School of Engineering & Design
Electronic & Computer Engineering

MSc Distributed Computing Systems Engineering

Brunel University West London

Simulation and Performance Analysis of a Distributed Position Correction Scheme for Unmanned Aerial Vehicles

Dionysios Satikidis

Supervisor: Dr. Marios Hadjinicolaou

March / 2011

A Dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science

School of Engineering & Design
Electronic & Computer Engineering

MSc Distributed Computing Systems Engineering

Brunel University West London

Simulation and Performance Analysis of a Distributed Position Correction Scheme for Unmanned Aerial Vehicles

Dionysios Satikidis

Signature: ___________________

Declaration: *I have read and I understand the MSc dissertation guidelines on plagiarism and cheating, and I certify that this submission fully complies with these guidelines.*

Abstract

Abstract summary ...

Acknowledgements

Acknowledgements ...

Contents

List of Abbreviations

API	Application Programming Interface(s)
CACD	Camera Calibration Domain
CCU	Central Control Unit
CLCS	Closed Loop Control System(s)
DOF	Degrees of Freedom
DSP	Digital Signal Processor
EOM	Equations of Motion
EVTF	Error-Value-Transfer-Function
FPGA	Field-Programmable Gate Array
GAV	Generalized Acceleration Vector
GPV	Generalized Position Vector
GUAV	Gliding Unmanned Aerial Vehicle(s)
GVV	Generalized Velocity Vector
HIL	Hardware In the Loop
HSE	University of Applied Sciences Esslingen
HUAV	Hovering Unmanned Aerial Vehicle(s)
HWBIP	Hardware based image processing
IMU	Inertial Measurement Unit

MAV	Micro Aerial Vehicle(s)
MBD	Model Based Design
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical System
MIMO	Multiple-Input and Multiple-Output
OFCA	Off-board camera
OFIP	Off-board image processing
ONCA	On-board camera
ONIP	On-board image processing
PID	Proportional-Integral-Derivative
RF	Radio Frequency
SBC	Single Board Computer(s)
SISO	Single-Input and Single-Output
SLAM	Simultaneous Localization And Mapping
SOAD	Shift of the Optical Axis Domain
SVTF	Set-Value-Transfer-Function
SWBIP	Software based image processing
TF	Transfer Function
UAV	Unmanned Aerial Vehicle(s)
WRT	with respect to

List of Symbols

X_E	X-axis WRT E-frame
Y_E	Y-axis WRT E-frame
Z_E	Z-axis WRT E-frame
ϕ	Angular quadrocopter position around X_E WRT E-frame (Roll)
θ	Angular quadrocopter position around Z_E WRT E-frame (Yaw)
ψ	Angular quadrocopter position around Y_E WRT E-frame (Pitch)
x_e	Linear quadrocopter position along X_E WRT E-frame
y_e	Linear quadrocopter position along Y_E WRT E-frame
z_e	Linear quadrocopter position along Z_E WRT E-frame
$\dot{x}_e$	Linear quadrocopter velocity along X_E WRT E-frame
$\dot{y}_e$	Linear quadrocopter velocity along Y_E WRT E-frame
$\dot{z}_e$	Linear quadrocopter velocity along Z_E WRT E-frame
$\ddot{x}_e$	Linear acceleration of quadrocopter along X_E WRT E-frame
$\ddot{y}_e$	Linear acceleration of quadrocopter along Y_E WRT E-frame
$\ddot{z}_e$	Linear acceleration of quadrocopter along Z_E WRT E-frame
X_B	X-axis WRT B-frame
Y_B	Y-axis WRT B-frame
Z_B	Z-axis WRT B-frame
p	Angular quadrocopter velocity around X_B WRT B-frame (Yaw)
q	Angular quadrocopter velocity around Y_B WRT B-frame (Roll)

r	Angular quadrocopter velocity around Z_B WRT B-frame (Pitch)
x_b	Linear quadrocopter position along X_B WRT B-frame
y_b	Linear quadrocopter position along Y_B WRT B-frame
z_b	Linear quadrocopter position along Z_B WRT B-frame
u	Linear quadrocopter velocity along X_B WRT B-frame
v	Linear quadrocopter velocity along Y_B WRT B-frame
w	Linear quadrocopter velocity along Z_B WRT B-frame
$\dot{u}$	Linear quadrocopter acceleration along X_B WRT B-frame
$\dot{v}$	Linear quadrocopter acceleration along Y_B WRT B-frame
$\dot{w}$	Linear quadrocopter acceleration along Z_B WRT B-frame
ω	Speed of rotation
$\Delta\omega$	Deviation of rotation speed
ξ	GPV WRT E-frame
$\dot{\xi}$	GVV WRT the E-frame
ν	GVV WRT the B-frame
J_Θ	Transformation matrix from B-frame to E-frame
R_Θ	Rotation transformation matrix from B-frame to E-frame
T_Θ	Translation transformation matrix from B-frame to E-frame
Γ^E	Linear position vector WRT the E-frame
Θ^E	Angular position vector WRT the E-frame
V^B	Linear velocity vector WRT the B-frame
ω^B	Angular velocity vector WRT the B-frame

$\dot{\Gamma}^E$	Linear velocity vector WRT the E-frame
M_B	System inertia matrix WRT the B-frame
M_H	System inertia matrix WRT the H-frame
$C_B(\nu)$	Coriolis-centripetal matrix WRT the B-frame
$C_H(\zeta)$	Coriolis-centripetal matrix WRT the H-frame
m	Mass of body
I_{nxn}	Identity matrix of dimension n times n
I	Body inertia matrix
$G_B(\xi)$	Gravitational vector WRT the B-frame
G_H	Gravitational vector WRT the H-frame
$O_B(\nu)$	Gyroscopic propeller matrix WRT B-frame
$O_H(\zeta)$	Gyroscopic propeller matrix WRT H-frame
Ω	Sum of propellers' speed
E_B	Movement matrix WRT B-frame
$E_H(\xi)$	Movement matrix WRT H-frame
b	Factor of thrust
d	Factor of drag
l	Distance between centre of quadrocopter and centre of propeller
$\dot{\nu}$	GAV WRT B-frame
ζ	GVV WRT H-frame
$\dot{\zeta}$	GAV WRT H-frame
F^B	Forces vector of quadrocopter WRT B-frame

τ^B	Torques vector of quadrocopter WRT B-frame
$\dot{V}^B$	Linear acceleration vector WRT the B-frame
$\dot{\omega}^B$	Angular acceleration vector WRT the B-frame
U_1	Vertical thrust WRT the B-frame
U_2	Roll torque WRT the B-frame
U_3	Pitch torque WRT the B-frame
U_4	Yaw torque WRT the B-frame
$F_{n\{X_B,Y_B,Z_B\}}$	Thrust component of rotor n along X_B, Y_B or Z_B

List of Figures

1 Introduction

1.1 Context of Project

1.2 Aims and Objectives

2 Methodology and Project Management

2.1 Model Based Design

2.2 Software Development Process

2.3 Tools and Architectures

2.4 Risks and Limitations

2.5 Project Management

3 Literature Review

The realization of an optical distributed error correction scheme for Unmanned Aerial Vehicle(s) (UAV) contains several challenges of different topics of technical information technology, which have to been introduced for this project in the context of a literature survey. These topics have to been with the focus to create a simulation. That means the theory of function and the complexity of these topics have to be examined and possibilities of simplification have to be given.

The first part of the survey will introduce the basic quadrocopter flight dynamics and approaches to derive the mathematical model of motion. The result of this model will be the Equations of Motion (EOM) in a form which can be used for simulation and reflects as much as possible the reality.

A comparison of the researched existing visual approaches for error correction of UAV will be the focus of the second part of this survey. Thereby the strengths and weaknesses in focus of different aspects have to be given. Furthermore this part will introduce optical movement detection technique which will be investigated here under consideration of the limitations and the abstraction of the reality.

Finally the third part of the survey will give an overview of control approaches in which the determined movement detection value can be processed for aircraft stabilisation. Further methods will be introduced which allow to measure and classify the performance of a control system and its characteristics under consideration of digitalisation aspects.

3.1 Quadrocopter Flight Dynamics

3.1.1 Principles

For a better appreciation of the quadrocopter flight dynamics, the following figure[1] visualizes the influence of the thrust in relation to the movements in the Degrees of Freedom (DOF). Thereby the values of ω [rad/sec]represents the least needed speed of rotation, for creating the required thrust for the hovering state. This state can be described as a state, in which forces in the x- and y-axis equals zero and the uplift force in direction of the z-axis has the same absolute value as the gravitational force. The value of $\Delta\omega$ characterizes the purposed deviation of the required rotation speed in hovering state and is used for navigation of the quadrocopter.

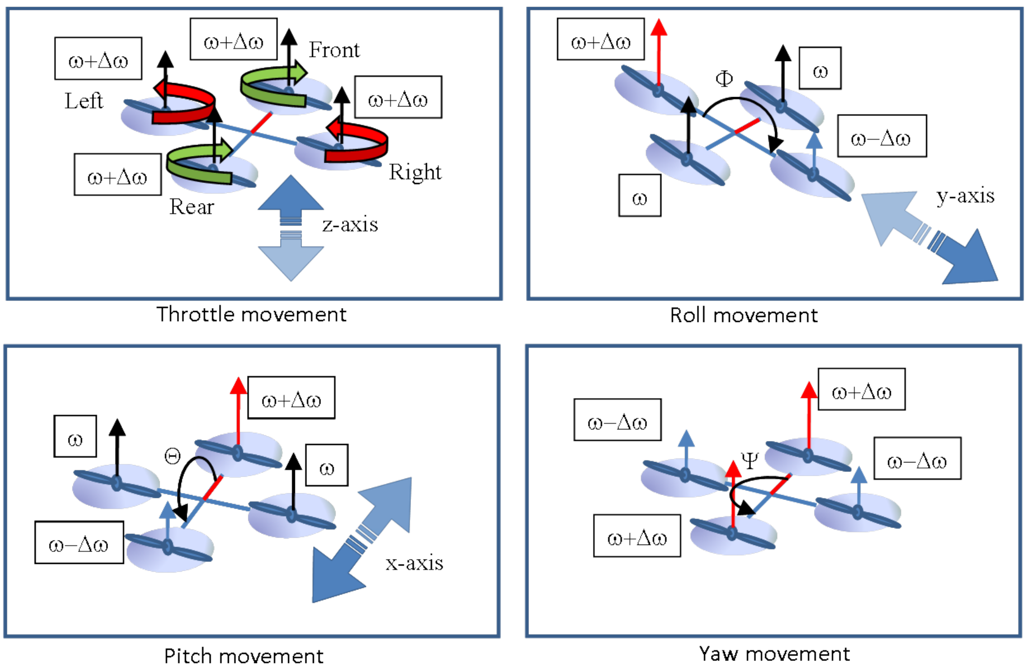

Figure 3.1: Degrees of freedom of a quadrocopter

[1]This picture extends the pictures presented in[Tom08, pp.8-11 Basic concepts]

To keep the equilibrium of rotational kinematics and to prevent self-rotation, the motor direction of rotation equals crosswise. The quadrocopter has six DOF which can be distinguished as angular and translational movements. Translational movements can be executed in x-, y- and z- axis. Accelerating all rotors with the same speed ω to the value of $\Delta\omega$ will affect a throttle movement U_1 [N] in z-direction. Movements to the negative direction of the z-axis are possible, if the thrust of the four rotors is smaller as the gravitational force of the aerial vehicle. The roll movement U_2 [N m] can be described as a change of the angle around the x-axis. Thereby the left and right rotors execute a force difference by slowing down the one and simultaneous increasing the other speed with $\Delta\omega$. Related to the thrust difference and angular movement, the aerial vehicle creates a force to the y-axis. Equivalent to roll, the pitch movement U_3 [N m] is executed with a change of the angle around the y-axis. Also the pitch movement creates a translational movement across the x-axis. Pitch and roll can only reach a stable angular state and accelerate to the x- or y-axis, if the value of $\Delta\omega$ is the same at the diagonal rotors. Otherwise, the quadrocopter would pure rotate across the corresponding axis. The yaw movement U_4 [N m] is a rotation around the z-axis. This angular movement results in combination of pairwise different thrusts and takes as long as these thrusts are different [Tom08, pp.8-11 Quadrotor model and system].

3.1.2 Equations of Motion

Simulations of dynamic systems are based on physical models which describe their motion. So the behaviour of the quadrocopter also can be described by EOM, with respect to (WRT) the input parameters of the model. Thanks to these equations, it is viable to predict and define the positions, velocities and accelerations of the quadrocopter by investigating the four motor speeds. Bouabdallah [Sam07, pp.15-24 System Modelling] demonstrates two methodologies to derive the EOM with the Newton-Euler and Euler-Lagrange formalism. The Euler-Lagrange formalism derives the EOM by calculating the difference of the kinetic and potential energy of the system under consideration of the general coordinates and forces [2]. The Newton-Euler formalism bases on the Newton-Euler equations [Roy08, p.106 The original recursive Newton-Euler equations] which describe the combined translational and rotational dynamics of a rigid-body with respect to the centre of mass. Both formalisms follow different approaches to derive the EOM, but are based on the same coordinate systems. These two coordinate systems have different origins and describe movements from different perspectives. One coordinate system can be considered as observer perspective to the quadrocopter because it describes the absolute position in space and is called earth inertial frame (E-frame). In contrast to that, the second coordinate system is a body fixed frame (B-frame) and is defined as system of relative movements which originates in the middle of the symmetric rigid quadrocopter body [3]. Bresciani [Tom08, pp.8-23 Quadrotor model and system] shows in his work the identification and derivation of the EOM by using the Euler-Newton formalism and gives reasons why EOM are more conveniently formulated in the B-frame or further in a mixture called hybrid frame (H-frame) [4]. A part of these congenial reasons focus the simplicity to convert on-board measurements to the B-frame coordinates and the matter of fact that the control forces almost are given in B-frame. The other part describes simplifications in the mathematical derivation

[2] See [Dav07, pp. 218-219 The Lagrangian Method]

[3] An origin which is coincidence to the centre of mass simplifies the Newton-Euler equation enormous, See [Roy08, p.98 The original recursive Newton-Euler equations]

[4] See [Tom08, p.19 Hybrid frame]

causes of B-frames symmetrical behaviour and time invariance of the inertias [5]. In the following figures[6], we can see the both coordinate systems and the movements in the DOF which are interesting.

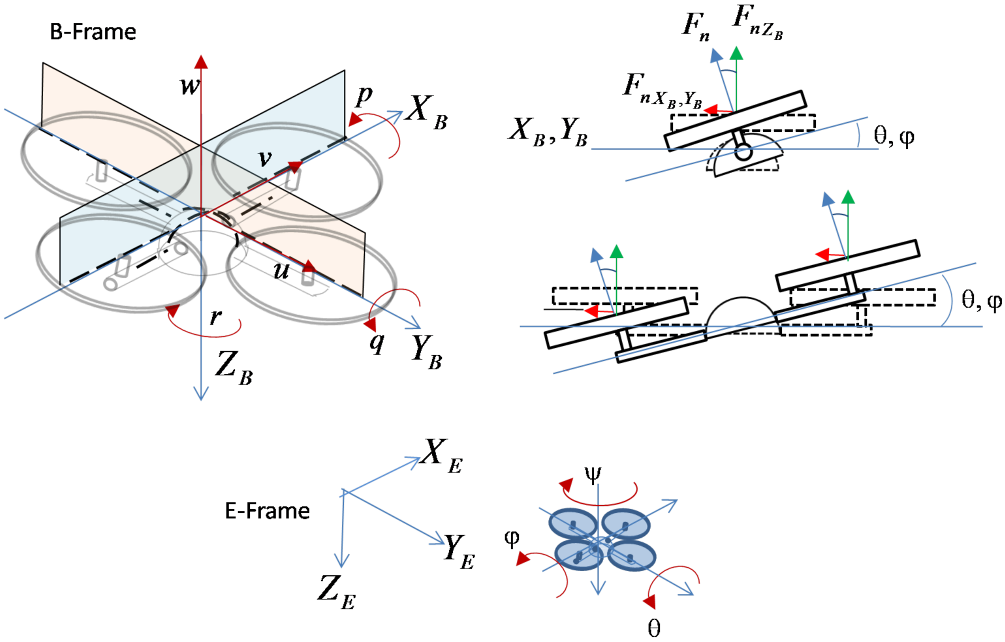

Figure 3.2: Quadrocopter Movements and Coordinate Systems

The starting point of Brescianis derivation is to define an equation, which describe a generic 6 DOF rigid-body (See 3.1) with the Generalized Velocity Vector (GVV) $\dot{\xi}$ WRT the E-frame, the GVV ν WRT the B-frame and the generalized matrix J_Θ which includes the rotation and the translation submatrix R_Θ, T_Θ which transforms ν to $\dot{\xi}$ (See 3.3). The definition of the GVV and the Generalized Position Vector (GPV) WRT the corresponding frames are visualized in figure 3.2 and can be described with the equations 3.4 and 3.2. Thereby ξ is composed of the linear position vector Γ^E [m] and the angular position vector Θ^E [rad]. Similar to the position vector, the velocity vector is composed of a linear velocity vector V^B [m/s] and an angular velocity vector ω^B [rad/sec]. The essence frame of in-

[5] See [Tom08, p.12 Conveniently formulated EOM]
[6] This picture extends the coordinate system picture presented in[Sam07, p.20 OS4 coordinate system]

terest is a mixture of the both mentioned frames, called H-frame, which contains the linear velocity vector $\dot{\Gamma}^E$ [m/s] from E-frame and the angular velocity vector ω^B [rad/s] from B-frame 3.5. Brescianis interest was to define a frame which can be transferred to an acceleration vector WRT the H-frame and to facilitate the EOM under consideration of typical Inertial Measurement Unit (IMU) values of a quadrocopter.

Bresciani uses the Euler-Newton equations (See 3.6)[7] to derive the matrix form which is composed of the system inertia matrix M_B and a Coriolis-centripetal matrix $C_B(\nu)$. A 6 DOF rigid-body dynamics system inertia matrix takes the mass of the body m [8] [kg] and its inertia I $[Nms^2]$ into account.

Furthermore Bresciani recognized that the quadrocopter dynamics can be divided to three contributors which describe the gravitational vector $G_B(\xi)$ [9], the gyroscopic torque $O_B(\nu)*\Omega$[10] and the movement vector with the inputs of the system $E_B*\Omega^2$(See 3.6) [11]. So it is possible to create a relation between the Newton-Euler equations (See 3.6) and the sum of these three contributors (3.6), to isolate the Generalized Acceleration Vector (GAV) $\dot{\nu}$, and to derive the EOM WRT the B-frame (See 3.8). Equivalent to that, the substitution of GAV $\dot{\nu}$ with GAV $\dot{\zeta}$ WRT the H-frame in 3.7, gives an equation which can be solved to $\dot{\zeta}$ for the determination of the required EOM and a quadrocopter model with typical IMU values (See 3.9).

[7] See [Tom08, p.13 The dynamics of a generic 6 DOF rigid-body]

[8] The notation I_{3x3} means 3 times 3 identity matrix (See I_{nxn})

[9] $G_B(\xi)$ considers the acceleration of gravity g $[m/s^2]$ and influences the linear forces, See [Tom08, p.15 The gravitational vector]

[10] $O_B(\nu)*\Omega$ considers the gyroscopic effects of the propeller rotation which influence the angular forces. Thereby $O_B(\nu)$ is the propeller matrix and Ω $[rad/s]$ the proppelers' speed vector See [Tom08, p.16 The gyroscopic torque]

[11] The constant matrix E_B 3.11 includes the thrust b $[Ns^2]$, $d[Nms^2]$and ll [m] (distance between the center of the quadrocopter and the and the center of a propeller) factors of the input system. The inputs are given with Ω^2 because the forces and torques are proportional to the squared propellers' speed, See [Tom08, pp.129-135 Aerodynamics calculation]

$$\dot{\xi} = J_{\Theta} * \nu \tag{3.1}$$

$$J_{\Theta} = \begin{pmatrix} R_{\Theta} & 0_{3x3} \\ 0_{3x3} & T_{\Theta} \end{pmatrix} \tag{3.3}$$

$$\xi = \begin{pmatrix} \Gamma^E \\ \Theta^E \end{pmatrix} = \begin{pmatrix} x_e \\ y_e \\ z_e \\ \phi \\ \theta \\ \psi \end{pmatrix} \tag{3.2}$$

$$\nu = \begin{pmatrix} V^B \\ \omega^B \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \end{pmatrix} \tag{3.4}$$

$$\zeta = \begin{pmatrix} \dot{\Gamma}^E \\ \omega^B \end{pmatrix} = \begin{pmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \\ p \\ q \\ r \end{pmatrix} \tag{3.5}$$

$$\begin{pmatrix} F^B \\ \tau^B \end{pmatrix} = \begin{pmatrix} m * I_{3x3} & 0_{3x3} \\ 0_{3x3} & I \end{pmatrix} \begin{pmatrix} \dot{V}^B \\ \dot{\omega}^B \end{pmatrix} + \begin{pmatrix} \omega^B \times (m * V^B) \\ \omega^B \times (I * \omega^B) \end{pmatrix} \tag{3.6}$$

$$= M_B * \dot{\nu} + C_B(\nu) * \nu = G_B(\xi) + O_B(\nu) * \Omega + E_B * \Omega^2$$

$$M_H * \dot{\zeta} + C_H(\zeta) * \zeta = G_H + O_H(\zeta) * \Omega + E_H(\xi) * \Omega^2 \tag{3.7}$$

$$\dot{\nu} = M_B^{-1} * (-C_B(\nu) * \nu + G_B(\xi) + O_B(\nu) * \Omega + E_B * \Omega^2) \tag{3.8}$$

$$\dot{\zeta} = M_H^{-1} * (-C_H(\zeta) * \zeta + G_H + O_H(\zeta) * \Omega + E_H(\xi) * \Omega^2) \tag{3.9}$$

The following equations show the derived EOM of the GAV $\dot{\zeta}$ WRT the H-frame. Apparent from the rotational DOF of the quadrocopter, the linear accelerations $\dot{\Gamma}^E$ WRT the E-frame are influenced by a trigonometrical equation of the angles Θ^E and the sum of the uplift forces U_1. This behaviour also is visualized in the cross-sectional and front side view WRT the specific angle and B-frame axis in the figure 3.2. Further the equations show the angular accelerations $\dot{\omega}^B$ WRT the B-frame and the corresponding influences which derive from the propellers' force constellation U_1, U_2, U_3, U_4(See chapter 3.1) the angular velocities from ω^B and the body inertias WRT the mass axis [12] [13] [14].

$$\dot{\zeta} = \begin{pmatrix} \ddot{x}_e \\ \ddot{y}_e \\ \ddot{z}_e \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} (sin\psi sin\phi + cos\psi sin\theta cos\phi)\frac{U_1}{m} \\ (-cos\psi sin\phi + sin\psi sin\theta cos\phi)\frac{U_1}{m} \\ -g + (cos\theta cos\phi)\frac{U_1}{m} \\ \frac{I_{YY}-I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega + \frac{U_2}{I_{XX}} \\ \frac{I_{ZZ}-I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega + \frac{U_3}{I_{YY}} \\ \frac{I_{XX}-I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}} \end{pmatrix} \tag{3.10}$$

$$U_B = E_B\Omega^2 = \begin{pmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ lb(-\Omega_2^2 + \Omega_4^2) \\ lb(-\Omega_1^2 + \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{pmatrix} \tag{3.11}$$

[12] The formula 3.10 bases on [Tom08, p.21 The GAV WRT H-frame]

[13] J_{TP} [N m s2] is the total rotational moment of inertia around the propeller axis, See [Tom08, p.16 Propeller inertia]

[14] I_{XX}, I_{YY}, I_{ZZ} [kg m2] are moments of inertias around the specific axis, See [Tom08, p.136 Inertia matrix]

3.2 Vision Sensors

3.2.1 Motivation

An enormous quantity of researches, which is sponsored by industry companies and universities, was executed to find a better approach to stabilize UAV by using different sensors. Movement detection approaches, which are ultrasonic, sonar or Radio Frequency (RF) based show that it is necessary to have known reference points to get a reliable result [Jue09, p.2 Ultrasound indoor localization with reference points][Nir02, pp.4-5 Radio Model Localization]. The problem of this approach is that the environment has to be prepared before the UAV flight. This preparation is a drawback in point of flexibility in different operation places. Anyway, approaches with ultrasonic, RF or sonar sensors show that the localization of UAV needs a kind of global feedback to correct the UAV absolute position. One of the first motivations for a vision based sensor was presented by Ettlinger et al. [Sco04, pp.1-2 Visual-Based Localization and Control]. In this paper, the authors suggest, that vision is the only practical solution for obstacles of reference free flight stability and showed an On-board approach for a Gliding Unmanned Aerial Vehicle(s) (GUAV) by detecting the horizon with a forward looking camera and estimating and control the flight attitude[Bea08, p.27 Vision Sensors]. The intension of a reference free vision based stability approach for solving the IMU error drift is the motivation for the investigation of visual approaches which will be introduced and compared in this chapter.

3.2.2 Distribution and implementation approaches

This chapter focus the comparison of different approaches of vision sensors' implementation and distribution, which were researched in several scientific works. These approaches are separated in distribution approaches (Off-board image processing (OFIP), On-board image processing (ONIP), On-board camera (ONCA), Off-board camera (OFCA)) which are visualized in figure 3.3, and the implementation approaches Hardware based image processing (HWBIP) and Software based image processing (SWBIP). Problems, such as a limited power resource, a poor level of algorithm complexity for ONIP resulting from the limited calculating On-Board performance and the endeavour to economise weight, lead to outsourcing the image processing to a remote system via a wireless communication, which is not concerned to the On-Board problems. A drawback of this approach was determined by Langer et al. [Sve08, pp.5-7 Off-Board Image Processing] which use OFIP to track a landing pad for autonomous landing and show that the wireless transmission delay has a impact on the sampling rate of the algorithm. Tippetts [Bea08, p.27 Vision Sensors] mentions in his work the limitation of the wireless communication in OFIP as a drawback for range of the aircraft [15]. The OFCA approach was researched by Altug et al. [Erd02, p.76 Localization and Control with an Off-Board camera], with the result of a less sensitive feature detection and position localization as the ONCA approach. OFCA tracking is also shown in the developments of extremely reliable and precise localization of a UAV and is used in the development of aggressive autonomous flights of multiple Micro Aerial Vehicle(s) (MAV) in the experiments of Mellinger et al. [Dan10, Trajectory Generation and Control with Off-Board cameras] [Dan07, pp.363-364]. The advantage of OFCA tracking system is that the image capturing and position tracking is ex-

[15]The comparison of OFIP and ONIP approaches is visualized in 3.4

ecuted outside the UAV and prevents complex On-Board calculations for movement and position estimation [16]. The disadvantage of this OFCA method is that they need a special flying environment, which is build up with an OFCA motion tracking system[17] [Nat10, pp.1-2 The GRASP Testbed].

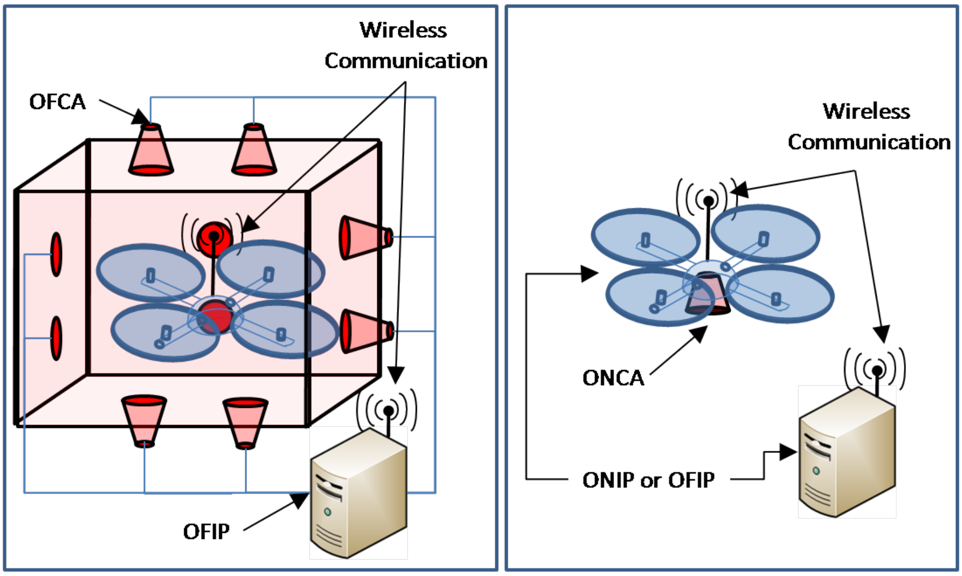

Figure 3.3: On- and Off-Board camera and motion tracking system approach

Tippetts [Bea08, pp.21-22 On-Board image processing with an FPGA] realized an ONIP approach with a Field-Programmable Gate Array (FPGA) which uses a complex feature tracking algorithm but runs with high sampling rate [18]. The characteristics of the feature tracking with a FPGA can result a fast movement tracking method, but it is not an efficient method in the behaviour of power consumption because the hardware is not optimized for the image processing tasks. In contrast to the drawbacks of FPGA, Langer et al. [Sve09, On-Board image processing with mice sensors] and Beyeler et al. [Ant09, pp.4-5] showed an approach for detecting the spatial movement of a UAV with optical mice sensors which have an optimized hardware for image processing and are lightweight.

[16]This approach focus the E-frame localisation and is robust against position drifts and errors
[17]The comparison of OFCA and ONCA approaches is visualized in 3.4
[18]In this case a high sampling rate means that the image processing runs nearly equal to the sampling rate of the IMU

These sensors calculate the optical flow of the captured images and estimate the movement direction of the UAV in hardware and can provide a high sample rate. In contrast to the fast movement detection, a disadvantage is that these sensors have limitations related to the operating environments. These limitations are the concrete light and distance range which is required from the manufacturer [ST 05, pp.7-15 Limitations of the operating environment][Sve09, p.6 Performance of the optical flow based position controller]

SWBIP approaches have a more flexible extension and change behaviour for prototyping of vision based solutions, but they don't work as fast as HWBIP approaches. Stowers et al. [Joh09]realized a heading estimation for a quadrocopter with an onboard Single Board Computer(s) (SBC) which runs the open source computer vision toolkit OpenCV[1]. This software based image processing approach shows strengths in the modularity of the image processing architecture and in the interchangeability of the vision system[19] [Joh09, pp.1-6 Software Based Vision Processing][Gar08, Software Structure and Portability]. Figure 3.4 includes the summary of the examined approaches of this chapter. We can see that no approach has outstanding set of beneficial characteristics. Each approach brings advantages and disadvantages, so that the decision of the appropriate approach is in the focus of the investigation.

3.2.3 Vision based movement detection algorithms

Sequential captured images contain a huge amount of information about the absolute and relative movement of objects in every direction. So several vision based movement detection approaches were researched in the topic of UAV sta-

[19] The comparison of HWBIP and SWBIP approaches is visualized in 3.4

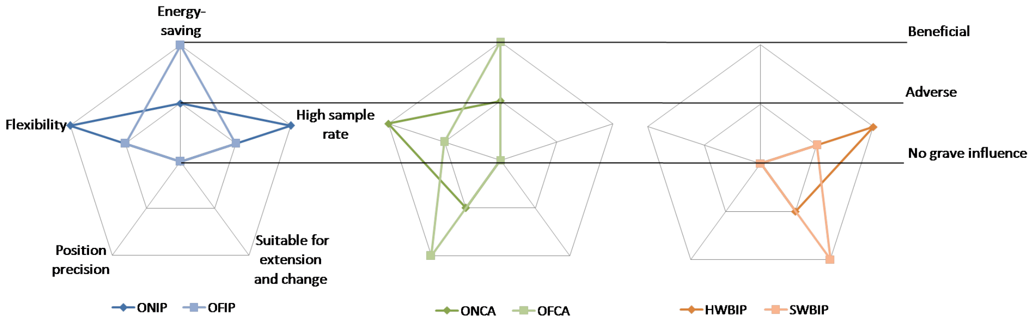

Figure 3.4: A comparison of the distribution behaviour of vision sensors

bility with different requirements to the information which extracted from the vision process[20]. A simple approach for a relative vision based control of a UAV was implemented by Boabdallah [Sam07, pp.110-114 Position Sensor] by using a down looking camera and the Canny edge detector [Joh86] and the Douglas Peuker Algorithm for curve equalisation [DP73]. The drawback of this approach is that the field of vision must contain forms with edges which mean that the approach cannot result a satisfactory result if no edges are detected. A further approach for detecting relative movements and to build up a map for autonomous navigation, is visual Simultaneous Localization And Mapping (SLAM). This approach tracks features in the field of vision and reconstructs a relation to tracked features of previous images. The realization of SLAM [Bri07] in a UAV was executed in the work of Bloesch et al. [Mic10] under consideration of real-time characteristics. The behaviour of the algorithm shows that the localization of the tracked features and the simultaneous mapping has a big impact at the time delay of the calculation. So the experiments and the control algorithms were researched and designed for 7.5Hz sample rate. Another popular tracking method for movement

[20] These investigated approaches are visualized in figure 3.5. This figure includes pictures presented in [3, SLAM][2, Edge detector][J.L94, p.27 Horn and Schunck optical flow]

detection is given with the Optical Flow, which can be described as movement observation of tracked objects or pixels in a sequence of images. Algorithms to calculate the Optical Flow were introduced by Horn and Schunk [Ber80], Lukas and Kanade [Bru81]. A few years after introducing the Lucas-Kanade-Algorithm, Lucas described the theoretical approach of visual navigation by using the Optical Flow. Thereby he described the possibility to detect movements and to calculate correspondence velocities. These velocities can be combined to a vector field which can describe movements in every direction [Bru84, pp.40-45 Optical Navigation Theory].

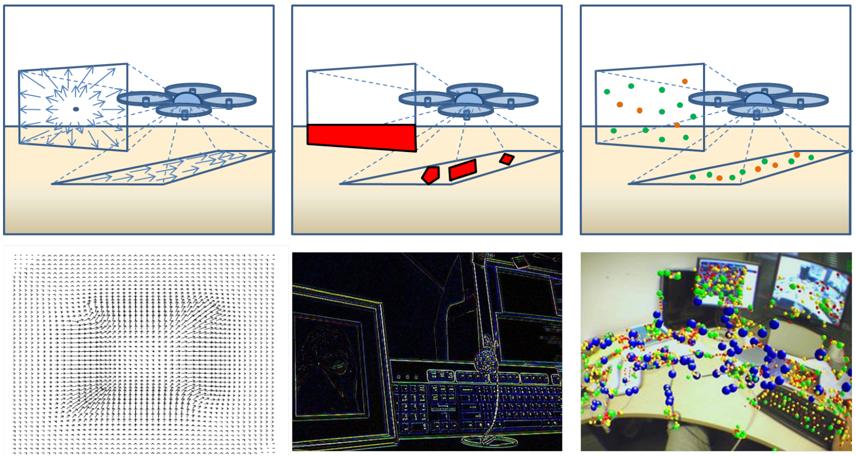

Figure 3.5: Samples for UAV stabilisation with Optical Flow, Edge detection and SLAM

3.2.4 Optical Flow

Researches which are related to image processing, mostly use camera models to reduce the complexity of the reality. This approach is also practicable in researches with optical flow. The figure 3.6 visualizes a camera model with a spherical, which illustrates a complex real camera and a flat image plane, which simplifies the complex model. Such simplification is achievable, if a mathematical transformation of the image plane can result a distortion-free image plane. Fach

[Mar08, pp.29-31 Chapter 3.3, Distortions] classifies reasons for distortions in a Camera Calibration Domain (CACD) and the Shift of the Optical Axis Domain (SOAD). Pincushion- and barrel- distortions are summarized as radial distortions and caused by focal distance in relation to the captured image dimension. Further decentering distortions are based on the wrong alignment of the optical axis to the projected plane. Further Fach [Mar08, pp.29-31 Matehmatical description of distortions] shows that these distortions of a physical camera can be eliminated with the equations shown in appendix 7.2. Thereby the problem of distortions is reduced to factorization of each distorted coordinate for the complete image in the initial camera calibration phase to eliminate distortions of the physical behaviour of the lenses and the image plane of the camera, and further in situations in which the camera is moved not planar to the projected plane. In the second case a fixed IMU of the camera system can provide the difference to the orthogonal gravity vector g, and can allow the transformation of the captured image in real time [21].

Following the optical flow projection suggested by Wei [Zha09, pp.71-96 Obstacle Detection Using Optical Flow], there is described as a 2D projection of a 3D motion in the world on the image plane. Thereby Wei uses the ideal perspective projection model to introduce the projection of a point $P = (X, Y, Z)$, from the camera frame into the image frame $p = (x, y, f)$ where f is the focal length. The optical flow definition is derived with the assumption that the point P moves in the time t' to the Point P' with the corresponding projection on the image plane p and p'. The velocity on the image plane can be calculated as $v = p - p' = (x, y, 0)^T - (x', y', 0)^T$, where the component of z is cancelled out causing the 2D projection. Wei uses this projection to describe the problem

[21] Such a camera was introduced in the year 2009 as standalone system from Fraunhofer Institute for Manufacturing Engineering and Automation ,See [Ber09, pp.1-2]

of retrieving the optical flow velocity with 6 DOF. Further Wei described that researches for optical flow velocity determination only can based on a reduced DOF model[22].

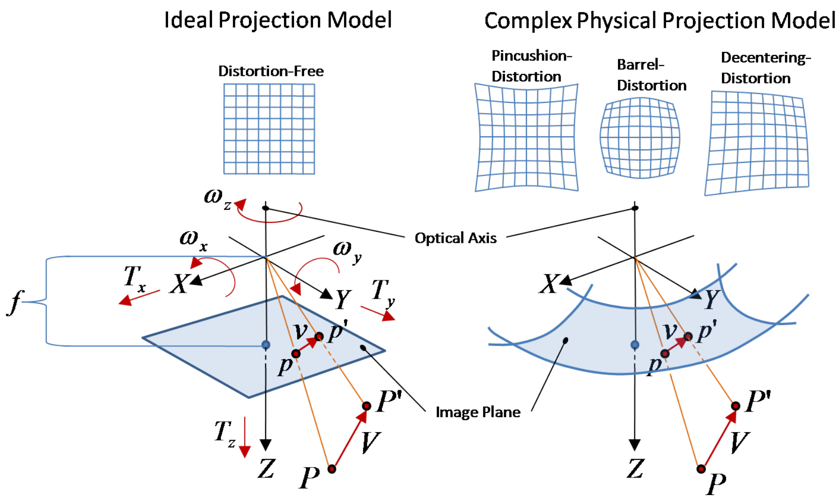

Figure 3.6: Complex Physical and reduced projection camera model

3.3 Control Systems Characteristics

3.3.1 Control approaches

The closed loop feedback architecture is an approved method for controlling systems and nearly used in the most of the systems controls in industry and society. So the most of the researches in the UAV stabilisation topic were and still are executed with closed loop control architectures which are built up as Multiple-Input and Multiple-Output (MIMO) or as multiple Single-Input and Single-Output (SISO) systems. The differences between the researched approaches are the

[22]The DOF of a ideal projection model are shown in figure 3.6 The can be summarized in a translation vector $T = (T_x, T_y, T_z)^T$ and a rotational vector $\omega = (\omega_x, \omega_y, \omega_z)^T$, See [Zha09, pp.74-77 Motion Model Deduction]

amount of the inputs and outputs of the physical process, the type of controller, the used measurements and the sampling rate. Thereby the type and behaviour of the controller is close related to the measurements and the physical process [Ric01, p.3]. The classical control approach using Proportional-Integral-Derivative (PID) controller was researched in several Hovering Unmanned Aerial Vehicle(s) (HUAV) projects [Ved08, pp.24-31] [Sam07, pp.43-68]. These researches show that the classical PID controller is not robust enough to handle with complex models which include several integration states, but has to be transformed to cascaded PID controllers. Basically the implementation of a PID controller can be executed with the determination of three parameters for the given process. These parameters are the proportional, integral and derivative gain of the controller, which have an impact on the dynamic behaviour of the controlled value [Ric01, pp. 695-698 PID Controllers]. Another approach for control improvement was introduced by Luenberger [Dav71] and describes a closed loop control approach, called state-observer, which simulates the process in real time parallel to the true process by using the input and output vector of the closed loop system and corrects the control strategy. Bloesch et al. [Mic10, p.5] have shown in their research, that the problematic of sensors with non-negligible time delay can be solved to an adequate result by using a state-observer. The reason for that is that the state-observer allows to configure and to control the stability behaviour of the closed loop feedback architecture. If a process can be observed or furthermore controlled, is related to the states in which it can be and furthermore to the measurements of the DOF [Ric01, pp.632-636 Controllability and Observability]. In the other hand, the state-observer has a big impact to the calculation capacity of the UAV target, because the real-time process simulation of a 6 DOF rigid-body becomes very complex.

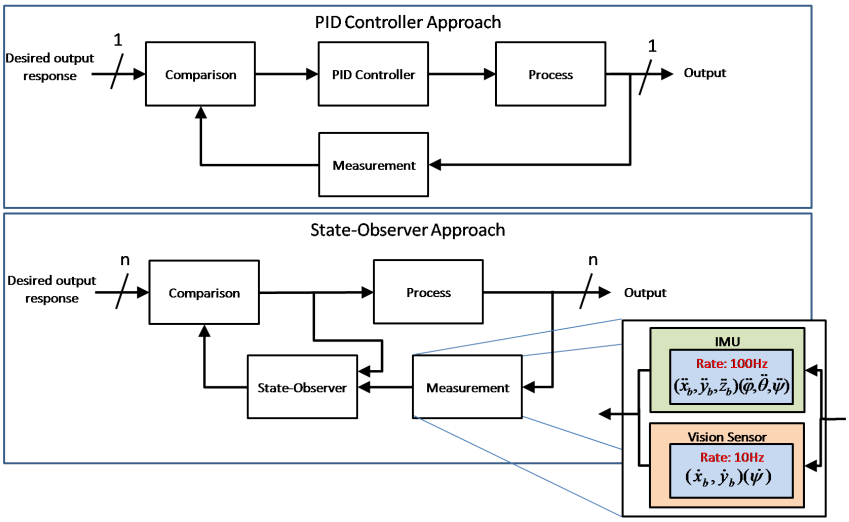

Figure 3.7: PID-Controler and State-Observer in closed loop System with variable sample sensor rates

3.3.2 Performance measures in the time domain

Control systems usual are designed and evaluated in different domains, under consideration of the special behaviour of these. The closed loop feedback architecture can be described with a transfer function which shows the behaviour of the closed loop feedback system in respect to the system boundaries. The time domain is used to visualize the time variant behaviour of the controlled value, and gives definitions of performance measures. These measures are visualized in figure 3.8[23] and describe the following characteristics [24].

- Stability: A control system is called stable, if the controlled value y(t) reaches a constant value and does not oscillate between a set of values.

- Precision: The precision describes the ability of steady state error minimiza-

[23]This picture extends the diagram presented in [Rei07, Chapter 2, p.11 Characteristics of control systems]
[24]The content of the characteristics is derived from [Rei07, Chapter 2, p.12 Requirements of control systems][Ric01, Chapter 5, pp.228-230 The Performance of Feedback Control Systems]

tion of a control system. The best case of precision is $e(\infty) = y_{set}(\infty) - y(\infty) = 0$.

- Transient oscillation: The oscillation process of a system is described with the needed time of the function is needed to reach the range of tolerance T_s and is called settling time. This time depends on the rise time which describes the first entrance point of the range of tolerance. A short rise time can bring the disadvantage of a big maximal overshoot y_{max}, which should be kept minimal.

- Robustness: A control system is characterized as robust, if it does not become instable if system values change over time in scope of their tolerances. Such tolerances can occur from temperature variations or tribological reasons.

- Load of actuator: The load of the actuator(s) should be minimal. That means that the maximum actuator value should be equal to the needed value to prevent unnecessary work.

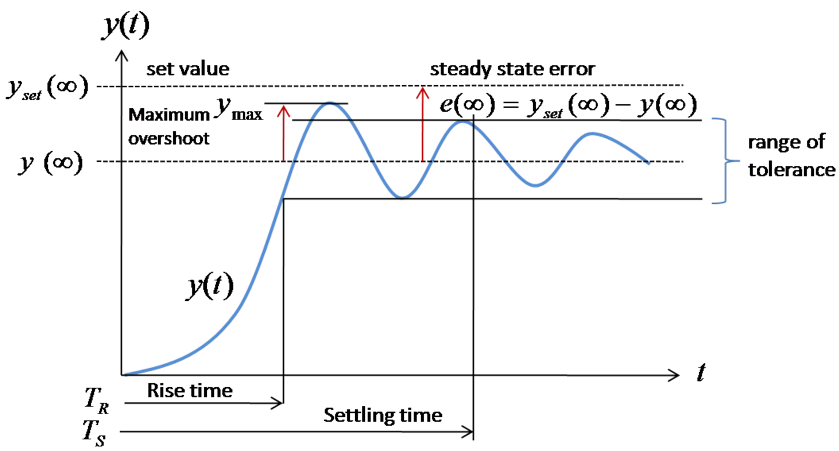

Figure 3.8: Performance measurement values of control systems

3.3.3 Continuous and discrete time and frequency domains

The necessity of the linear approximation and the advantage of simplification of systems, led the analysts to the use of the Laplace transformation. This method transfers relative easily solved algebraic equations in the Laplace domain for the more differential equations in the time domain[Ric01, pp.41-47 The Laplace Transform]. Combined with the fact that the sampling rate of a computing system can have a big impact to the stability of it, the z-transformation allows transferring functions from the continuous frequency domain to the discrete frequency domain [Ric01, pp.749-754 The z-Transform]. These two transformations and the corresponding retransformations are visualized in the figure 3.9[25] and can be executed in a simple form by using transformation tables [26]. The frequency domain equations are describes as functions of the complex number s. This continuous frequency parameter is discredited by using the following abbreviation $z = e^{sT}$ where T is the sampling rate of the discrete system. The factor k in the discrete domain symbolizes the k^{th} sample value of the function and can be described as parameterization factor of the sample time T [Rei07, Chapter 4, p.19 z-Domain set value transfer function of a closed loop control system].

3.3.4 Stability criteria of transfer functions

The mentioned simplification of Closed Loop Control System(s) (CLCS) in the following chapter can be shown by reducing the complete system to a quotient which in contains polynomials for the numerator and denominator. This charac-

[25]This figure bases on the theorems in [Rei07, Chapter 4, p.17 Definitions and Rules of the z-Transform]

[26]See, Transformation tables in [Ric01, pp.42 Important Laplace Transform Pairs] and [Ric01, pp.751 z-Transforms]

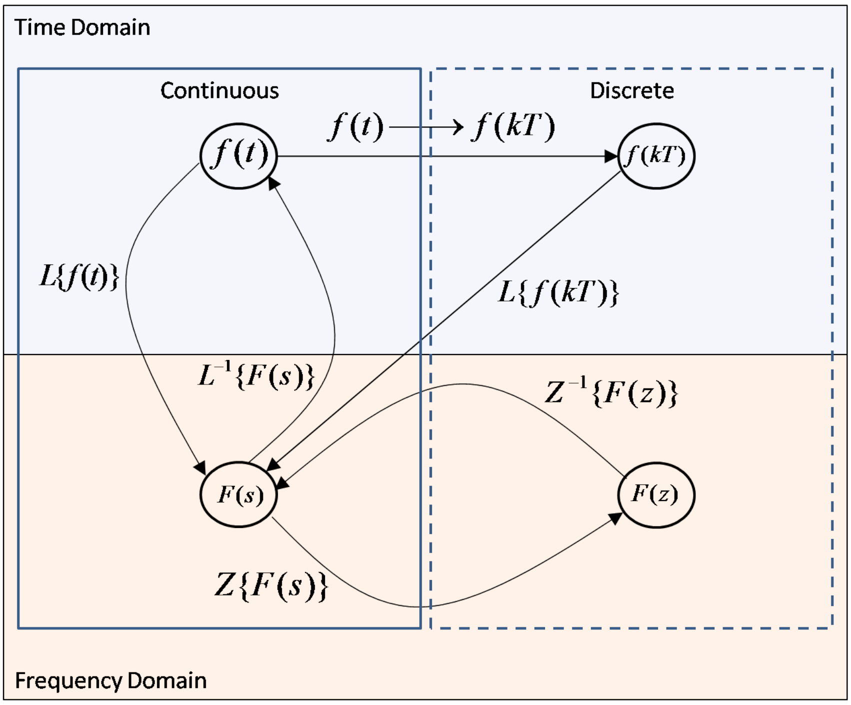

Figure 3.9: Transformation possibilities of continuous and discrete time and frequency domains

teristically quotient is called Transfer Function (TF) and describes the behaviour of a system in relation to its boundaries. In the continuous domain the TF can be created as Set-Value-Transfer-Function (SVTF) (See formula 3.12) or as Error-Value-Transfer-Function (EVTF) (See 3.13) which allows investigations from two different input perspectives of the CLCS. The discrete closed loop control system can be used to describe the dynamic control behaviour of a discrete controller which controls a continuous process from the SVTF perspective (See formula 3.14). The interface between the process and the controller in this case is realized with sample and hold components which discretize the continuous process [Ric01, p.747 Sampled-Data Systems]. The figure 3.10[27] visualizes the generic architecture of CLCS in the continuous and discrete domain. Thereby the figure

[27]This picture extends diagrams presented in [Rei07, Chapter 4, p.2 Generic linear Closed Loop Control System],[Rei07, Chapter 4, p.19 z-Domain set value transfer function of a closed loop control system], [Rei07, Chapter 4, p.21 Stability]

shows that the components of a global TF of a CLCS also can be denoted as TF which describe the specific behaviour of the controller, the process [28] and measurement[29].

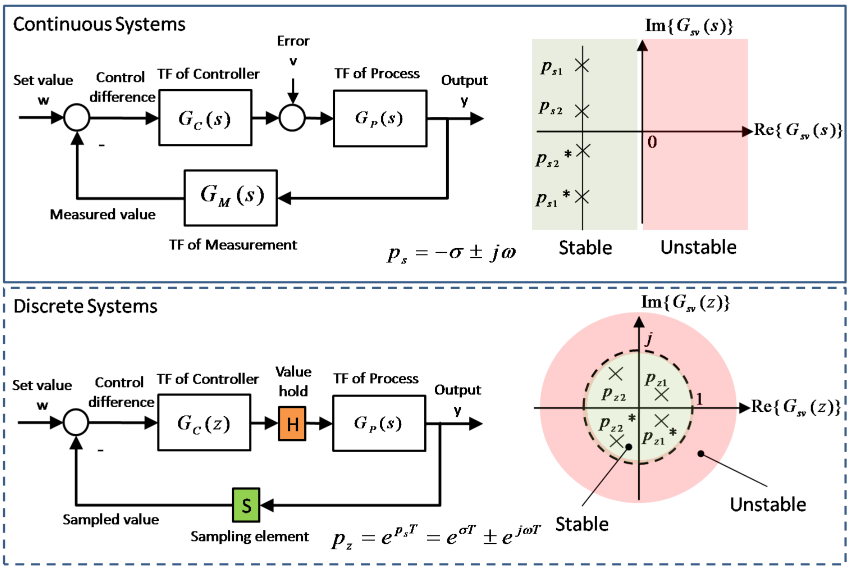

Figure 3.10: TF of continuous and discrete CLCS and their stability behaviour

The corresponding TF of the introduced domains are shown in the formulas 3.12, 3.13, 3.14[30]. Especially the SVTF contains important information of the stability behaviour. So continuous SVTF are stable, if the poles p_s of the quotient are located at the real negative section of the complex plane. Equivalent to that the stability of a discrete SVTF is given, if the poles p_z are located in the unit circle of the complex plane [31][32]. This stability behaviour in the complex plane is visualized in the figure 3.10. Consequently the sample rate of digital systems and the corresponding sensors has a big impact to the stability of a CLCS [33].

[28]In specific literature of control systems, sometimes the synonym plant is used for process
[29]The measurement component in the discrete domain is substituted with a sample component which holds the value of the process in the desired appearance
[30]See, [Ric01, p.754 Closed-Loop Feedback Sampled-Data Systems]
[31]This causes the quantisation relation of the frequency $z = e^{sT}$
[32]CLCS which have poles located on the edges of the stability sections are called semi-stable. Such CLCS are totally not robust and also assigned to the set of unstable systems
[33]See, [Ric01, p.756 Stability Analysis in the z-Plane]

$$G_W(s) = \frac{Y(s)}{W(s)} = \frac{G_C(s) * G_P(s)}{1 + G_C(s) * G_P(s) * G_M(s)} \tag{3.12}$$

$$G_V(s) = \frac{Y(s)}{V(s)} = \frac{G_S(s)}{1 + G_C(s) * G_P(s) * G_M(s)} \tag{3.13}$$

$$G_W(z) = \frac{Y(z)}{W(z)} = \frac{G_C(z) * G_P(z)}{1 + G_C(z) * G_P(z)} \tag{3.14}$$

4 Design and Implementation

4.1 Existing Quadrocopter Architecture

4.1.1 Central Control Unit

4.1.2 Software Components

4.2 Adopted approach

4.2.1 Problems and Limitations

4.3 Simulation of Embedded System Quadrocopter

4.3.1 Flight dynamics

4.3.2 Control

4.3.3 Sensors

4.4 Simulation of Base Station

4.4.1 Video Sources

4.4.2 Synthetic Video vs. Real Video

5 Experimental Results and Analysis

6 Conclusions and Further Work

7 Appendix

7.1 Moment of inertia

The physical modelling and simulation of a rigid body, needs to be described with the dynamic behaviour of that. So the quadrocopter UAV can be described with the EOM described in chapter 3.1.2 in relation to the moment inertias which describe the resistance of the UAV rigid body against changes in rotational directions. Such moment inertia is described as a $3x3$ matrix, also called tensor, for a rigid body with 6 DOF. The assumption that the B-frame originates in the point of mass of the quadrocopter [1], the matrix is simplified to a matrix with diagonal entries $(Ixx, Iyy, Izz)^2$.

$$I = \begin{pmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{pmatrix} = \begin{pmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{pmatrix} \tag{7.1}$$

7.2 Mathematical description of distortions

Distortions can occur if the distance of image points to the centre of projection varies or if the centre of projection is not in the centre of the image. These kinds

[1]See chapter 3.2

[2]The derivation of the moment inertia and the corresponding pictures are derived from [Joa10, pp.24 -33 Determination of the moment of inertia]

of distortions are defined as radial $x_d r, y_d r$ and decentring distortions $x_d d, y_d$d, and represent the components of the complete distortion of images x_d, y_d.

The following formulas[3] describe the possible distortion of images and show that the complexity of an image equalisation is proportional to the image points e.g. the image dimensions. Thereby the values x_n, y_n describe a point of an image which has to been equalized. The distance radius to the centre of the image 7.3 is a important component and is used in combination of the distortion coefficients $k_{cn} : n \in \{1, 2, 3, 4, 5\}$ to eliminate the mentioned components of distortion.

$$\begin{aligned} x_d &= x_{dr} + x_{dd} \\ y_d &= y_{dr} + y_{dd} \end{aligned} \tag{7.2}$$

$$r = \sqrt{(x_n^2 + y_n^2)} \tag{7.3}$$

$$\begin{aligned} x_{dr} &= (1 + k_{c1} * r^2 + k_{c2} * r^4 + k_{c5} * r^6) * x_n \\ y_{dr} &= (1 + k_{c1} * r^2 + k_{c2} * r^4 + k_{c5} * r^6) * y_n \end{aligned} \tag{7.4}$$

$$\begin{aligned} x_{dd} &= 2 * kc3x_n * y_n + k_{c4} * (r^2 + 2 * x_n^2) \\ y_{dd} &= k_{c3} * (r^2 + 2 * y_n^2) + 2 * k_{c4} * x_n * y_n \end{aligned} \tag{7.5}$$

These calculations can be executed initial for the camera calibration, and further for the real-time equalization for the trajectory movement of the camera. Popular calibration processes transform the distortions to a value in which they can

[3]This formulas are derived from the formulas presented in [Mar08, p.30 Mathematical description of distortions]

be considered as equalized. Such a calibration process is described by Tsai et al. [Tsa05, Camera Calibration by Tsai] and extended by Zhang [Zha99, Visual Servoing with Dynamics], and describes a iterative way for the determination of the coefficients k_{cn}. Thereby a squared image is used as reference for the camera and the algorithm, described in figure 7.1[4], is executed iteratively. Real-time equalization processes mostly are implemented with an array of distortion coefficients and can effective determine these with a lookup operation[5].

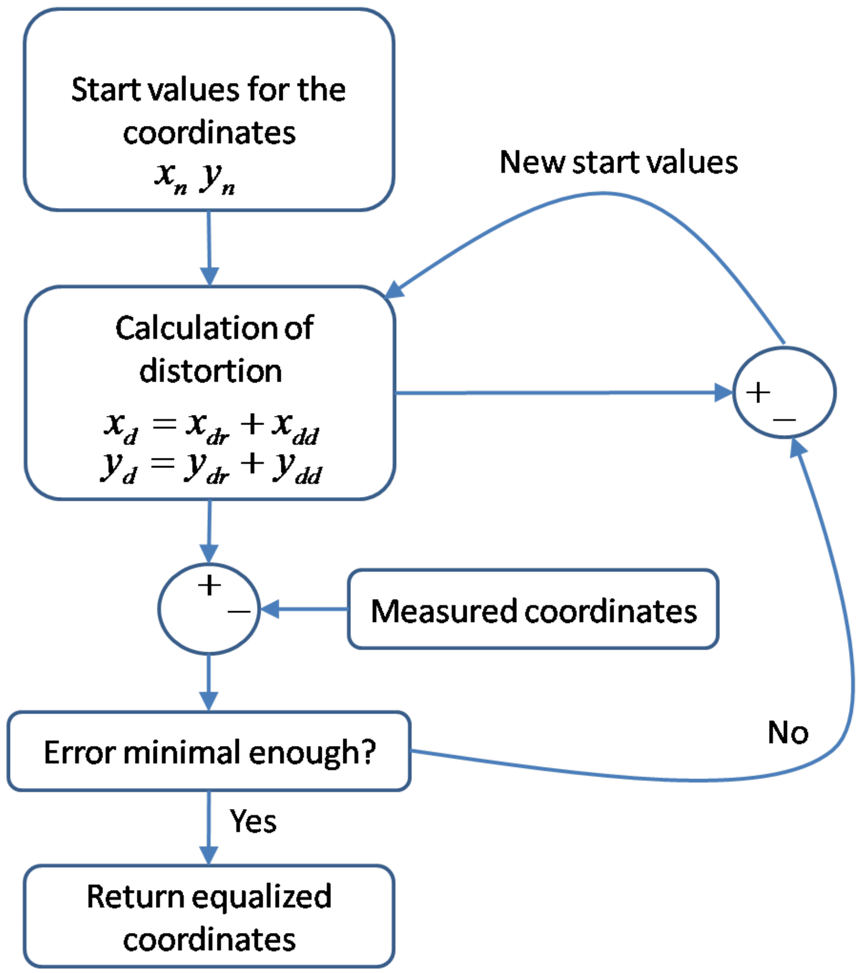

Figure 7.1: Equalization Algorithm of distorted images

[4]This picture extends the scheme picture presented in [Mar08, p.32 Equalization of a captured image]
[5]See, Distortion Correction with FPGA [Gri03, pp. System Design]

8 References

[1] OpenCV 2.0 C Reference. http://opencv.willowgarage.com/, 2010.

[2] George Koharchik. Image Processing with OpenGL and Shaders. http://www.linuxjournal.com/content/image-processing-opengl-and-shaders, February 2011.

[3] Jean-Bernard Hayet. Exploration strategies for simultaneous localization and mapping. http://www.cimat.mx/ jbhayet/temas.php, February 2011.

9 Bibliography

[Ant09] Antoine Beyeler, Jean-Christophe Zufferey, Dario Floreano. optiPilot: control of take-off and landing using optic flow. Technical report, Laboratory of Intelligent Systems, École Polytechnique Fédérale de Lausanne (EPFL), 2009.

[Bea08] Beau J. Tippetts. Real-Time Implementation of Vision Algorithms For Control, Stabilization, and Target Tracking, for a Hovering Micro-UAV. Master's thesis, Department of Electrical and Computer Engineering, Brigham Young University, August 2008.

[Ber80] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1980.

[Ber09] Bernhard Kleiner. Kameragestuetztes inertiales Navigations-system mit Sensorfusion. Technical report, Fraunhofer Institute for Manufacturing Engineering and Automation IPA, 2009.

[Bri07] Brian Williams, Georg Klein and Ian Reid. Real-Time SLAM Relocalisation. Technical report, Department of Engineering Science, University of Oxford, 2007.

[Bru81] Bruce D. Lucas, Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages pp.121–130, 1981.

[Bru84] Bruce D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Robotics Institute, Carnegie Mellon University, July 1984.

[Dan07] Daniel Gurdan, Jan Stumpf, Michael Achtelik, Klaus-Michael Doth, Gerd Hirzinger, Daniela Rus. Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. *IEEE International Conference of Robotics and Automation, Roma, Italy* , April 2007.

[Dan10] Daniel Mellinger, Nathan Michael and Vijay Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *International Symposium on Experimental Robotics, Delhi, India*, December 2010.

[Dav71] David G. Luenberger. An Introduction to Observers. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-16, NO. 6*, December 1971.

[Dav07] David Morin. *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2007. ISBN 978-0-521-87622-3 First Published Edition 2008.

[DP73] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, pages 112–122, October 1973.

[Erd02] Erdinc Altug, James P. Ostrowski, Robert Mahony. Control of a Quadrotor Helicopter Using Visual Feedback. *IEEE International Conference of Robotics and Automation, Washington, DC* , 2002.

[Gar08] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly, First Edition edition, 2008.

[Gri03] Gribbon K.T., Johnston C.T. and Bailey D.G. A Real-time FPGA Implementation of a Barrel Distortion Correction Algorithm with Bilinear Interpolation. Technical report, Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand, 2003.

[J.L94] J.L. Barron, D.J. Fleet and S.S. Beauchemin. Performance of Optical Flow Techniques. Technical report, Dept. of Computer Science, University of Western Ontario, Queen's University, 1994.

[Joa10] Joachim Schwab, Sreekanth Sundaresh, Nina Vetlugina, Marc Weber, Joerg Fiedrich and Dionysios Satikidis. Project Quadrocopter, Automotive Systems Master Software Based Automotive Systems ASM2-SB / SS 2010 Documentation. Technical report, University of Applied Sciences Esslingen, Department of Information Technology, June 2010.

[Joh86] John Canny. A Computational Approach to Edge Detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6*, November 1986.

[Joh09] John Stowers, Andrew Bainbridge-Smith, Michael Hayes and Steven Mills. Optical Flow for Heading Estimation of a Quadrotor Helicopter. Technical report, University of Canterbury, New Zealand, 2009.

[Jue09] Juergen Eckert, Falko Dressler and Reinhard German. An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes. Technical report, University of Erlangen Dept. of Computer Science, February 2009.

[Mar08] Martin Fach. *Navigation und Regelung eines Luftschiffes mittels optischer, inertialer und GPS Sensoren*. PhD thesis, Institut für Flugmechanik und Flugregelung, University of Stuttgart, 2008.

[Mic10] Michael Bloesch, Stephan Weiss, Davide Scaramuzza and Roland Siegwart. Vision Based MAV Navigation in Unknown and Unstructured Environments. Technical report, Autonomous Systems Lab, ETH Zuerich, 2010.

[Nat10] Nathan Michael, Daniel Mellinger, Quentin Lindsey and Vijay Kumar. The GRASP Multiple Micro UAV Testbed. Technical report, GRASP Laboratory, University of Pennsylvania, 2010.

[Nir02] Nirupama Bulusu, John Heidemann and Deborah Estrin. GPS-less Low Cost Outdoor Localization For Very Small Devices. Technical report, University of Southern California / Information Sciences Institute, 2002.

[Rei07] Reinhard Keller, Hermann Kull, Walter Lindermeier, Werner Zimmermann. System Dynamics, Modelling, Simulation, Control engineering, July 2007. VZ1.2.

[Ric01] Richard C. Dorf, Robert H. Bishop. *Modern Control Systems*. Prendice-Hall, Inc., ninth edition edition, 2001.

[Roy08] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer Science+Business Media, LLC, 2008. ISBN 978-0-387-74314-1.

[Sam07] Samir Bouabdallah. *Design and Control of Quadrotors with Application to*

Autonomous Flying. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2007.

[Sco04] Scott M. Ettinger, Michael C. Nechyba, Peter G. Ifju and Martin Waszak. Vision-Guided Flight Stability and Control for Micro Air Vehicles. Technical report, Intel Corporation, University of Florida, NASA Langley Research Center, March 2004.

[ST 05] ST Microelectronics. MEMS Inertial Sensor LIS3LV02DQ Rev 1 Datasheet, October 2005.

[Sve08] Sven Lange, Niko Sünderhauf, and Peter Protzel. Autonomous Landing for a Multirotor UAV Using Vision. *Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice, Italy*, November 2008.

[Sve09] Sven Lange, Niko Sünderhauf, Peter Protzel. A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments. Technical report, Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, 2009.

[Tom08] Tommaso Bresciani. Modelling, Identification and Control of a Quadrotor Helicopter. Master's thesis, Department of Automatic Control, Lund University, October 2008.

[Tsa05] Tsai R. Y. and Lenz R. K. . A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. *Robotics and Automation. IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, June 1989*, December 2005.

[Ved08] Vedran Sikiric. Control of Quadrocopter. Master's thesis, Royal Institute of Technology Sweden, School of Computer Science and Communication, 2008.

[Zha99] Zhang H. and Ostrowsi J. P. Visual Servoing with Dynamics: Control of an Unmanned Blimb. Technical report, General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory; University of Pennsylvania, 1999.

[Zha09] Zhaoyi Wei. *REAL-TIME OPTICAL FLOW SENSOR DESIGN AND ITS APPLICATION ON OBSTACLE DETECTION*. PhD thesis, Department of Electrical and Computer Engineering, Brigham Young University, 2009.

School of Engineering & Design – Electronic & Computer Engineering

MSc Distributed Computing Systems Engineering

Brunel University West London

Simulation and Performance Analysis of a Distributed Position Correction Scheme for Unmanned Aerial Vehicles

Interim Report

Author: Dionysios Satikidis

Supervisor: Dr Marios Hadjinicolaou

Date: 10/2010

Contents

List of Abbreviations

API	Application Programming Interface(s)
CCU	Central Control Unit
DOF	Degrees of Freedom
FPGA	Field-Programmable Gate Array
GUAV	Gliding Unmanned Aerial Vehicle(s)
HIL	Hardware In the Loop
HSE	University of Applied Sciences Essslingen
HUAV	Hovering Unmanned Aerial Vehicle(s)
IMU	Inertial Measurement Unit
MAV	Micro Aerial Vehicle(s)
MBD	Model Based Design
MEMS	Micro-Electro-Mechanical System
MIMO	Multiple-Input and Multiple-Output
PID	Proportional-Integral-Derivative
RF	Radio Frequency
SBC	Single Board Computer(s)
SLAM	Simultaneous Localization And Mapping
UAV	Unmanned Aerial Vehicle(s)

List of Figures

1 Introduction

During the last years the interest of robotic science and the development of robots increase enormous. Reasons for that are, the unstoppable technological progress of hard and software techniques, but also the necessity to replace humans with machines in dangerous, monotonous or unreachable industrial environments (medical, space, aviation and so far). One area of these interests is the aerial platform and the realization of Unmanned Aerial Vehicle(s) (UAV), which are mostly controlled via remote control or fly autonomous. These aircraft vehicles have several capabilities like in military or rescue operations with special environments like a burning house. For such indoor operation it is important to focus, that some kind of feedback sensors like GPS-sensors could not work satisfactory. In such cases UAVs faces problems with the self stabilization, because their physical behaviour is in generally unstable [Mic10, p.1 Introduction]. Most of the approaches to stabilize UAVwork with a clever combination of sensor equipment and control algorithms. Mostly this controller uses a Inertial Measurement Unit (IMU)which is mounted on board and includes accelerations sensors to detect movements in the given Degrees of Freedom (DOF). Actual acceleration sensors, which are used for this purpose, are Micro-Electro-Mechanical System (MEMS)or fiberoptic sensors which have a finite precision and unacceptable error propagation in case of integration for velocity or position detection [Mar08, p.p.11-13 Function Principles of MEMS, Sources of Error].

This problem also is a field of research at the Quadrocopter project of the University of Applied Sciences Essslingen (HSE)[HSE10, Website].

Figure 1.1: HSE Quadrocopter

1.1 Project State

The quadrocopter project of the HSEwas launched with the goal to build a system from the scratch, which is developed by students, scientific workers and Professors. In a first project the Printed Circuit Board which hosts the parts necessary to control the quadrocopter was designed by students at the faculty of Mechatronics and Electrical Engineering in Goeppingen. This project groups' focus was then on the simulation, implementation of the basic functions and visualization of the actual condition of the aircraft. The first two project groups already show the benefit of this project, a multidisciplinary development including hardware design, application development, embedded programming, simulation and the interfaces between these special fields. In the year 2009, the Faculty of Information technology adopted the development of the project with the aim to solve problems which came up in the previous development and to redesign the soft- and hardware architecture.

So a new hardware design was developed in a corporation between the two faculties with the outcome of a Central Control Unit (CCU)which can detect inertial movements in six DOFand can control the four actuators of the quadrocopter via so called brushless controllers. One of the biggest unresolved problems since yet is the development of a robust controller and the elimination of drifts which especially come up at the hovering state. The practice and the experience of previous developments show, that it is indispensable to proof new developments with a simulation before they will be realized at the real UAV. So the main topic and the focus of this Master´s Thesis will be on the development of a simulation which shows potential solutions of the mentioned problems and the research and evaluation of the outcome results.

2 Background to the project

2.1 Problem Description

Improvements of high density power storage, integrated miniature actuators and sensors, facilitate the development of Micro Aerial Vehicle(s) (MAV)and new areas of research for unmanned and autonomous flying systems [Sam04, p.1 Introduction]. This new area of interest brought also a new area of problems. One of these is the fact that the pilot of the aerial vehicle does not exist, because the UAVflights autonomous or the pilot observes and controls the UAVvia remote control. In both cases it is necessary that the UAVsystem can detect its absolute position, to provide the pilot a better quality of control or furthermore to manoeuvre autonomous. The following articles also describe this problem with different views and approaches [Sla09, p.2 Localization and path planning] [Kar10, p.1 High precision aircraft positioning] [Seb08, p.2 Teleoperated Robot Control].

This necessity of location determination requires measurement unit, which provides the detection of the movements in the given DOF. Because of low-cost and low-complexity reasons, the most popular components which are used to reach a nearly satisfactorily level of flight stabilization are inertial sensors. These sensors combined together called IMUand have the ability to detect the acceleration of translational or rotational movements.

Furthermore a control system can be used in a closed loop together with the IMUand can correct theoretically nearly every disturbance in a continual system [Sam07, p.p.45-64 System Control] [Ped05, p.p.49-51 Control Strategy]. The limitation of resolution of the sensors and the necessity of integrating the acceleration values for position and velocity determination, have in real systems a big impact in aspects of error propagation and detection of smoothly movements [Bea08, p.5 Low-acceleration Drift]. For a better appreciation of the movements, the following figure visualizes the influence of the thrust in relation to the movements in the DOFs. Thereby the values of ω [rad/sec]represents the least needed speed of rotation, for creating the required thrust for the hovering state. This state can be described as a state, in which forces in the x- and y-axis equals zero and the uplift force in direction of the z-axis has the same absolute value as the gravitational force. The value of $\Delta\omega$ characterizes the purposed deviation of the required force in hovering state and is used for navigation of the quadrocopter.

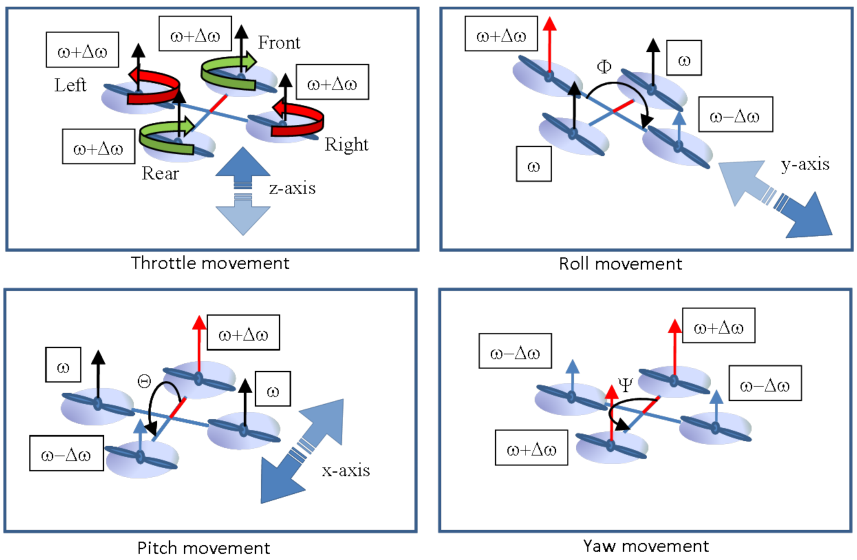

Figure 2.1: Degrees of Freedom of a quadrocopter

To keep the equilibrium of rotational kinematics and to prevent self-rotation, the motor direction of rotation equals crosswise. The quadrocopter has six DOFwhich can be distinguished as angular and translational movements. Translational movements can be executed in x-, y- and z- axis. Accelerating all rotors with the same thrust ω to the value of $\Delta\omega$ will affect a movement in z-direction. Movements to the negative direction of the z-axis are possible, if the thrust of the four rotors is smaller as the gravitational force of the aerial vehicle.

The roll movements can be described as a change of the angle around the x-axis. Thereby the left and right rotors execute a force difference by slowing down the one and simultaneous increasing the other thrust with $\Delta\omega$. Related to the thrust difference and angular movement, the aerial vehicle creates a force to the y-axis. Equivalent to roll, the pitch movement is executed with a change of the angle around the y-axis. Also the pitch movement creates a translational movement across the x-axis. Pitch and roll can only reach a stable angular state and accelerate to the x- or y-axis, if the value of $\Delta\omega$ is the same at the diagonal rotors. Otherwise, the quadrocopter would pure rotate across the corresponding axis.

The yaw movement is a rotation around the z-axis. This angular movement results in combination of pairwise different thrusts and takes as long as the thrusts are different [Ped05, p.p.8-11 Quadrotor model and system].
Synthesizing the weaknesses of MEMSlike noise or the limited resolution, together with the characteristics of the kinematics of the quadrocopter shows the problems which come up in case of MAVcontrol with a IMUwith MEMS. As mentioned smoothly accelerations could be a drawback for the control system, because this kind of accelerations gets lost in the sensor noise.

Regarding the kinematics of the quadrocopter, we have seen that translational movements can only be reached with angular movements across the x- and y-axis of with increasing or decreasing the thrust on every rotor in the z-axis. Imprecise noisy sensors would affect a wrong thrust regulation at the actuators. Thereby it is highly improbable, that all four rotors affected by the noise equal and accelerate in the z-axis. More probable as a uncontrollable acceleration in the direction of the z-axis, is that pairwise two rotors have the some drift and create so an unexpected yaw rotation. Finally the most probable will be a wrong thrust regulation at one rotor, but for a short time interval. So the most likely error drifts will exist at the x- and y-axis. So the hovering state will be nearly impossible to reach just with MEMSacceleration sensors, because the x- and y- movements have to be zero.

The following figure 2.2 shows the described problem area of MEMSsensors in the concrete case of the sensors which are mounted at the quadrocopter CCUof the HSE. The mechanical characteristics of the translational acceleration sensor [ST 05, pp.37-39 Typical performance characteristics] were simulated under consideration of the change of the position of the quadrocopter and the applied acceleration.

We can see, that accelerations which are small enough, get lost in the sensor noise and are invisible for the quadrocopter. Furthermore the quadrocopter reacts in t=12 if the acceleration transcends the threshold value of the sensor noise.

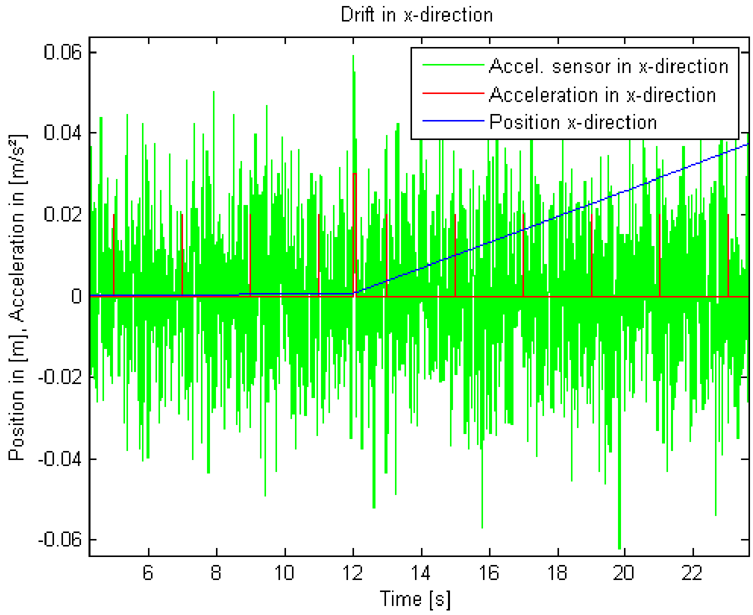

Figure 2.2: The impact of sensor noise

3 Initial Survey

An enormous quantity of researches, which is sponsored by industry companies or universities, was executed to find a better approach to stabilize UAVby using different sensors. Movement detection approaches, which are ultrasonic, sonar or Radio Frequency (RF)based, show that it is necessary to have known reference points to get a reliable result [Jue09, p.2 Ultrasound indoor localization with reference points][Nir02, pp.4-5 Radio Model Localization]. The problem of this approach is that the environment has to be prepared before the UAV flight. This preparation is a drawback in point of flexibility in different operation places. Anyway, approaches with ultrasonic, RFor sonar sensors show that the localization of UAVneeds a kind of global feedback to correct the UAVabsolute position. One of the first motivations for a vision based sensor was presented by Ettlinger et al. [Sco04, pp.1-2 Visual-Based Localization and Control]. In this paper, the authors suggest, that vision is the only practical solution for obstacles of flight stability and showed an On-board approach for a Gliding Unmanned Aerial Vehicle(s) (GUAV)by detecting the horizon with a forward looking camera and estimating and control the flight attitude. Further vision based attitude control approaches also were used for Hovering Unmanned Aerial Vehicle(s) (HUAV)with the difference of a down looking camera for reference point free movement detection or an Off-Board camera which tracks the global position of the UAV.

3.1 On- vs. Off-Board Camera and Image Processing

The Off-Board camera approach was researched by Altug et al. [Erd02, p.76 Localization and Control with an Off-Board camera], with the result of a less sensitive feature detection and position localization as the On-Board camera approach. Off-Board camera tracking is also shown in the developments of extremely reliable and precise localization of a UAVand it is actually used in the development of aggressive autonomous flights of multiple MAVin the experiments of Mellinger et al. [Dan10, Trajectory Generation and Control with Off-Board cameras] [Dan07, pp.363-364]. The advantage of the Off-Board camera tracking system is that the image capturing and position tracking is executed outside the UAVand prevents complex On-Board calculations for movement and position estimation. The drawbacks of this Off-Board Tracking methods are that they need a previous prepared testbed for the flying environment, which is build up with a motion tracking system [Nat10, pp.1-2 The GRASP Testbed].

Problems, such as a limited power resource, a poor level of algorithm complexity for image processing resulting from the limited calculating On-Board performance and the endeavour to economise weight, lead to outsourcing the image processing to a remote system which is not concerned to the On-Board problems. Langer et al. [Sve08, pp.5-7 Off-Board Image Processing] uses an Off-Board image processing to track a landing pad for autonomous landing. Thereby the images are transmitted in this approach over Wifi communication to a base station, which tracks the landing pad and sends back information of position control. Thereby it was shown that the landing algorithm has to run at maximal 100Hz to work with the transmitting delays of the images.

This landing algorithm shows that it is possible to outsource successfully the image processing in consideration of the sample rate which the algorithm needs. By regarding the drawback of the image transmission delay of the Off-Board image processing, On-Board approaches were developed with the focus to the Real-Time behaviour. Tippetts [Bea08, pp.21-22 On-Board image processing with an FPGA] realized an On-Board approach with a Field-Programmable Gate Array (FPGA)which uses a complex feature tracking algorithm but runs with 100MHz sampling rate. The characteristics of the feature tracking with a FPGAcan result a fast movement tracking method, but it is not an efficient method in the behaviour of power consumption because the hardware is not optimized for the image processing tasks.

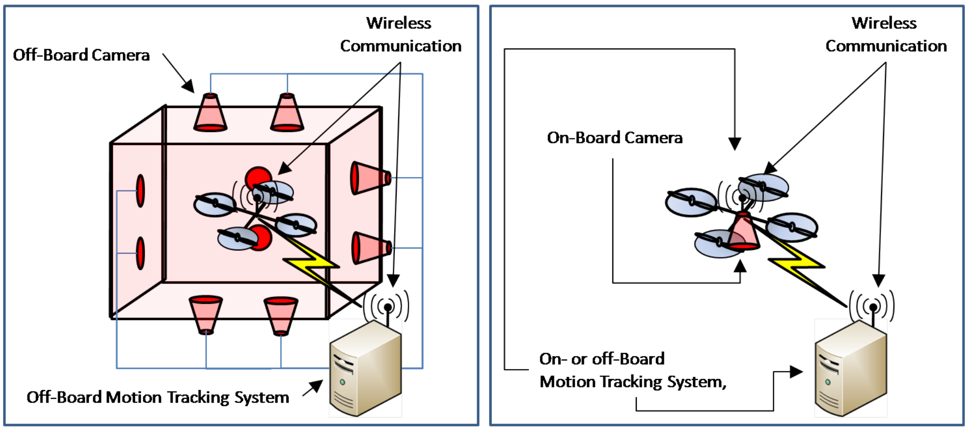

Figure 3.1: On- and Off-Board camera and motion tracking system approach

3.2 Hardware vs. Software-based image processing

In contrast to the drawbacks of FPGAs, Langer et al. [Sve09, On-Board image processing with mice sensors] and Beyeler et al. [Ant09, pp.4-5] showed an approach for detecting the spatial movement of a UAVwith optical mice sensors which have an optimized hardware for image processing and are lightweight.

These sensors calculate the optical flow of the captured images and estimates the movement direction of the UAVin hardware and can provide a high sample rate. In contrast to the fast movement detection, a disadvantage, is that these sensors have limitations related to the operating environments. These limitations are the concrete light and distance range which is required from the manufacturer [ST 05, pp.7-15 Limitations of the operating environment]. The most of the experiments with a stabilisation approach with mice sensors uses optical lens to reach a higher degree of operating distance, but the effort in most of the cases is not satisfactorily to the results as Langer et. al have shown [Sve09, p.6 Performance of the optical flow based position controller]. Software based image processing approaches have a more flexible extension and change behaviour, but they work not as fast as hardware based approaches. Further advantages of software based image processing are, that the most of the commercial and open source computer vision Application Programming Interface(s) (API)also provide implemented estimators and filters to correct the captured input and to decrease noise of the images. Stowers et al. realized a heading estimation for a quadrocopter with an onboard Single Board Computer(s) (SBC)which runs the open source computer vision toolkit OpenCV[Ope10, The OpenCV Reference]. This software based image processing approach shows strengths in the modularity of the image processing architecture and in the interchangeability of the vision system [Joh09, pp.1-6 Software Based Vision Processing][Gar08, Software Structure and Porability].

3.3 Vision based movement detection approaches

Sequential captured images contain a huge amount of information about the absolute and relative movement of objects in every direction. So several vision based movement detection approaches were researched in the topic of UAVstabilisation with different requirements to the information which extracted from the vision process. A simple approach for a relative vision based control of a UAVwas implemented by Boabdallah [Sam07, pp.110-114 Position Sensor] by using a down looking camera and the Canny edge detector [Joh86] and the Douglas Peuker Algorithm for curve equalisation [DP73]. The drawback of this approach is that the field of vision must contain forms with edges which mean that the approach cannot result a satisfactory result if no edges are detected. A further approach for detecting relative movements and to build up a map for autonomous navigation, is visual Simultaneous Localization And Mapping (SLAM). This approach tracks features in the field of vision and reconstructs a relation to tracked features of previous images. The realization of SLAM[Bri07] in a UAVwas executed in the work of Bloesch et al. [Mic10] under consideration of real-time characteristics. The behaviour of the algorithm shows that the localization of the tracked features and the simultaneous mapping has a big impact at the time delay of the calculation. So the experiments and the control algorithms were researched and designed for 7.5Hz sample rate. Another popular tracking method for movement detection is given with the Optical Flow, which can be described as movement observation of tracked objects or pixels in a sequence of images. Algorithms to calculate the Optical Flow were introduced by Horn and Schunk [Ber80], Lukas and Kanade [Bru81].

A few years after introducing the Lucas-Kanade-Algorithm, Lucas described the theoretical approach of visual navigation by using the Optical Flow. Thereby he described the possibility to detect movements and to calculate correspondence velocities. These velocities can be combined to a vector field which can describe movements in every direction [Bru84, pp.40-45 Optical Navigation Theory].

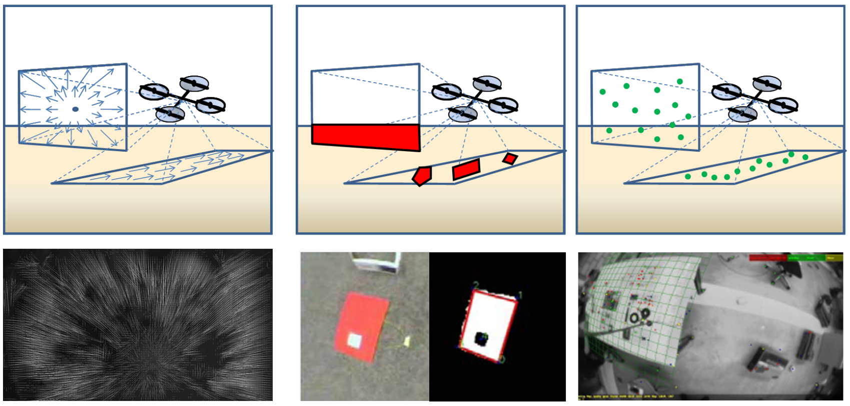

Figure 3.2: Samples for UAV stabilisation with Optical Flow, Edge detection and SLAM

3.4 Control Approaches

The closed loop feedback architecture is an approved method for controlling systems and nearly used in the most of the systems controls in industry and society. So the most of the researches in the UAVstabilisation topic were and still are executed with closed loop control architectures which are built up as Multiple-Input and Multiple-Output (MIMO)systems. The differences between the researched approaches are the amount of the inputs and outputs of the physical process, the type of controller, the used measurements and the sampling rate. Thereby the type and behaviour of the controller is close related to the measurements and the physical process [Ric01, p.3]. The classical control approach using Proportional-Integral-Derivative (PID)controler was researched in several HUAVprojects [Ved08, pp.24-31] [Sam07, pp.43-68]. These researches show that the PIDcontrol is not robust enough to handle with measurement errors and to control multiple DOFs. So the most of the released control approaches with PIDtechnique extends the control architecture with estimators and filters to get more precise measurements. Another approach for control improvement was introduced by Luenberger [Dav71] and describes a closed loop control approach, called state-observer, which simulates the process in real time parallel to the true process by using the input and output vector of the closed loop system and corrects the control strategy. Bloesch et al. [Mic10, p.5] have shown in their research, that the problematic of sensors with non-negligible time delay can be solved to an adequate result by using a state-observer.

The reason for that is that the state-observer allows configurate and to control the stability behaviour of the closed loop feedback architecture. If a process can be observed or furthermore controlled, is related to the states in which it can be and furthermore to the measurements of the DOF[Ric01, pp.632-636 Controllability and Observability]. In this project it has to be determined how correction values which are provided from the optical sensor with a slower sampling rate, can include into the control algorithm of the aircraft.

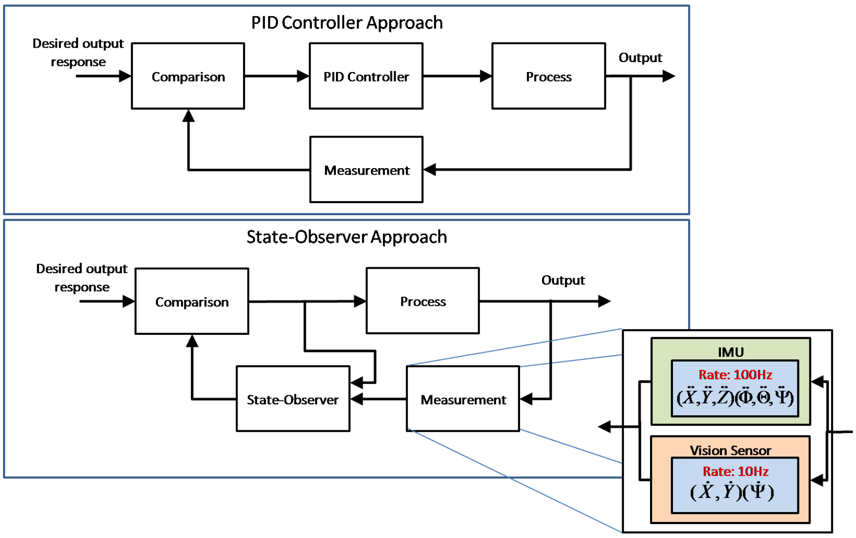

Figure 3.3: PID-Controler and State-Observer in closed loop System with variable sample sensor rates

3.5 Proposed Approach

This chapter shows several approaches for movement detection and stabilisation of UAV. The found methods were critically analyzed and assessed with the result to investigate the following techniques which are shown in figure 3.4. The proposed solution to eliminate the stabilisation problems of the quarocopter has to be vision-based for achieving the goal of flexibility and independence of operational environment. Related to the off-board image processing, the problem of the processing delay, could be solved with a state-observer. Furthermore the optical movement detection should work without reference points. So the optical flow method has to be researched by using a software-based detection approach for more flexibility and replaceability. Because the approach of the flight control is teleoperated and the flight stabilisation does not have to provide aggressive flight maneuvers, it is simpler to realize the camera view on-board. This proposed solution has to be simulated to check the feasibility and to determine the limitations.

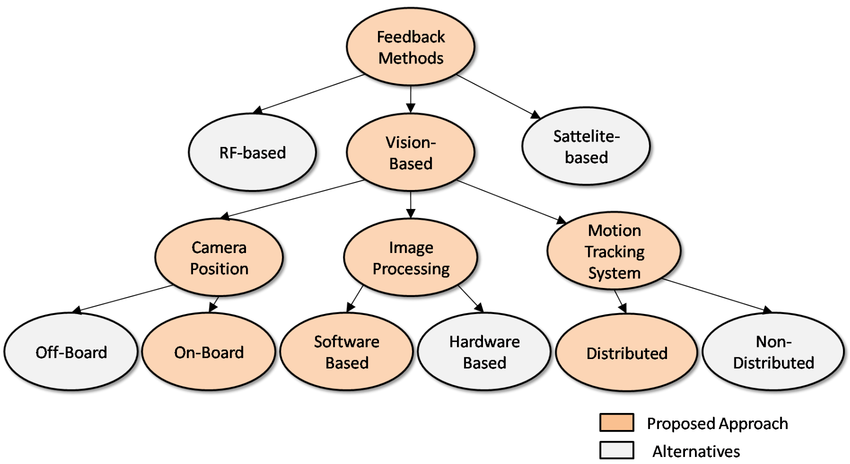

Figure 3.4: Proposed Approach

4 Aims and Objectives

The aim of this dissertation is to provide a simulation architecture that can be used as prototype development platform for a distributed visual movement detection and control of a quadrocopter. Furthermore the characteristics of the distributed image processing and movement detection have to be analyzed in relation to the variation of configurations and critically assessed.

One essential objective is that the configuration of the simulating components provides the option to simulate a range of hardware components which are not purchased until now. By way of example the simulation of the on-Board camera has to provide options of configuration for the resolution, color intensity, blur and so far. The simulation of the communication between UAVand host also has to provide a variation of transmission rate and further behaviour which could affect the visual movement detection at the base station. The efficiency in relation with the quality of function is an important indicator for the success and acceptance of the distributed movement detection approach. So it is important to get an insight to the possible characteristics of the simulated components with the result to find a way which satisfies the efficiency and quality aspects.

Another important objective is that the interfaces between the simulating components are clearly specified and allow a way of modular exchangeability of simulation components with the real objects.

This purpose has to allow a more precise investigation of the behaviour of the real hardware related components and the option to test software for the UAVtarget, like the On-Board control algorithm, at the base station.

The realization of the simulation therefore has to provide an encapsulated and flexible architecture and has to simulate behaviour like delays and jitters for simulated components. Thereby the simulation has to adjust a real-time-behaviour in the complete simulated environment and to allow so measurements and prediction of feasibilities with the simulated configuration.

5 Experimental and Investigative Methods

5.1 Development Process

During the initial survey we have seen many approaches for aircraft stability and movement detection of UAV. The most of these approaches were developed iterative under consideration of the upcoming problems and obstacles [Erd02, Erd03, Iterative Development, Single and Dual Camera Feedback] [Sve08, Sve09, Iterative Development, Landing and Position Control Development]. This procedure model also will be appropriate to this project, because the potential risks are difficult to identify. So the outcome of the development process have to be a prototype which can be evaluated, tested and extended. A process model, which provides an appropriate structure to face the iterative prototyping strategy under the consideration of the risk aspects, is given with the spiral model. The classical spiral model has typically four phases in which the product is developed in an incremental evolutionary process. Derivates of this classical model which focus the customer evaluation for quality improvements may three, five or six phases. In the context of this project, the classical four phases spiral model is used for scientific and feasibility study and does not have to provide further customer communication phases [Rog01, pp.36-38 The Spiral Model].

A typical cycle of the four phases spiral model (5.1) begins with the identification of the objectives which have to be elaborated like performance, functionality, flexibility and so far. The next step is to evaluate the alternatives relative to the objectives and constraints and to determine significant sources of risks. After that, the next level iteration of the product is planned. In the special case of this project it is comfortable to use Model Based Design (MBD)in this phase by using the results of the previous phase as input. This input can be a planed prototype or requirements which describe the changes to execute. The output of the MBDphase can be used again in the planning phase for the next iteration [Bar88, pp.64-69 Spiral Model of the Software Process].

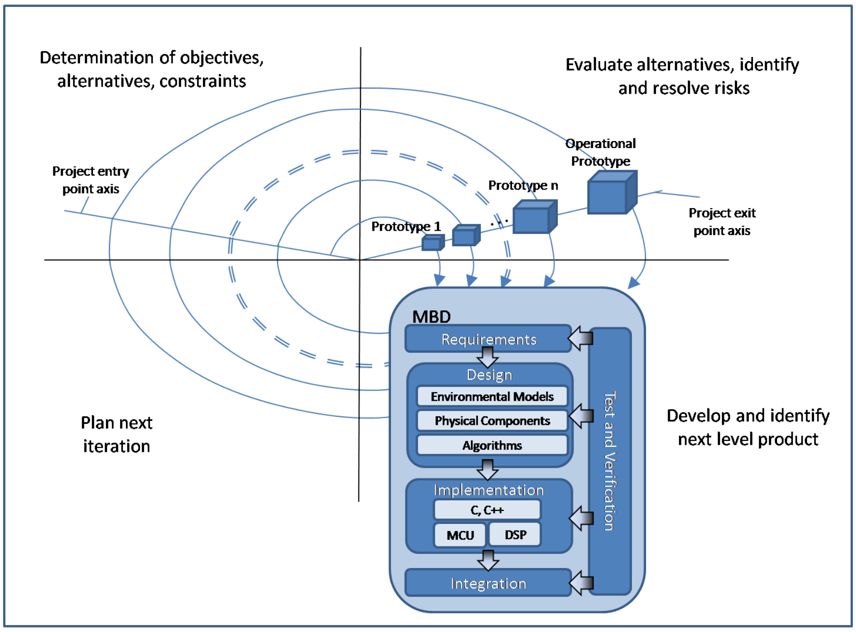

Figure 5.1: The Spiral Model in combination with Model Based Design

5.2 Model Based Design

MBDis a popular method to encounter problems which come up in the development of mechatronic products. These systems almost involve mechanical, electrical, control and embedded components which are developed in different teams of engineers with a specialized focus on the part of the complete project. Some of these components are related on the results of other components before they can be developed. This problem can be solved with Model Based Design and the advantage to develop modules of the complete project by simulating their environment. So the development can run highly parallel with the benefit that the modules can be continuous tested in each phase of the project [Dou09, pp.1-2 Challenges of mechatronic product development].

The development section in figure 5.1 includes the phases and the corresponding key capabilities of MBD. The first phase of MBDis the realization of the researched and required components into a simulation environment. Thereby physical components, environmental model and algorithms are abstracted to systems by using domain-specific modeling tools with a well defined edges and intercommunication. The developed systems in the design phase can be tested simultaneously to analyze the system performance and correctness. Other key capabilities of MBDare given in the implementation phase. The MBDtool Matlab allows to generate embedded code from the designed systems or to combine handwritten code with the build simulation of the design phase. So the implemented modules can be similarly tested in the adopted simulation environment. Finally components which have passed the tests at the implementation phase can be integrated together.

Ultimately the final product can also be tested with the MBDtool by simulating the environment of the product e.g. in a Hardware In the Loop (HIL)test bench. [Mat10, Model Based Design, MathWorks]

5.3 Overall Design Model

The overall design model gives an overview of the components of the simulation which has to be realized in the context of this project and the corresponding applied techniques. As visualised in figure 5.2, the left side of the simulation abstracts the embedded system of the quadrocopter. Inside these components the plant or physical model has to simulate the movement behaviour in the DOFof the quadrocopter. Furthermore the controller has to correct the position of the UAVand to compensate outside disturbances with information of the IMUand the distributed correction value of the base station. The environment simulation which has to include the underground and disturbances simulation will be triggered to start just as the rest of the simulation with a task which describes the ideal flight manoeuvre. The transmissions of the camera IMUand correction data have to be delayed in relation to a configurable transmission rate. The image processing has to be executed with an application, realized in OpenCV [Ope10, OpenCV Project] and invoked by Matlab [Rac10, OpenCV and MEX-Functions in Matlab]. The calculated drift from image processing finally has to be compared with the received values of the IMUto determine the position correction value. This correction has to be transmitted to the embedded system and included in the correction of the controller.

The novel aspect of the project here is reflected in the approach to simulate the complete embedded system, the transmission process, the base station and to realize the image processing in an application. The strength of this approach is that the simulation gives better results and a deep insight to the concurrent processes and helps to understand the challenges and limitations of the investigated scheme.

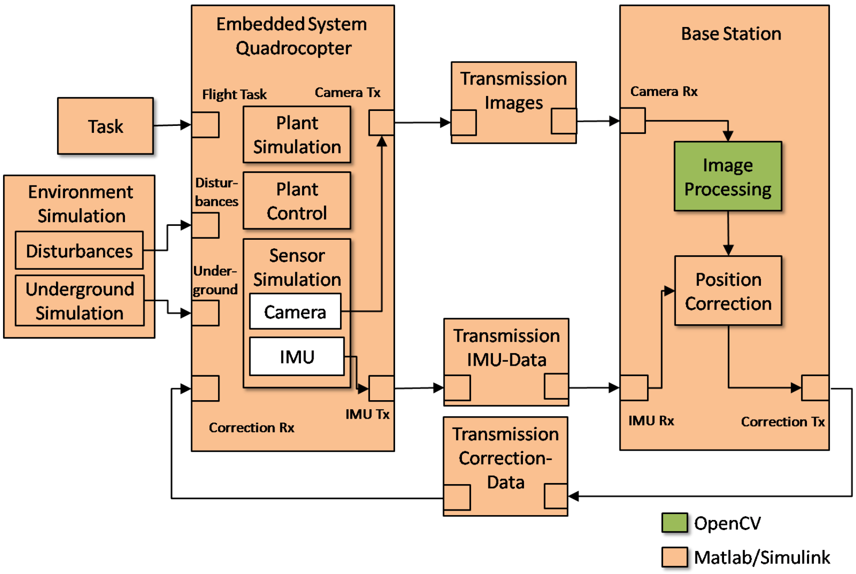

Figure 5.2: The Overall Design Model of the Simulation

6 Project Plan

The Project Plan of this Master's Dissertation is visualized in figure 6.1. The critical, red path is the result of the parallelization of some work packages. These packages mostly are independent and include idle time delays, so that it is more efficient to parallelise them. For example we can regard the simulation and research together with the Master's Thesis documentation. These work packages can be parallelized, because the simulations which have to be executed are time intensive and not a basic part of the complete documentation. Furthermore we can see that one cycle of the spiral model will be executed in the context of this Master's Thesis including an iteration of the MBDprocess.

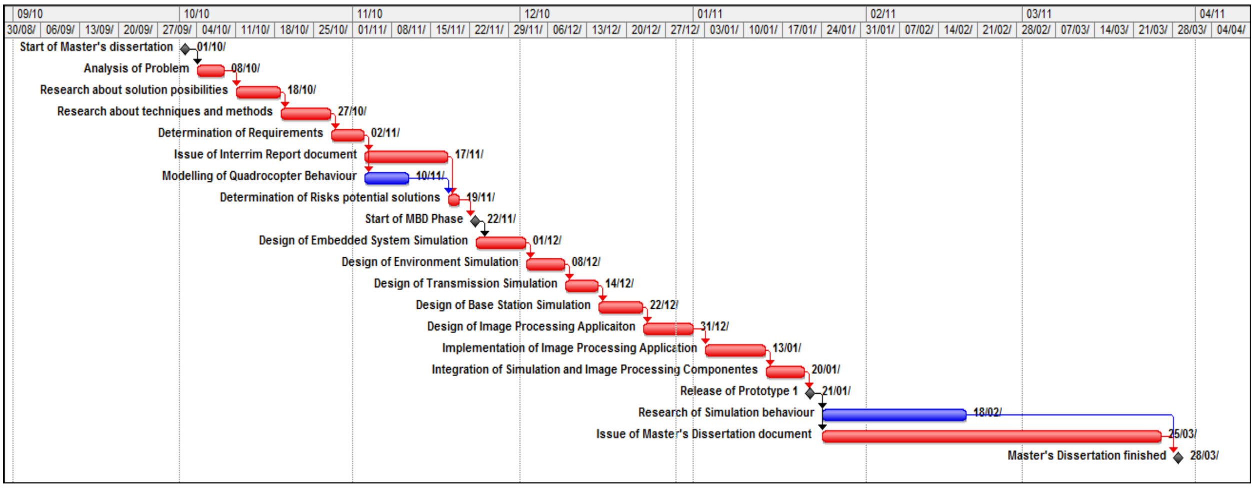

Figure 6.1: The Project Plan

7 Deliverables and Outcomes

The expected outcome of the Master's Thesis is the research and feasibility study of the distributed position correction scheme which has to reflect the improvements of drift eliminations. This has to be realized with a simulation prototype which reflects behaviour of the real components and allows a look insight the complete system. Another outcome has to be the image processing application at the base station which has to collaborate with the simulation.

8 Bibliography

[Ant09] Antoine Beyeler, Jean-Christophe Zufferey, Dario Floreano. optiPilot: control of take-off and landing using optic flow. Technical report, Laboratory of Intelligent Systems, École Polytechnique Fédérale de Lausanne (EPFL), 2009.

[Bar88] Barry Boehm. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5):pp.61–72, 1988.

[Bea08] Beau J. Tippetts. Real-Time Implementation of Vision Algorithms For Control, Stabilization, and Target Tracking, for a Hovering Micro-UAV. Master's thesis, Department of Electrical and Computer Engineering, Brigham Young University, August 2008.

[Ber80] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1980.

[Bri07] Brian Williams, Georg Klein and Ian Reid. Real-Time SLAM Relocalisation. Technical report, Department of Engineering Science, University of Oxford, 2007.

[Bru81] Bruce D. Lucas, Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages p.p.121–130, 1981.

[Bru84] Bruce D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Robotics Institute, Carnegie Mellon University, July 1984.

[Dan07] Daniel Gurdan, Jan Stumpf, Michael Achtelik, Klaus-Michael Doth, Gerd Hirzinger, Daniela Rus. Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. *IEEE International Conference of Robotics and Automation, Roma, Italy* , April 2007.

[Dan10] Daniel Mellinger, Nathan Michael and Vijay Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *International Symposium on Experimental Robotics, Delhi, India*, December 2010.

[Dav71] David G. Luenberger. An Introduction to Observers. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-16, NO. 6*, December 1971.

[Dou09] Douglas Eastman, Paul Lambrechts and Arkadiy Turevskiy. Design and Verification of Motion Control Algorithms Using Simulation. *The MathWorks, Inc.*, 2009.

[DP73] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, pages 112–122, October 1973.

[Erd02] Erdinc Altug, James P. Ostrowski, Robert Mahony. Control of a Quadrotor Helicopter Using Visual Feedback. *IEEE International Conference of Robotics and Automation, Washington, DC* , 2002.

[Erd03] Erdinc¸ Altug, James P. Ostrowski, Camillo J. Taylor. Quadrotor Control Using Dual Camera Visual Feedback. *IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, 2003.

[Gar08] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly, First Edition edition, 2008.

[HSE10] Hochschule Esslingen University of Applied Sciences. http://www.hs-esslingen.de, September 2010.

[Joh86] John Canny. A Computational Approach to Edge Detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6*, November 1986.

[Joh09] John Stowers, Andrew Bainbridge-Smith, Michael Hayes and Steven Mills. Optical Flow for Heading Estimation of a Quadrotor Helicopter. Technical report, University of Canterbury, New Zealand, 2009.

[Jue09] Juergen Eckert, Falko Dressler and Reinhard German. An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes. Technical report, University of Erlangen Dept. of Computer Science, February 2009.

[Kar10] Karl E. Wenzel, Andreas Masselli and Andreas Zell. Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle. Technical report, Department of Computer Science, University of Tuebingen, 2010.

[Mar08] Martin Fach. *Navigation und Regelung eines Luftschiffes mittels optischer,* *inertialer und GPS Sensoren*. PhD thesis, Institut für Flugmechanik und Flugregelung, University of Stuttgart, 2008.

[Mat10] Model Based Design, The MathWorks, Inc. http://www.mathworks.co.uk/model-based-design/, September 2010.

[Mic10] Michael Bloesch, Stephan Weiss, Davide Scaramuzza and Roland Siegwart. Vision Based MAV Navigation in Unknown and Unstructured Environments. Technical report, Autonomous Systems Lab, ETH Zuerich, 2010.

[Nat10] Nathan Michael, Daniel Mellinger, Quentin Lindsey and Vijay Kumar. The GRASP Multiple Micro UAV Testbed. Technical report, GRASP Laboratory, University of Pennsylvania, 2010.

[Nir02] Nirupama Bulusu, John Heidemann and Deborah Estrin. GPS-less Low Cost Outdoor Localization For Very Small Devices. Technical report, University of Southern California / Information Sciences Institute, 2002.

[Ope10] Opencv 2.0 c reference. http://opencv.willowgarage.com/, 2010.

[Ped05] Pedro Castillo, Rogelio Lozano and Alejandro Dzul. Stabilization of a Mini Rotorcraft with Four Rotors. *IEEE Control Systems Magazine*, December 2005.

[Rac10] Rachely Esman and Yoad Snapir. Opencv and mex-files quick guide, 2010.

[Ric01] Richard C. Dorf, Robert H. Bishop. *Modern Control Systems*. Prendice-

Hall, Inc., ninth edition edition, 2001.

[Rog01] Roger R. Pressman. *Software Engineering, A Practitioner's Approach*. McGraw-Hill, Inc., fifth edition edition, 2001.

[Sam04] Samir Bouabdalla, Pierpaolo Murrieri, Roland Siegwart. Design and Control of an Indoor Micro Quadrotor. Technical report, Autonomous Systems Laboratory Swiss Federal Institute of Technology, University of Pisa, 2004.

[Sam07] Samir Bouabdallah. *Design and Control of Quadrotors with Application to Autonomous Flying*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2007.

[Sco04] Scott M. Ettinger, Michael C. Nechyba, Peter G. Ifju and Martin Waszak. Vision-Guided Flight Stability and Control for Micro Air Vehicles. Technical report, Intel Corporation, University of Florida, NASA Langley Research Center, March 2004.

[Seb08] Sebastian Blumenthal, Dirk Holz, Thorsten Linder, Peter Molitor, Hartmut Surmann and Viatcheslav Tretyakov. Teleoperated Visual Inspection and Surveillance with Unmanned Ground and Aerial Vehicles. Technical report, Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), 2008.

[Sla09] Slawomir Grzonka, Giorgio Grisetti, Wolfram Burgard. Towards a Navigation System for Autonomous Indoor Flying. Technical report, Autonomous Systems Lab, Department of Computer Science, University of Freiburg, 2009.

[ST 05] ST Microelectronics. MEMS Inertial Sensor LIS3LV02DQ Rev 1 Datasheet, October 2005.

[Sve08] Sven Lange, Niko Sünderhauf, and Peter Protzel. Autonomous Landing for a Multirotor UAV Using Vision. *Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice, Italy*, November 2008.

[Sve09] Sven Lange, Niko Sünderhauf, Peter Protzel. A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments. Technical report, Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, 2009.

[Ved08] Vedran Sikiric. Control of Quadrocopter. Master's thesis, Royal Institute of Technology Sweden, School of Computer Science and Communication, 2008.