to $k-1$, to

$$\hat{\boldsymbol{\eta}}_k = \lambda \left(\hat{\boldsymbol{\eta}}_{k-1} + \Delta t \cdot \boldsymbol{\omega}_{gyro} \right) + (I - \lambda)\hat{\boldsymbol{\eta}}_{acc} \tag{32}$$

abbreviating $\lambda = 1/(\frac{\Delta t}{T} + 1)$. Mathematically this is equivalent to a convex combination of the single sensor estimates,

$$\hat{\boldsymbol{\eta}} = \lambda_1 \hat{\boldsymbol{x}}_1 + \lambda_2 \hat{\boldsymbol{x}}_2 \, , \quad \sum_i \lambda_i = 1 \, , \quad \lambda_i \geq 0 \, , \tag{33}$$

where $\hat{\boldsymbol{x}}_i$ are estimates from different sensors. The parameters λ represent a "trust" or "probability" in the estimates; see [12].

Equation (32) is referred to as complementary filter. The origin of this name is obvious if one compares equation (32) and (33). In the complementary filter the constraint on the parameters λ_i in equation (33) is omitted and one parameter is replaced by the complement to the others.

A Kalman filter uses a state space representation of the dynamics of the system to estimate its state. Conventionally the system is written in the (discrete time) form

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + B\boldsymbol{u}_{k-1} + \boldsymbol{v} \tag{34a}$$

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \boldsymbol{w} \tag{34b}$$

where $\boldsymbol{x}$ is the state-, $\boldsymbol{u}$ the input- and $\boldsymbol{z}$ the measurement vector, A is the process matrix, B specifies how the input influences the state and H defines what is measured of the state. Note that because most real systems cannot react instantly to any kind of input, a direct input-measurement term is omitted. k is the time step/index, which relates to the time t by $t(k+i) = t(k) + i \cdot \Delta t$, $\forall i$ and trivially $t(k=0) = 0$. Since the update equation (34a) is not perfect, nor perfectly known, an error term $\boldsymbol{v}$ is added. The variance of $\boldsymbol{v}$ has to be known, or estimated and will be denoted by Q. Analogously $\boldsymbol{w}$ represents errors in the measurements of $\boldsymbol{y}$ and the variance is R. Note that no further requirement on the two stochastic terms $\boldsymbol{v}$ and $\boldsymbol{w}$ is needed. Under the assumption of normally distributed signals, however, the Kalman filter is guaranteed to be stable for properly defined problems.

A Kalman filter, which is a mean squared error optimal Bayesian estimation algorithm, consists of two logical steps. First equation (34a) is used to predict the state $\boldsymbol{x}$ one step into the future. Following this prediction step the so-called measurement update corrects the prediction based on new knowledge of the state, which is inferred from a measurement. Derivations and further details can be found in for example [13].

The algorithm not only keeps track of the state, but also of the "trust" or "probability" of the prediction (first step) and measurement update (second step), in the form of the variances P_p and P_m respectively.

A possible formulation of the Kalman filter equations is

$$\hat{\boldsymbol{x}}_p = A\hat{\boldsymbol{x}}_m + B\boldsymbol{u} \tag{35a}$$

$$P_p = AP_mA^T + Q \tag{35b}$$

for the prediction step with $\hat{\boldsymbol{x}}_p$ as predicted state vector and

$$K = P_pH^T(\underbrace{HP_pH^T + R}_{=:S})^{-1} \tag{36a}$$

$$\hat{\boldsymbol{x}}_m = \hat{\boldsymbol{x}}_p + K(\boldsymbol{z} - H\hat{\boldsymbol{x}}_p) \tag{36b}$$

$$P_m = (I - KH)P_p \tag{36c}$$

$$= (I - KH)P_p(I - KH)^T + KRK^T \tag{36d}$$

where K is the optimal Kalman gain. The time indices are omitted for clarity, since in the prior equations (35) the left-hand-side is defined at timestep k and the right-hand-side at time-step $k-1$ and in the update equations (36) all variables and matrices are defined at time instant k. An important relation when dealing with variances is that their inverse is a measure for information. The inverse of P_p is proportional to the probability of $\hat{\boldsymbol{x}}_p$ being true and P_m is the variance of the error of the Kalman filter. To provide more intuition, S is defined to be the covariance of the term $\boldsymbol{z} - H\hat{\boldsymbol{x}}_p$. The inverse is the information gained from the measurements $\boldsymbol{z}$ compared to the prediction $\hat{\boldsymbol{x}}_p$ alone as estimated in P_p. Equation (36a) therefore defines K to be the information of the prediction compared to the new knowledge and H^T maps the prior variance to the measurement space to make S and P_p comparable. Thus the Kalman gain weights measurements against what is predicted to be measured in equation (36b). Equations (36c) and (36d) are mathematically equivalent, however the latter formulation is numerically more stable and is used for other algorithms, whereas the former is computationally cheaper. It is required, that the mean errors are subtracted from the measurements $\boldsymbol{z}$ and the prediction $\hat{\boldsymbol{x}}_p$ before using equations (35) and (36).

Just like for the complementary filter, the Kalman filter equations can be implemented for each axis individually. This is especially easy if object oriented programming is used and the filter is instantiated using 3 objects of a single class. Therefore the following equations can be understood as stacked vector equations for all axes at once or per axis.

The most basic model for system (34) has to provide information how the measurements are related to the state. Such a simple model is

$$\hat{\boldsymbol{\eta}}_k = \hat{\boldsymbol{\eta}}_{k-1} + \Delta t \cdot \boldsymbol{\omega}_{gyro} \tag{37a}$$

$$\boldsymbol{z}_k = \hat{\boldsymbol{\eta}}_{acc} \tag{37b}$$

where the state consists of the angles to be tracked $\boldsymbol{x} = \hat{\boldsymbol{\eta}}$, the prediction matrix $A = 1$, the inputs are the gyroscope measurements $\boldsymbol{u} = \boldsymbol{\omega}_{gyro}$ and $B = \Delta t$. Furthermore for the observation equation (34b), the angle estimate from the accelerometers $\hat{\boldsymbol{\eta}}_{acc}$ is the measurement and H is 1.

This model uses the gyroscope as “input” and the accelerometer angle estimate as observed measurement. This model only tells the Kalman filter, that the measured rates shall be added to the current angles and that $\hat{\boldsymbol{\eta}}_{acc}$ is an measurement of those angles.

Using only the equations of the Kalman filter depending on the state with this model yields

$$\hat{\boldsymbol{\eta}}_p = \hat{\boldsymbol{\eta}}_m + \Delta t \cdot \boldsymbol{\omega}_{gyro} \tag{38a}$$

$$\hat{\boldsymbol{\eta}}_m = (I - K)\hat{\boldsymbol{\eta}}_p + K \cdot \hat{\boldsymbol{\eta}}_{acc} \tag{38b}$$

and concatenated

$$\hat{\boldsymbol{\eta}}_{m,k} = (I - K)\hat{\boldsymbol{\eta}}_{m,k-1} + K \cdot \hat{\boldsymbol{\eta}}_{acc} \tag{39}$$

which is equivalent to (32) with the parameter $\lambda = I - K$ This shows, that a Kalman filter with constant K with the simple model (37) is equivalent to a complementary filter. In other words, a convex combination is, if no further knowledge provided the mean squared error optimal way to combine two measurements, provided a specific choice of the parameter.

The Kalman gain is constant if and only if the matrix P_p converges to a fixed value. This is guaranteed for a linear time-invariant system if 3 requirements are met. First, Q and R the variances of the noise/error signals v and w have to be positive semi-definite, positive definite respectively.

$$Q \succeq 0, \quad R \succ 0 \tag{40}$$

Second, the pair (A, H) has to be detectable, thus

$$rank \begin{bmatrix} A - \lambda_i I \\ H \end{bmatrix} = n, \quad \forall i \; with \; |\lambda_i| \geq 1 \tag{41}$$

with λ_i the eigenvalues of A and n is the dimension of A. Third the pair (A, G) has to be stabilizable, which is equivalent to the rank condition of equation (41), when replacing H with G such that $GG^T = Q$. In words the first condition makes sure, that there actually is noise, because otherwise the Kalman gain would not be positive definite and finite, but rather 0 or ∞. The second condition is the most critical, which is met only, if the measurements are enough to estimate all the states. If two states are related by the model equations, the Kalman filter can estimate both by measurements of only one of them. However if an independent state is not measured, equation (41) requires, that this state is at least stable. The third condition is met, if the noise in the prediction model excites every state, which is guaranteed if $Q \succ 0$ automatically. Practically Q is always positive definite, since this says, that our equations are not perfect, which they never are. The steady state optimal Kalman gain, which can be calculated, if the 3 conditions above are met, can be calculated by solving the discrete time algebraic Riccati equation

$$P_p = AP_pA^T - AP_pH^T \left(HP_pH^T + R \right)^{-1} HP_pA^T + Q \tag{42}$$

and solving for K with equation (36a). This can be done easily with a decent numerical calculation program.

3.2.5 Kalman Filter

The equivalence of the Kalman filter and the complementary filter implies, that the Kalman filter can not do any better than to fine tune a complementary filter gain, given the model (37). Providing a more sophisticated model (more knowledge about the process/phsysics) to the filter however, the Kalman filter can implicitly solve some fundamental problems.

In the following a physically motivated model will be proposed. Because the physics for yaw are different from pitch and roll, the model is presented for one axis (pitch or roll) and then differences for yaw are discussed. The state vector of the used model is

$$\boldsymbol{x} = \begin{bmatrix} \eta \\ {}_I\omega_{IB} \\ e \end{bmatrix}, \tag{43}$$

consisting of the estimated angle, the corresponding rate and e, an estimate of the bias and other additive errors of the gyroscope .

The matrices

$$A = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = {}_BJ_x^{-1} \cdot l \cdot k_F \begin{bmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{bmatrix} \tag{44}$$

and

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{45}$$

are identical for pitch and roll due to their symmetry and the input vectors for pitch and roll

$$\boldsymbol{u}_x = \begin{bmatrix} \gamma_1^2 \\ \gamma_3^2 \end{bmatrix} \quad \boldsymbol{u}_y = \begin{bmatrix} \gamma_2^2 \\ \gamma_4^2 \end{bmatrix} \tag{46}$$

respectively use different motor signals, as described in section 3.1. The measurement vector $\boldsymbol{z}$ differs for pitch and roll from yaw, by their angle estimates

$$\boldsymbol{z} = \begin{bmatrix} \hat{\eta}_{acc} \\ \omega_{gyro} \end{bmatrix}, \quad \boldsymbol{z} = \begin{bmatrix} \hat{\eta}_{mag} \\ \omega_{gyro,z} \end{bmatrix} \tag{47}$$

where yaw uses $\hat{\eta}_{mag}$, the estimate of the yaw angle using the magnetometer, compared to the pitch and roll estimates from the accelerometer. Furthermore

$$B = k_M \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \ , \quad u_z = \begin{bmatrix} \gamma_1^2 \\ \gamma_2^2 \\ \gamma_3^2 \\ \gamma_4^2 \end{bmatrix} \tag{48}$$

have to be used as input matrix and vector for the yaw model.

Substituting these model into the system (34) reveals a discrete model of the equations from section 3.1.

The first row adds the discrete integral of the angular rates to the current angle. This is equivalent to equation (22). Note, that the second state in equation (43) is the angular velocity expressed in inertial coordinates, which we want to find. It is bias free by definition. Thus we only “tell” the Kalman filter, what we want and the bias correction is done implicitly.

The second equation/row is a linearisation of equation (16a). It describes the physics of the vehicle and with the input $\tau_i, \ i = x, y, z$ of equations (17), it is possible to predict the angular velocity, using the motor inputs, which are known.

The third row is an artificial equation to compensate the gyroscopes bias. Written out the model equation is $e_k = e_{k-1}$ telling, that the bias is to be expected constant. And the observation equation is

$$z_2 = \omega_{gyro} = {}_I\omega_{IB} + e \ , \tag{49}$$

which says, that the gyroscopes measure the angular rate, but corrupted by the error e. Using those equations, the Kalman filter implicitly estimates the error of the gyroscopes by comparing their integration, according to equation (22), to the angle estimate from the accelerometer or magnetometer. It is then used to correct the estimated error of the gyroscope, again implicitly. Therefore the first two state variables can be used directly as angle and angular rate. These two completely define the state of the attitude of the vehicle, and therefore solve the state estimation problem.

3.3 Simulation

A simulation of the presented mathematical model was developed using explicit one step forward numerical integration. To obtain realistic results and make them comparable with the RC transmitter PWM input, the motor thrust was statically measured. The used moments of inertia were calculated using a simple point mass model and a box in the center of the vehicle. For simulation purposes the non-linear system of equations (16) was used. The simulated model also includes the nonlinearities of the rotational equations and the nonlinear transformation of the velocities into the inertial frame according to equation (10).

Figure 3 shows the results of the simulation. The initial condition of the shown simulation include a random disturbance in pitch and yaw angular velocities. This corresponds to a disturbance acting upon the vehicle and introducing an angular motion, from which the system has to return to the setpoint. At $t = 0.1$ the setpoint is changed to investigate the systems step response. The system follows fast and accurately.

It is observed, that the simulation shows unrealistically fast time constants, most certainly caused by the following effects. Uncertainties resulting from the estimated moments of inertia and aerodynamic effects like extra mass moved with the spinning rotors. The neglected motor dynamics play a role here too and will be discussed further below. A hardware validation of the model is a next step to be taken. The complex damping effects of the real life model could be incorporated by a system parameter identification. Another approach to accomplish more realistic results would be to add virtual moments of inertia to the mathematical model to compensate for not modelled effects.

Figure 4 shows the root locus of the system as presented above. It shows two poles originating at the origin, with higher gains, moving through a circle as complex conjugated poles before recombining on the negative real axis. One branch thereafter gets faster, whereas the other one gets slower and reaches a finite value as the gain goes to infinity. There is no branch reaching into the right half plane, thus with this simple model it is impossible to destabilize the system. Obviously this is a sign of an oversimplified model.

For realistic gains the motor dynamics are important, since the torque the controller needs to apply can not be provided instantly. With this fact in mind a simplified motor model was included. Since we know that the motor has low-pass

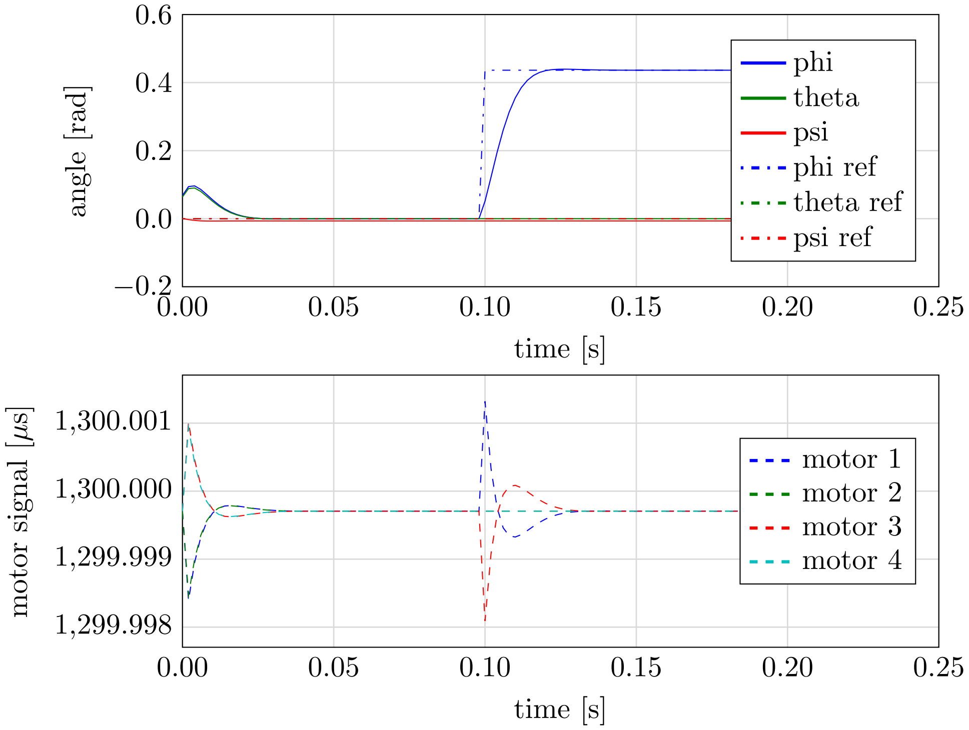


Figure 3: The above plot shows the 3 attitude angles for a random disturbance rejection and input step. The lower plot shows the corresponding changes in motor speeds for the 4 motors.

behaviour, a PT1 low pass filter is used to represent the motor response. Figure 5 shows the modified root locus, where the two conjugate complex branches reach into the unstable region. With a faster zero the two conjugate complex branches join and yield a smaller stable region and less damping.

4 Hardware

4.1 Wiring

High currents (up to 80 Ampere for 15 seconds, depending on the motor type and size of the vehicle) and the danger of getting a wire into the rotors, demand special care for an appropriate wiring. A minimalistic approach in terms of signal wiring and a well-suited power supply and distribution wiring has to be guaranteed.

Before testing single parts, one has to make sure not only that each part is

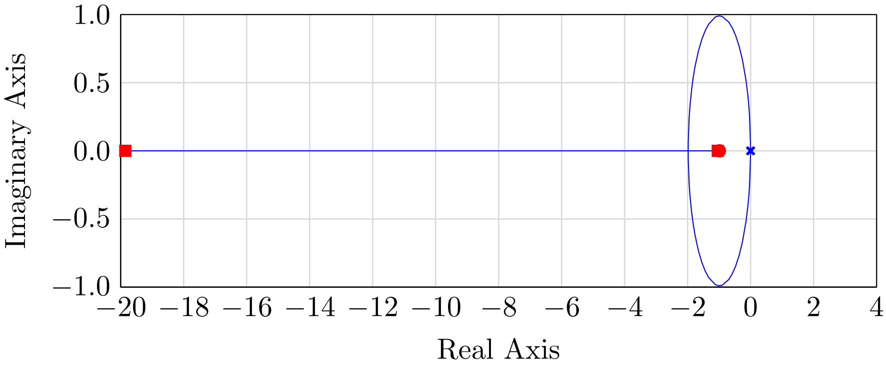


Figure 4: The root locus of the simple system with no branch going into the right half of the complex plane.

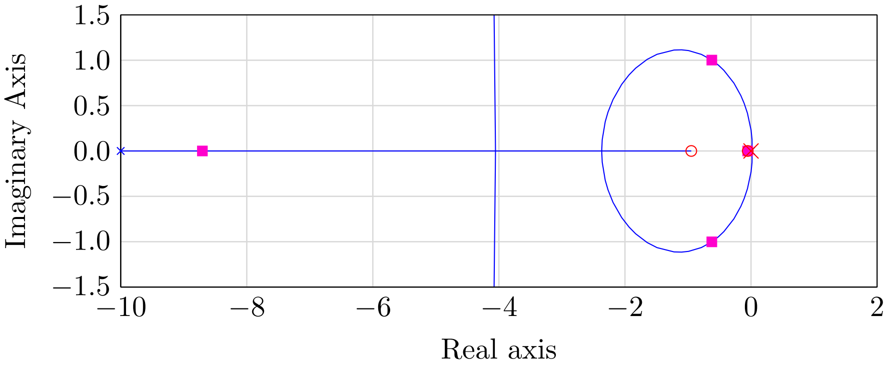


Figure 5: Root locus with simple motor model incorporated. Another branch reaches out at about −4 and bifurcates with different zero position of the controller.

wired correctly, but also that a security switch is placed in the circuit and software. Especially when dealing with 4 motors one has to take precautions in the mechanical setup (fix motors) and logical setup (emergency stop).

A common ground has to be set up for all connected components. A different ground reference can lead to undetermined behaviour. Especially with signal wiring this can lead to a fatal system failure.

4.2 Motors

In the past few years the market for motors for model aircraft was conquered by brushless DC motors. This type of motor offers an unmatched power to weight ratio and is therefore well-suited for various types of small air vehicles.

The major drawback of brushless motors is the complexity of controlling their speed. Unlike brushed motors, a brushless motor has to be controlled by an alternating 3-phase current. The electronic speed controller (ESC) alternates the 3-phased current to rotate the magnetic field in the stator to turn the motor with the desired speed. Also electromagnetic effects are used to determine the effective speed and thereby control the motors appropriately. The availability of ESC made the use of brushless motors viable. Even with the extra weight of the ESC, brushless motors surpass conventional motors in power to weight ratio and thus are the preferred motor type for aerial vehicles. The specifications of the used motors are given in table 1.

type	outrunner
cellnumber	2-3 (LiPo)
propellers	2S 12x8; 3S 10x6
dimensions body/shaft	30/3,2 mm
mass	55 g
“kV”-value	980 rpm/volt
max current (15s)	17 A
max permanent current	13 A
max power (15s)	200 W
internal resistance	132 MΩ

Table 1: Specifications of the used motors.

4.3 Battery

The principal concern in battery selection, just like with motors, is the power-to-weight ratio. The greatest power density battery type is Lithium Polymer (LiPo). These batteries are available in a wide variety of sizes, capacities and maximum current drawn. A typical LiPo consists of 3.7 Volt cells coupled in series. A 3-Cell (labeled 3S) LiPo pack therefore has 11.1 Volts nominal voltage. On model battery packs the maximum discharge rate is given by the “C” rating. The “C” rating has to be multiplied by the capacity of the pack in Ampere hours to give the maximum current to be drawn in Amperes. These specifications for the battery used on the current setup are given in Table 2.

Since LiPo packs are sensible to varying voltages across individual cells, they have to be balanced. They should never reach voltages below 3.7 Volts per cell nor higher voltages than 4.2 Volts per cell. Especially during charging good care has to be taken, as LiPo batteries can catch fire if handled incautiously.

cellnumber	3 S
voltage	11.1 V
capacity	2200 mAh
C rating	30 C

Table 2: Specifications of the utilized battery pack.

4.4 Propellers

The choice of propellers has a great effect on the efficiency and thrust of an aerial vehicle. There are fixed or variable pitch propellers available. A variable pitch actuation features a fast response to changes and can be used for 3D flight. This, however, comes at the expense of a more complex mechanical setup. The asymmetric shape of a fixed pitch propeller has advantages in efficiency, since it is designed to produce a maximum thrust for the running direction. With fixed pitch rotors, the motor has to vary the speed of its rotor. This leads to a slower dynamic response and demands electric motors. Since the dynamics of recent motors are sufficiently fast and for ease of use, fixed pitch propellers were used here.

The most important parameters of a propeller are pitch and the diameter of the blade. A steeper pitch allows for more thrust, but also demands a higher motor torque (lower KV rating on brushless motors). The diameter should be chosen large for greater efficiency but is limited by the available torque. The diameter needs to fit the size of the vehicle as well. For this project propellers with a pitch of 4.7 inches (119.38 mm) and 8 inches (203.2 mm) diameter were chosen.

4.5 Structure

There are many possible configurations of quadrotors and other multirotors. Amongst them are setups with 2 longer and 2 shorter arms. Here a design with two lines of symmetry is used. All arms are of the same length as can be seen in Figure 1. This allows for more predictable dynamics and a wider range of use.

The material used for the connection between motor and base has to be stiff enough to hold the motor in place while still providing enough flexibility to reduce damage in case of crashes. Possible materials are different kinds of wood and aluminium amongst others. Carbon fiber however fulfils these demands and also provides an excellent stiffness to weight ratio. Additionally carbon fiber rods reduce vibrations introduced by the motors and propellers.

4.6 Microcontroller

To implement the controller and calculate data from sensors and the remote control, a micro controller with low power hardware has to be used. Due to high availability and the ease of implementation of a prototype platform, the Arduino was used. This micro controller consists of a AT-mega 328p processor and has a built-in voltage divider as well as various other tools to simplify rapid prototyping. Code for the Arduino hardware is written in C++ or Assembly. Since it is wide spread and adopted to multiple fields, there are many libraries and many convenient functions available.

4.7 Sensors

To control a flying vehicle, responsive and small sensors are mandatory. In the past years there have been great advances reducing sensor sizes, based on micro-electro-mechanical systems (MEMS). For the current setup a GY-86 sensor was used. It integrates a gyroscope, accelerometer, magnetometer and barometer as well as a thermometer in a single breakout board. While a gyroscope and an accelerometer would suffice to estimate the roll and pitch angles, a drift of the gyroscope about the yaw axis can just be compensated by an extra magnetometer. The barometer can be used to roughly estimate the height of the vehicle by using models of the atmospheric pressure. The thermometer can be used to compensate for the temperature drift of all the mentioned sensors. The sensor communicates over the I^2C communication protocol.

5 Software

The code presented is optimized for speed on the AVR microcontroller. An example optimization can be found in chapter 5.1, where the low level registers of the microcontroller are used to make direct use of the hardware by bypassing the Arduino-API. Other techniques for speed-optimizations are given in [14].

Particularly useful for the given task is the unrolling of loops. This is avoiding short loops and hard code each iteration. As a side effect the use of arrays can be avoided this way.

Another method which finds wide application in the presented code is to reduce the use of multi-byte data types and floating point variables. Care was taken to use the minimal suitable byte datatype. For example instead of boolean type variables, a single char (uint8_t) was used to store up to 8 boolean values. Binary operations have to be used to deal with these.

5.1 RC Receiving

Common 2.4 GHz radio control receivers are built to work with electronic speed controllers (ESC), used with brushless motors, as well as with servo motors. Both of these act upon the same digital signal type, which will be described in the following section.

5.1.1 Signal Type - PWM

PWM stands for Pulse Width Modulation. Since this term is used with other Hardware in a different manner, it has to be noted, that the actual signal of a 2.4 GHz receiver is neither typical PWM nor PPM (Pulse Position Modulation). PWM is defining a signal by the relation between the time a digital signal is held high and low. A receiver however sets the outgoing signal high for a specific amount of time for each channel. This timespan, typically between 1000 and 2000 μs is directly used. The specific span depends on settings (digital trim) and the hardware used.

PPM receivers on the other hand emit the so-called PPM train, which is comprised of multiple PWM signals on one line. The difference is that the signal is not held high, but only the rising edges are taken into account. For the last channel an artificial edge is used to terminate the signal. Thereafter, the signal is held low to guarantee a constant sampling rate. Since many channels can be transmitted on a single wire this method would be preferable. PPM signals are, however, rarely found on model receivers, since the end user (or device) would have to de-multiplex the signal for different motors.

5.1.2 Receiving Technique

The RC receiver used ("Spektrum AR600") operates all channels separately. Therefore the signal can be read by reading individual channels, or multiplexing the channels and produce a PPM stream and read that. Since the latter requires extensive use of electronic components and works only for PWM signals, that follow a constant channel order, the chosen method was to read the individual channels.

The Arduino Uno can handle only 2 external interrupts. To access 4 channels (the minimum for a quadcopter, since this is the number of inputs) another type of interrupt has to be used: internal interrupts. This was done with basic Arduino functions and libraries and later by exploiting the processors registers directly.

Internal interrupts can be accessed by using the Arduino 'Pin Change Interrupts' Library. Hereby interrupts are registered to the desired pin and callback functions are defined which measure the signal time of the respective channel.

This method has a severe drawback: the interrupts are handled within the library, which has to determine which pin triggered the interrupt. This takes some microseconds to calculate. If more than 1 interrupt occurs simultaneously, the microprocessor queues the interrupts and processes them with a "first come, first serve" principle. This delays the following interrupt callbacks and thereby directly distorts the received signal. The effect was observed by analyzing the signal during a one minute test, without changes on the transmitter. For the test just 3 channels were used, of which 2 are shown here. Shown are the channels for pitch and roll angles, which are the most critical inputs.

To avoid the clashing effect observed with the given library, the internal interrupt functions were used directly through the processors registers. Not only does this method speed up the interrupt service routines (callbacks) but it also eliminates the overhead of calling a user-defined function.

The most significant difference of the two presented methods is that, with common functions, the pin which triggered the interrupt has to be evaluated. Since the Arduino pins are divided into 3 ports, which trigger their own interrupt, only one pin of each port was used to eliminate the need for defining which pin of the respective port triggered the interrupt. This choice has a significant effect as shown in Table 3. Measurements of a constant signal (1500 μs) are received on two channels over a one minute period.

	exploit	exploit	lib	lib
min	1484	1488	1480	1480
max	1500	1504	1512	1512
std	1.184	1.152	2.046	2.339
mean	1492.3	1495.8	1495	1498

Table 3: Constant signal readings with Arduino Library code and AVR Register usage.

The offset of a few micro seconds for both methods appear to be the same. This is a result of the internal oscillators relative error of up to 10%. The standard deviation however is drastically reduced. The difference between maximal and minimal reading is reduced by a factor of two as well. These differences which were measured without any other calculations and interrupts, become greater when more channels and other interrupts are being used.

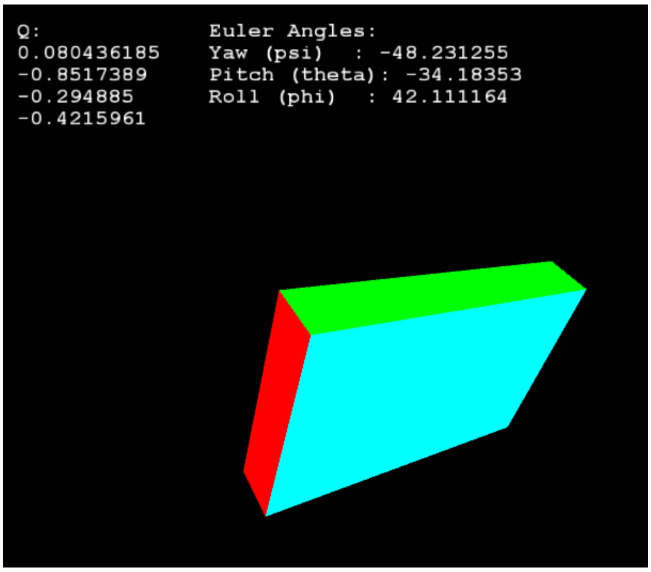
5.2 Motor Commands

The signal type for the electronic speed controllers (ESC) for the motors is the same as the one coming from the RC receiver. It can be produced by timing the output pins of the microcontroller to be set high/low in a specific interval. The outgoing signal was produced using the Arduino library “servo”. The library was written by Michael Margolis and others [15]. It is widely spread and is proven to work for various tasks.

The “servo” library uses the 16 bit timer TCNT1 on the AVR microcontroller. For a more precise way of reading the RC receiver, this timer would be used for timing the outgoing and incoming signals. As an example this approach can be found at [16], in another context. This however is not compatible with the servo library. The small gain in accuracy was considered not to be of priority for the overall behaviour, so for the output the “servo” library was used.

5.3 Sensors

To validate the used estimation method, a graphical representation of the sensor was used (see Figure 6). A steady-state Kalman filter with constant coefficients was used to reduce the payload on the micro-controller. All it has to process is the control algorithm and a few floating point calculations. This can be traded to using the computationally more expensive extended Kalman filter (EKF). Another possible setup would use EKFs for pitch and roll, while the rather cheap steady state Kalman filter for yaw estimation, which is less time critical. To guarantee manoeuvrability and stable operation in acrobatic or 3D flight, the non-linearity of the rotations (the Coriolis terms) can't be neglected or linearized. For advanced filtering techniques see for example [17, 18] (EKF) or [7] (Lyapunov estimation).



```
Q:             Euler Angles:
0.080436185    Yaw (psi)    : -48.231255
-0.8517389     Pitch (theta): -34.18353
-0.294885      Roll (phi)   : 42.111164
-0.4215961
```

Figure 6: Graphical representation of the sensor used to validate the fusion algorithm.

6 Conclusion

The goal of constructing and building a quadcopter at low costs has been achieved. Also the dynamics of the systems were investigated experimentally and numerically. Due to the high complexity of the interaction between electronics, software and hardware as well as complex side effects not all features which came up along the a-priori research could be implemented. Along with not yet implemented features there are various research topics in the field of quadcopters and their dynamics and control. Already mentioned extending features include further mathematical modelling and a reduced size flight controller by using the microcontroller directly on an integrated circuit board. One of the most promising improvements is a more sophisticated control algorithm. With the precise knowledge about the system state a state space controller can be used and control performance can be improved. Further measurements are done to model the dynamics of the motor. Along with these algorithm changes the motors can be tilted to improve yawing performance. The use of the barometer would lead into transversal dynamics and guided or planned flight. With GPS autonomous flight can be extended in range and precision. Furthermore a vast number of controlling observation and route tracking methods can be implemented and tested on a cheap but versatile system. There is a research field in life sized multirotor vehicles revived since the beginning of the VTOL (vertical take off and landing) vehicles. A new approach to this would include variable pitch blades used along with combustion engines. This would on the one hand lower the mechanical complexity to a minimum and eliminate the need for higher powered batteries. Another possibility is using impellers and deflectors instead of motors with blades. Limited capacities are the most limiting factor for bigger multirotors. Therefore the two proposed setups could lead to very versatile and stable VTOL vehicles with various advantages over conventional helicopters.

A Code Content

The appendix describes the content of the code files appended this work.

The program **flip.ino** contains Arduino compatible C++ code to control the quadcopter. It contains high level logics, such as a security lock and the control algorithm. Multiple classes are used in this top-level file.

The **RC.cpp** and **RC.h** files contain hardware specific code for the Atmel ATmega AVR328p and AVR32u4, where the model can be toggled. It is used to read the radio controller values on the respective platform. The used pins and further details can be found in the comments of the header file.

GY86.cpp and its header **GY86.h** define a high level class for the sensor breakout board GY86, containing 3-axes gyroscopes, accelerometers and magnetometers. For the communication to the sensor the file **I2C.h** contains I^2C protocol wrapper functions.

uMPU6050 is the sub-class for the gyroscopes and accelerometers, the breakout board MPU6050. It contains only the most basic features to reduce the workload.

HMC5883L is the corresponding sub-class for the magnetometers.

Kalman.cpp and **Kalman.h** contain a time-variying Kalman filter for the proposed system. It can easily be modified to an extended Kalman filter.

sskf.cpp and **sskf.h** can be used to replace the Kalman filter by a time-invariant steady state version. It is computationally optimized and uses the same system, thus giving the same estimates, in the limit of time going to infinity, as the Kalman class.

References

[1] Gabriel Hoffmann Haomiao Huang. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, August 2007. AIAA, AIAA Guidance, Navigation and Control Conference and Exhibit.

[2] William Premerlani and Paul Bizard. *Direction Cosine Matrix IMU: Theory*, 2014.05.17 edition, May 2009.

[3] Ardupilot - open source unmanned aerial vehicle (uav) platform, http://ardupilot.com/.

[4] Starlino-Electronics. DCM tutorial - an introduction to orientation kinematics, http://www.starlino.com/dcm_tutorial.html, May 2011.

[5] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles.* Springer Publishing Company, Incorporated, 1st edition, 2010.

[6] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *Full Quaternion Based Attitude Control for a Quadrotor*, European Control Conference, July 2013.

[7] Joao Cafe. Flight control and attitude estimation of a quadrotor. Technical report, Instituto Superior Tecnico.

[8] Christoph Glocker. Multibody dynamics. ETH.

[9] Reza N. Jazar. *Theory of Applied Robotics.* Springer, second edition, 2010.

[10] S. Widnall J. Peraire. 3d rigid body dynamics: Euler angles. Technical report, MIT, 2009.

[11] Talat Ozyagcilar. Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. Technical report, Freescale Semiconductor, 2012.

[12] R.T. Rockafellar. *Convex Analysis.* Convex Analysis. Princeton University Press, 1997.

[13] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* Wiley, 2006.

[14] Atmel Corporation, Parkring 4 D-85748 Garching by Munich. *Atmel AVR4027: Tips and Tricks to Optimize Your C Code for 8-bit AVR Microcontrollers*, rev. 8453a-avr-11/11 edition, 2011.

[15] Arduino servo library http://arduino.cc/en/Reference/Servo.

[16] Duane B. RC-Arduino Blog http://rcarduino.blogspot.co.at/2012/11/how-to-read-rc-receiver-ppm-stream.html.

[17] Sergei Lupashin and Raffaello D'Andrea. Adaptive fast open-loop maneuvers for quadrocopters. *Autonomous Robots*, 33:89–102, 2012.

[18] Bernardo Sousa Machado Henriques. *Estimation and Control of a Quadrotor Attitude.* PhD thesis, University of Lissabon, June 2011.