

MASTER OF SCIENCE THESIS

Design, implementation and flight test of indoor navigation and control system for a quadrotor UAV

Menno Wierema B.Sc.

December 11, 2008



Design, implementation and flight test of indoor navigation and control system for a quadrotor UAV

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace
Engineering at Delft University of Technology

Menno Wierema B.Sc.

December 11, 2008



Delft University of Technology

Copyright © Menno Wierema B.Sc.
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
CONTROL AND SIMULATION

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled "**Design, implementation and flight test of indoor navigation and control system for a quadrotor UAV**" by **Menno Wierema B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: December 11, 2008

Professor: _____
Prof. dr. ir. J.A. Mulder

Assistant-Professor: _____
Dr. Q.P. Chu

Supervisor: _____
Ir. C. de Wagter

Reader: _____
J. Breeman (NLR)

Abstract

An unmanned aerial vehicle (UAV) is an unpiloted aircraft which can either be remotely controlled or fly autonomously based on pre-programmed flight plans. Unmanned aerial vehicles are often used in military applications for missions that are too dull, dirty, or dangerous for manned aircraft. They are also used in a growing number of civil applications such as aerial photography.

Rotating wing (or helicopter) UAVs have the advantage above fixed wing UAVs that they are able to take off and land vertical, and that it is possible to hover at a fixed point. One very successful design for smaller UAVs is an helicopter with four horizontal rotors and no tailrotor, a such called *quadrotor*. Quadrotors have the advantage that they can be controlled by varying the speed of the rotors and thus fixed-pitch blades can be used which simplifies the design and control of the vehicle. Secondly, the use of four rotors allows each individual rotor to have a smaller diameter than the equivalent helicopter rotor, for a given vehicle size, allowing them to store less kinetic energy during flight.

In order to do autonomous flight with a quadrotor an accurate estimation of the current position of the vehicle is required. In outdoor situations a Global Position System (GPS) receiver can be used to measure velocity and position with respect to the earth. However the use of GPS is not possible in indoor environments. Therefore an alternative measurement of position is required indoor. A straightforward approach would be the double integration of the accelerometers measurements of an Inertial Measurement Unit (IMU), this is however not possible because not only the body acceleration is measured, but also the gravitational vector, the accelerations due to rotation and a bias.

In this work it is proposed to combine three gyroscopes, three accelerometers and three magnetometers of an IMU with infrared sensors to achieve accurate state estimation. This tightly coupled sensor integration allows estimation of all the states of the system and the biases of both the gyroscopes and accelerometers and estimation of all states of the quadrotor. The relationships between the measurements and states are described by non-linear equations, therefore the Extended Kalman Filter (EKF) is used to perform state estimation.

To validate this approach, first the state estimation is tested in simulation. Subsequently, a quadrotor UAV is built to perform flight tests and test the filter with real measurements.

The simulations show that using six infrared sensors in orthogonal directions, this tightly coupled sensor integration can accurately track the states of the system. Also the biases of both the gyroscopes and accelerometers can be correctly estimated. The filter is still observable using a minimum of four sensors in any direction, however for accurate tracking a minimum of five

infrared sensors are preferred. Due to the linearisations in the EKF the result is not optimal, but results of the filtering can be improved by tuning the initial settings of the process and measurement noises.

A quadrotor UAV is built to be able to do flight test, equipped with three gyroscopes, three accelerometers and three magnetometers and six infrared sensors. Onboard the fixed point processor of the quadrotor filtering and control is implemented, which stabilises the attitude angles of the quadrotor. The attitude feedback allows even inexperienced pilots to fly the quadrotor. Measurements in a real-world test environment show that the state estimation algorithm is stable and able to give an estimate of the states.

"In theory there is no difference between theory and practice. In practice there is."

Yogi Berra

Preface

Finishing a master of science thesis is a great achievement, which would not have been possible without any help. Writing this thesis felt like a rollercoaster ride, with many ups and down, and I would like to thank every person that helped me to finalise it.

There are two persons in particular to whom I wish to express my sincere gratitude. First of all, Dr. Q.P. Chu, my direct supervisor, for his endless support, patience and nice discussions not only about navigation and control, but also outside the field of aerospace engineering. He has an admirable passion for modern flight test technologies and system identification, which gave me the motivation to pursue a thesis in this field.

Secondly, ir. C. de Wagter, one of the founders of the Micro Aerial Vehicle Lab (MAVLab), who showed me his enthusiasm in unmanned aerial vehicles (UAVs) already during my Bachelor Project. Without his daily support, critical judgement and flexibility the building of the actual quadrotor would never be a fact.

I would also like to thank the other members of the MAVLab: B. Remes, R. Ruijsink and G. de Croon for their very practical help in building the quadrotor and for asking every day again: *Does it already fly?*. The people of the MAVLab show their unparalleled enthusiasm in applying theory in practice, which is a very valuable addition to the Faculty of Aerospace Engineering of Delft Technical University. Next, I would like to thank Prof. dr. ir. J.A. Mulder for his great supervision of the Department of Control and Simulation and bringing over his enthusiasm for the research in this field to students.

Without the very brave boys and girl from the *Egg* this thesis would never be a fact, thanks to their the day-to-day support and for leaving a spot free for me, even when I had my thesis break. Special thanks go to my girlfriend Sibrich, who helped me to get through the tough times of writing a thesis.

Last, but certainly not least, I would like to thank my parents, for their life-long support and sometimes necessary pressure to finalise this thesis.

Delft, University of Technology
December 11, 2008

Menno Wierema B.Sc.

Table of Contents

| | |
|--|----------|
| Abstract | v |
| Preface | viii |
| Nomenclature | xxi |
| Acronyms | xxv |
| | |
| I Introduction | 1 |
| 1 Introduction | 3 |
| 1-1 Quadrotor concept | 4 |
| 1-2 Autonomous flight | 5 |
| 1-3 Problem statement | 5 |
| 1-4 Literature overview | 6 |
| 1-5 Objective | 7 |
| 1-6 Outline of the report | 8 |
| | |
| 2 Overview of quadrotor platforms | 9 |
| 2-1 Early history | 9 |
| 2-2 Modern successors | 10 |
| 2-3 Main research projects | 11 |
| 2-4 Commercially available | 16 |
| 2-5 Open source projects | 17 |
| 2-6 Conclusions | 17 |

| | |
|--|-----------|
| II Quadrotor modelling and control design | 19 |
| 3 Quadrotor modelling | 21 |
| 3-1 Reference frames | 21 |
| 3-2 Kinematic relations | 22 |
| 3-3 Assumptions | 23 |
| 3-4 Aerodynamic forces and moment of a rotor | 23 |
| 3-5 Dynamic equations | 24 |
| 3-6 Additional effects | 26 |
| 3-7 Simplified model for control & filter design | 29 |
| 3-8 Motor dynamics | 29 |
| 3-9 Matlab-Simulink model | 30 |
| 3-10 Conclusions and recommendations | 30 |
| 4 Control design | 31 |
| 4-1 Selection of the control method | 31 |
| 4-2 Classical control | 35 |
| 4-3 Quadrotor controller structure | 36 |
| 4-4 Results | 39 |
| 4-5 Conclusions and recommendations | 41 |
| III Indoor navigation using state estimation | 43 |
| 5 Sensor modelling | 45 |
| 5-1 3-Axis gyroscope | 45 |
| 5-2 3-Axis accelerometer | 47 |
| 5-3 3-Axis magnetometer | 48 |
| 5-4 Infrared sensors | 50 |
| 5-5 Conclusions and recommendations | 54 |
| 6 Indoor navigation using state estimation | 55 |
| 6-1 Extended Kalman filter (EKF) | 55 |
| 6-2 Validation of the Kalman Filter | 58 |
| 6-3 Attitude and Heading Reference System (AHRS) | 59 |
| 6-4 Full positioning system using IR-sensors | 62 |
| 6-5 Conclusions | 64 |
| 6-6 Recommendations | 64 |
| IV Design and flight tests | 65 |
| 7 Design of the test platform | 67 |
| 7-1 Original hardware | 67 |
| 7-2 Requirements | 67 |
| 7-3 New hardware design | 68 |
| 7-4 Results & Conclusions | 73 |
| 7-5 Recommendations | 74 |

| | |
|--|------------|
| 8 Onboard filtering and control | 75 |
| 8-1 Optimised onboard processing | 75 |
| 8-2 Steady state fixed point Kalman filter for roll and pitch estimate | 77 |
| 8-3 Yaw estimation | 79 |
| 8-4 Feedback controller | 80 |
| 8-5 Results | 81 |
| 8-6 Conclusions | 82 |
| 8-7 Recommendations | 82 |
| 9 Indoor navigation with real data | 83 |
| 9-1 Calibration and Preprocessing | 83 |
| 9-2 Results of the AHRS Kalman filter | 86 |
| 9-3 Results of the full Kalman filter | 87 |
| 9-4 Conclusions | 87 |
| 9-5 Recommendations | 88 |
| V Conclusions | 91 |
| 10 Conclusions | 93 |
| 10-1 Conclusions | 93 |
| 10-2 Discussion | 94 |
| 10-3 Future work | 94 |
| VI Appendices | 95 |
| Bibliography | 103 |
| A Simulation parameters | 103 |
| A-1 Quadrotor | 104 |
| A-2 Sensors | 105 |
| B Simulink Model | 107 |
| C Calibration parameters | 111 |
| C-1 Raw measurements | 112 |
| C-2 Preprocessed and calibrated measurements | 113 |
| D Schematics | 115 |
| E Comparison between complete model and simplified model | 119 |
| F AHRS Kalman filtering tested with simulated data | 123 |
| F-1 Using realistic sensor parameters | 124 |
| F-2 Using 1/10 random walk on sensors | 126 |

| | |
|---|------------|
| G AHRS Kalman filtering tested with real data | 129 |
| G-1 Roll, pitch, yaw movements, with motors off | 130 |
| G-2 Roll, pitch, yaw movements, with motors on | 132 |
| G-3 Actual flight data | 134 |
| H Full Kalman filtering tested with simulated data | 137 |
| H-1 Using 6 infrared sensors, directions X1-X2-Y1-Y2-Z1-Z2 | 138 |
| H-2 Using 5 infrared sensors, directions X1-X2-Y1-Y2-Z1 | 141 |
| H-3 Using 4 infrared sensors, directions X1-X2-Y1-Z1 | 144 |
| H-4 Using 4 infrared sensors, directions X1-Y1-Y2-Z1 | 147 |
| H-5 Using 4 infrared sensors, directions X1-Y1-Z1-Z2 | 150 |
| H-6 Using 3 infrared sensors, unobservable, directions X1-Y1-Z1 | 153 |
| H-7 Using 6 infrared sensors and 1/10 random walk, directions X1-X2-Y1-Y2-Z1-Z2 | 156 |
| I Full Kalman filtering tested with real data | 159 |
| I-1 Using 5 infrared sensors - Pendulum movement - Motors off | 160 |
| I-2 Using 5 infrared sensors - Pendulum movement - Motors on | 164 |
| I-3 Using 5 infrared sensors - Hand movement Motors off | 168 |
| I-4 Using 5 infrared sensors - Actual flight data | 172 |

List of Figures

| | | |
|-----|---|----|
| 1-1 | Quadrotor motion, the arrow width is proportional to propeller rotational speed | 4 |
| 1-2 | Steps in control of a quadrotor and the perception of the control task by a human | 5 |
| 2-1 | Early quadrotors | 10 |
| 2-2 | Modern manned quadrotors | 11 |
| 2-3 | Stanford University quadrotor projects | 12 |
| 2-4 | The EPFL OS4 quadrotor | 14 |
| 2-5 | The ANU MARK II X4 Flyer | 15 |
| 2-6 | Some commercially available quadrotors | 16 |
| 3-1 | Earth fixed reference frame and body fixed reference frame | 22 |
| 3-2 | Forces and moments acting on a rotor | 23 |
| 3-3 | The flapping properties of a stiff, fixed-pitch rotor blade | 28 |
| 3-4 | Blade flapping angles. | 28 |
| 4-1 | Example of a typical response to a unit step input of a system. | 35 |
| 4-2 | Parallel PID control structure | 36 |
| 4-3 | Yaw ψ control loop | 37 |
| 4-4 | Roll ϕ controller | 37 |
| 4-5 | Horizontal position y nested controller | 38 |
| 4-6 | Vertical position z controller | 38 |
| 4-7 | Controller stability in hover | 39 |
| 4-8 | Simulation of a flight tracking waypoints generated by interpolation. | 40 |
| 4-9 | Simulation of a flight tracking waypoints generated by interpolation. | 41 |
| 5-1 | IDG-300 MEMS Sensor | 46 |
| 5-2 | Optical and SEM images of ADXL330 MEMS structure | 48 |
| 5-3 | Definition of the magnetic field parameters | 50 |
| 5-4 | Functioning of the IR sensor - measurement by triangulation | 51 |

| | | |
|-----|--|-----|
| 5-5 | GP2Y0A02YK - Example of output characteristics from the datasheet | 52 |
| 5-6 | Derivation of the measurement equations of the infrared sensors | 54 |
| 6-1 | The operation of the Extended Kalman Filter | 57 |
| 6-2 | AHRS Extended Kalman Filtering setup | 60 |
| 6-3 | EKF Full positioning system using IR-sensors | 62 |
| 7-1 | Original Hardware ESL Quadrotor | 68 |
| 7-2 | New hardware layout | 69 |
| 7-3 | Flying quadrotor | 73 |
| 8-1 | Processes that run in the main loop. | 76 |
| 8-2 | Time invariant Kalman filters | 78 |
| 8-3 | Control activity with only rate feedback | 81 |
| 8-4 | Control activity with rate and angle feedback | 81 |
| 9-1 | Comparison of the output of the accelerometer | 84 |
| 9-2 | 3D - Comparison of the output of the accelerometer | 85 |
| 9-3 | Polynomial estimation of magnetometer throttle effect. | 85 |
| 9-4 | Testroom setup | 87 |
| B-1 | The implementation of the quadrotor dynamics, control and sensors in Simulink | 108 |
| B-2 | Simulink model - Quadrotor UAV block | 109 |
| B-3 | Simulink model - Control block | 109 |
| B-4 | Simulink model - Sensor block | 110 |
| C-1 | Raw measurements | 112 |
| C-2 | Preprocessed and calibrated measurements | 113 |
| D-1 | Schematics of the connections of the Atmega 2560 | 116 |
| D-2 | Schematics of the power group | 117 |
| D-3 | Schematics of the sensors group | 117 |
| D-4 | Schematics of the external connectors | 118 |
| E-1 | Comparision of complete model (blue) with simplified model (red) using same simulation parameters. | 119 |
| E-2 | Comparision of complete model (blue) with simplified model (red) using same simulation parameters. | 120 |
| E-3 | Comparision of complete model (blue) with simplified model (red) using same simulation parameters. | 121 |
| F-1 | AHRS filter results - states | 124 |
| F-2 | AHRS filter results - error | 124 |
| F-3 | AHRS filter results - measurements | 125 |
| F-4 | AHRS filter results - observability degree | 125 |
| F-5 | AHRS filter results - states | 126 |
| F-6 | AHRS filter results - error | 126 |
| F-7 | AHRS filter results - measurements | 127 |
| F-8 | AHRS filter results - observability degree | 127 |

| | |
|--|-----|
| G-1 AHRS filter results - states | 130 |
| G-2 AHRS filter results - error | 130 |
| G-3 AHRS filter results - measurements | 131 |
| G-4 AHRS filter results - states | 132 |
| G-5 AHRS filter results - error | 132 |
| G-6 AHRS filter results - measurements | 133 |
| G-7 AHRS filter results - states | 134 |
| G-8 AHRS filter results - error | 134 |
| G-9 AHRS filter results - measurements | 135 |
| | |
| H-1 FULL filter results - states | 138 |
| H-2 FULL filter results - error | 139 |
| H-3 FULL filter results - measurements | 140 |
| H-4 FULL filter results - observability degree | 140 |
| H-5 FULL filter results - states | 141 |
| H-6 FULL filter results - error | 142 |
| H-7 FULL filter results - measurements | 143 |
| H-8 FULL filter results - observability degree | 143 |
| H-9 FULL filter results - states | 144 |
| H-10 FULL filter results - error | 145 |
| H-11 FULL filter results - measurements | 146 |
| H-12 FULL filter results - observability degree | 146 |
| H-13 FULL filter results - states | 147 |
| H-14 FULL filter results - error | 148 |
| H-15 FULL filter results - measurements | 149 |
| H-16 FULL filter results - observability degree | 149 |
| H-17 FULL filter results - states | 150 |
| H-18 FULL filter results - error | 151 |
| H-19 FULL filter results - measurements | 152 |
| H-20 FULL filter results - observability degree | 152 |
| H-21 FULL filter results - states | 153 |
| H-22 FULL filter results - error | 154 |
| H-23 FULL filter results - measurements | 155 |
| H-24 FULL filter results - observability degree | 155 |
| H-25 FULL filter results - states | 156 |
| H-26 FULL filter results - error | 157 |
| H-27 FULL filter results - measurements | 158 |
| H-28 FULL filter results - observability degree | 158 |
| | |
| I-1 FULL filter results - states | 160 |
| I-2 FULL filter results - error | 161 |
| I-3 FULL filter results - measurements | 162 |
| I-4 FULL filter results - observability degree | 162 |
| I-5 FULL filter results - 3D plot. | 163 |
| I-6 FULL - Tracing of the relative motion by a camera. | 163 |
| I-7 FULL filter results - states | 164 |

| | | |
|------|---|-----|
| I-8 | FULL filter results - error | 165 |
| I-9 | FULL filter results - measurements | 166 |
| I-10 | FULL filter results - observability degree | 166 |
| I-11 | FULL filter results - 3D plot. | 167 |
| I-12 | FULL - Tracing of the relative motion by a camera. | 167 |
| I-13 | FULL filter results - states | 168 |
| I-14 | FULL filter results - error | 169 |
| I-15 | FULL filter results - measurements | 170 |
| I-16 | FULL filter results - observability degree | 170 |
| I-17 | FULL filter results - 3D plot. | 171 |
| I-18 | FULL filter results - states | 172 |
| I-19 | FULL filter results - error | 173 |
| I-20 | FULL filter results - measurements | 174 |
| I-21 | FULL filter results - observability degree | 174 |
| I-22 | FULL filter results filtered states positio states. | 175 |

List of Tables

| | | |
|-----|--|----|
| 4-1 | PID gains in simulation | 39 |
| 7-1 | Hardware changes | 68 |
| 7-2 | Infrared sensors overview | 71 |
| 7-3 | Cost estimation | 73 |
| 8-1 | ADC read order | 76 |
| 8-2 | Downlink address map | 78 |
| 8-3 | Gains of the implemented feedback controller | 80 |

Nomenclature

Greek Symbols

| | |
|------------------|--|
| $\dot{\phi}$ | Roll rate in the earth reference frame [rad/s] |
| $\dot{\phi}_d$ | Desired roll rate in the earth reference frame [rad/s] |
| $\dot{\psi}$ | Yaw rate in the earth reference frame [rad/s] |
| $\dot{\psi}_d$ | Desired yaw rate in the earth reference frame [rad/s] |
| $\dot{\theta}$ | Pitch rate in the earth reference frame [rad/s] |
| $\dot{\theta}_d$ | Desired pitch rate in the earth reference frame [rad/s] |
| ν | Air viscosity at 20° Celsius [Pa s] = 1.810^{-5} Pa s |
| Ω | Propeller speed [rad/s] |
| Ω_H | Propeller speed at hoover [rad/s] = $\sqrt{\frac{W_{prop}}{b}}$ |
| Ω_r | Residual propellers rotational speed [rad/s] = $-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ |
| Ω_{tot} | Total propellers rotational speed [rad/s] = $\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4$ |
| ϕ | Roll angle in the earth reference frame [rad] |
| ϕ_d | Desired roll angle in the earth reference frame [rad] |
| ϕ_d | Desired yaw angle in the earth reference frame [rad] |
| ψ | Yaw angle in the earth reference frame [rad] |
| ρ | Air density at 20° Celsius [kg/m ³] = 1.293 kg/m ³ |
| σ | Solidity ratio (rotor fill ratio) [rad ⁻¹] = $\frac{N_c}{\pi R}$ |
| θ | Pitch angle in the earth reference frame [rad] |
| θ_0 | Pitch of incidence [rad] |
| θ_d | Desired pitch angle in the earth reference frame [rad] |
| θ_{tw} | Twist pitch [rad] |

Latin Symbols

| | |
|--------------------------|---|
| \bar{C}_d | Airfoil drag coefficient of the section corresponding to the 70 percent radial station [-] |
| Δt | Sampling period [s] $= 1/100 \text{ s}$ |
| \dot{p} | Roll rate acceleration in the body reference frame [rad/s] |
| \dot{q} | Pitch rate acceleration in the body reference frame [rad/s] |
| \dot{r} | Yaw rate acceleration in the body reference frame [rad/s] |
| \dot{u} | Horizontal acceleration in forward direction in the body reference frame [m/s^2] |
| \dot{v} | Horizontal acceleration in right sideward direction in the body reference frame [m/s^2] |
| \dot{w} | Vertical acceleration in downward direction in the body reference frame [m/s^2] |
| \dot{x} | Horizontal velocity in forward direction in the earth reference frame [m/s] |
| \dot{x}_d | Desired horizontal velocity in forward direction in the earth reference frame [m/s] |
| \dot{y} | Horizontal velocity in right sideward direction in the earth reference frame [m/s] |
| \dot{y}_d | Desired horizontal velocity in right sideward direction in the earth reference frame [m/s] |
| \dot{z} | Vertical velocity in downward direction in the earth reference frame [m/s] |
| \dot{z}_d | Desired vertical velocity in downward direction in the earth reference frame [m/s] |
| λ | Inflow ratio [-] $= \frac{v_i + \dot{z}}{\Omega_H R}$ |
| μ | Advance ratio [-] $= \frac{\bar{V}}{\Omega_H R}$ |
| A | Propeller disk area [m^2] $= \pi R^2$ |
| a | Lift slope |
| A_u | Aerodynamic area [m^2] |
| b | Thrust factor in hover [N s^2] |
| C_u, C_v, C_w | Translational drag coefficients [-] |
| d | Drag factor in hover [N $\text{m} \text{s}^2$] |
| e_{ss} | Steady state error |
| g | Gravity constant $= 9.8065 \text{ m/s}^2$ |
| h | Vertical distance between CoG and propellers plan [m] |
| I | Identity matrix |
| I_{xx}, I_{yy}, I_{zz} | Inertia components [kg m^2] |
| J | Inertia matrix |
| J_m | Rotor inertia [kg m^2] |
| K_D | Derivative gain |
| K_I | Integral gain |
| K_P | Proportional gain |
| l | Arm length [m] |
| m | Total mass [kg] |
| M_p | Maximum overshoot |
| p | Roll rate in the body reference frame [rad/s] |
| q | Pitch rate in the body reference frame [rad/s] |
| R | Propeller radius [m] |
| r | Yaw rate in the body reference frame [rad/s] |

| | |
|------------|--|
| t_r | Rise time |
| V | Sideways velocity [m/s] $= \sqrt{u^2 + v^2}$ |
| v_i | Inflow velocity [m/s] $= \sqrt{-\frac{1}{2}V^2 + \sqrt{\left(\frac{1}{2}V^2\right)^2 + \left(\frac{W_{prop}}{2\rho A}\right)^2}}$ |
| W_{prop} | Weight of the quadrotor per propeller [N] $= \frac{mg}{4}$ |
| x | Horizontal position in forward direction in the earth reference frame [m] |
| x_d | Desired horizontal position in forward direction in the earth reference frame [m] |
| y | Horizontal position in right sideward direction in the earth reference frame [m] |
| y_d | Desired horizontal position in right sideward direction in the earth reference frame [m] |
| z | Vertical position in downward direction in the earth reference frame [m] |
| z_d | Desired vertical position in downward direction in the earth reference frame [m] |
| u | Horizontal velocity in forward direction in the body reference frame [m/s] |
| v | Horizontal velocity in right sideward direction in the body reference frame [m/s] |
| w | Vertical velocity in downward direction in the body reference frame [m/s] |

Acronyms

| | |
|--------------|--|
| AAHRS | Attitude and Heading Reference System |
| AHRS | Attitude and Heading Reference System |
| ANU | Australia National University |
| CCD | Charge-Coupled Device |
| CEA | Commissariat l'Énergie Atomique |
| CoG | Centre of Gravity |
| DC | Direct Current |
| DCAV | Dynamics and Control of Aerospace Vehicles |
| DGPS | Differential GPS |
| DoF | Degrees Of Freedom |
| EEMCS | Electrical Engineering, Mathematics and Computer Science |
| EKF | Extended Kalman Filter |
| EPFL | Ecole Polytechnique Fédérale de Lausanne |
| ESL | Embedded System Lab |
| FBL | FeedBack Linearisation |
| FET | Field Effect Transistor |
| GPS | Global Positioning System |
| HKF | Hybrid Kalman Filter |
| IEKF | Iterated Extended Kalman Filter |
| ISM | Integral Sliding Mode |
| IGE | In Ground Effect |
| IMU | Inertial Measurement Unit |

| | |
|-----------------|---|
| INS | Inertial Navigation System |
| LQR | Linear-Quadratic Regulator |
| MAVLab | Micro Arial Vehicle Lab |
| MARG | Magnetic, Angular Rate, and Gravity |
| MEMS | Micro-machined ElectroMechanical Systems |
| MRML | Modified Recursive Maximum Likelihood |
| NDI | Non-linear Dynamic Inversion |
| NN | Neural Network |
| OGE | Out of Ground Effect |
| OS4 | Omnidirectional Stationary Flying OUTstretched Robot |
| PID | Proportional-Integral-Derivative |
| PI | Proportional-Integral |
| PD | Proportional-Derivative |
| PCB | Printed Circuit Board |
| PCH | Pseudo-Control Hedging |
| PPM | Pulse-Position Modulation |
| PWM | Pulse-Width Modulation |
| RC | Remote Control |
| RML | Recursive Maximum Likelihood |
| RL | Reinforcement Learning |
| SMC | Sliding Mode Control |
| SPI | Serial Peripheral Interface |
| SODAR | SOnic Detection And Ranging |
| SLAM | Simultaneous Localization And Mapping |
| STARMAC | Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control |
| SVD | Singular Value Decomposition |
| TU Delft | Delft University of Technology |
| TWI | Two-wire Serial Interface |
| UKF | Unscented Kalman Filtering |
| UAV | Unmanned Aerial Vehicle |
| V/STOL | Vertical/Short Take Off and Landing |
| VTOL | Vertical Take Off and Landing |

Part I

Introduction

Chapter 1

Introduction

An unmanned aerial vehicle ([UAV](#)) is an unpiloted aircraft which can either be remotely controlled or fly autonomously based on pre-programmed flight plans. Unmanned aerial vehicles are often used in military applications for missions that are too dull, dirty, or dangerous for manned aircraft. They are also used in a growing number of civil applications such as aerial photography.

Rotating wing (or helicopter) UAVs have the advantage above fixed wing UAVs that they are able to take off and land vertical, and that it is possible to hover at a fixed point. One very successful design for smaller UAVs is an helicopter with four horizontal rotors and no tailrotor, a such called *quadrotor*. Quadrotors have the advantage that they can be controlled by varying the speed of the rotors and thus fixed-pitch blades can be used which simplifies the design and control of the vehicle. Secondly, the use of four rotors allows each individual rotor to have a smaller diameter than the equivalent helicopter rotor, for a given vehicle size, allowing them to store less kinetic energy during flight.

At the Delft University of Technology ([TU Delft](#)) recently interest is gained in the quadrotor UAV design, starting with a Bachelor of Science design exercise in 2007 at the faculty of Aerospace Engineering: *The Insight project* [[Thus et al., 2007](#)]. The goal of this design exercise was to develop an UAV which can be used in life threatening situations to explore terrains and buildings before entering them. In the first phase of the project a quadrotor was designed for this tasks. After the success of the first phase it was decided to continue the project and built the quadrotor as designed with the help of the Micro Aerial Vehicle Lab ([MAVLab](#)), and successive steps were made. However the project was never finished.

For the development of the hardware onboard the Insight quadrotor there has been a close cooperation between the Insight project team and the Embedded System Lab ([ESL](#)) of the faculty of Electrical Engineering, Mathematics and Computer Science ([EEMCS](#)) also at TU Delft. Under supervision of Professor dr.ir. A.J.C. van Gemund, Marc de Hoop, a M.Sc. student, developed the embedded systems capable of controlling both the Insight quadrotor and the quadrotor used in the Embedded Real-Time Systems Course. This quadrotor is already capable of basic flight by an experienced pilot.

The faculty of Aerospace Engineering and the MAVLab of the TU Delft have been successfully working together in several projects with autonomous UAVs. The faculty provides the theoretical background and the MAVLab provides the possibility of applying theory into practice on real flying UAVs.

This project combines the theory of state estimation and flight control in the *design, implementation and flight test of an indoor navigation and control system for a quadrotor UAV*.

The chapter starts with the background of the project and explains the concept of a quadrotor and autonomous flight. Then the problem is stated and the solutions that are found in literature are discussed. From this the objectives of the project are set. Finally an overview of the chapters in this report is given.

1-1 Quadrotor concept

From the first days in helicopter development, the quadrotor layout was seen as an alternative. In a regular helicopter configuration the torque is counteracted by the tail rotor. When using two rotors the torques of the rotors can compensated by each other. But two rotors create still challenges in control as the rotational and translational motions are still highly coupled. With four rotors the control becomes easier and the rotational motions can be decoupled for the gyroscopic effects. Translational motions are achieved by tilting the vehicle.

In the most common layout for quadrotor, the two pairs of rotors (1, 3) and (2, 4) turn in opposite directions as shown in Figure 1-1. By varying the rotor speeds, one can change the lift forces and create motion. Increasing or decreasing the four rotor speeds together generates vertical motion. Changing the 2 and 4 propellers speed conversely produces roll rotation coupled with lateral motion. Pitch rotation and the corresponding lateral motion result from 1 and 3 propellers speed conversely modified. Yaw rotation results from the difference in the counter-torque between the pairs of rotors.

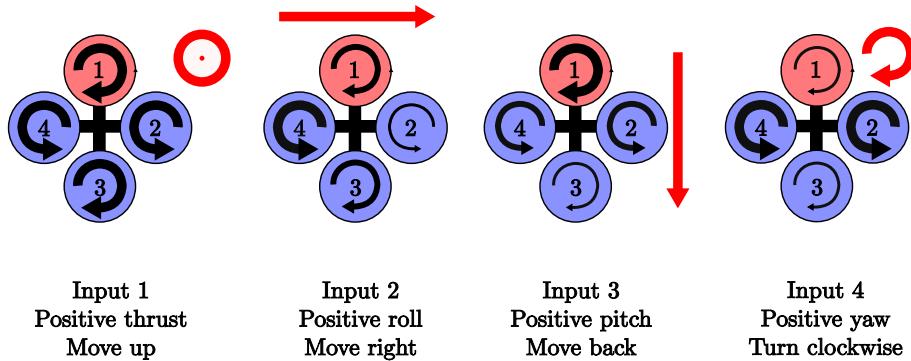


Figure 1-1: Quadrotor motion, the arrow width is proportional to propeller rotational speed

1-1-1 Advantages and disadvantages

As only thrust control can be used to change position, blade-pitch control of the rotors is not necessary and thus rotor mechanics are simplified, which make the quadrotor layout interesting for smaller sized Vertical/Short Take Off and Landing (**V/STOL**) UAVs. Another advantage is that with a quadrotor the thrust is solely used to compensate for the weight and not to counteract the torque, because the four rotors eliminate the gyroscopic effect, so thrust is completely used to carry the payload.

For micro UAVs, four rotors results in a very small rotor diameter, which penalises efficiency and increases energy consumption to get similar lift. Also the size and weight of a quadrotor with similar payload is higher than a normal helicopter. The simplifications in building and control which is offered by the concept, makes it still a very favourable design for UAVs.

1-2 Autonomous flight

Depending on the controlled variable the perception of the control of a quadrotor by a pilot is perceived differently, see figure 1-2. Autonomous control is the most easiest for the pilot, where the only variables to be set by the human controller are the desired positions (x_d, y_d, z_d) of the quadrotor, the rest of the control will be done automatically.

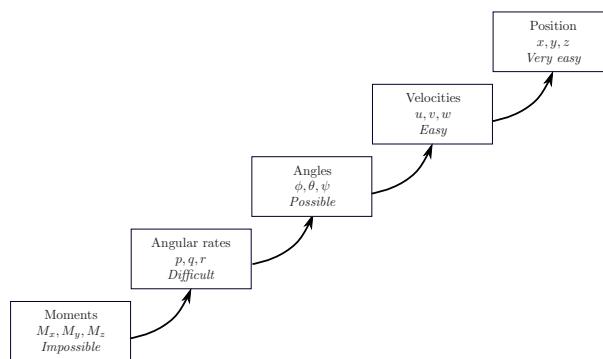


Figure 1-2: Steps in control of a quadrotor and the perception of the control task by a human

One step further is complete autonomous flight, where the pilot can set desired begin and endpoint, and the quadrotor will automatically move from begin to end, finding the best way around obstacles.

1-3 Problem statement

Most quadrotor systems are equipped with 3-axis gyroscopes to measure the rotational rates p, q, r and with 2 or 3-axis accelerometers to get an estimate of the angles ϕ and θ . Some quadrotor system are equipped with 2 or 3-axis magnetometer to measure the heading ψ . This combination of sensors in an Inertial Measurement Unit ([IMU](#)) allow the quadrotor to be stabilised in attitude, but imperfections in the blades and motors, differing air flows over certain components on the craft and outside forces such as wind can cause the craft to move without any noticeable change in pitch or roll. These motions cannot be detected by the [IMU](#) and so it cannot correct for this gentle drift [Roberts et al., 2007, Fowers et al., 2007]. Various errors in sensors, such as sensor misalignment and sensor biases, can cause slight a difference between measured state and true state, which also result in a drift.

In order to do autonomous flight an accurate estimation of the current position of the quadrotor is required to close the position feedback loop. Double integration of the accelerometers measurements of the IMU is not possible because not only the body acceleration is measured, but also the gravitational vector, the accelerations due to rotation and an bias.

In outdoor situations a Global Positioning System (**GPS**) can be used to measure velocity and position with respect to the earth. However the use of GPS is not possible in indoor environments. Therefore an alternative measurement of position is required indoor.

1-4 Literature overview

Additionally to the **IMU** sensors to estimate *attitude*, in literature several methods can be found that are used for estimation of the *position* of a quadrotor.

1-4-1 Vision

In [Tournier et al., 2006] the six Degrees Of Freedom (**DoF**) state estimation of a quadrotor vehicle based on moire patterns is demonstrated. From a single camera a moire pattern is observed and the x, y, z and ψ information is extracted and combining this information with the attitude information from the **IMU** allowed a quadrotor to fly an autonomous hover for an limited time span. The controlled movement is only possible within a space given the target remains in view. Further studies are necessary to test usability in larger spaces or for landing pads.

Camera's are also used in [Altuğ et al., 2002, Altuğ, 2003] and a successive paper [Altuğ et al., 2005]. Two six **DoF** state estimation methods are proposed: initially a single camera was used on the ground to estimate the state of the quadrotor using detection of blobs. However this did not allow full autonomous flight. Therefore next a combination of two cameras is used, one on-board the quadrotor and an other on the ground. This helped to decrease the errors in estimating the tilt angles. Initial experiments on a *tethered* system have shown that the vision-based control is effective in controlling the helicopter.

At the University of Cambridge one successful quadrotor project using a single camera for state estimation is elaborated [Kemp, 2006]. In this thesis, a multiple hypothesis filter which employs a dynamic programming algorithm is used. This filter is responsive enough, while yielding stable and accurate determines the position x, y and z and attitude $\phi \theta \psi$. Onboard sensors measure the angular rates p, q, r . The attitude estimates and onboard sensors are used in simple controller to reliably control the orientation of the helicopter. This controller is in turn *nested* inside a controller which chooses the orientation required to reach or hold at a desired position. This combined controller is able to reliably fly the helicopter and results from a variety of test flights are presented.

In the OS4 project at Ecole Polytechnique Fédérale de Lausanne (**EPFL**) a CCD camera is used to determine horizontal positions x, y and the yaw angle ψ [Bouabdallah, 2007, Epiney et al., 2006]. Pattern recognition with a Canny edge detector and the Douglas-Peucker algorithms are implemented. A red A4 paper can be robustly detected, and horizontal position and yaw angle can be determined.

At Aalborg University [Petersen et al., 2008] a Vicon Motion Tracking System is used as sensors to close the control loops of an Draganfly quadrotor. Tracking of the attitude and position of the quadrotor is possible but autonomous control still lacks performance and is not yet possible.

In the STARMAC II project of Stanford University [Waslander et al., 2005] for indoor flights, an overhead USB camera is used, with blob tracking software, to provide position sensing in place of **GPS**. The camera system gives 1-2 cm accuracy at a rate of 10 Hz.

1-4-2 Vertical position sensors

In several quadrotors, such as the Microdrones MD4-200 and in the Mikrokopter project a pressure sensor is built in. Pressure sensors can be used to estimate the height, but are not preferable for indoor environments because their measurement can be disturbed by airflows such as the rotor flow or air condition systems. Furthermore their accuracy is not very high (in the order of 10 cm).

Most quadrotor systems, such as the OS4 [Bouabdallah, 2007] and the STARMAC [Hoffmann et al., 2007] use ultrasonic range finders for direct height measurement. Ultrasonic sensors have a wide measurement range (0.17 - 5.0 m) and a 20 Hz update rate. Test at the MAVLab and in [Hoffmann et al., 2007] show that these sensors can interfere with the rotor subsystem of a quadrotor. In indoor environments the sound of the rotor subsystem is reflected by the walls and cause an incorrect measurement. Another problem with ultrasonic range finders is that they have a wide beam angle so they do not measure the exact distance to a wall. Infrared sensors have a small beam width, and give an accurate measurement of the distances also when measuring under an angle. In the OS4 project [Bouabdallah, 2007] IR-sensors were tried for height control, but not used because they inhibit a nonlinearity in the measurement, however with correct calibration this problem can be solved.

1-4-3 Horizontal position sensors

At the EPFL another experiment is done using four infrared sensors, two in x and two in y direction. This allows autonomous hover of the quadrotor [Roberts et al., 2007], but the performance of this set up is not optimal as the velocity of the quadrotor is derived by differentiating the noisy signal from the infrared sensors and the onboard sensors are not integrated.

1-4-4 Optic flow sensors

Recently at Brigham University [Brigham Young University, 2008] optic flow sensors are implemented onboard a quadrotor UAV. Integration of the velocity measurements of the optic flow sensors give a position and allow the quadrotor to hover but give no information with respect to the walls in the indoor environment.

1-5 Objective

In order to perform indoor navigation and control of a quadrotor UAV it is proposed to integrate the measurements of the three gyroscopes, three accelerometers and three magnetometers of an IMU with the distance measurements of infrared sensors. This tightly coupled sensor integration allows estimation of all the states of the system and the biases of both the gyroscopes and accelerometers. This indoor navigation and control should be tested on a real quadrotor UAV.

To reach this objective:

1. First the *dynamical model* of the quadrotor is required to test the sensor integration in simulation.
2. As the dynamic model of the quadrotor is open loop unstable, a *controller design* is required to do flight test in simulation
3. Next step is the development of the state estimation algorithm to be able to do *indoor navigation* with the selected sensors and test this with the simulation.

4. A quadrotor UAV needs to be built with the selected sensors, able to perform flight tests and send the measurements to a desktop computer for processing. These measurements can then be used to test the state estimation algorithm.
5. An extra challenge is to use as much inexpensive parts as possible.

1-6 Outline of the report

In this chapter an introduction is given to this project and the quadrotor platform. Chapter 2 contains an overview of other main quadrotor platforms.

In the next part the dynamic model of the quadrotor is described, which is used to test the indoor navigation and control in simulation. Because the quadrotor is open-loop unstable system, a controller is required to do flight tests in simulation. This controller is designed in chapter 4.

In part III first the selected sensors that are used in the sensor integration are modelled. Chapter 6 explains the state estimation method used to integrate the sensors of the IMU and the IR sensors and discusses the results of this state estimation in simulation.

To perform flight tests with actual sensors, a quadrotor platform is designed. Part IV starts with the development of this platform in chapter 7. Because without any control feedback the quadrotor cannot fly, onboard filtering and control is required, which is described in chapter 8. Next in chapter 9 the real data from the quadrotor UAV is used to test the state estimation method developed.

Finally, a conclusions and discussion of the work and recommendations for future work are given.

Chapter 2

Overview of quadrotor platforms

The quadrotor layout is a popular design for UAVs and a number of research projects already tackled the design, navigation and control of this platform, for indoor and outdoor use. This proves that this layout is a successful one and it is important to learn from these previous projects, therefore this chapter will discuss previous research in the design of quadrotors.

This chapter starts with the early history of manned quadrotors and its modern day successors. Next the main unmanned quadrotor platforms developed at universities are evaluated. Finally the commercially available quadrotors and the open-source projects will be summarised. From this also a conclusion for research at the TU Delft will be drawn.

2-1 Early history

In 1907 the Breguet Brothers built the first helicopter, which was a quadrotor. Their *Gyroplane No. 1* consisted of four long girders arranged in the form of a horizontal cross. The flights of the Gyroplane No. 1 are considered to be the first manned flight of a helicopter, but not a free or untethered flight. The machine never flew completely freely because it lacked stability and proper means of control.

Later, in 1922, Georges de Bothezat built one of the largest helicopters of the time also a quadrotor. His *Flying Octopus* flew successfully many times, at low altitudes and slow forward speeds. However, because of insufficient performance, high financial costs, and the increasing military interest in autogiros at the time, the project was cancelled.

In the same year Etienne Oemichen built his quadrotor, the *Oemichen No.2*. It was equipped with four 2-blade paddle-shaped rotors of which the angles could be varied by warping. Additional rotors were necessary for stabilisation and lateral movement, due to this the Oemichen No.2 exhibited, for its time, a considerable degree of stability and controllability, and in all made more than a thousand test flights during the middle 1920s.

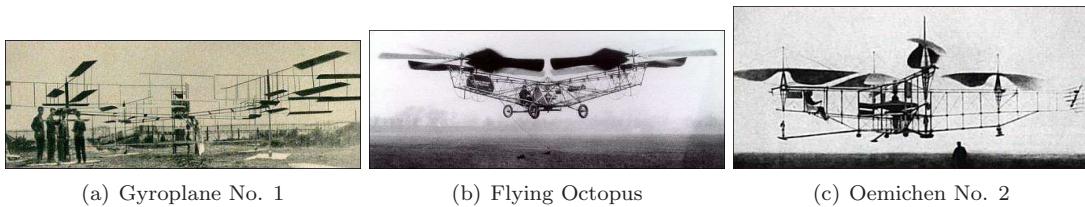


Figure 2-1: Early quadrotors

2-2 Modern successors

After these first attempts, designers focused more on the single rotor helicopter layout, because of the stability problems with the quadrotor layout. Interest in the quadrotor layout for manned flight returned when knowledge on control strategies increased. Also the quadrotor layout with tilttable rotors became an option to provide more flexibility and faster forward flight.

Convertawings Model A revived the concept tried out in 1922 by Oemichen and de Bothezat. A first prototype flew in 1956. Despite successful testing and development, military support for the quadrotor ceased after cutbacks in defence spending. However, the design, particularly its control system, was a precursor of current experimental vertical-rising aircraft designs that incorporate tandem wings or a square configuration of four fans, ducts, or jets.

In order to develop a passenger aircraft using four tilting propellers, the *Curtiss X-19* was designed. The first hover flight was in November 1963, but the aircraft never completed transition from hover to flight because poor mechanical control system characteristics severely penalised low-speed operation. Lack of stability augmentation systems and upsets due to random flow In Ground Effect (**IGE**) further increased pilot workload in hover.

From the start, in the *Bell X-22A* variable stability and control features were incorporated for flight research on **V/STOL** handling qualities. Hover operation Out of Ground Effect (**OGE**) in no wind was rated excellent but wind effects were quite noticeable, because of the large side forces generated by the ducts. High-speed performance was limited by relatively high drag associated with the four large ducts. The second prototype has generated significant **V/STOL** handling qualities and is currently on flight status.

In 24 September 2005 the team of Bell Helicopter and Boeing signed a contract by the U.S. Army to perform conceptual design and analysis of the *Quad Tilt Rotor* (QTR) aircraft, to provide for the critical need for long range, high speed, heavy lift without access to runways. Bell Boeing's QTR is an evolutionary application of its tiltrotor technology utilised in the V-22 Osprey. The QTR is a tandem-wing, four-propotor aircraft with a large cargo fuselage and a rear-loading ramp.

The Moller company has been developing several flying vehicles with the final goal of designing a aircraft to replace the car as the common way of transport which not able to land and hover in a small areas and to use the sky and fly over the crowded highways. Their last design is the *M400 Skycar* with four ducted rotors. This **V/STOL** aircraft with a capacity of four passengers, a cruise speed of 275 mph and its maximum range of 750 miles it has the features to be able to reach their goal. However, currently they have still some problems with validation, but these are expected to be solved next year.

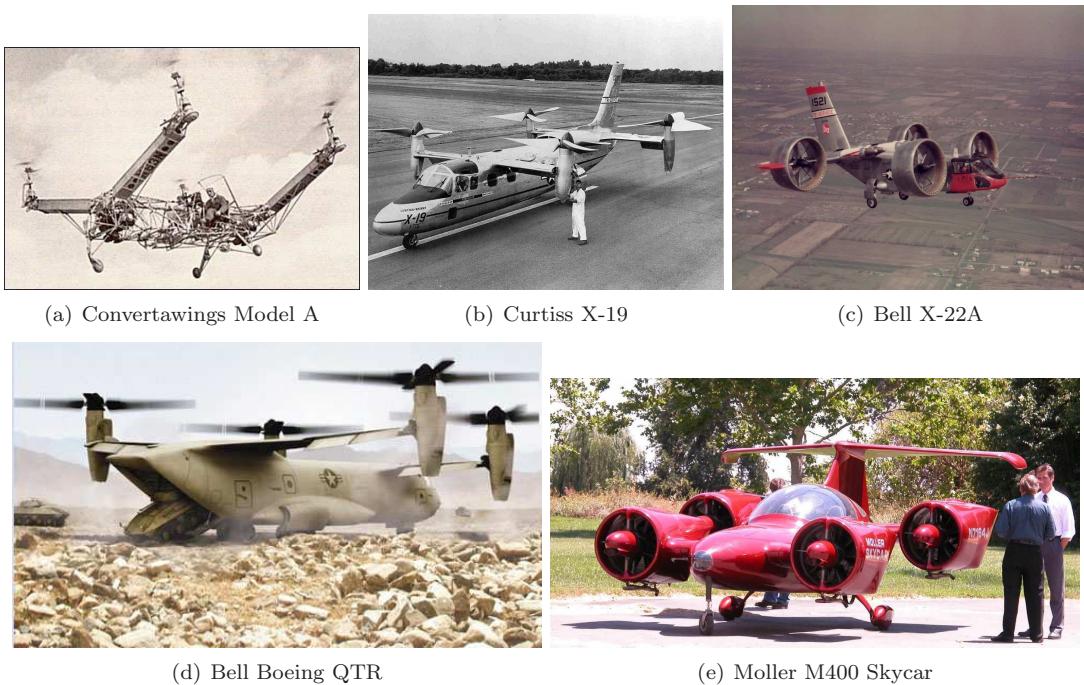


Figure 2-2: Modern manned quadrotors

2-3 Main research projects

In time the knowledge of control theory improved together with the introduction of better computing technology and sensors, which allowed the development of unmanned quadrotors. Several research laboratories and universities started projects with quadrotors, but the development of full autonomous flight in all environments is still a challenge.

Next subsections describe the main three research projects, done at Stanford University, EPFL and Australia National University (ANU). Those three have developed a complete testbed for quadrotor research. At other research laboratories and universities also studies are done with the quadrotor design, but these are build from commercial off-the-shelf quadrotors or not as far developed as these projects. Some of the other platforms are mentioned in the following chapters if they are interesting objects to study for state estimation and control, because it is vital to learn from others and build on already available knowledge.

2-3-1 Stanford University

At Stanford University in 1999 the *Mesicopter* project started [Kroo and Prinz, 2001a, Kroo and Prinz, 2001b] and ended in 2001. It studied the feasibility of a centimetre scaled quadrotor to use in huge numbers to investigate large areas or even planets. For this project an extensive study was done into the aerodynamics effects of a quadrotor. Finally a prototype was built of which the rotors could turn, but no lift off was achieved.

Next the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (**STARMAC**) was set up. This is a multi vehicle test bed used to demonstrate new concepts in multi agent control on a real-world platform [Hoffmann et al., 2004, Hoffmann et al., 2007]. STARMAC consists of

up to eight quadrotor vehicles that autonomously track a given waypoint trajectory.

The project is divided in three phases. In the first phase the initial vehicle and testbed design for proof-of-concept flights is done. This phase was completed in the fall of 2004 with the development of *STARMAC I*. In phase two the complete vehicle and testbed is redesigned to create a full functioning testbed, this phase is now nearing completion with the *STARMAC II*. In the final phase the multi agent control will be demonstrated.

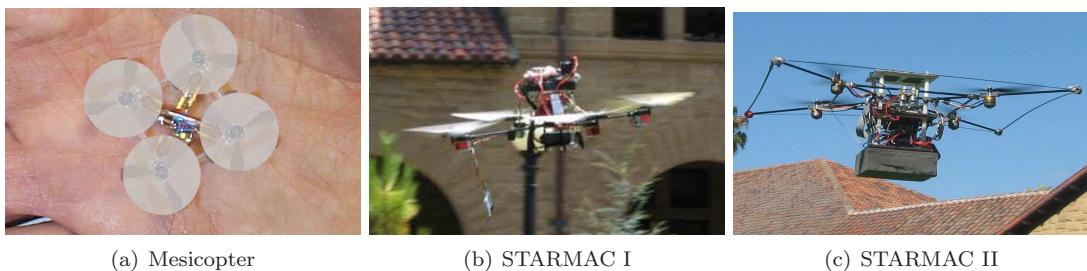


Figure 2-3: Stanford University quadrotor projects

STARMAC I

The base vehicle for STARMAC I was originally an off-the-shelf DraganFlyer III, which has a total of 1 kg of thrust and can sustain hover for about ten minutes at full throttle. The onboard electronics were replaced with own printed circuit board so as to have complete control over motor commands, power supply and sensor measurements.

An upgrade to lithium-polymer batteries has increased both payload and flight duration and has greatly enhanced the abilities of the system. For attitude measurement the Microstrain 3DM-G motion sensor was used, with 3-axis gyro, accelerometer and magnetometer information. For position and velocity measurement, a Trimble Lassen Low Power GPS receiver was used. To improve altitude information the downward-pointing SONic Detection And Ranging (**SODAR**) the Devantech SRFO8 was used, especially for critical tasks such as take off and landing. All of the onboard sensing is coordinated through two Microchip 40 MHz microcontrollers programmed in C.

Position estimation uses both GPS position and velocity measurements, as well as attitude information in a Extended Kalman Filter (**EKF**) to update the position and velocity estimates at 10 Hz. The GPS data is used to correct for integration bias of the small angle approximated accelerations derived from the attitude information.

Attitude stabilisation is performed on board at 50 Hz and relayed to a central base station on the ground, communicating via a Bluetooth Class II device with a range of over 150 ft. Designed as a serial cable replacement and therefore operates at a maximum bandwidth of 115.2 kbps. This bandwidth of 115.2 kbps is divided among all (of four) flyers. The base station on the ground performs Differential GPS (**DGPS**) and waypoint tracking tasks for all four flyers, and sends commanded attitude values to the flyers for position control. Manual flight and waypoint control is performed via the ground station laptop using Labview.

For control of the altitude a sliding mode control is applied and the attitude loop was designed using Linear-Quadratic Regulator (**LQR**) techniques. Angular deviations were penalised more than rate deviations. Also, the pitch and roll loops were penalised identically, with a reduced

penalty on the yaw deviations. On the STARMAC I platform also more advanced controller types are tested, such as integral sliding mode and reinforcement learning [Waslander et al., 2005].

STARMAC II

The STARMAC II testbed is a improvement of the first version at several points:

1. *Thrust capabilities*: Brushless motors and more rigid plastic propellers were combined to double the efficiency and increase the total available thrust up to a total of 4 kg of thrust.
2. *On-board computation resources*: With the added lift capabilities, it was possible to significantly increase onboard computing power. Computation and control are managed at two separate levels. The low level control, which performs real-time control loop execution and outputs PWM motor commands, occurs on a microcontroller board based on the Atmega 128 processor. The high level planning, estimation and control occurs on a lightweight Crossbow Stargate 1.0 single board computer (SBC) running embedded Linux. Alternatively an Advanced Digital Logic ADL855 PC104 running Kubuntu Linux can also be implemented, providing more computational power, at the cost of additional weight and hence shortened flight times.
3. *Communication reliability and bandwidth*: The testbed communication channel was changed from Bluetooth to a WiFi network, enabling much greater bandwidth and improved communication channel management through a wireless router at the basestation. This has shown significant improvements in communication robustness between base station and vehicle as compared with the Bluetooth capabilities of STARMAC I.
4. *Position measurement accuracy*: Three-dimensional position and velocity measurements are now done using a carrier phase differential Novatel Superstar II [GPS](#) unit with an resulting position accuracy is 1-2cm. For indoor flights, an overhead USB camera is used, with blob tracking software, to provide position sensing in place of [GPS](#). The camera system gives 1-2 cm accuracy at a rate of 10 Hz, and provides a drop-in replacement for [GPS](#) input to the [EKF](#).

Instead of the [LQR](#) used in the *STARMAC I* in the *STARMAC II* Proportional-Integral-Derivative ([PID](#)) control was implemented for attitude, altitude and position control.

Not only the hardware was improved but also the mathematical model of the quadrotor. Three extra effects where quantified that relate to motion of the vehicle relative to the free stream.

2-3-2 Ecole Polytechnique Fédérale de Lausanne (EPFL)

At the [EPFL](#) also an extensive quadrotor project is set up by Samir Bouabdallah. His PhD. project focused on simultaneous consideration of the design and control problems of a quadrotor UAV [Bouabdallah, 2007, Bouabdallah et al., 2005]

Two platforms were developed during this project. The first one is a quadrotorlike test-bench with off-board data processing and power supply. It was used to safely and easily test control strategies. The second one, the Omnidirectional Stationary Flying OUtstretched Robot ([OS4](#)), is a highly integrated quadrotor with on-board data processing and power supply. It has all the necessary sensors for autonomous operation.



Figure 2-4: The EPFL OS4 quadrotor

The hardware of the **OS4** consists of a self designed frame with the brushless sensorless outrunner DC motors. Onboard is a miniature computer module, based on Geode 1200 processor running at 266MHz with 128M of RAM and flash memory was developed. The controller includes a microcontroller for interfacing Bluetooth with the computer module. The same chip is used to decode the Pulse Position Modulation (PPM) signal picked up from a 5 channels Remote Control (**RC**) receiver. For attitude sensing the Microstrain 3DM-GX **IMU** is used. Also five range sensors, Devantech SRF10, are used. One for altitude and four for obstacle avoidance. Additionally the position is determined using an on-board down-looking CCD camera and a simple pattern on the ground. The camera provides a motion-blur free image of 320x240 at up to 25 fps. The algorithm detects the pattern, estimates the pose and provides the camera position (x,y) and yaw angle (ψ), also described in [Roberts et al., 2007].

Several control theories were tested, the first one, based on Lyapunov theory, was applied for attitude control [Bouabdallah et al., 2004a]. The second and the third controllers are based on **PID** and **LQR** techniques [Bouabdallah et al., 2004b, Bouabdallah and Siegwart, 2006]. These were compared for attitude control. The fourth and the fifth approaches use backstepping and sliding-mode concepts [Bouabdallah and Siegwart, 2005]. They are applied to control attitude. Finally, backstepping is augmented with integral action and proposed as a single tool to design attitude, altitude and position controllers. This approach is validated through various flight experiments conducted on the OS4.

A dynamic model to simulate the quadrotor evolved from a simple set of equations, valid only for hovering, to a complex mathematical model with more realistic aerodynamic coefficients and sensor and actuator models.

2-3-3 Australia National University (ANU)

The Australia National University developed in 2002 the *MARK I* prototype which is a quadrotor with an innovative design using inverted teetering rotors [Pounds et al., 2002]. The two key challenges facing the development of the Australian National Universitrys X-4 Flyer were thrust generation and dynamic stability. The successor *MARK II* X4 Flyer has weight of 2 kg with a length

of 70 cm an 11 inch diameter rotors [Ng et al., 2004, Pounds et al., 2006, Pounds et al., 2004]. The electronics are substantially the same as the Mark I, although a lighter sensor unit has replaced the original Crossbow IMU. The craft is stabilised by an onboard embedded HC12 controller. The controller reads attitude from a CSIRO Eimu IMU that provides angular rate and acceleration measurements and angular position estimates at 50 Hz. The Eimu is a full six-axis IMU with magnetometer.

The rotors are designed to lift the flyer with an additional 30% control margin and were designed and fabricated at the ANU. The geometry is designed so that the rotor tips flex to the optimal operating angle under load. The ANUX2 airfoil used is a custom section made specially for the rotors. The rotors are driven by Jeti Phasor 30-3 three-phase brushless motors. They offer high torque performance that allows for direct drive of the rotors, eliminating the need for gearing. Custom motor control boards commutate the motors. These were developed by the CSIRO Queensland Centre for Advanced Technology ICT group. The boards are based around the Freescale HC12D60A microprocessor and Toshiba TB9060 brushless motor speed control chip.

Power is provided by 24 Li-Poly 2000 mAh high discharge cells with a nominal voltage of 3.7 V, ranging from 4.2 V fully charged and dropping to 3 V at depletion. Each cell can deliver up to 20 A. The batteries are connected to a power bus of six parallel sets of four cells in series combining the batteries to 14.8 V nominal voltage and 120 A of current draw per motor. This gives the flyer an expected flight time of 11 minutes at hover speed.

Communication is done using a long-range Bluetooth serial module connected to a laptop base station running Linux, with a range of up to 100 m. Telemetry from the flyer is logged by the base station and displayed on-screen. The user can issue commands via the laptop using the keyboard and a JR-X3810 radio handset. The radio handset can also trigger a safety kill switch on the X-4, independently of the Bluetooth communications channel, using an onboard radio receiver. In an emergency the kill switch can stop the rotors instantly by disabling the motor control boards, even if data communications is lost.

Unlike the *MARK I*, the *MARK II* incorporates simple onboard proportional-integral-derivative control. The previous iteration used a slow, off-board control system connected to the flyer by a tether. It is anticipated that the convenient aerodynamics of the X-4 had made sophisticated control unnecessary. In conjunction with onboard power, this allowed the flyer to be entirely self-contained.

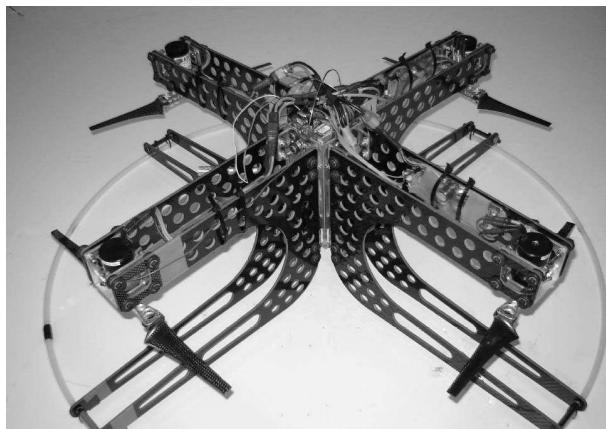


Figure 2-5: The ANU MARK II X4 Flyer

2-4 Commercially available

Microdrones MD4-200

<http://www.microdrones.com/md4-200.html>

The Microdrones MD4-200 is the most advanced quadrotor available on the customer market. Interesting fact is that the Dutch police is also owner of one of those quadrotors for future use in crowd control. It uses accelerometers, gyroscopes, magnetometer, airpressure, humidity and temperature sensors, combined in a Attitude and Heading Reference System (**AHRS**). The optional **GPS** provides position hold and autonomous waypoint navigation. It has an onboard flight recorder (microSD card) and a downlink decoder to provide all important data at the base station (battery state, altitude, attitude, position, flighttime). Security features prevent it from crashing, featuring autonomous landing on low battery or missing radio signal. Depending on payload, temperature and wind the vehicle achieves up to 20 minutes of flight time.

Draganfly

<http://www.rctoys.com/>

The Draganfly quadrotor has been developed in successive steps into its current version *V Ti*. This version has a self leveling system with 4 infrared sensors and a three axis gyroscopes. It can be equipped with a wireless video camera system, to use it as a stable aerial video system.

Silverlit X-UFO

<http://www.silverlit-flyingclub.com/xufo-body.htm>

This winner of the Toy Innovation award is the cheapest quadrotor platform available, and therefore has only limited performance using mechanical gyroscopes and a flighttime of only 5 minutes. The frame is from carbon fiber and has an foam protection ring. The frame and engines of this quadrotor are used as a basis for this project.



(a) Microdrones MD4-200



(b) Draganflyer V Ti



(c) Silverlit X-UFO

Figure 2-6: Some commercially available quadrotors

X-UFO Brushless

<http://www.xufo-shop.de>

The X-UFO Brushless is a further development of the Silverlit X-UFO using brushless motors. It is smaller and quieter than the open source project *Mikrokopter*. It can be bought in parts, and the framework can be designed separately. The X-UFO contains no expansion options, such as GPS module or digital cameras.

Quattrokopter

<http://www.lcc-shop.de>

The Quattrokopter is based on the open source project Mikrokopter. Either a complete Quattrokopter can be bought or only the separate X-frame. The flight is controlled with three gyroscopes, one accelerometer, a barometer height sensor.

BigQuadro

<http://www.tt-tronix.de>

The BigQuadro is a project using a special control board, the QuadroControl II, with onboard control 3 gyroscopic sensors for the stabilization of the flight. There are two different versions, one with brushed motors and one with brushless motors for increased payload. It is not possible to buy a complete quadrotor, but the manual gives clear instructions how to construct your own quadrotor.

2-5 Open source projects

Mikrokopter

<http://www.mikrokopter.de>

This is the most developed open source project for a quadrotor, where the electronics have been completely designed, including software, and all knowledge is freely available. It uses a combination of a 3-axis gyroscope and accelerometers for control.

It is possible to equip the quadrotor with brushless motors, for which a special controller Printed Circuit Board (**PCB**) is developed. The Mikrokopter is optional expandable with barometer (MPX4115) for a height control / limitation, an electronic compass, a **GPS** module (uBlox), a digital camera and wireless radio transmission.

UAVP

<http://www.uavp.ch>

The UAVP (Universal Aerial Video Platform) is a remotely controlled, and also optional autonomous flying quadrotor. If not controlled by the pilot, it retains height and direction, and using GPS even position. This allows this platform to be used for video flights. It uses brushless electric engines, has a variable diameter of 360 to 1000mm and a weight of less than 1,000 grams. It can be expanded with magnetic sensors, an acceleration sensor for drop, a camera and a GPS module.

The new UAVP Next Generation (<http://uavp.ch/moin>) is a full reimplementation of the UAVP using a newer processor with more computing power and more memory capacity. This project is still in development phase.

2-6 Conclusions

This chapter gives a good overview of the various manned and unmanned quadrotor platforms developed. It is clear that the quadrotor platform is a multi purpose platform which requires integrated development in electronics, informatics, aerodynamics, state estimation and control. Already quite some knowledge is available, but still some great challenges are there to be found, such as complete autonomous indoor flight. Therefore it is an ideal platform for research combining knowledge on the various faculties at **TU Delft**.

Part II

Quadrotor modelling and control design

Chapter 3

Quadrotor modelling

In order to test the indoor navigation and control in simulation a dynamical model of the quadrotor is required. The model of dynamic equations describing the attitude and position of the quadrotor are basically those of a rotating rigid body with six degrees of freedom and four inputs.

The dynamic model of the quadrotor system has already thoroughly been investigated by [Fay, 2001, Hoffmann et al., 2007, Pounds et al., 2006, McKerrow, 2004] and others. The model described in [Bouabdallah, 2007] is very complete and validated with real in flight data. Therefore it is reliable enough to be used as basis for the simulation model. However it does not include some additional effects that are investigated in other studies. In this chapter all those aspects will be combined. Recently in [Martinez, 2007] the Draganfly quadrotor is modeled in more detail including test in wind-tunnels, this model is not yet validated with real inflight data.

In this chapter first the reference frames and the kinematic relations between the reference frames are defined. After stating the assumptions used in this model, the forces and moments acting on the rotors are modelled. All forces and moments are then combined in the full dynamic equations of motion. In the final section the implementation in the Matlab/Simulink model will be elaborated and a conclusion given.

3-1 Reference frames

Modelling starts with defining the reference frames, these are according to [Mulder, 2006] and depicted in figure 3-1:

- The earth-fixed inertial reference frame ($0_e X_e Y_e Z_e$) is a right-handed orthogonal axis-system with the origin at the quadrotor's centre of gravity at the beginning of the considered motion. This reference frame is fixed to the earth and is considered as the inertial frame of reference under simplifying conditions.
- The body-fixed reference frame ($0_b X_b Y_b Z_b$) is a right-handed orthogonal axis-system with the origin at the quadrotor's centre of gravity. The reference frame remains fixed to the quadrotor even in perturbed motion.

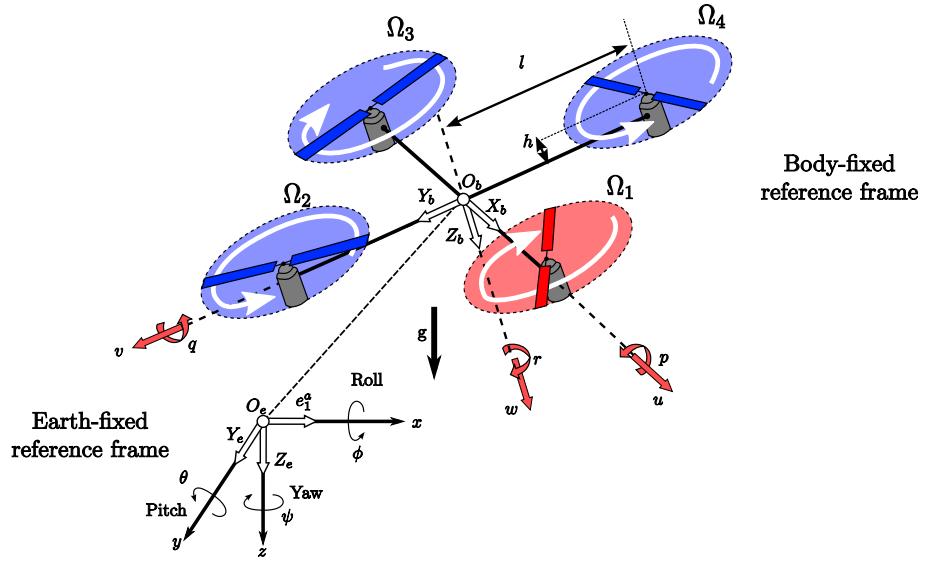


Figure 3-1: Earth fixed reference frame and body fixed reference frame

The *absolute position* of the quadrotor is described by the three coordinates (x,y,z) of the centre of mass with respect to the earth reference frame.

Its *absolute attitude* is described by the three Euler's angles (ψ, θ, ϕ) . These three angles are respectively called yaw angle ($-\pi \leq \psi < \pi$), pitch angle ($-\frac{\pi}{2} < \theta < \frac{\pi}{2}$) and roll angle ($-\frac{\pi}{2} < \phi < \frac{\pi}{2}$). Because for now the quadrotor will not perform aerobatics, it is possible to use the Euler angles in the boundaries given.

3-2 Kinematic relations

The kinematic relations relate the movements and rotations in the earth-fixed inertial reference to the body-fixed reference frame.

The derivatives with respect to time of the angles (ψ, θ, ϕ) can be expressed in the form:

$$[\dot{\psi} \quad \dot{\theta} \quad \dot{\phi}]^T = N(\psi, \theta, \phi)\omega \quad (3-1)$$

in which $\omega = [p \quad q \quad r]^T$ are the angular velocities with respect to the body reference frame and $N(\psi, \theta, \phi)$ is the 3x3 matrix given by:

$$N(\psi, \theta, \phi) = \begin{bmatrix} 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \\ 0 & \cos \phi & -\sin \phi \\ 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \end{bmatrix} \quad (3-2)$$

This matrix depends only on (ψ, θ, ϕ) and is invertible if the boundaries on (ψ, θ, ϕ) for non-singularity hold.

Similarly, the derivative with respect to time of the position (x, y, z) is given by

$$[\dot{x} \quad \dot{y} \quad \dot{z}]^T = V_0 \quad (3-3)$$

where $V_0 = [u_0 \ v_0 \ w_0]^T$ is the absolute velocity of the quadrotor with respect to an earth-fixed inertial reference frame. Let $V = [u \ v \ w]^T$ be the absolute velocity of the quadrotor expressed in a body-fixed reference frame. V and V_0 are related by

$$V_0 = R(\psi, \theta, \phi)V \quad (3-4)$$

where $R(\psi, \theta, \phi)$ is the rotation matrix given by:

$$R(\psi, \theta, \phi) = \begin{bmatrix} \cos \theta \cos \psi & (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) & (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \cos \theta \sin \psi & (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) & (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (3-5)$$

3-3 Assumptions

It is not possible to create a model that is completely conform reality, some assumptions need to be made. In this model the following is assumed:

- The quadrotor structure is rigid
- The quadrotor structure is symmetrical
- The propellers are rigid
- The cross products of the inertia matrix can be neglected
- The ground effect is neglected

3-4 Aerodynamic forces and moment of a rotor

Using the blade element theory [Fay, 2001, Prouty, 2002] it is possible to calculate the forces acting parallel and perpendicular to the rotor shaft as well as the moments acting about the rotor shaft and hub. Assuming the rotors are rigid, the force parallel to the rotor shaft is defined as the thrust T of the rotor and the forces perpendicular to the rotor shaft are defined as the hub forces H . Also two moments work on the rotor; the drag moment M_Q and the rolling moment M_R . It can be assumed that the lift acting on the blade is about an order of magnitude higher than the drag. In the following subsections, the thrust, hub forces, and rotor moments will be defined for each rotor. In figure 3-2 these forces and moments are clearly visible.

The explanation of the various symbols used in the equations below can be found in the list of symbols.

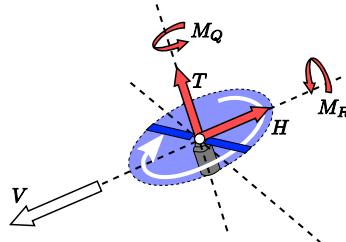


Figure 3-2: Forces and moments acting on a rotor

Thrust force

The thrust force is the resultant of forces acting on all blade elements perpendicular to the rotor shaft.

$$T = C_T \rho A (\Omega R)^2 \quad (3-6)$$

$$\frac{C_T}{\sigma a} = \left(\frac{1}{6} + \frac{\mu^2}{4} \right) \theta_0 - (1 + \mu^2) \frac{\theta_{tw}}{8} - \frac{\lambda}{4} \quad (3-7)$$

Hub force

The hub force is the resultant of forces acting on all blade elements in the horizontal plane. Notice that the hub force is zero if the velocity is zero.

$$H = C_H \rho A (\Omega R)^2 \quad (3-8)$$

$$\frac{C_H}{\sigma a} = \frac{\mu \bar{C}_d}{4a} + \frac{1}{4} \lambda \mu \left(\theta_0 - \frac{1}{2} \theta_{tw} \right) \quad (3-9)$$

Drag moment

The drag moment is the resultant moment of all horizontal forces acting about the centre of the rotor. This torque determines the power required for the motor to keep the rotor spinning, and thus providing a *drag* moment.

$$M_Q = C_{M_Q} \rho A (\Omega R)^2 R \quad (3-10)$$

$$\frac{C_{M_Q}}{\sigma a} = \frac{1}{8a} (1 + \mu^2) \bar{C}_d + \lambda \left(\frac{\theta_0}{6} - \frac{\theta_{tw}}{8} - \frac{\lambda}{4} \right) \quad (3-11)$$

Rolling moment

Because the blades move horizontally through the air, the advancing blade will produce more lift than the retracting blade, resulting in an overall rolling moment being produced on the rotor.

$$M_R = C_{M_R} \rho A (\Omega R)^2 R \quad (3-12)$$

$$\frac{C_{M_R}}{\sigma a} = \mu \left(\frac{\theta_0}{6} - \frac{\theta_{tw}}{8} - \frac{\lambda}{8} \right) \quad (3-13)$$

Note that in [Fay, 2001] this equation is negative, because an other reference system is used.

3-5 Dynamic equations

It is assumed that the cross products of the inertia matrix J can be neglected, due to the layout of the quadrotor.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3-14)$$

Using the general equations of motions, found in [Mulder, 2006]:

$$\begin{aligned}
 F_x &= -W \sin \theta + X &= m(\dot{u} + qw - rv) \\
 F_y &= W \cos \theta \cos \psi + Y &= m(\dot{v} + ru - pw) \\
 F_z &= W \cos \theta \sin \psi + Z &= m(\dot{w} + pv - qu) \\
 M_x &= L &= I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\
 M_y &= M &= I_{yy}\dot{q} + (I_{xx} - I_{zz})rp \\
 M_z &= N &= I_{zz}\dot{r} + (I_{yy} - I_{xx})pq
 \end{aligned} \tag{3-15}$$

Where F_x, F_y, F_z are the external forces in the body-fixed frame, and M_x, M_y, M_z are the external moments in the body fixed frame.

In the next subsections the external forces and torques are summarised. Note that the quadrotor rotor plane is above the centre of gravity so the hub forces will also create a moment.

3-5-1 External forces in body fixed frame

Forces along u axis

$$\begin{aligned}
 \text{Hub forces} &= -\sum_{i=1}^4 H_{u_i} \\
 \text{Friction} &= -\frac{1}{2} C_u A_u \rho u |u|
 \end{aligned}$$

Forces along v axis

$$\begin{aligned}
 \text{Hub forces} &= -\sum_{i=1}^4 H_{v_i} \\
 \text{Friction} &= -\frac{1}{2} C_v A_v \rho v |v|
 \end{aligned}$$

Forces along w axis

$$\begin{aligned}
 \text{Thrust} &= -\sum_{i=1}^4 T_i \\
 \text{Friction} &= -\frac{1}{2} C_w A_w \rho w |w|
 \end{aligned}$$

3-5-2 External torques in body fixed frame

Roll moments

$$\begin{aligned}
 \text{Propeller gyro effect} &= J_r q \Omega_r \\
 \text{Pitch actuators action} &= l(-T_2 + T_4) \\
 \text{Hub force due to sideways flight} &= -h(\sum_{i=1}^4 H_{v_i}) \\
 \text{Rolling moment due to forward flight} &= (-1)^i \sum_{i=1}^4 M_{R_{u_i}}
 \end{aligned}$$

Pitch moments

$$\begin{aligned}
 \text{Propeller gyro effect} &= -J_r p \Omega_r \\
 \text{Pitch actuators action} &= l(T_1 - T_3) \\
 \text{Hub force due to forward flight} &= h(\sum_{i=1}^4 H_{u_i}) \\
 \text{Pitch moment due to sideways flight} &= (-1)^i \sum_{i=1}^4 M_{R_{v_i}}
 \end{aligned}$$

Yaw moments

| | |
|--|-------------------------------|
| Inertial counter-torque | $J_r \Delta \dot{\Omega}_r$ |
| Counter-torque unbalance | $(-1)^i \sum_{i=1}^4 M_{Q_i}$ |
| Hub force unbalance in forward flight | $l(H_{u_2} - H_{u_4})$ |
| Hub force unbalance in sideways flight | $l(-H_{v_1} + H_{v_3})$ |

The explanation of the symbols can be found in the nomenclature in the front of this report.

3-5-3 Complete dynamic equations of the quadrotor

Combining the above the complete dynamic equations of the quadrotor are obtained:

$$m\dot{u} = -m(g \sin \theta - qw + rv) - \sum_{i=1}^4 H_{u_i} - \frac{1}{2} C_u A_u \rho u |u| \quad (3-16)$$

$$m\dot{v} = m(g \cos \theta \sin \phi - ru + pw) - \sum_{i=1}^4 H_{v_i} - \frac{1}{2} C_v A_v \rho v |v| \quad (3-17)$$

$$m\dot{w} = m(g \cos \theta \cos \phi - pw + qu) - \sum_{i=1}^4 T_i - \frac{1}{2} C_w A_w \rho w |w| \quad (3-18)$$

$$I_{xx}\dot{p} = (I_{yy} - I_{zz})qr + J_r q \Omega_r + l(-T_2 + T_4) - h(\sum_{i=1}^4) H_{v_i} + (-1)^i \sum_{i=1}^4 M_{R_{u_i}} \quad (3-19)$$

$$I_{yy}\dot{q} = (I_{zz} - I_{xx})rp - J_r p \Omega_r + l(T_1 - T_3) + h(\sum_{i=1}^4) H_{u_i} + (-1)^i \sum_{i=1}^4 M_{r_{v_i}} \quad (3-20)$$

$$I_{zz}\dot{r} = (I_{xx} - I_{yy})pq + J_r \dot{\Omega}_r + (-1)^i \sum_{i=1}^4 M_{Q_i} + l(H_{u_2} - H_{u_4}) + l(-H_{v_1} + H_{v_3}) \quad (3-21)$$

3-6 Additional effects

The dynamic equations of section 3-5-3 give the complete model of the quadrotor under the assumptions given in 3-3. However for completeness some additional effects are studied. These effects are less straightforward to model but could be implemented in the model after further experiments.

3-6-1 Ground effect

If an rotor of the quadrotor approaches the ground the thrust increases due to better rotor efficiency, this is called the ground effect [Prouty, 2002]. The airflow from the rotor is impacting with the ground and causing a small build up of air pressure in the region below the rotor disk. The rotor is then "floating" on a cushion of air. This means that less power is required with respect to the Out of Ground Effect (OGE). In Ground Effect(IGE) conditions are usually found within heights about 0.5 to 1.0 times the diameter of the main rotor. The height will vary depending on the type of rotor, the slope and nature of the ground, and any prevailing winds.

In [Bouabdallah, 2007] a basic model of Cheeseman [Cheeseman and Bennett, 1957] is used

to calculate the thrust coefficient **IGE** C_T^{IGE} . Cheeseman stated that at constant power the following relationship can be used:

$$T_{OGE}v_{i_{OGE}} = T_{IGE}v_{i_{IGE}} \quad (3-22)$$

where T_{OGE} and T_{IGE} is the thrust **IGE** and **OGE**, and $v_{i_{OGE}}$ and $v_{i_{IGE}}$ the induced velocity in **IGE** and **OGE**.

Then with the variation of the induced velocity:

$$\delta v_i = \frac{Av_i}{16\pi z^2} = \frac{v_i}{(4z/R_{rad})^2} \quad (3-23)$$

a relation between the thrust **IGE** and **OGE** can be derived, assuming v_i and δv_i are constant over the disk which allows $v_{i_{IGE}} = v_i - \delta v_i$:

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \frac{R_{rad}^2}{16z^2}} \quad (3-24)$$

Consider the inflow ratio in the ground effect as:

$$\lambda_{IGE} = \frac{(v_{i_{OGE}} - \delta v_i - \dot{z})}{\Omega R_{rad}} \quad (3-25)$$

then the thrust coefficient **OGE** in equation 3-7 can be rewritten for the **IGE**:

$$T_{IGE} = C_T^{IGE} \rho A (\Omega R_{rad})^2 \quad (3-26)$$

$$\frac{C_T^{IGE}}{\sigma a} = \frac{C_T^{OGE}}{\sigma a} + \frac{\delta v_i}{4\Omega R_{rad}} \quad (3-27)$$

Note that as in movements the attitude angles change, so the thrust **IGE** are not similar for all four rotors at the same time. In [Bouabdallah, 2007] a simple test is performed for one rotor near ground, where for the above formulas an indication of validity is given. As the ground effect is influenced by several parameters, mentioned above, extensive testing should be performed to properly validate this relationship for the quadrotor platform. For now only the flight regime **OGE** is modelled.

3-6-2 Blade flapping

When the rotors translate horizontally there is a difference in lift between the advancing and retreating blades, which caused the rolling moment as described in 3-4, but in practice it also induces another effect; blade flapping. When using *non-rigid* unhinged stiff blades, blade flapping causes the rotor tip path plane to tilt. The dynamics of rotor flapping are very fast, occurring within one revolution of the rotor, compared to the rigid body dynamics of the quadrotor. Consequently, the blade flapping equations can be written as instantaneous functions of the quadrotor horizontal velocity.

In [Pounds et al., 2006] the theory of blade flapping as used in the model for the X4 quadrotor is described. This X4 is equipped with hinged blades, so this model cannot be fully used for our situation with unhinged stiff blades as it over-predicts the blade flapping effect. In the STARMAC II project [Hoffmann et al., 2007] this model is expanded for use in these situations. The flapping properties of a stiff, fixed-pitch rotor blade can be analysed by modelling the blade as being hinged at an effective offset e_f from the centre of rotation (expressed as a percentage of the rotor radius) and a torsional spring with stiffness k_β Nm/rad at the hinge as is shown in 3-3.

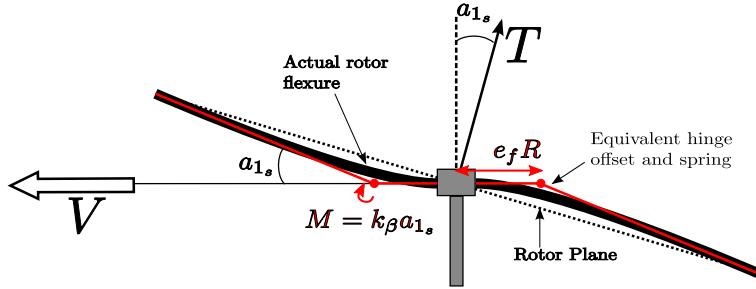


Figure 3-3: Change of the direction of the forces on the rotors is caused due to the longitudinal deflection, a_{1s} . The flapping properties of a stiff, fixed-pitch rotor blade can be analysed by modelling the blade as being hinged at an effective offset e_f from the centre of rotation (expressed as a percentage of the rotor radius) and a torsional spring with stiffness k_β Nm/rad at the hinge.

Coning is the upward flexure of the rotor blades from the lift force on each blade. For stiff two bladed rotors, the moments due to the coning angles are symmetric about the rotor hub and cancel.

The flap angle β of a rotor blade is in literature typically defined as the total deflection of a rotor blade away from the horizontal in body coordinates at any point in the rotation:

$$\beta = a_{0s} - a_{1s} \cos \Psi + b_{1s} \sin \Psi \quad (3-28)$$

where a_{0s} is the blade deflection due to coning and a_{1s} and b_{1s} are the longitudinal and lateral blade deflection amplitudes due to flapping. Ψ is the azimuth angle of the blade, and is defined as zero at the rear opposite to the velocity of the rotor V , see figure 3-4.

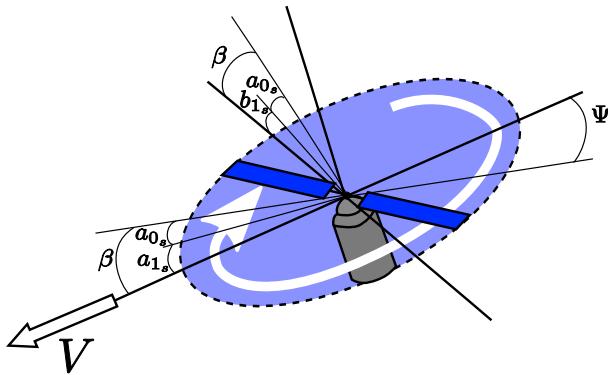


Figure 3-4: Blade flapping angles.

Since the effect of coning is cancelled by using stiff two bladed rotors and the effect of the lateral deflection cancels out for quadrotors, the change of direction of the forces on the rotor are entirely due to the longitudinal deflection a_{1s} .

From experiments in [Hoffmann et al., 2007] it is shown that the model with stiff blades corresponds reasonably with the measured values. In our model however for now we assume that the rotor blades are rigid. The non-rigid effect of the rotor blades will effect flight only in higher velocity flights, and our initial research will not touch that flight regime. It is shown by [Bouabdallah, 2007] that the rigid-rotor model is sufficient for lower velocity flight regimes. Implementing the blade flapping also requires extensive experiments as is shown in the STARMAC II project.

3-7 Simplified model for control & filter design

For control and filter design purposes it is a simplified model can be usefull, as this enables easier design of the state estimation filter and control laws. Neglecting all forces and moments except for the actuator forces and moments and weight, the model is simplified as follows:

$$m\dot{u} = -m(g \sin \theta - qw + rv) \quad (3-29)$$

$$m\dot{v} = m(g \cos \theta \sin \phi - ru + pw) \quad (3-30)$$

$$m\dot{w} = m(g \cos \theta \cos \phi - pv + qu) - U_1 \quad (3-31)$$

$$I_{xx}\dot{p} = (I_{yy} - I_{zz})qr + lU_2 \quad (3-32)$$

$$I_{yy}\dot{q} = (I_{zz} - I_{xx})rp + lU_3 \quad (3-33)$$

$$I_{zz}\dot{r} = (I_{xx} - I_{yy})pq + U_4 \quad (3-34)$$

Where U_1 , U_2 , U_3 and U_4 are the inputs, described in equation 3-44.

In hover the thrust T_i for each rotor can be estimated as:

$$T_i \approx b\Omega_i^2 \quad (3-35)$$

where b is the thrust factor in hover.

For the torque M_{Q_i} a similar relation holds:

$$M_{Q_i} \approx d\Omega_i^2 \quad (3-36)$$

where d is the drag factor in hover.

Then the inputs can be rewritten as:

$$\begin{aligned} U_1 &= \sum_{i=1}^4 T_i &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= (-T_2 + T_4) &= b(-\Omega_2^2 - \Omega_4^2) \\ U_3 &= (T_1 - T_3) &= b(\Omega_1^2 - \Omega_3^2) \\ U_4 &= (-1)^i \sum_{i=1}^4 M_{Q_i} &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{aligned} \quad (3-37)$$

Simplification of the model of course induces an error. In appendix E a comparison is shown between the the simplified and complete model, where can be noticed that the complex model is more damped.

3-8 Motor dynamics

The Direct Current (DC) motors can be modelled using the known equations [Franklin and Emami-Naeini, 2006]:

$$\dot{\Omega}_m = -\frac{1}{\tau}\Omega_m - \frac{d}{\eta r^3 J_t} \Omega_m^2 + \frac{1}{k_m \tau} u \quad (3-38)$$

$$\frac{1}{\tau} = \frac{k_m^2}{R_m J_t} \quad (3-39)$$

where d is the drag factor in hover, τ the motor-time constant, k_m the torque constant, R_m the motor internal resistance, η motor efficiency and u the motor voltage input. Linearizing the rotor speeds around an operating point Ω_0 , the motor rotational speed in hover, gives:

$$\dot{\Omega}_m = 0 = -\left(\frac{1}{\tau} + \frac{2d\Omega_0}{\eta r^3 J_t}\right)\Omega_m + \left(\frac{1}{k_m \tau}\right)u + \frac{d\Omega_0^2}{\eta r^3 J_t} \quad (3-40)$$

With this the relationship between the required angular speed and motor voltage can be written:

$$u = k_m \tau \left(\left(\frac{1}{\tau} + \frac{2d\Omega_0}{\eta r^3 J_t} \right) \Omega_m - \frac{d\Omega_0^2}{\eta r^3 J_t} \right) \quad (3-41)$$

Using equation 3-35 and 3-36 and simplifying gives:

$$u_{thrust} = \alpha_{thrust} \sqrt{T} + \beta \quad (3-42)$$

$$u_{torque} = \alpha_{torque} \sqrt{M_Q} + \beta \quad (3-43)$$

Experiments can be used to determine this relation to calculate the voltage setting from the required thrust and torque.

By inverting the equations 3-37 and combining it with equation 3-41, a formula is derived to calculate the motor voltages from the inputs:

$$u = \begin{bmatrix} u_{motor_1} \\ u_{motor_2} \\ u_{motor_3} \\ u_{motor_4} \end{bmatrix} = k_m \tau \left(\left(\frac{1}{\tau} + \frac{2d\Omega_0}{\eta r^3 J_t} \right) \sqrt{\begin{bmatrix} \frac{1}{4b} & 0 & \frac{1}{2b} & -\frac{1}{4b} \\ \frac{1}{4b} & -\frac{1}{2b} & 0 & \frac{1}{4b} \\ \frac{1}{4b} & 0 & -\frac{1}{2b} & -\frac{1}{4b} \\ \frac{1}{4b} & \frac{1}{2b} & 0 & \frac{1}{4b} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}} - \frac{d\Omega_0^2}{\eta r^3 J_t} \right) \quad (3-44)$$

This relationship is implemented in the motor inversion block in Simulink.

3-9 Matlab-Simulink model

The dynamical model derived in this chapter is used to simulate the quadrotor in a Simulink model. This model is a modified version of the model as developed the OS4 project [Bouabdallah, 2007], where the dynamic equations are corrected for the North East Down reference frame. The parameters from the OS4 project are used as these are already validated and give the most realistic quadrotor simulation (listed in appendix A-1). The simulations are split in separate blocks, allowing easy adjustments of the various parts and can be used for various quadrotor platforms by adjusting the parameters. An overview of the simulation set up can be found in appendix B.

With this model it is possible to simulate the measurements of the onboard sensors. Additional modules showing the 3D view of the quadrotor and connecting a joystick for input are also designed.

3-10 Conclusions and recommendations

This chapter describes the dynamical model of the quadrotor system that is valid for low speed flights, starting with the aerodynamic forces and moments of the rotors and combining these moments together with other external forces and moments into a dynamical equations using Newton's laws. The dynamical model is implemented in Simulink and this creates an ideal tool for evaluating various state estimation and control techniques before implementing them on the real quadrotor.

In future research this model can be expanded to implement additional effects such as blading flapping and the ground effect. An other effect that is not modelled are the lags in the motor system, to implement this motor identification is required.

Chapter 4

Control design

A quadrotor has the four actuators, and in free flight six degrees of freedom so it is an under-actuated and dynamically unstable system, therefore the controller design is a challenging task. The open loop response of the quadrotor is unstable and therefore control feedback is necessary to be able to fly the quadrotor in simulation and in reality.

This chapter starts with analysing the control strategies already applied to the quadrotor layout in previous experiments and simulations. For control of the quadrotor, classical control is selected as the preferred method. Following up, the theory of classical control will be explained with respect to its application to the quadrotor system. Next the results in simulation will be discussed.

4-1 Selection of the control method

It is wise to have a look at the various control methods already applied on the quadrotor layout so it is possible to learn by the experiments already done and use this knowledge to select the controller that will be used in this project or even in future projects. Each controller type will be explained briefly together with the projects involved.

4-1-1 Literature overview

Classical control

In classical control the controller consist of the three components: the Proportional, Integral and Derivative part (P-I-D). The Proportional part provides the basic feedback, the Integral part removes the steady state and the Derivative part adds damping to the controller.

In the OS4 project at the EPFL [Bouabdallah, 2007, Bouabdallah et al., 2004b, Bouabdallah and Siegwart, 2006] PID control was applied successfully on a quadrotor on a ball-joint. Simulations and the experiments have shown that their OS4 quadrotor can be controlled efficiently in hover using a classical approach. This controller will not be able to stabilize the quadrotor in presence of strong perturbation. The results with the classical approach were better than with a Linear-Quadratic Regulator. An integrator was added to remove the

steady state error.

Flight tests indoors in the STARMAC project [Hoffmann et al., 2007] show that good tracking was obtained with classical control even without an integrator for pitch and roll giving tracking errors on the order of 2 - 3 degrees.

At low velocities and with small aerodynamic disturbances (for example in indoor flight), PID control is fully sufficient for good tracking of commanded attitude since the vehicle approximates a double-integrator with a first-order lag from the motor dynamics. In higher velocity translational flight, the pitch and roll dynamics of a quadrotor are very sensitive to additional effects, and thus other control strategies should be used.

At Lakehead University [Tayebi and McGilvray, 2006] an alternative to the standard classical controller is implemented. The proposed controller is based upon the use of a PD² feedback structure, where the proportional action is in terms of the vector quaternion and the two derivative actions are in terms of the airframe angular velocity and the vector quaternion velocity. Also is shown that the model-independent Proportional-Derivative (PD) controller, where the proportional action is in terms of the vector-quaternion and the derivative action is in terms of the airframe angular velocity, without compensation of the Coriolis and gyroscopic torques, provides asymptotic stability. The proposed controller as well as some other controllers have been tested experimentally on a small-scale quadrotor aircraft.

In the project at at the University of Cambridge using a single camera for state estimation also a nested PID controller is used [Kemp, 2006]. An Proportional-Integral (PI) controller is *nested* inside a controller which chooses the orientation required to reach or hold at a desired position. This combined controller is able to reliably fly the quadrotor and results from a variety of test flights are presented.

Linear-Quadratic Regulator

The LQR is an optimal control solution that minimises a certain cost function. In discrete time the algebraic Riccati equation is used. At Heudiasyc [Castillo et al., 2005] test are done applying a LQR on a quadrotor platform. The roll and pitch angle of the quadrotor oscillates considerably, so that the helicopter could not hover. After numerous trials this oscillations were reduced, nevertheless, the obtained performance is not adequate to perform autonomous hovering.

Nested Saturations

As performance of the LQR was not adequate, Heudiasyc [Castillo et al., 2004, Castillo et al., 2005, Escareno et al., 2006, Salazar-Cruz et al., 2005] focused on the use of the nested saturations technique for quadrotor control. The nested saturation technique developed can exponentially stabilize a chain of integrators with bounded input. In successive papers the control strategy was developed and implemented on a microcontroller on-board a quadrotor. The experimental results showed that the low-cost embedded control system performs satisfactorily.

Backstepping

In backstepping the control law is designed using the Lyapunov stability criterium. Using a virtual control law the idea is to *step back* through the system to find a control law. Backstepping controllers are especially useful when some states are controlled through other states [Altug et al., 2002], as it is the case with the quadrotor dynamics.

After the unsatisfying results applying [LQR](#) on the quadrotor platform the application of pure backstepping on the quadrotor is tested at the [EPFL](#) [Bouabdallah and Siegwart, 2005, Bouabdallah, 2007]. The controller is able to stabilize the orientation angles even for relatively critical initial conditions. Next integral backstepping was applied [Bouabdallah, 2007], which combines integral action of [PID](#) with pure backstepping. Robustness against disturbances offered by backstepping and robustness against model uncertainties offered by the integral action. This integral backstepping controller was chosen as the final control method as it had the optimum performance.

At the Commissariat l'Énergie Atomique ([CEA](#)) pure backstepping is applied to control the translational velocities of the vehicle as well as the yaw angle [Guenard et al., 2005]. The simplicity of the proposed control law and the use of quaternions have allowed us to implement the proposed controller in an embedded microcontroller for which the computing power is limited. Flight experimentations show good performance of the proposed controller for the flight stability.

The Robotics Laboratory of Versailles also investigates a full state backstepping controller [Madani and Benallegue, 2006b, Madani and Benallegue, 2006a]. The system is divided into an under-actuated subsystem, fully-actuated subsystem and a propeller subsystem. Simulation and experimentation results show the adequate performance of the proposed controller.

In an interesting paper of the GRASP Laboratory at the University of Pennsylvania backstepping is compared against a mode based feedback linearisation controller in simulation [Altug et al., 2002]. From this simulation the backstepping controller is selected as the favourable option. This selected controller is then successfully applied in a quadrotor set-up on a stand.

Sliding mode

In control theory, Sliding Mode Control ([SMC](#)) is a type of variable structure control where the dynamics of a nonlinear system is altered via application of a high-frequency switching control. This is a state feedback control scheme where the feedback is not a continuous function of time.

After testing the pure backstepping control law at the [EPFL](#) this controller is altered to use a sliding mode design with a signum function [Bouabdallah, 2007, Bouabdallah and Siegwart, 2005]. As it can be seen from the experiments the sliding-mode approach provides average results, because it induces a chattering effect. This is partly due to switching nature of the controller which introduces high frequency, low amplitude vibrations causing the sensor to drift.

At Ohio State University [[Xu and Özgüner, 2006](#)] the control problem is divided into a fully-actuated subsystem (z and ψ) and an under-actuated subsystem (x, y, θ, ϕ) in simulation. The results show that their proposed approach with a sliding mode controller is able to robustly stabilize the quadrotor helicopter and move it to any position with any yaw angle. The chattering effect of the sliding mode control, as also was found by the [EPFL](#), is avoided in the control input by using the continuous approximation of the sign function. This makes this approach applicable in real applications.

To provide robust control throughout the flight envelop and overcome the chattering effect of the [SMC](#) the Integral Sliding Mode ([ISM](#)) technique is applied in the STARMAC project [[Waslander et al., 2005](#)]. An [ISM](#) altitude control with integral [LQR](#) position control technology was implemented real-time on the platform. Compared to linear control design techniques the [ISM](#) control proves a significant enhancement. By explicitly incorporating bounds on the

unknown disturbance forces in the derivation of the control law, it is possible to maintain stable altitude on a system that has evaded standard approaches.

Feedback Linearisation

In FeedBack Linearisation (**FBL**) control the known non-linear system is transformed into an equivalent linear system, through a change of variables and a suitable control input. A specific case of **FBL** is known as Non-linear Dynamic Inversion (**NDI**), where the non-linear model is linearised and inverted. This linearised system is placed as the *inner loop* and an outer loop is added to control this inner loop.

At the Robotics Laboratory of Versailles [Mokhtari and Benallegue, 2004, Mokhtari et al., 2005, Mokhtari et al., 2006, Benallegue et al., 2006, Benallegue et al., 2007] a feedback linearisation controller for the quadrotor model is designed in several successive design steps. To avoid singularity in the Lie transformation matrices when using exact linearisation, one of the control variables is delayed by a double integrator. The outer loop is designed with a classical polynomial control law. Additionally an higher-order sliding mode observer is designed to find the additional states from the position (x,y,z) and the yaw angle ψ . Simulations show that this combination of controller and observer performs satisfactory also when aerodynamic disturbances applied. In [Bouadi et al., 2007] the inner loop feedback linearisation is combined with an outer loop $GH\infty$ controller. The results show that the overall system becomes robust when weighting functions are chosen judiciously.

Neural-Network

Using purely Neural Network (**NN**), trained offline, the Royal Military College of Canada [Dunfied et al., 2004] tried to create a **NN** based control system that could reside in an on-board lightweight microcontroller system. Success was achieved for the control of the roll and pitch, using tilt sensor inputs only and tilt sensor and gyro inputs. When the yaw axis was added, control of all three axes was achieved but some work remains to be done to integrate more smoothly the drift of the gyro data.

Reinforcement Learning

On the **STARMAC** platform next to the **ISM** an Reinforcement Learning (**RL**) controller was tested for *altitude* control [Waslander et al., 2005]. For this work, a nonlinear, nonparametric model of the system is first constructed using flight data, approximating the system as a stochastic Markov process. Then a model-based reinforcement learning algorithm uses the model in policy-iteration to search for an optimal control policy that can be implemented on the embedded microprocessors. The **RL** law is susceptible to system disturbances for which it is not trained. In particular, varying battery levels and blade degradation may cause a reduction in stability or steady state offset. Comparison of the step response for **ISM** and **RL** control reveals both stable performance and similar response times, although the transient dynamics of the **ISM** control are more pronounced.

4-1-2 Conclusion

In the previous sections the applied control strategies on the quadrotor design are discussed. Clearly not all control strategies give similar results. Classical **PID** control is the most simple control type for the quadrotor that performs very good as well, therefore it is decided to start with this type of control. In later phases it is possible to use more advanced controllers.

4-2 Classical control

Classical PID control is selected as the preferred control method, this section the basic theory is given of this controller type.

4-2-1 Theory

The classical control method or PID control structure in its complete form consists of three components: the Proportional, Integral and Derivative part. In this section the separate parts are described with their respective characters. The response of a system can best be described by looking at the response of a unit step input on a system, shown in figure 4-1.

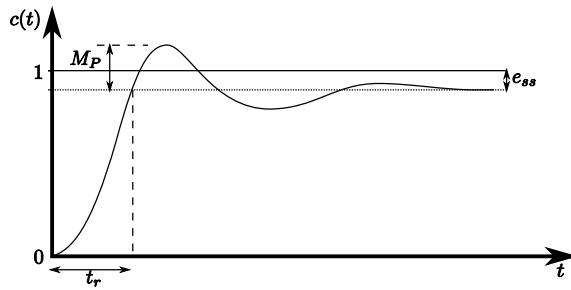


Figure 4-1: Example of a typical response to a unit step input of a system. M_p is the maximum overshoot, t_r the rise time and e_{ss} is the steady state error.

Proportional control

The proportional control reduces the rise time of the system. With pure proportional control, it is typical that steady state errors occur in response to a step input.

Proportional-Integral control

If an integral action is added, then the steady state error will be reduced. Note that integral action, while removing the offset or steady-state error, may lead to oscillatory response of slowly decreasing amplitude or even increasing amplitude, both of which are usually undesirable. Note that when control saturation occurs integrator windup can happen, this should be avoided as it creates instability of the controller.

Proportional-Derivative control

Derivative control when added to the proportional control, provides a means of obtaining a controller with high sensitivity, or in other words it improves the transient response. For a step input this means that the overshoot is reduced. An advantage of using derivative control is that it responds to the rate of change of the actuating error and can produce a significant correction before the magnitude of the actuating error becomes too large. Derivative control thus increases the stability of the system. Although derivative control does not affect the steady-state error directly, it adds damping to the system and thus permits the use of a larger value of the proportional gain, which will result in an improvement in the steady-state accuracy.

Proportional-Integral-Derivative control

Combining the Proportional, Integral and Derivative control results in a **PID** controller, which can be applied to reduce rise time, reduce maximum overshoot and remove the steady state error for many systems. In aerospace engineering mainly the parallel **PID** controller structure is used, depicted in figure 4-2. The control law is described with the transfer function:

$$K(s) = K_P + \frac{K_I}{s} + K_D s \quad (4-1)$$

where K_P is the proportional gain, K_I the integral gain and K_D the derivative gain.

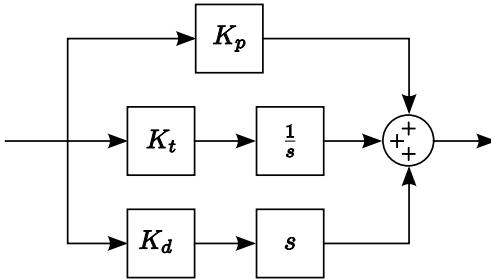


Figure 4-2: Parallel PID control structure

4-2-2 Controller tuning

For tuning of the PID controller, several tuning algorithms are developed, such as Ziegler-Nichols and Lambda tuning. From the Insight project [Thus et al., 2007] it is known best results for the quadrotor are obtained by iterative tuning.

When iterative tuning is used, the following algorithm can be used to get the desired overall response:

1. Investigate step response
2. Add gain K_P to reduce the rise time t_r if necessary
3. Add gain K_D to improve the maximum overshoot M_p if necessary
4. Add gain K_I to eliminate steady state error e_{ss} if necessary
5. Adjust gains till desired overall response is obtained

4-3 Quadrotor controller structure

The proposed controller structure is a **PID** controller for each of the states, with a special nested **PID** controller for position control [Buskey et al., 2003].

Additionally a motor inversion block is added, which is the calculation of equation 3-44, to translate the control command to a motor voltage.

4-3-1 Yaw control

The yaw is the least critical of the controls since it has no direct effect on the quadrotor's motion. It can be independently tuned and tested, while having manual control in the remaining channels.

It is important to note that disturbances have a relatively small effect on yaw, so no large gains are required to control the yaw direction and only low bandwidth yaw control is required.

The PID control law for yaw is described by the following equation:

$$U_4 = -K_{\psi_P}(\psi - \psi_d) - K_{\psi_I} \int (\psi - \psi_d) \delta t - K_{\psi_D}(\dot{\psi} - \dot{\psi}_d) \quad (4-2)$$

And also shown schematically in figure 4-3

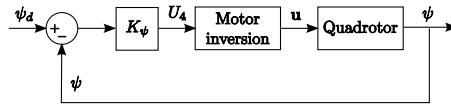


Figure 4-3: Yaw ψ control loop

4-3-2 Roll and pitch controller

Because the quadrotor's layout is symmetrical the roll and pitch control can be assumed independent for small attitude angles. Relative to the yaw control a higher bandwidth is required, since its value has also a direct relationship to the lateral acceleration in x and y direction.

For roll the following control law is used, also shown schematically in figure 4-4

$$U_2 = -K_{\phi_P}(\phi - \phi_d) - K_{\phi_I} \int (\phi - \phi_d) \delta t - K_{\phi_D}(\dot{\phi} - \dot{\phi}_d) \quad (4-3)$$

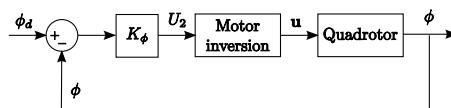


Figure 4-4: Roll ϕ controller

Similar for the pitch control:

$$U_3 = -K_{\theta_P}(\theta - \theta_d) - K_{\theta_I} \int (\theta - \theta_d) \delta t - K_{\theta_D}(\dot{\theta} - \dot{\theta}_d) \quad (4-4)$$

4-3-3 Horizontal position nested controller

The horizontal position (x, y) of the quadrotor is controlled by adjusting the roll and pitch. The total thrust produces a lift acceleration approximately equal to the gravity g in a direction normal to the plane of the rotors. A small angle of roll ϕ results therefore in a lateral acceleration $\ddot{y} \approx \phi g$. Note that the commanded roll should be in the plane of rotors, otherwise incorrect commands are given when the yaw does not equal zero.

A nested controller is used to correct the position of the vehicle, as shown in figure 4-5. The commands in x and y direction:

$$U_x = -K_{x_P}(x - x_d) - K_{x_I} \int (x - x_d) \delta t - K_{x_D}(\dot{x} - \dot{x}_d) \quad (4-5)$$

$$U_y = -K_{y_P}(y - y_d) - K_{y_I} \int (y - y_d) \delta t - K_{y_D}(\dot{y} - \dot{y}_d) \quad (4-6)$$

Corrected for the yaw angle they give the commands for the desired roll and pitch angle:

$$\phi_d = -\sin(\psi)U_x + \cos(\psi)U_y \quad (4-7)$$

$$\theta_d = \cos(\psi)U_x + \sin(\psi)U_y \quad (4-8)$$

The schematics of the y position controller is shown in figure 4-5.

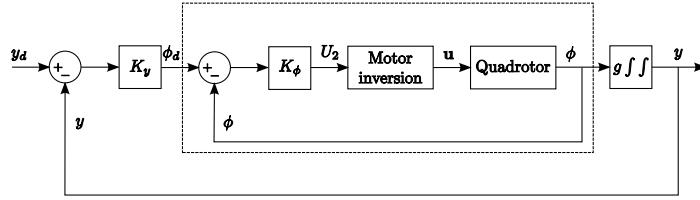


Figure 4-5: Horizontal position y nested controller. For simplification it is assumed that $\psi \approx 0$.

4-3-4 Vertical position z controller

When operating above the ground effect, the control output U_1 is approximately proportional to the vertical acceleration in the body reference frame. To remain at constant height, a large value of U_1 is required to counteract gravity, additionally a PID controller is added to stabilise the motion in z direction. The control law can be described as:

$$r_1 = -K_{z_P}(z - z_d) - K_{z_I} \int (z - z_d) \delta t - K_{z_D}(\dot{z} - \dot{z}_d) \quad (4-9)$$

$$U_1 = \frac{r_1 + mg}{\cos \phi \cos \theta} \quad (4-10)$$

And its schematics are shown in figure 4-6

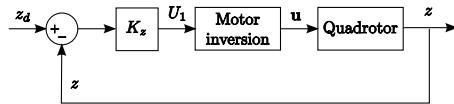


Figure 4-6: Vertical position z controller

4-3-5 Waypoint following

Waypoints are coordinates that identify a point in physical space where the quadrotor should fly to (so x_d , y_d and z_d). In order to hover only a single waypoint is necessary, in order to fly a certain track waypoints with an timestamp t_d need to be generated, this timestamp gives the moment the command of x_d , y_d and z_d is set. It is possible to either use discrete waypoints, or generate waypoints by interpolation. The latter is preferred as this bounds the error that is fed into the controller significantly and thus results in a much smoother response. Using discrete waypoints with to large step size can result in instability.

4-4 Results

Using the Simulink model developed in chapter 3, we test the proposed controller before implementing it into the quadrotor. Using the gains from table 4-1, this results in a stable hover of the quadrotor model. The results of the simulation is shown in figure 4-7.

Figures 4-8 and 4-9 show the result of flying the quadrotor using interpolated waypoints. This flight plan will be used to test the indoor navigation.

| movement | P | I | D |
|----------|-------|------|------|
| roll | 0.9 | 0.3 | 0.2 |
| pitch | 0.9 | 0.3 | 0.2 |
| yaw | 0.06 | 0.01 | 0.02 |
| x | -0.09 | 0 | -0.1 |
| y | 0.09 | 0 | 0.1 |
| z | 8 | 0.5 | 10 |

Table 4-1: PID gains in simulation

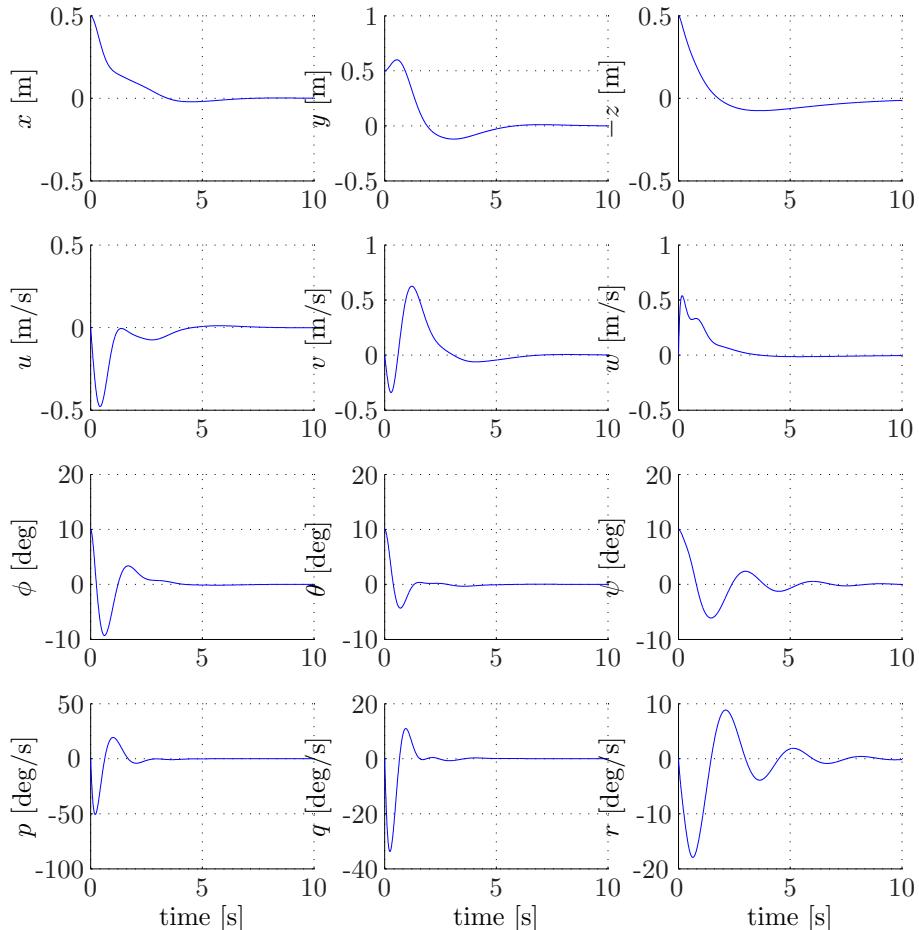


Figure 4-7: Controller stability in hover. Initial states: 10 degrees pitch, roll, yaw and 0.5 meters in x, y, z position.

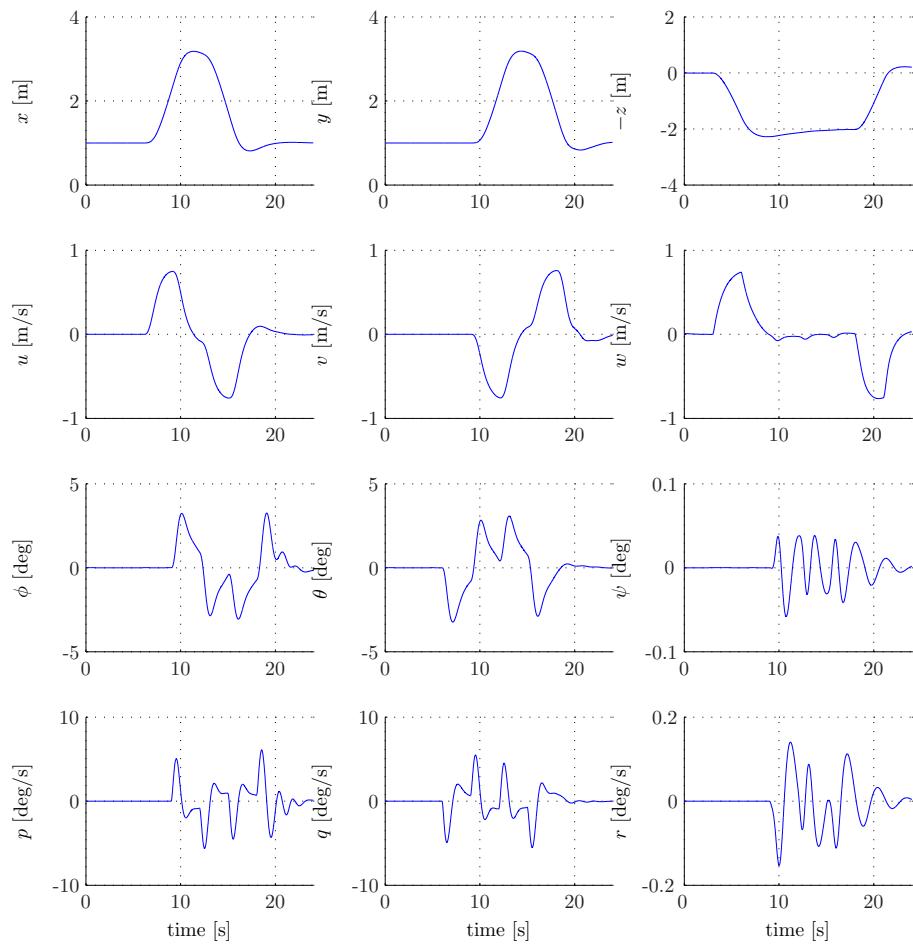


Figure 4-8: Simulation of a flight tracking waypoints generated by interpolation.

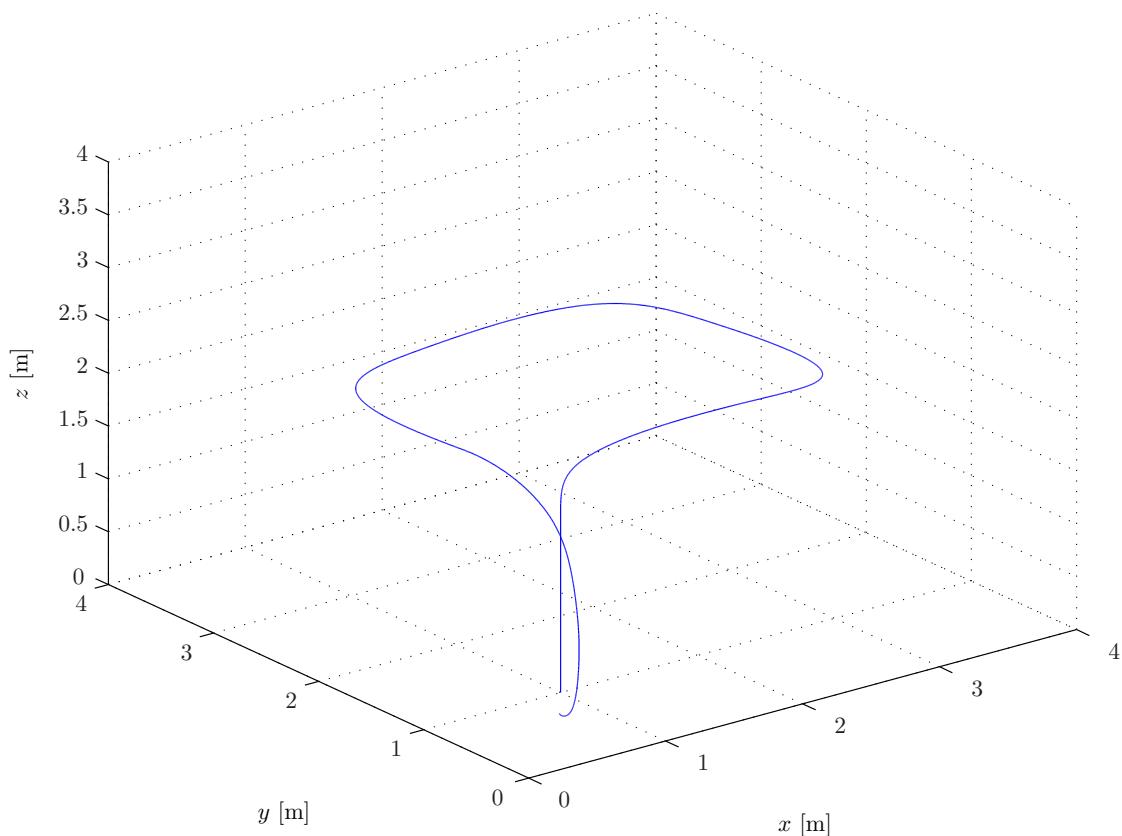


Figure 4-9: Simulation of a flight tracking waypoints generated by interpolation.

4-5 Conclusions and recommendations

The selected controller structure, a PID controller for each of the states, with a special nested PID controller for position control is able to successfully stabilises the quadrotor and is also able to fly a track in space with selected waypoints.

At the TU Delft in the department of Dynamics and Control of Aerospace Vehicles ([DCAV](#)) extensive research is done in the field of using [NDI](#) in flight control. To the knowledge of the author, this has only been tested in simulation for the quadrotor layout, therefore it would be interesting to apply [NDI](#) in practice on the quadrotor developed. However due to time limitations this is not possible within this thesis project.

Part III

Indoor navigation using state estimation

Chapter 5

Sensor modelling

In chapter 1 the sensors that are used in the proposed sensor integration are introduced: three gyroscopes, three accelerometers, three magnetometers and IR sensors. This chapter describes the models used to simulate and estimate these measurement of these sensors. This modelling cannot be done without knowledge of the technologies behind the sensors, therefore a explanation is given of the technology of each of the sensors and their error characteristics.

5-1 3-Axis gyroscope

The gyroscopes used onboard the quadrotor are Micro-machined ElectroMechanical Systems (**MEMS**) sensors, which have compared to mechanical and optical gyroscopes a low part count (as few as three parts) and are relatively cheap to manufacturer. At present **MEMS** sensors cannot match the accuracy of optical devices, however they are expected to do so in the future. Some advantages of **MEMS** sensors are: small size, low weight rugged construction and low power consumption.

MEMS gyroscopes make use of the Coriolis effect, which describes the effect that in a frame of reference rotating at an angular velocity ω , a mass m moving with velocity \mathbf{v} experiences a force:

$$F_c = -2m(\omega \times \mathbf{v}) \quad (5-1)$$

In a **MEMS** gyroscope the Coriolis effect is measured using vibrating elements [Woodman, 2007]. Many vibrating element geometries exist, but the simplest geometry consist of a single mass which is driven to vibrate along a drive axis, as shown in image 5-1(a). When the gyroscope is rotated a secondary vibration is induced along the perpendicular sense axis due to the Coriolis force. The angle velocity can be calculated by measuring the secondary rotation. In image 5-1(b) the detailed layout of the Y sensor in the IDG-300 is shown.

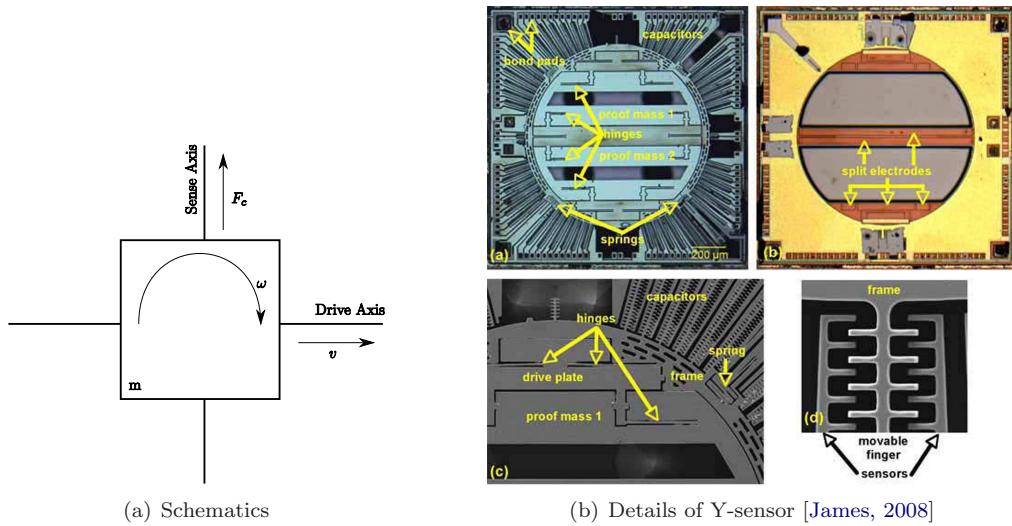


Figure 5-1: IDG-300 MEMS Sensor

5-1-1 MEMS error characteristics

As mentioned [MEMS](#) gyroscopes are far less accurate than optical gyroscopes. To model them correctly it is necessary to first describe the errors that arise in them. The ability to compensate for these inaccuracies is an important factor determining the quality of the device.

- **Constant bias**

The bias of a rate gyro is the average output from the gyroscope when it is not undergoing any rotation (i.e: the offset of the output from the true value). A constant bias error of ϵ , when integrated, causes an angular error which grows linearly with time $\theta(t) = \epsilon \cdot t$. The constant bias error of a rate gyro can be estimated by taking a long term average of the gyro's output whilst it is not undergoing any rotation. Once the bias is known it is trivial to compensate for it by simply subtracting the bias from the output.

- **Thermo-Mechanical White Noise / Angular Random Walk**

The output of a MEMS gyro will be perturbed by some thermo-mechanical noise which fluctuates at a rate much greater than the sampling rate of the sensor. As a result the samples obtained from the sensor are perturbed by a white noise sequence, which is simply a sequence of zero-mean uncorrelated random variables [Woodman, 2007]. Each random variable is identically distributed and has a finite variance σ^2 . This effect of the noise on the integrated signal is that the noise introduces a zero-mean random walk error into the integrated signal, whose standard deviation $\sigma_\theta(t) = \sigma \cdot \sqrt{\delta t \cdot t}$ grows proportionally to the square root of time. Here a random walk is defined as a process consisting of a series of steps, in which the direction and size of each step is randomly determined.

- **Flicker noise / Bias drift**

The bias of a MEMS gyroscope wanders over time due to flicker noise in the electronics and in other components susceptible to random flickering. The effects of flicker noise are usually observed at low frequencies in electronic components. At high frequencies flicker noise tends to be overshadowed by white noise. Bias fluctuations which arise due to flicker noise are usually modelled as a random walk.

In reality bias fluctuations do not really behave as a random walk. If they did then the uncertainty in the bias of a device would grow without bound as the timespan increased.

In practice the bias is constrained to be within some range, and therefore the random walk model is only a good approximation to the true process for short periods of time.

- **Temperature Effects**

Temperature fluctuations due to changes in the environment and sensor self heating induce movement in the bias. Note that such movements are not included in bias stability measurements which are taken under fixed conditions.

Any residual bias introduced due to a change in temperature will cause an error in orientation which grows linearly with time. The relationship between bias and temperature is often highly nonlinear for MEMs sensors.

- **Calibration Errors**

The term *calibration errors* refers collectively to errors in the scale factors, alignments, and linearities of the gyros. Such errors tend to produce bias errors that are only observed whilst the device is turning.

Such errors lead to the accumulation of additional drift in the integrated signal, the magnitude of which is proportional to the rate and duration of the motions.

5-1-2 Assumptions

Assumptions for the gyroscope model:

- Noise with zero mean and variance σ^2
- Bias with random walk model

5-1-3 Mathematical model

The measured angular rates are assumed to have a bias and white noise:

$$\begin{bmatrix} p_m \\ q_m \\ r_m \end{bmatrix} = \begin{bmatrix} p + \lambda_{p_m} + w_{p_m} \\ q + \lambda_{q_m} + w_{q_m} \\ r + \lambda_{r_m} + w_{r_m} \end{bmatrix} \quad (5-2)$$

The derivatives of the gyroscope biases are modelled as white noise:

$$\begin{bmatrix} \dot{\lambda}_{p_m} \\ \dot{\lambda}_{q_m} \\ \dot{\lambda}_{r_m} \end{bmatrix} = \begin{bmatrix} w_{\dot{\lambda}_{p_m}} \\ w_{\dot{\lambda}_{q_m}} \\ w_{\dot{\lambda}_{r_m}} \end{bmatrix} \quad (5-3)$$

5-2 3-Axis accelerometer

The ADXL330 sensor is a monolithic, three-axis, force-balanced, capacitive accelerometer. The proof mass, supported by the four springs, is movable along any axis (X-Y-Z) in response to an inertial force. The proof mass is attached to the common electrode of sets of differential capacitors, with the other electrodes of each differential capacitor pair being fixed. Each sense axis requires a separate set of differential capacitors oriented to the appropriate direction. The differential capacitance is measured by applying one of two complementary (180° out-of-phase) square-waves to each of the fixed plates of the differential capacitors. [James, 2008]

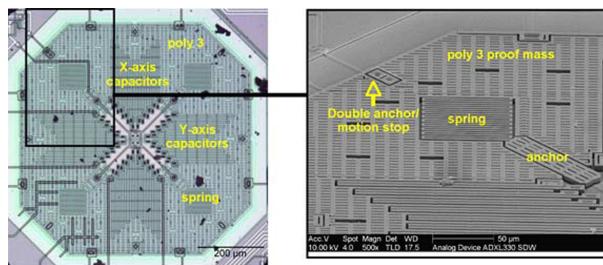


Figure 5-2: Optical and SEM images of ADXL330 MEMS structure

5-2-1 Specific Accelerometer Error characteristics

Additional to the errors mentioned for the MEMS gyroscopes, the accelerometer measurement is influenced by the position of the sensor with respect to the centre of gravity of the vehicle. Because our sensor is placed close to the centre of gravity, this effect is ignored.

5-2-2 Assumptions

Assumptions for the accelerometer model:

- Noise with zero mean and variance σ^2
- Bias with random walk model
- Placed in the centre of gravity of the vehicle

5-2-3 Mathematical model

For a perfect accelerometer the measurement can be calculated with:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv & + & g \sin \theta \\ \dot{v} + ru - pw & - & g \cos \theta \sin \phi \\ \dot{w} + pv - qu & - & g \cos \theta \cos \phi \end{bmatrix} \quad (5-4)$$

However, in reality also a biases and noise are also measured:

$$\begin{bmatrix} A_{x_m} \\ A_{y_m} \\ A_{z_m} \end{bmatrix} = \begin{bmatrix} A_x + \lambda_{A_{x_m}} + w_{A_{x_m}} \\ A_y + \lambda_{A_{y_m}} + w_{A_{y_m}} \\ A_z + \lambda_{A_{z_m}} + w_{A_{z_m}} \end{bmatrix} \quad (5-5)$$

5-3 3-Axis magnetometer

The magnetometer measures the strength and direction of the local magnetic field, allowing the north direction to be found. Magnetometers are not accurate enough to replace gyroscopes completely in a Inertial Navigation System (INS). The main advantage of the magnetometer is that it does not rely on any external infrastructure.

5-3-1 Error characteristics

Local disturbances in the earth's magnetic field due to nearby magnetic objects have influence on the measurement of the magnetometer. It was found that the magnetometer is very sensitive to power cables and radiators. The motors also change the direction of the magnetic vector, which should be adjusted for.

5-3-2 Assumptions

Assumptions for the magnetometer model:

- No bias on the measurement, only noise due to vibrations of the motors.
- The direction of the magnetic field vector M_0 is not changing over time.

5-3-3 Mathematical model

For a perfect magnetometer the measurement can be calculated with:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \times \begin{bmatrix} \cos(\psi + \psi_{room}) & \sin(\psi + \psi_{room}) & 0 \\ -\sin(\psi + \psi_{room}) & \cos(\psi + \psi_{room}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{x_0} \\ M_{y_0} \\ M_{z_0} \end{bmatrix} \quad (5-6)$$

In this formula the yaw angle of the room is implemented ψ_{room} , see also figure 5-6, which is the angle of the room with respect to the magnetic field. M_{x_0} , M_{y_0} , M_{z_0} are defined as:

$$\begin{bmatrix} M_{x_0} \\ M_{y_0} \\ M_{z_0} \end{bmatrix} = \begin{bmatrix} -\cos(i) \cos(d) \\ -\cos(i) \sin(d) \\ \sin(i) \end{bmatrix} \quad (5-7)$$

Where i is the inclination and d is the declination, see figure 5-3. The inclination can be calculated using $i = \sin^{-1}(M_z)$ when holding the magnetometer static and horizontal using the average over a number of measurements. The declination d can be measured can be measured by aligning the magnetometer exactly north and comparing the measured heading with the true heading. However in practice this is hard to achieve without exact knowledge of the true heading. Literature states (see figure 5-3) that zero is a good approximation for our location on the earth. The real magnetometer measurement is assumed to be only disturbed by white noise:

$$\begin{bmatrix} M_{x_m} \\ M_{y_m} \\ M_{z_m} \end{bmatrix} = \begin{bmatrix} M_x + w_{M_{x_m}} \\ M_y + w_{M_{y_m}} \\ M_z + w_{M_{z_m}} \end{bmatrix} \quad (5-8)$$

Each timestep the measurements of the magnetometer are normalised, to overcome problems with error spikes in the measurements.

Note that for determining M_{x_m} , M_{y_m} and M_{z_m} the room parameters ϕ_{room} should be known.

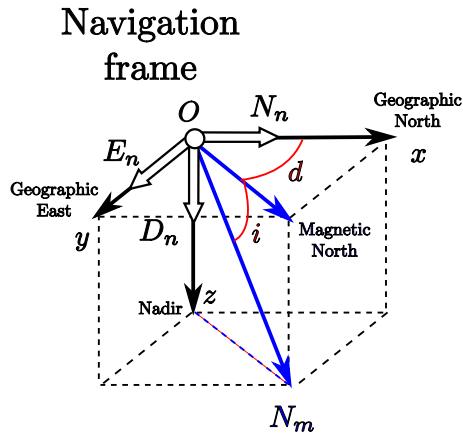


Figure 5-3: Definition of the magnetic field parameters. The inclination of the magnetic field i is the angle between the magnetic north and the vector of the magnetic field measurement N_m . In the Netherlands the inclination is approximately 60° degrees downwards. The declination d is the difference between the true geographic north and the magnetic north, in the Netherlands about 0° - 2° degrees west [van der Merwe, 2004]. The deviation (not depicted) is the angle between the geographic north and the measured geographic north, this is caused due to disturbances onboard the vehicle.

5-4 Infrared sensors

The selected infrared rangers use triangulation and a small linear CCD array to compute the distance and/or presence of objects in the field of view. The basic idea is that a pulse of IR light is emitted by the emitter. This light travels out in the field of view and either hits an object or just keeps on going. In the case of no object, the light is never reflected and the reading shows no object. If the light reflects off an object, it returns to the detector and creates a triangle between the point of reflection, the emitter, and the detector.

The angles in this triangle vary based on the distance to the object, see image 5-4. The receiver portion of these detectors is a precision lens that transmits the reflected light onto various portions of the enclosed linear CCD array based on the angle of the triangle described above. The CCD array can then determine what angle the reflected light came back at and therefore, it can calculate the distance to the object.

This method of ranging is reasonably immune to interference from ambient light and offers indifference to the colour of object being detected. Detecting a black wall in full sunlight is possible with this technique. The wavelength of the pulse is in the range of 850nm +/-70nm.

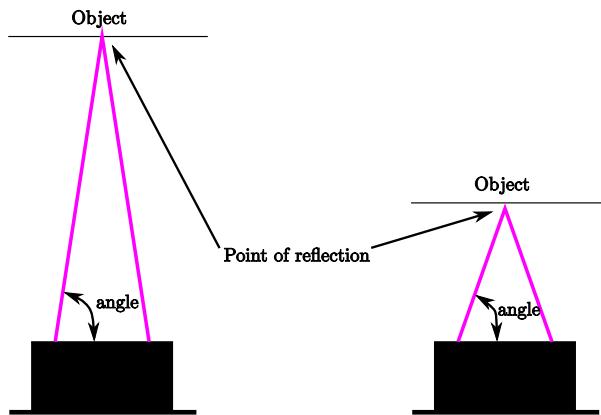


Figure 5-4: Functioning of the IR sensor - measurement by triangulation

5-4-1 Infrared error characteristics

The infrared sensors have some specific errors that cause the measurement to be disrupted:

- **Nonlinearity**

Because of the measurement method, the output of these detectors is non-linear with respect to the distance being measured, see 5-5. Secondly, within a close range inside of the stated distance range (for the GP2Y0A02YK less than 20cm), the output drops rapidly and resembles a longer range reading. By placing the sensors in such way that these measurement are prevented (as in the x-y direction) or by combining sensors with different ranges, this problem can be avoided.

- **Reflectivity of surface**

In figure 5-5 the difference is shown between the measurements on white paper and gray paper, which differ slightly. It was found that the reflectivity of most surfaces is sufficient, except for very dark carpet.

- **Ambient light**

As stated in the specifications, this method of ranging is almost immune to ambient light. However initial test with the quadrotor show that some interference is measured with fluorescent lamps which are quite common in office environments.

- **Calibration Errors**

The quadrotor is equipped with several infrared sensors. Test have shown that each sensor has slightly different characteristics.

- **Aligning Errors**

Due to the incorrect placement of the sensors on the vehicle an other distance is measured than expected.

- **Measurement noise**

Because of the non linear relationship of the measurements the noise increases with measured distance.

- **Interference with other IR sensors**

The IR sensors also measure the reflection of the beams on opposite walls of the other IR sensors onboard the quadrotor.

- Not aligned with ADC update rate

The GP2Y0A02YK updates its measurement at about 26 Hz and the GP2Y0A0700 updates its measurement at about 70 Hz, the ADC is read at about 380 Hz so effectively the measurements can only be used at the original update rate of the sensor.

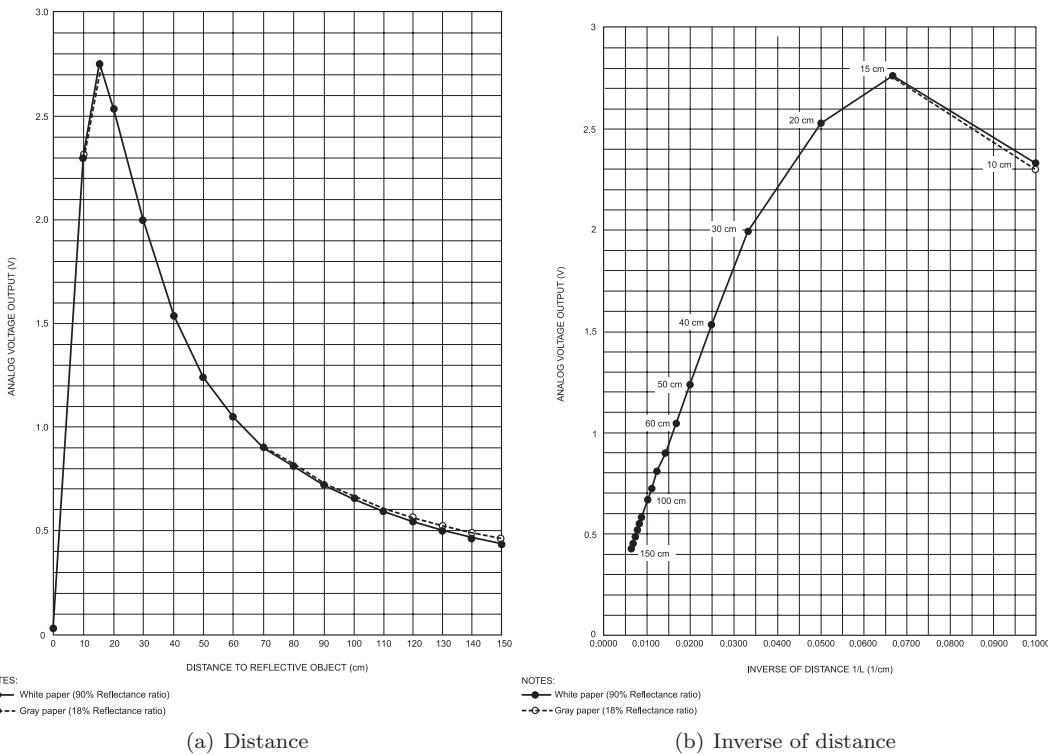


Figure 5-5: GP2Y0A02YK - Example of output characteristics from the datasheet

5-4-2 Assumptions

Assumptions for the IR model:

- Constant noise, worst case scenario;
- No measurement outside minimum and maximum range;
- Room is not changing and room size is known;
- The position of the sensor with respect to the quadrotor's centre of gravity is neglected;
- Each sensor same characteristics;
- The yaw angle of the quadrotor does not change more than 40 degrees.

5-4-3 Mathematical model

The infrared sensor measures the distance from the sensor to a fixed reflection point, for example a wall.

A room is defined as a box with four walls, a floor and a ceiling. The walls are perpendicular to each other and a distance of x_{room} and y_{room} apart, the distance between the ceiling and floor is z_{room} , see figure 5-6. Two walls are crossing the origin O . The distance from any point P in the room to the wall it sees, when measuring with 6 infrared sensors from the origin of the body reference frame along the axis of this frame without any rotations, equals:

$$\begin{bmatrix} IR_{x_1} \\ IR_{x_2} \\ IR_{y_1} \\ IR_{y_2} \\ IR_{z_1} \\ IR_{z_2} \end{bmatrix} = \begin{bmatrix} x_{room} - x_P \\ x_P \\ y_{room} - y_P \\ y_P \\ -z_P \\ z_{room} + z_P \end{bmatrix} \quad (5-9)$$

Notice the negative sign of the z , because the position is specified in the North-East-Down Earth Fixed reference frame.

When the body reference frame rotates, see figure 5-6 and if point P is the position of the quadrotor, the distance measurements of the infrared sensors using basic goniometry yield:

$$\begin{bmatrix} IR_{x_1} \\ IR_{x_2} \\ IR_{y_1} \\ IR_{y_2} \\ IR_{z_1} \\ IR_{z_2} \end{bmatrix} = \begin{bmatrix} \frac{x_{room}-x}{\cos \theta \cos \psi} \\ \frac{x}{\cos \theta \cos \psi} \\ \frac{y_{room}-y}{\cos \phi \cos \psi} \\ \frac{y}{\cos \phi \cos \psi} \\ \frac{-z}{\cos \phi \cos \theta} \\ \frac{z_{room}+z}{\cos \phi \cos \theta} \end{bmatrix} \quad (5-10)$$

Because the measurements are not perfect, also a noise is measured:

$$\begin{bmatrix} IR_{x_1m} \\ IR_{x_2m} \\ IR_{y_1m} \\ IR_{y_2m} \\ IR_{z_1m} \\ IR_{z_2m} \end{bmatrix} = \begin{bmatrix} IR_{x_1} + w_{IR_{x_1m}} \\ IR_{x_2} + w_{IR_{x_2m}} \\ IR_{y_1} + w_{IR_{y_1m}} \\ IR_{y_2} + w_{IR_{y_2m}} \\ IR_{z_1} + w_{IR_{z_1m}} \\ IR_{z_2} + w_{IR_{z_2m}} \end{bmatrix} \quad (5-11)$$

Note that for determining IR_{x_1m}, IR_{y_1m} and IR_{z_2m} the room parameters x_{room}, y_{room} and z_{room} should be known.

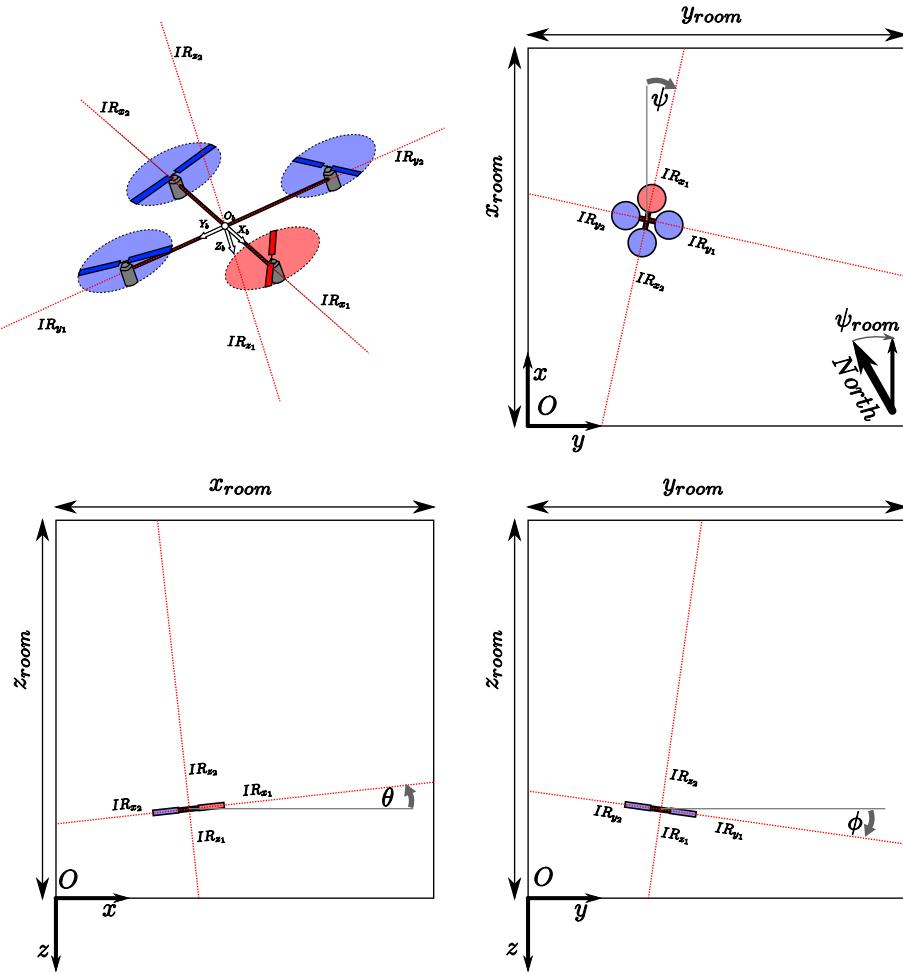


Figure 5-6: Derivation of the measurement equations of the infrared sensors

5-5 Conclusions and recommendations

In this chapter the models of the sensors are described. The models are simplified versions of the actual measurements of the sensors, some effects such as changing noise parameters, placing of the sensors with respect to the centre of gravity and calibration errors are neglected. In future work, further testing of the sensors could be done to enhance these models.

Chapter 6

Indoor navigation using state estimation

The goal of this project is to combine gyroscopes, accelerometers and infrared sensors, to get a state estimation in an indoor environment. The relationships between the measurements and states are described by non-linear equations, and therefore the Extended Kalman Filter([EKF](#)) is used to perform state estimation.

This chapter starts with explaining the theory of the Extended Kalman Filter and discusses the results in simulation of a standard AHRS Kalman filter and a full Kalman filters that combines all measurement. In the chapter [9](#) these filters are then tested on real sensor data.

6-1 Extended Kalman filter ([EKF](#))

The standard Kalman filter can be used to estimate the state $x \in \Re^n$ of a discrete-time *linear* process. For a *non-linear* process the Extended Kalman Filter([EKF](#)) is used that linearises about the current mean and covariance [[Welch and Bishop, 2001](#)].

6-1-1 Formulation

Assume the process has a state vector $x \in \Re^n$ described by the non-linear stochastic difference equation:

$$\mathbf{x}_k \approx f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_{k-1}) \quad (6-1)$$

with a measurement $z \in \Re^m$:

$$\mathbf{z}_k \approx h(\mathbf{x}_k, \mathbf{v}_k) \quad (6-2)$$

where \mathbf{w}_k and \mathbf{v}_k are the process and measurement noise respectively. There are assumed to be independent of each other, white noise and with normal probability distributions:

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}_k) \quad (6-3)$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}_k) \quad (6-4)$$

In practice the *process noise covariance* \mathbf{Q}_k and *measurement noise covariance* \mathbf{R}_k might change with each time step of measurement, but are assumed to be constant in this filter.

Because one does not know the values of this noise at each time step, it is possible to approximate the state and measurement vector:

$$\tilde{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0) \quad (6-5)$$

$$\tilde{\mathbf{z}}_k = f(\tilde{\mathbf{x}}_k, 0) \quad (6-6)$$

where $\hat{\mathbf{x}}_k$ is an *a posteriori* estimate of the state (from previous time step).

Then equations 6-1 and 6-2 can be linearised using:

$$\mathbf{x}_k \approx \tilde{\mathbf{x}}_k + \mathbf{A}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{W}\mathbf{w}_{k-1} \quad (6-7)$$

$$\mathbf{z}_k \approx \tilde{\mathbf{z}}_k + \mathbf{H}(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}\mathbf{v}_{k-1} \quad (6-8)$$

where:

- \mathbf{x}_k and \mathbf{z}_k are the actual state and measurement vectors
- \mathbf{A} is the Jacobian matrix of partial derivatives of f with respect to \mathbf{x} :

$$\mathbf{A}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0}$$

- \mathbf{W} is the Jacobian matrix of partial derivatives of f with respect to \mathbf{w} :

$$\mathbf{W}_k = \left. \frac{\partial f}{\partial \mathbf{w}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0}$$

- \mathbf{H} is the Jacobian matrix of partial derivatives of h with respect to \mathbf{x} :

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\tilde{\mathbf{x}}_k, 0}$$

- \mathbf{V} is the Jacobian matrix of partial derivatives of h with respect to \mathbf{v} :

$$\mathbf{V}_k = \left. \frac{\partial h}{\partial \mathbf{v}} \right|_{\tilde{\mathbf{x}}_k, 0}$$

Note that this Jacobian is often equal to the identity matrix.

The operation of the Kalman filter is divided into two steps: a *prediction* step and a *correction* step. For each of the steps a number of equations need to be solved, see figure 6-1. For each time step the Jacobians \mathbf{A} , \mathbf{W} , \mathbf{H} and \mathbf{V} need to be recomputed.

State prediction

In practice the state prediction from equation 6-1 is calculated using integration of the state derivate function:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0) = \mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} g(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0) \delta t \quad (6-9)$$

where $g(\hat{\mathbf{x}}_{k-1})$ are the equations that describe the derivatives of the states. Note that in the actual calculation simple Euler integration is used:

$$\hat{\mathbf{x}}_k^- \approx \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1} \Delta t \quad (6-10)$$

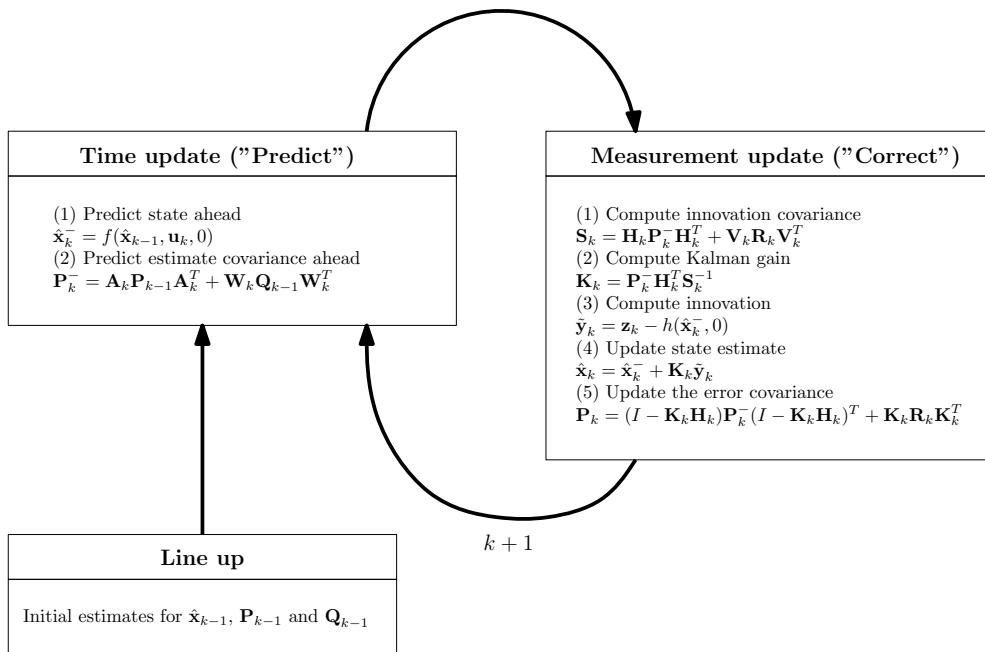


Figure 6-1: The operation of the Extended Kalman Filter

Covariance matrix of the state prediction error

The prediction of the covariance ahead is calculated with:

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \quad (6-11)$$

Because integration of the state variables is used, the equations of section 6-1-1 cannot be used, and:

$$\mathbf{A}_k = e^{\mathbf{F}_k \delta t} \quad (6-12)$$

$$\mathbf{W}_k = \left(\int \mathbf{A}_k \delta t \right) \mathbf{G}_k \quad (6-13)$$

Where \mathbf{F}_k is the Jacobian:

$$\mathbf{F}_k = \frac{\partial g}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0} \quad (6-14)$$

And \mathbf{G}_k is the noise Jacobian:

$$\mathbf{G}_k = \frac{\partial g}{\partial \mathbf{w}} \Big|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0} \quad (6-15)$$

The Jacobians \mathbf{A}_k and \mathbf{W}_k are the transform of the continuous functions \mathbf{F}_k and \mathbf{G}_k to discrete time.

6-1-2 Critism on the Extended Kalman filter

Unlike its linear counterpart, the extended Kalman filter is not an optimal estimator. In addition, if the initial estimate of the state is wrong, or if the process is modelled incorrectly, the filter may quickly diverge, owing to its linearisation. Another problem with the extended Kalman filter is that the estimated covariance matrix tends to underestimate the true covariance matrix and

therefore risks becoming inconsistent in the statistical sense without the addition of stabilising noise.

Several alternatives for this filter are possible:

- Iterated Extended Kalman filter [Chu, 2006, Section 2, page 58].
- Unscented Kalman Filter [Julier and Uhlmann, 1997],[Julier and Uhlmann, 2001, Chapter 13].
- Hybrid Kalman - Minimax filter [Simon, 2007].

In this project, in spite of the its limitations, the Extended Kalman Filter is used because it is believed that the process can be modelled correctly and initial state can be estimated.

6-2 Validation of the Kalman Filter

To validate the performance of the Kalman filter, several tools are available. Most important is the observability analysis, to analyse if the filter can diverge to a meaningful solution.

6-2-1 Observability

A linear system described by the matrices A, B, C, D in state space form:

$$x(n+1) = Ax(n) + Bu(n) \quad (6-16)$$

$$y(n) = Cx(n) + Du(n) \quad (6-17)$$

and is **observable** if a finite time $t_1 > 0$ exists such that for each admissible input function u , it follows from $y(t, x_0, u) = y(t, x_1, u)$ for all $t \in [0, t_1]$, that $x_0 = x_1$.

The observability of A, B, C and D can be completely characterized by the matrices A and C . A $np \times n$ matrix W can be defined, called the *observability matrix* as:

$$W = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \quad (6-18)$$

The following statements are equivalent:

1. The system (A, B, C, D) is observable.
2. W has rank n

Similarly for the Extended Kalman filter the observability is characterised with the observability matrix, using H and F :

$$W = \begin{pmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{pmatrix} \quad (6-19)$$

A Kalman filter built around a system with unobservable states will simply not work. By definition, an unobservable state is one about which no information may be obtained through the observation equations; in the absence of information, the filter estimate for that state will not converge on a meaningful solution.

6-2-2 Convergence of the covariances matrix of state estimation error

In order for the Kalman filter to be stable the covariances matrix of state estimation error should stabilise, changes in this matrix are a sign that the observability changes. The values on diagonal of the covariances matrix should converge to a number, divergence is a sign that the Kalman filter is unstable. The square root of the covariance have the same units as the quantity being estimated. For an perfect estimator the error of the estimation should be normally distributed with the standard deviation calculated by the square root of the covariance and with a zero mean.

6-2-3 Mean squared error

The mean squared error (MSE) quantifies the amount by which an estimator differs from the true value of the quantity being estimated. The MSE of an estimator $\hat{\theta}$ with respect to the estimated parameter θ is defined as:

$$\text{MSE}(\hat{\theta}) = \text{E}((\hat{\theta} - \theta)^2). \quad (6-20)$$

The MSE can be written as the sum of the variance and the squared bias of the estimator

$$\text{MSE}(\hat{\theta}) = \text{Var}(\hat{\theta}) + (\text{Bias}(\hat{\theta}, \theta))^2. \quad (6-21)$$

For an unbiased estimator, the MSE is the variance. The MSE thus assesses the quality of an estimator in terms of its variation and unbiasedness. The square root of MSE yields the root mean squared error or RMSE, which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance.

6-2-4 Observability degree

If a system is observable the quality of the observability can be determined with the condition number of the observability matrix. This number is calculated by first taking the singular values from the diagonal matrix S of the Singular Value Decomposition ([SVD](#)). The singular value decomposition of a matrix X produces a diagonal matrix S of the same dimension as the matrix X , with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that $X = U * S * V'$.

By dividing the highest singular value by the lowest singular value a number is calculated that gives an indication of the difference in observability of the various parameters and should be a factor 10^{10} or lower. An lower number indicates that the observability of the various parameters is more similar and thus the estimation will give better results.

6-3 Attitude and Heading Reference System (AHRS)

The standard Attitude and Heading Reference System (Attitude and Heading Reference System ([AHRS](#))) consists of a 3 axes angular rate gyroscope, a 3 axes accelerometer and a 3 axis magnetometer. In literature this combination is also known as Magnetic, Angular Rate, and Gravity ([MARG](#))-sensor. In this section the measurements of these 9 sensors will be combined using an [EKF](#), which is depicted in figure 6-2.

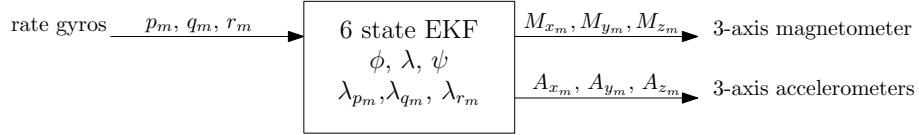


Figure 6-2: AHRS Extended Kalman Filtering setup

In this thesis Euler angles are used to calculate the translations between the reference frames. This is because the movements of the quadrotor in this project stay within the singularity boundaries and the use of Euler angles also simplifies development. Future work on the quadrotor platform could involve the use of quaternions.

6-3-1 State equations

The [EKF](#) is used to determine 6 states in the state vector \mathbf{x} , including the attitude angles and the biases of the gyroscope measurements

$$\mathbf{x} = [\phi \quad \theta \quad \psi \quad \lambda_{p_m} \quad \lambda_{q_m} \quad \lambda_{r_m}]^T \quad (6-22)$$

The derivatives of these states are calculated with:

$$\begin{aligned} \dot{\phi} &= (p_m - \lambda_{p_m} - w_{p_m}) + \sin \phi \tan \theta (q_m - \lambda_{q_m} - w_{q_m}) \\ &\quad + \cos \phi \tan \theta (r_m - \lambda_{r_m} - w_{r_m}) \end{aligned} \quad (6-23a)$$

$$\dot{\theta} = \cos \phi (q_m - \lambda_{q_m} - w_{q_m}) + \sin \phi (r_m - \lambda_{r_m} - w_{r_m}) \quad (6-23b)$$

$$\dot{\psi} = \frac{\sin \phi}{\cos \theta} (q_m - \lambda_{q_m} - w_{q_m}) + \frac{\cos \phi}{\cos \theta} (r_m - \lambda_{r_m} - w_{r_m}) \quad (6-23c)$$

$$\dot{\lambda}_{p_m} = w_{\dot{\lambda}_{p_m}} \quad (6-23d)$$

$$\dot{\lambda}_{q_m} = w_{\dot{\lambda}_{q_m}} \quad (6-23e)$$

$$\dot{\lambda}_{r_m} = w_{\dot{\lambda}_{r_m}} \quad (6-23f)$$

Where the inputs \mathbf{u} are the measurements of the rate gyroscopes:

$$\mathbf{u} = [p_m \quad q_m \quad r_m]^T \quad (6-24)$$

And the process noise vector \mathbf{w}

$$\mathbf{w} = \left[w_{p_m} \quad w_{q_m} \quad w_{r_m} \quad w_{\dot{\lambda}_{p_m}} \quad w_{\dot{\lambda}_{q_m}} \quad w_{\dot{\lambda}_{r_m}} \right]^T \quad (6-25)$$

6-3-2 Measurement equations

The measurements \mathbf{z} are from the 3 axis accelerometer and the 3 axis magnetometer:

$$\mathbf{z} = [A_{x_m} \quad A_{y_m} \quad A_{z_m} \quad M_{x_m} \quad M_{y_m} \quad M_{z_m}]^T \quad (6-26)$$

$$A_{x_m} = g \sin \theta + v_{A_{x_m}} \quad (6-27a)$$

$$A_{y_m} = -g \cos \theta \sin \phi + v_{A_{y_m}} \quad (6-27b)$$

$$A_{z_m} = -g \cos \theta \cos \phi + v_{A_{z_m}} \quad (6-27c)$$

$$M_{x_m} = \cos \theta \cos(\psi + \psi_{room}) M_{x_0} + \cos \theta \sin(\psi + \psi_{room}) M_{y_0} - \sin \theta M_{z_0} + v_{M_{x_m}} \quad (6-27d)$$

$$\begin{aligned} M_{y_m} = & (\sin \phi \sin \theta \cos(\psi + \psi_{room}) - \cos \phi \sin(\psi + \psi_{room})) M_{x_0} \\ & + (\sin \phi \sin \theta \sin(\psi + \psi_{room}) + \cos \phi \cos(\psi + \psi_{room})) M_{y_0} \\ & + \sin \phi \cos \theta M_{z_0} + v_{M_{y_m}} \end{aligned} \quad (6-27e)$$

$$\begin{aligned} M_{z_m} = & (\cos \phi \sin \theta \cos(\psi + \psi_{room}) + \sin \phi \sin(\psi + \psi_{room})) M_{x_0} \\ & + (\cos \phi \sin \theta \sin(\psi + \psi_{room}) - \sin \phi \cos(\psi + \psi_{room})) M_{y_0} \\ & + \cos \phi \cos \theta M_{z_0} + v_{M_{z_m}} \end{aligned} \quad (6-27f)$$

$$(6-27g)$$

Note that the equations of the accelerometers are simplified and the velocity and acceleration parts of the equations are removed and thus a static measurement is assumed. In real flight this is not the case, so the measurement equations are not a correct model in this case.

And the measurement noise vector \mathbf{v}

$$\mathbf{v} = [v_{A_{x_m}} \quad v_{A_{y_m}} \quad v_{A_{z_m}} \quad v_{M_{x_m}} \quad v_{M_{y_m}} \quad v_{M_{z_m}}]^T \quad (6-28)$$

6-3-3 Results

For testing of the state estimation the quadrotor is simulated to fly a path with movements in all directions.

In appendix F-1 the results of the filter using these data is shown. The filter is observable and stable, but it is clear that the tracking of the angles is not possible, due to the bias of the accelerometers. When the bias walk of both accelerometer and gyroscopes is decreased by a factor 10 as shown in appendix F-2, the filter is better able to track the angles much better.

6-3-4 Discussion

In a static case ideal accelerometers give a direct indication of the attitude. The accelerometers measure the gravitation of the earth which points downwards. But in a dynamic case onboard the quadrotor, the measurement of the accelerometer is also influenced by the acceleration and velocity of the quadrotor. Also the bias of an MEMS quadrotor is in reality not equal to zero and shows random walk. Therefore, for correct estimation of the angles of the quadrotor an additional measurement is necessary in order to estimate this bias of the accelerometer and determine position and velocity, which will be done in the next section.

6-4 Full positioning system using IR-sensors

The full positioning system using IR-sensors extends the standard Attitude and Heading Reference System (**AHRS**) with a number of IR-sensors, see [6-3](#).

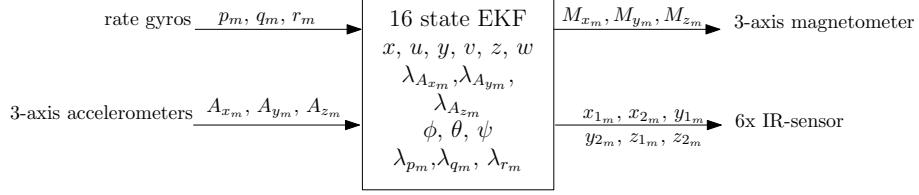


Figure 6-3: EKF Full positioning system using IR-sensors

6-4-1 State equations

The state vector \mathbf{x} includes position (in earth fixed frame), velocity (in body-fixed frame), the biases of the accelerometers, the attitude angles (in earth fixed frame) and the biases of the gyroscopes:

$$\mathbf{x} = [x \ y \ z \ u \ v \ w \ \lambda_{A_{x_m}} \ \lambda_{A_{y_m}} \ \lambda_{A_{z_m}} \ \phi \ \theta \ \psi \ \lambda_{p_m} \ \lambda_{q_m} \ \lambda_{r_m}]^T \quad (6-29)$$

The derivatives of these states are calculated with:

$$\dot{x} = (u \cos \theta + (v \sin \phi + w \cos \phi) \sin \theta) \cos \psi - (v \cos \phi - w \sin \phi) \sin \psi \quad (6-30a)$$

$$\dot{y} = (u \cos \theta + (v \sin \phi + w \cos \phi) \sin \theta) \sin \psi + (v \cos \phi - w \sin \phi) \cos \psi \quad (6-30b)$$

$$\dot{z} = -u \sin \theta + (v \sin \phi + w \cos \phi) \cos \theta \quad (6-30c)$$

$$\dot{u} = A_{x_m} - \lambda_{A_{x_m}} - w_{A_{x_m}} - (q_m - \lambda_{q_m} - w_{q_m})w + (r_m - \lambda_{r_m} - w_{r_m})v - g \sin \theta \quad (6-30d)$$

$$\dot{v} = A_{y_m} - \lambda_{A_{y_m}} - w_{A_{y_m}} - (r_m - \lambda_{r_m} - w_{r_m})u + (p_m - \lambda_{p_m} - w_{p_m})w + g \cos \theta \sin \phi \quad (6-30e)$$

$$\dot{w} = A_{z_m} - \lambda_{A_{z_m}} - w_{A_{z_m}} - (p_m - \lambda_{p_m} - w_{p_m})v + (q_m - \lambda_{q_m} - w_{q_m})u + g \cos \theta \cos \phi \quad (6-30f)$$

$$\dot{\lambda}_{A_{x_m}} = w_{\lambda_{A_{x_m}}} \quad (6-30g)$$

$$\dot{\lambda}_{A_{y_m}} = w_{\lambda_{A_{y_m}}} \quad (6-30h)$$

$$\dot{\lambda}_{A_{z_m}} = w_{\lambda_{A_{z_m}}} \quad (6-30i)$$

$$\dot{\phi} = (p_m - \lambda_{p_m} - w_{p_m}) + \sin \phi \tan \theta (q_m - \lambda_{q_m} - w_{q_m}) + \cos \phi \tan \theta (r_m - \lambda_{r_m} - w_{r_m}) \quad (6-30j)$$

$$\dot{\theta} = \cos \phi (q_m - \lambda_{q_m} - w_{q_m}) + \sin \phi (r_m - \lambda_{r_m} - w_{r_m}) \quad (6-30k)$$

$$\dot{\psi} = \frac{\sin \phi}{\cos \theta} (q_m - \lambda_{q_m} - w_{q_m}) + \frac{\cos \phi}{\cos \theta} (r_m - \lambda_{r_m} - w_{r_m}) \quad (6-30l)$$

$$\dot{\lambda}_{p_m} = w_{\lambda_{p_m}} \quad (6-30m)$$

$$\dot{\lambda}_{q_m} = w_{\lambda_{q_m}} \quad (6-30n)$$

$$\dot{\lambda}_{r_m} = w_{\lambda_{r_m}} \quad (6-30o)$$

Where the inputs \mathbf{u} are the measurements of the gyroscopes and the accelerometer:

$$\mathbf{u} = [p_m \ q_m \ r_m \ A_{x_m} \ A_{y_m} \ A_{z_m}]^T \quad (6-31)$$

And the process noise vector \mathbf{w}

$$\mathbf{w} = \begin{bmatrix} w_{p_m} & w_{q_m} & w_{r_m} & w_{\dot{\lambda}_{p_m}} & w_{\dot{\lambda}_{q_m}} & w_{\dot{\lambda}_{r_m}} & w_{A_{x_m}} & w_{A_{y_m}} & w_{A_{z_m}} & w_{\dot{\lambda}_{A_{x_m}}} & w_{\dot{\lambda}_{A_{y_m}}} & w_{\dot{\lambda}_{A_{z_m}}} \end{bmatrix}^T \quad (6-32)$$

6-4-2 Measurement equations

The measurements \mathbf{z} are from the 3 axis magnetometer and the 6 infrared sensors:

$$\mathbf{z} = [M_{x_m} \ M_{y_m} \ M_{z_m} \ IR_{x_{1m}} \ IR_{x_{2m}} \ IR_{y_{1m}} \ IR_{y_{2m}} \ IR_{z_{1m}} \ IR_{z_{2m}}]^T \quad (6-33)$$

$$M_{x_m} = \cos \theta \cos(\psi + \psi_{room}) M_{x_0} + \cos \theta \sin(\psi + \psi_{room}) M_{y_0} - \sin \theta M_{z_0} + v_{M_{x_m}} \quad (6-34a)$$

$$\begin{aligned} M_{y_m} = & (\sin \phi \sin \theta \cos(\psi + \psi_{room}) - \cos \phi \sin(\psi + \psi_{room})) M_{x_0} \\ & + (\sin \phi \sin \theta \sin(\psi + \psi_{room}) + \cos \phi \cos(\psi + \psi_{room})) M_{y_0} \\ & + \sin \phi \cos \theta M_{z_0} + v_{M_{y_m}} \end{aligned} \quad (6-34b)$$

$$\begin{aligned} M_{z_m} = & (\cos \phi \sin \theta \cos(\psi + \psi_{room}) + \sin \phi \sin(\psi + \psi_{room})) M_{x_0} \\ & + (\cos \phi \sin \theta \sin(\psi + \psi_{room}) - \sin \phi \cos(\psi + \psi_{room})) M_{y_0} \\ & + \cos \phi \cos \theta M_{z_0} + v_{M_{z_m}} \end{aligned} \quad (6-34c)$$

$$IR_{x_{1m}} = \frac{x_{room} - x}{\cos \theta \cos \psi} + v_{IR_{x_{1m}}} \quad (6-34d)$$

$$IR_{x_{2m}} = \frac{x}{\cos \theta \cos \psi} + v_{IR_{x_{2m}}} \quad (6-34e)$$

$$IR_{y_{1m}} = \frac{y_{room} - y}{\cos \phi \cos \psi} + v_{IR_{y_{1m}}} \quad (6-34f)$$

$$IR_{y_{2m}} = \frac{y}{\cos \phi \cos \psi} + v_{IR_{y_{2m}}} \quad (6-34g)$$

$$IR_{z_{1m}} = \frac{-z}{\cos \phi \cos \theta} + v_{IR_{z_{1m}}} \quad (6-34h)$$

$$IR_{z_{2m}} = \frac{z_{room} + z}{\cos \phi \cos \theta} + v_{IR_{z_{2m}}} \quad (6-34i)$$

And the measurement noise vector \mathbf{v}

$$\mathbf{v} = [v_{M_{x_m}} \ v_{M_{y_m}} \ v_{M_{z_m}} \ v_{IR_{x_{1m}}} \ v_{IR_{x_{2m}}} \ v_{IR_{y_{1m}}} \ v_{IR_{y_{2m}}} \ v_{IR_{z_{1m}}} \ v_{IR_{z_{2m}}}]^T \quad (6-35)$$

6-4-3 Results

In appendix H-1 the results are shown using the simulation data in the full filter. It is clear that the filter is stable and able to estimate the states. The estimate of the angles and gyroscope biases is much more accurate than of the AHRS filter. When the bias walk of both accelerometer and gyroscopes is decreased by a factor 10 as shown in appendix H-7, the filter is still able to track the states within a certain bound.

The condition number of the observability (or observability degree) is in the order of 10^4 , which is excellent.

6-4-4 Observability analysis - Reduction of the number of IR sensors

In practice it may be possible that not all IR sensors can be used for measurements. In the quadrotor built for test flights only 5 IR measurements are available (section 7-3-6). Therefore also simulations are done with only using 5, 4 and 3 sensors, by multiplying the \mathbf{H} matrix with an matrix $\text{diag}([\mathbf{H}_{\text{select}}])$. Simulations in appendices H-2, H-3, H-4, H-5 show that for 5 and 4 infrared sensors the filter is observable and the states can be estimated. When using four infrared sensors worst results are obtained when using 2 measurements in the Z direction. However for three infrared sensors (appendix H-6) the filter is not observable and in this way the states cannot be properly estimated, the condition number of the observability matrix is also very high (10^{17}).

6-5 Conclusions

In this chapter the Extended Kalman Filter theory is applied with the design of two different filters, an AHRS filter and a full Kalman filter using infrared sensors. It is shown in simulation that the AHRS filter is stable when estimating the attitude angles and biases of the gyroscopes when the bias walk of the accelerometers and gyroscopes is not too large. However when introducing larger bias walks on the inertial sensors the filter becomes less stable. The angles estimated in this filter are also not optimal as the measurement equation of the accelerometers are simplified.

Therefore an other filter is introduced using infrared sensors to get an position measurement. Simulation shows that this filter can correctly estimate the states and thus improves the estimation of the attitude angles. When the bias walk parameters of the accelerometers and gyroscopes are decreased it is shown that the filter is still capable of tracking the states. Furthermore it is shown that the filter is still observable when not using 6 infrared sensors but also with 5 or 4 sensors. When using only 3 infrared sensors, the filter is unobservable and correct estimation of the states cannot be performed.

6-6 Recommendations

The Extended Kalman Filter has is drawbacks as mentioned in 6-1-2, so instead of the using the Extended Kalman Filter, either the Iterated Extended Kalman Filter, the Unscented Kalman Filter or the Hybrid Kalman - Minimax filter could be evaluated.

In more practical implementations it is very hard to estimate the initial noise parameters or the noise parameters change over time, therefore evaluation of the modified recursive maximum likelihood (MRML) adaptive filter would be interesting as described in section 9-5-3.

Part IV

Design and flight tests

Chapter 7

Design of the test platform

A test platform is designed in order to teste the filter with real data.

For this project the quadrotor hardware developed at the [ESL \[de Hoop, 2007\]](#) is used as a basis. This design includes a 5 [DoF IMU](#), a compass and a pressure sensor implemented on a Silverlit X-ufo frame and allows the quadrotor to be manually controlled via a FPGA board. After the lab project for which this board was initially developed, it was found that the initial design could be improved. Together with Christophe de Wagter, one of the founders of the MAVLab and my daily supervisor, a new version of the hardware is developed, which is an improvement on the previous design and allows indoor navigation.

This chapter starts with describing the original hardware onboard the ESL quadrotor. Then the requirements of the hardware for this project and the decisions in improving the hardware are elaborated. A final conclusion is given and recommendation for future developments.

7-1 Original hardware

The original hardware onboard the ESL quadrotor included a 5 [DoF IMU](#), a compass and a pressure sensor implemented on a toy Silverlit X-ufo frame and allowed the quadrotor to be manually controlled via a FPGA board by an experienced pilot. An overview of this hardware can be seen in figure 7-1 and is listed in table 7-1.

7-2 Requirements

From results with the previous design and the requests of this research project the following requirement for the new design are set:

Full AHRS With the current sensors no complete [AHRS](#) is possible, where three axis measurements are available for accelerometers, gyroscopes and magnetometer. A measurement of the body yaw rate r and the z axis magnetic force is necessary. Note that the compass is also capable of outputting the x and y axis magnetic forces.

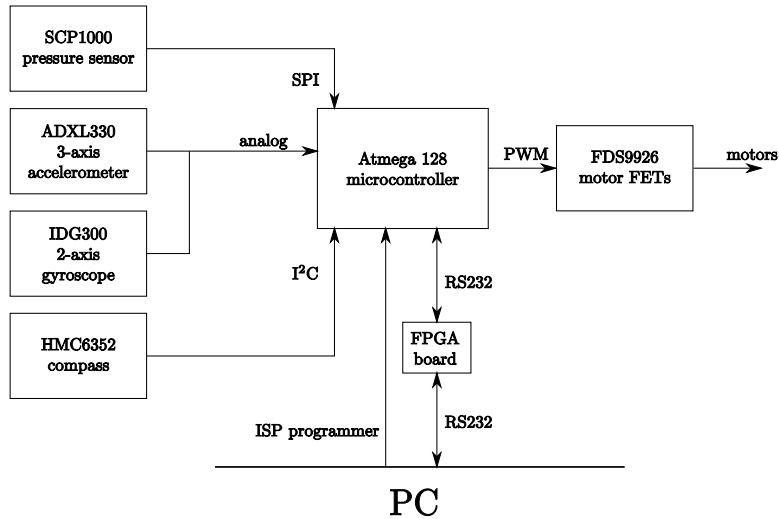


Figure 7-1: Original Hardware ESL Quadrotor

Indoor position measurement To be able to fly autonomous, a position measurement is necessary. This position measurement can then also be used to obtain full indoor state estimation.

Improvement of the engines The engines on the Silverlit X-Ufo experienced two problems.

First of all they do not provide enough lift for the old design to be equipped with a battery pack that allows a flighttime for more than 5 minutes. Also the engines on the Silverlit X-Ufo are *toy* engines, so not designed for frequent use as the lift power of the engines were found to be decreasing quickly in time.

Keep within reasonable budget Another challenging requirement is to keep costs as low as possible, as funds for this project are limited.

7-3 New hardware design

In table 7-1 the components of the new design are listed. How they are connected is shown in figure 7-2. The next subsections will explain the choice for each component.

| Part | Original | New | Measurements |
|----------------------|-----------------|------------------|--|
| Microprocessor | Atmega 128 | Atmega 2560 | - |
| 3 Axes accelerometer | ADXL 330 | ADXL 330 | $A_{x_m}, A_{y_m}, A_{z_m}$ |
| 2 Axes gyroscope | IDG 300 | IDG 300 | p_m, q_m |
| 1 Axis gyroscope | - | ADXRS 150 | r_m |
| Compass | HMC6352 | - | ψ_m or M_{x_m}, M_{y_m} |
| 3 Axes magnetometer | - | MicroMag3 | $M_{x_m}, M_{y_m}, M_{z_m}$ |
| Pressure sensor | SCP-1000 | - | P_{a_m} |
| 6x Infrared Sensor | - | Sharp GP2Y0A02YK | $IR_{x_{1m}}, IR_{x_{2m}}, IR_{y_{1m}}$ $IR_{y_{2m}}, IR_{z_{1m}}, IR_{z_{2m}}$ |
| Frame | Silverlit X-UFO | Draganfly | - |
| FETs | FDS9926 | IRLZ44NS | - |

Table 7-1: Hardware changes

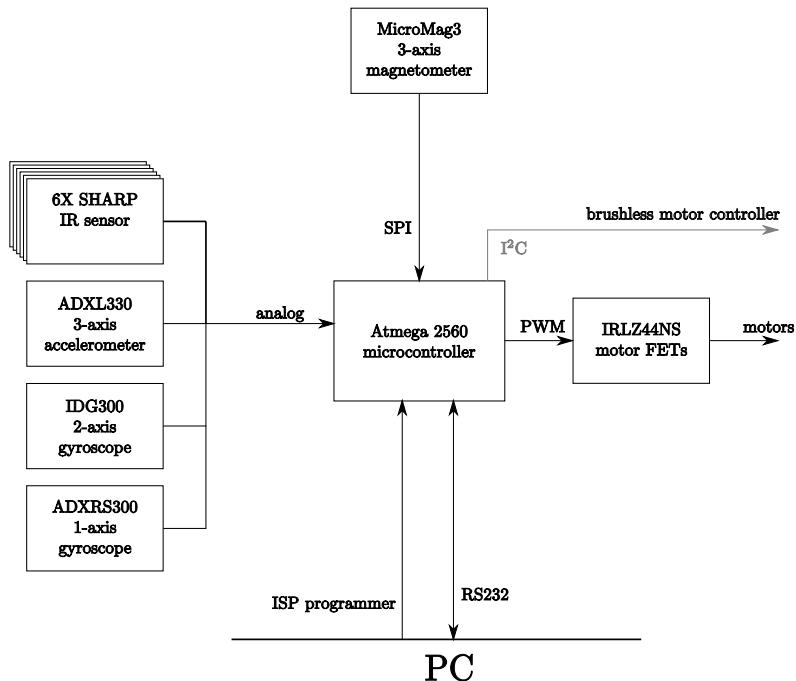


Figure 7-2: New hardware layout

7-3-1 Microprocessor: Atmega 2560

Originally the print was equipped with the Atmega 128 microprocessor, however for the implementation of more analog sensor more A/D channels were required. After tests with the Maxim BCAP+ A/D converter it was found these also induced some problems. For the easy of implementation the Atmega 2560 was chosen with 16 A/D channels.

7-3-2 3-Axis accelerometer: ADXL 330

In the original design included the 5 DoF IMU as sold by Sparkfun. This is a breakout board which includes an ADXL330 three axis accelerometer and an IDG300 dual axis gyroscope. Initially the accelerometers were set up too sensitive, and filtering needs to be applied to get proper results. Experience at the MAVLab shows it is important to perform the filtering in the correct order:

1. Mechanical filtering

Mechanical filtering filters out the direct vibrations by the engines due to the rigid connection of the parts parts, this can be done by separating the vibrating frame from the sensor with a spring-damper system. It is important to select the correct properties for this system, as too much damping results in no measurements at all, too little damping will result in measurement outside the sensor range. In the test set-up the PCB is separated from the frame with foam, which was selected from the various types of foams available at the MAVLab. This solved most sensitivity issues with the accelerometers.

2. Electronic filtering

The output of the ADXL330 has a typical bandwidth of greater than 500 Hz. The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution. The ADXL330

has provisions for band limiting the output pins. Capacitors must be added at these pins to implement low-pass filtering for anti aliasing and noise reduction. The cutoff frequency for such a filter is given by:

$$f_c = \frac{1}{2\pi RC} \text{Hz} \quad (7-1)$$

The internal resistor has a value of nominal value of $32 \text{ k}\Omega$. On the breakout board capacitors of $0.1\mu\text{F}$ are used, which results in a cutoff frequency of about 50 Hz. Tests with full engines running showed that noise reduction was not satisfactory, therefore capacitors of $0.47\mu\text{F}$ are installed to reduce the frequency to around 10 Hz, which gives proper noise reduction.

3. Software filtering

This is the final step in filtering, as described in chapter 6 and 8

After filtering the signal mechanically and electronically, the ADXL 330 could be properly used. As the combination of sensors on the IMU of Sparkfun formed a reliable, inexpensive and easy to implement solution, it was decided to keep this part in the new design.

7-3-3 2-Axis gyroscope: IDG 300

The gyroscopes in the IDG 300 are also mechanically filtered as the whole **PCB** is separated from the frame with the same foam. The electronic filtering of the gyroscopes was already mentioned in [de Hoop, 2007] and are adjusted: the capacitor on the breakout board is $0.1\mu\text{F}$ and the resistor is replaced with a 33kW resistor so the new cutoff frequency is about 48Hz.

7-3-4 1-Axis gyroscope: ADXRS 300

An extra gyroscope is added in order to have measurement in all three directions. The gyroscopes on the IDG 300 have a $500^\circ/\text{s}$ range. For the gyroscope measuring the body yaw rate this high range is not necessary, as this is a slower motion. For the new version the ADXRS 300 is used, with an $300^\circ/\text{s}$ range as it was already available at the MAVLab.

7-3-5 3-Axis magnetometer: Micromag3

Originally the compass module HMC6352 from Sparkfun was used, but some issues were found:

- Measurements had low refresh rate of 20 Hz.
- In yaw angle mode, the compass only gave correct yaw angle when pitch and roll are zero.
- Compass can be read out with separate X and Y value in sequence and also Z can be estimated to get a correct yaw angle in pitch and roll (explanation in [Cho and Park, 2003]), but still there are some issues:
 - An higher higher refresh rate is preferred: now it is only 10 Hz maximum, because of two measurements X and Y at 20 Hz.
 - The magnetometer Z axis value is an estimate and validity depends on estimated inclination angle, which is not optimal.

To solve this issues the Micromag3 from PNI was selected, with 3 axis magnetometer measurements at a selectable refresh rate. The refresh rate and resolution can be selected by adjusting the command send to the Micromag to set the division ratio of the L/R oscillator. The division ratio applied to the L/R oscillator is set to /256 resulting in a maximum sampling delay of 4.0 ms

per channel when the sensor being sampled is in a zero field, which allows the three measurements to be done within the 100Hz and having sufficient resolution.

7-3-6 Infrared sensors: 5x Sharp GP2Y0A700K0F, 1x Sharp GP2Y0A02YK

After some initial modifications, the pressure meter on the old design did measure the pressure properly, but the measurements itself had no use in indoor environments to estimate height. In ideal situations the pressure sensor can be used to measure with an precision up to of 8cm two times a second or with a precision of 20 cm nine times a second. In indoor environments the pressure is disturbed by indoor airflows such as air-conditioning.

For full state estimation also a more accurate measurement are preferred at a higher refresh rate.

Next, experiments with Maxbotic ultrasonic range finders (or sonars) were performed, also available at the MAVLab. These lightweight sensors have a 0.17-6.54 meter measurement range and readings can occur up to every 50ms (20Hz rate). The sensors are available with various beam widths [MAXIM, 2008].

Theses test showed that their measurements are disturbed by sounds of the rotor and gears, which are apparently in the same sonic range. Test show that the LV-MaxSonar®-EZ1, with a smaller beam than the LV-MaxSonar®-EZ0 has much better results, because of its narrower beam, but still the measurements needed a lot of filtering and do not allow measurements at higher refresh rates than 10Hz after filtering.

IR sensors were ultimately chosen because of its, low cost, light weight, and unlike sonar, it does not give erroneous result when it is sensing at an angle with target surface.

Initially the Sharp GP2Y0A02YK was selected for the range measurements, due to payload constraints on the EWI quadrotor. It was found out their range was too small for practical implications. Therefore the the IRLI-300 would be a good option, which is a improvement of the Sharp GP2Y0A02YK, however manufacturing of this sensor is halted. The Sharp GP2Y3A003K0F is also a good option, BUT three times as expensive as the GP2Y0A02YK.

With increased payload capacities of the new Draganfly frame it is possible to use the Sharp GP2Y0A700K0F which is much larger but has also a much larger range. In the practical application it was found that the IR measurement pointing upwards is too much disturbed by the fluorescent lamps in the testing environment, so this IR was not installed. A big drawback of the GP2Y0A700K0F is that it has an effective range of minimum 100cm. Below 100cm the measurement cannot be used, therefore an GP2Y0A02YK to measure in the range of 20-100cm. In table 7-2 the infrared sensors are compared.

The new version will be equipped with 6 infrared sensors (2 times x, y and z) for double measurements in the x and y direction and a single measurement in the z downwards direction. Note that in initial simulations the IR sensor in upward direction is also simulated.

| Brand | Type | Size (lxwxh) [mm] | Weight (est.) [g] | Range [cm] | Notes |
|-------|--------------|-------------------|-------------------|------------|-----------------|
| Sharp | GP2Y0A700K0F | 60x37x22 | 16 | 100-550 | |
| Sharp | GP2Y0A02YK | 37x22x19 | 5 | 20-150 | |
| Sharp | GP2Y3A003K0F | 53x18x20 | 9 | 40-300 | Wide angle |
| IRLI | IRLI-300 | 45x25x25 | 7 | 20-300 | No availability |

Table 7-2: Infrared sensors overview

7-3-7 Frame: Draganfly

At the start of this project it was found that the Silverlit X-Ufo frame was not sold any more and only available via the second hand market. In case of a crash, extra parts are also scarcely available. However the major problem with the Silverlit X-Ufo frame is due to the engines. These *toy* engines are very quickly deteriorating, resulting in a decrease in lift. Because engine wear down was not similar for all engines, lift decrease was also not evenly over all engines. This effected flight quality very seriously. Because a minimum lift could not be guaranteed with the engines, there was not enough headroom using this engines to increase payload. Also replacement engines were limited available, which is not an ideal situation.

The frame of the Draganfly as described in 2-4 is also separately available. It is not more expensive than the Silverlit X-ufo. The Silverlit X-Ufo frame set is not sold separately, but the package is sold for €100,- versus the standalone Draganfly frame for €140,-. The frame is designed for professional use and very robust. Parts and expansions sets are also available for separate order, so ideal for this ongoing project. The Draganfly kit is designed to carry larger payloads such as cameras, so there is enough lift power available to equip the hardware with more sensors.

On the testbed the **PCB** is separated from the frame by a piece of foam for mechanical filtering, as described in section 7-3-2.

7-3-8 FETs: IRLZ44NS

The standard engines on the Draganfly frame are also DC engines, but are for more energy consuming than the Silverlit engines. The method selected by Marc de Hoop to control the speed of DC motors is by pulse-width-modulating the supply voltage of the motors. The current that flows through the Draganfly motors is estimated to be maximum of 10A, so a Field Effect Transistor (**FET**) is selected combine fast switching with a low resistance. The FDS9926 FETs on the original design, cannot handle the high current required by the Draganfly engines.

On the original Draganfly board, the IRLZ44NS FETs are used, which were also selected for the new design. With an $R_{DS(on)}$ of 0.022mΩ at a gate-source voltage of 4.0 V, this has similar properties as the FDS9926 and can also be used with the Silverlit engines.

7-3-9 Sensor placement

From previous experiments at the MAVLAB it is known that in a Magnetic, Angular Rate, and Gravity (**MARG**) system the placement of the sensors is very important. The sensors should be placed as close as possible to the centre of gravity of the vehicle. Of course it is not possible to place all sensors in the Centre of Gravity (**CoG**), and a decision should be made which sensor is most critical. Also the **PCB** will not be placed in exactly the **CoG**. Therefore the sensors are placed in the centre of the print as good as possible. Because the accelerometers are most critical in placement, these will be placed closest to the centre point of the print.

7-3-10 Miscellaneous

Originally the quadrotor was powered with an external power cable. Because the power cable had its own magnetic field, it distorted the measurement of the magnetometer. As the frame now provides more lift it is replaced with a 1800mAh 11.1V battery.

To be able to fly the quadrotor wireless, a radio controller receiver is connected, now the quadrotor can be controlled with the radio controller Graupner MC22. On the new design an I²C

connector is available, so the board can be extended with a brushless motor controller in the future.

7-4 Results & Conclusions

A new version of the quadrotor is built implementing the hardware changes described in this chapter with all hardware and onboard software working and possible of measuring all the sensors. In the next chapter the onboard filtering and control will be elaborated, in order to make it flyable for unexperienced pilot, see figure 7-3. In appendix D the schematics of the hardware can be found. The total costs of the all the hardware is about 645 €(table 7-3) using as much inexpensive parts as possible. This quadrotor is a very inexpensive solution to test various state estimation and control strategies in practice in the future.



Figure 7-3: Flying quadrotor

| Item | Estimated costs |
|---------------------------------|-----------------|
| Sparkfun IMU 5 DoF | 90 € |
| Sparkfun gyro ADXRS300 | 55 € |
| Sparkfun magnetometer MicroMag3 | 50 € |
| 6 x Infrared sensors | 95 € |
| Draganfly frame | 175 € |
| PCB fabrication | 50 € |
| PCB components | 30 € |
| Battery 1800mAh 11.1 V | 60 € |
| Remote controller receiver | 40 € |
| Total | 645 € |

Table 7-3: Cost estimation

7-5 Recommendations

With the new design there are many new options for further development:

- The current sensors biases of the IMU are very unstable, therefore improving the IMU sensors will improve navigation accuracy and thus flight quality.
- Because of the positioning of the IR sensors, the measurements of each sensors shows interference with the other IR sensors. To overcome this problem, only switching on the IR sensor that is sampled by the ADC removes this effect. When using the current IR sensor, the GP2Y0A700K0F, an extra switch board need to be implemented as the average supply current of the sensor is high (30mA). It would be interesting to use the GP2Y3A003K0F wide angle IR sensor, which already has this switching implemented. This sensor has a smaller range, 3 meters, but then wide angle measurements possible, which could also be used for simultaneous localization and mapping of the environment, which is for example described in [Arras, 2003].
- Several sensors could be added, such as an optical flow sensor, GPS or camera's.
- Changing the engines to brushless variants to provide more thrust.
- Instead of the cable to send the data to the computer, a high bandwidth wireless link could be used. Or the data could be logged on a flash memory card.
- To provide more onboard calculation power, a better micro controller can be installed.

Chapter 8

Onboard filtering and control

In chapter 6 it is shown that with the use of three gyroscopes, three accelerometers, three magnetometers and five infrared sensors it is possible to perform full indoor state estimation. Also an control method is designed that is able to control the quadrotor using these states in simulation in 4

Initially the plan was to send the measurements from the quadrotor to a desktop computer, run the state estimation off board and send the motor commands back to the quadrotor. However test showed that this sequence of downlink, offboard processing and uplink induced a too large lag in the control. Therefore in order to do test flights also filtering and control needed to be implemented onboard the quadrotor, in such way that the motors are updated at their maximum speed. In the design of the onboard filtering and control *a quick noisier response is preferred above any delay*. Therefore onboard filtering for control is optimised for speed and not for quality.

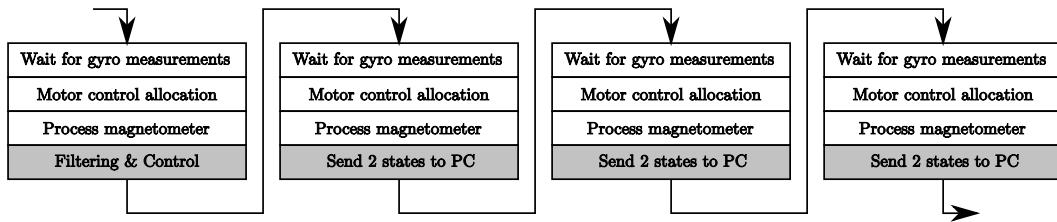
Calculation of the full fifteen state Kalman filter is not possible onboard of the quadrotor as computation power onboard the current quadrotor is limited. The quadrotor hardware is equipped with the Atmel Atmega2560 8-bit Microcontroller with 256K Bytes flash memory which is optimised for fixed point arithmetics. This core does not have floating point arithmetics built in those are emulated by software. The use of fixed point arithmetics in filtering and control is highly recommended in order to obtain maximum performance, therefore only basic filtering is implemented onboard which allows the quadrotor to be stabilised in the attitude loop in order to perform flight tests. The data from the sensors is then processed offline in a filter. In future design of the quadrotor with more available calculation power, the full filter can be applied onboard.

This chapter starts with explaining the optimised onboard processing. The next section explains the onboard fixed point Kalman filtering to calculate the attitude angles for roll and pitch and correct the gyroscopes for their biases. Next the calculation of the yaw angle is explained.

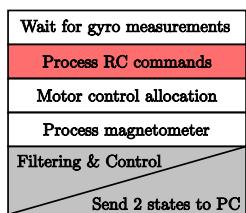
8-1 Optimised onboard processing

Onboard the quadrotor one main loop is running and several processes run within this loop, which is depicted in 8-1. Each process is triggered at a different moment so not all processes are always

At about 1526 Hz



At about 50Hz



At 100.8605 Hz

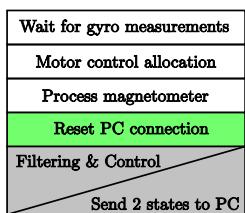


Figure 8-1: Processes that run in the main loop. The main loop runs at about 1526 Hz. One out of four loops the filtering of the signals and calculation of additional control commands is processed, effectively at 381 Hz. In each of the other three loops two measurements are send down to the PC. At about 50Hz the RC commands are processed, at 100.8605 Hz the PC connection reset is triggered.

executed in the main loop. The processes cannot take too long otherwise the main loop is not running at the optimised speed.

Main loop

The main loop waits on the measurement of the all three gyroscopes by the analog-digital converter (ADC) at about 1526 Hz (+/- 20). This frequency is not constant, as the processing of the interrupt is not equal for every time step. In this fast loop the commands of the control feedback are allocated to the motors, see also section 8-4.

| | | |
|--------------------|--------------------|--------------------|
| 1. Gyroscope X | 2. Gyroscope Y | 3. Gyroscope Z |
| 4. Accelerometer X | 5. Accelerometer Y | 6. Accelerometer Z |
| 7. Gyroscope X | 8. Gyroscope Y | 9. Gyroscope Z |
| 10. Battery | 11. Channel 1 | 12. Channel 2 |
| 13. Gyroscope X | 14. Gyroscope Y | 17. Gyroscope Z |
| 16. Channel 3 | 17. Channel 4 | 18. Channel 5 |
| 19. Gyroscope X | 20. Gyroscope Y | 21. Gyroscope Z |
| 22. Channel 6 | 23. Channel 7 | 24. Channel 8 |

Table 8-1: Read order of the analog digital converter. The channels are the free connectors where IR sensors or other sensors can be connected. After reading channel 8 a flag is set that all measurements are read. Each pack of 6 channels is sent down at about 1526 Hz = (Clock frequency / prescaler / nr. of channels / (time of one ADC cycle + average time of ADC interrupt) = $16000000/128/6/(13.5 + 0.15)$). All channels are sent down every $1526/4 = 381$ Hz.

Process radio controller commands

The radio controller Graupner MC22 sends Pulse-Position Modulation (**PPM**) signals and these are received with the RC receiver, connected to an input capture pin. Every time the input capture pin is triggered the pulse width is checked. After six correct signals a flag is set. When this flag is set the current state of the radio controller is used as input for the feedback controller to be able to control the quadrotor. New measurements in all channels are received at about every 50 Hz.

Processes magnetometers

The magnetometer is connected to the Serial Peripheral Interface (**SPI**) of the processor. Because the measurement of the magnetometer takes some time (depending of the resolution setting) the data ready pin of the magnetometer is connected to an input capture pin, that is triggered as soon as the data is ready. In this way there is no need for the main loop to wait until the measurement is ready. Each of the three magnetometer channels (M_x , M_y and M_z) is processed subsequently: first the command for the channel is sent, when the data ready pin is ready the the result is read (2 bytes) and then the next channel is read.

Filtering of the signals and calculation of additional control commands

This process is triggered when all the measurements of the ADC are read once, and runs at 381 Hz, see table 8-1. Next an fixed point Kalman filter is applied to estimate the roll and pitch angles and the biases of the gyroscopes (section 8-2) . The roll and pitch angles and the bias corrected angular velocities p and q are also send back to the feedback controller.

Downlink

When the filtering and control process is not triggered, measurements are sent to the PC in pairs of two addresses, shown in table 8-2, when all 19 measurements are send down, nothing is done until the next downlink reset. A timer clocked with the crystal runs at 100.8065 Hz which resets the downlink, to ensure a constant time step between sequential packs of data. In order to be able to sent down the all 19 addresses the original protocol [de Hoop, 2007] is modified to use a 5 bit address and a maximum data size of 16 bit. The protocol is changed to: 1aaaaadd Oddddddd Oddddddd, where **a** denotes an address bit, and **d** denotes the actual data.

8-2 Steady state fixed point Kalman filter for roll and pitch estimate

To improve handling of the quadrotor in flight tests feedback on the angles is implemented. Using for example a Butterworth (low-pass) filter on the X and Y accelerometer channels, these measurements can be filtered. Then using the following measurement equations of the accelerometer in the static case, an estimate can be derived of the roll and pitch angles:

$$A_{x_m} = g \sin \theta \quad (8-1a)$$

$$A_{y_m} = -g \cos \theta \sin \phi \quad (8-1b)$$

$$A_{z_m} = -g \cos \theta \cos \phi \quad (8-1c)$$

| Address | Data | Data format |
|---------|-------------------------------|-------------------------------|
| 0 | ADXL330 acceleration X-axis | 10 bits unsigned |
| 1 | ADXL330 acceleration Y-axis | 10 bits unsigned |
| 2 | ADXL330 acceleration Z-axis | 10 bits unsigned |
| 3 | Micromag X-axis | 16 bits 2's complement signed |
| 4 | Micromag Y-axis | 16 bits 2's complement signed |
| 5 | Micromag Z-axis | 16 bits 2's complement signed |
| 6 | Infrared measurement X-axis 1 | 10 bits unsigned |
| 7 | Infrared measurement X-axis 2 | 10 bits unsigned |
| 8 | Infrared measurement Y-axis 1 | 10 bits unsigned |
| 9 | Infrared measurement Y-axis 2 | 10 bits unsigned |
| 10 | Infrared measurement Z-axis 1 | 10 bits unsigned |
| 11 | Infrared measurement Z-axis 2 | 10 bits unsigned |
| 12 | IDG300 gyroscope X-axis | 10 bits unsigned |
| 13 | IDG300 gyroscope Y-axis | 10 bits unsigned |
| 14 | ADXRS300 gyroscope Z-axis | 10 bits unsigned |
| 15 | Battery level | 10 bits unsigned |
| 16 | Remote controller - Throttle | 16 bits unsigned |
| 17 | Remote controller - Roll | 16 bits unsigned |
| 18 | Remote controller - Pitch | 16 bits unsigned |
| 19 | Remote controller - Yaw | 16 bits unsigned |

Table 8-2: Downlink address map, the address are sent in pairs of two. The whole pack of data is sent down at 100.8065 Hz

For small angles the roll and pitch angles can be approximated:

$$\phi_m \approx -\sin\left(\frac{180}{\pi}\right) \frac{A_{y_m}}{g} \quad (8-2a)$$

$$\theta_m \approx \sin\left(\frac{180}{\pi}\right) \frac{A_{x_m}}{g} \quad (8-2b)$$

However the Butterworth filtering induces a phase lag. Therefore in the quadrotor two Kalman filters (figure 8-2) are implemented to have an minimal lag, using the measured angular rates p and q as input, the states are roll and pitch and the biases of the gyroscopes and the measurements are the accelerometers in X and Y direction using the approximated equations (equation 8-2).

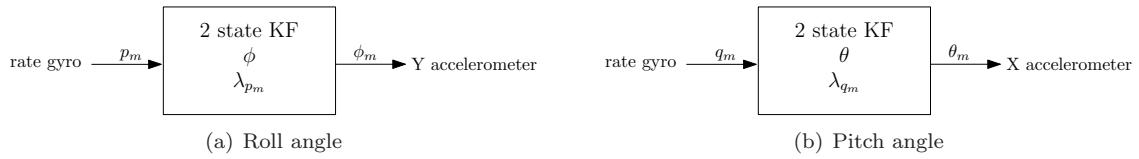


Figure 8-2: Time invariant Kalman filters

In order to reduce calculation time a simplified time invariant (steady state) Kalman filter is used with a constant gain K . The Kalman filter runs in three steps:

1. State prediction

$$\begin{pmatrix} \phi \\ \lambda_{p_m} \end{pmatrix} = \begin{pmatrix} 1 & -dt \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi \\ \lambda_{p_m} \end{pmatrix} + \begin{pmatrix} dt \\ 0 \end{pmatrix} p_m \quad (8-3)$$

2. Measurement update

$$\phi_m = C \quad A_{y_m} \quad (8-4)$$

3. State correction

$$\begin{pmatrix} \phi \\ \lambda_{p_m} \end{pmatrix} = \begin{pmatrix} \phi \\ \lambda_{p_m} \end{pmatrix} + \mathbf{K} (\phi - \phi_m) \quad (8-5)$$

where:

- dt is the timestep between the measurements. In the actual calculation onboard this is set to 1, increasing the resolution of the estimation of ϕ by scaling it with a factor $1/dt$.
- p_m is the raw input from the gyroscope, already corrected for the ADC bias using a static measurement;
- A_{y_m} is the raw input from the accelerometers, already corrected for the ADC bias using a static measurement;
- K is a 2x1 matrices containing the fixed Kalman gains. The Kalman gains are tuned by trial and error: $K = [-1/4001/4000000]^T$
- C is the constant that defines the linearised relationship between the measurements of the accelerometers and the angles. $C = -100$;

Using the same equations θ and λ_q are estimated, replacing ϕ , ϕ_m , p_m , λ_{p_m} and A_{y_m} with θ , θ_m , q_m , λ_{q_m} and A_{x_m} using the same dt and K and using $C = 100$. The estimated angles are bounded in order to overcome overflow of the integrator. Using this simplified Kalman filter induces some errors:

- The body angular rates from the body-fixed frame are integrated to the angles in earth-fixed frame and not transformed to the earth-fixed frame before integration. Only for small angles this approximation is possible.
- The measurement equations are linearised version for small angles of the actual relationship between the accelerometer and attitude angles in steady state.
- The timestep of the loop is not equal for each loop, so the integration step is not constant.

8-3 Yaw estimation

Onboard the quadrotor the r angular rate is measured by a better gyroscope, the ADXRS300. Tests showed that this gyroscope has a much smaller bias and therefore bias estimation of this gyroscope is less vital to the flight performance. Also the yaw angle can be estimated accurately with the magnetometer.

The yaw angle can be calculated with [Cho and Park, 2003]:

$$\psi = \tan^{-1} \left(\frac{-Y_H}{X_H} \right) \quad (8-6)$$

| | |
|-----------------------------------|--------|
| P_p | 10 |
| P_q | 10 |
| P_r | 5 |
| P_θ | 20/128 |
| P_ϕ | 20/128 |
| P_ψ | 5 |
| $P_{\text{RC}_{\text{Throttle}}}$ | 1/6 |
| $P_{\text{RC}_{\text{Roll}}}$ | -1/8 |
| $P_{\text{RC}_{\text{Pitch}}}$ | -1/8 |
| $P_{\text{RC}_{\text{Yaw}}}$ | 1/8 |

Table 8-3: Gains of the implemented feedback controller

Where:

$$Y_H = M_{y_m} \cos \phi - M_{z_m} \sin \phi \quad (8-7)$$

$$X_H = M_{x_m} \cos \theta + M_{y_m} \sin \theta \sin \phi + M_{z_m} \sin \theta \cos \phi \quad (8-8)$$

For small angles this can be simplified to:

$$\psi = \tan^{-1} \left(\frac{-M_{y_m}}{M_{x_m}} \right) \quad (8-9)$$

This calculation of the arctan is done onboard with a lookup table, in order to maximize speed.

8-4 Feedback controller

The control commands to the motors are calculated in two steps. First in equation 8-10 the commands U for throttle, roll pitch and yaw are calculated.

$$\begin{aligned} U_{\text{Throttle}} &= P_{\text{RC}_{\text{Throttle}}} \text{RC}_{\text{Throttle}} + P_z(z - z_d) + P_{\dot{z}}\dot{z} \\ U_{\text{Roll}} &= P_{\text{RC}_{\text{Roll}}} \text{RC}_{\text{Roll}} + P_\phi(\phi - \phi_d) + P_{pp} \\ U_{\text{Pitch}} &= P_{\text{RC}_{\text{Pitch}}} \text{RC}_{\text{Pitch}} + P_\theta(\theta - \theta_d) + P_{qq} \\ U_{\text{Yaw}} &= P_{\text{RC}_{\text{Yaw}}} \text{RC}_{\text{Yaw}} + P_\psi(\psi - \psi_d) + P_{rr} \end{aligned} \quad (8-10)$$

The part $P_z(z - z_d) + P_{\dot{z}}\dot{z}$ in the throttle command is the height feedback. Horizontal position control can be done by adjusting the desired roll ϕ_d and pitch θ_d . These are calculated with:

$$\phi_d = P_y(y - y_d) + P_y\dot{y} \quad (8-11)$$

$$\theta_d = P_x(x - x_d) + P_x\dot{x} \quad (8-12)$$

Horizontal position and height feedback is not implemented at the moment. The desired yaw angle ψ_d is set to zero. The gains used onboard are summarized in table 8-3. In the next step the motor commands are calculated:

$$\begin{aligned} U_{\text{motor}_1} &= U_{\text{Throttle}} + U_{\text{Pitch}} - U_{\text{Yaw}} \\ U_{\text{motor}_2} &= U_{\text{Throttle}} - U_{\text{Roll}} + U_{\text{Yaw}} \\ U_{\text{motor}_3} &= U_{\text{Throttle}} - U_{\text{Pitch}} - U_{\text{Yaw}} \\ U_{\text{motor}_4} &= U_{\text{Throttle}} + U_{\text{Roll}} + U_{\text{Yaw}} \end{aligned} \quad (8-13)$$

The actual relationship between the motor command and the forces generated by the rotor is non-linear. In [Martinez, 2007] this relationship is measured for the Draganfly motor-rotor system, and also described in section 3-8 and 4-3. For the motor settings in the flight envelope of hover and near-hover this relationship is assumed linear.

8-5 Results

In figures 8-3 and 8-4 two logs show the input of a experienced pilot in pitch, roll and yaw from the remote control when flying the quadrotor using different controllers. The pilot was given the task to hold the quadrotor on a fixed horizontal position with constant thrust setting. When attitude feedback is enabled the bias of the accelerometer causes an error in the estimation of the pitch and roll angles, which need to be corrected with a constant value by the pilot. This can also be done by using the trim settings on the remote controller.

It is clearly visible that enabling only the rate feedback causes the inputs of the pilot to be much larger than when the attitude feedback is also enabled, also when taking a look in the frequency domain. For the pilot, when only rate feedback is enabled, the task is experienced as constantly correcting the attitude angle. When attitude feedback is enabled, the pilot experiences the task as actual position control. Also, there is no need to control the yaw angle.

With attitude feedback enabled, the quadrotor can be hit on one of the motorhubs in flight and will automatically stabilize attitude after the disturbance.

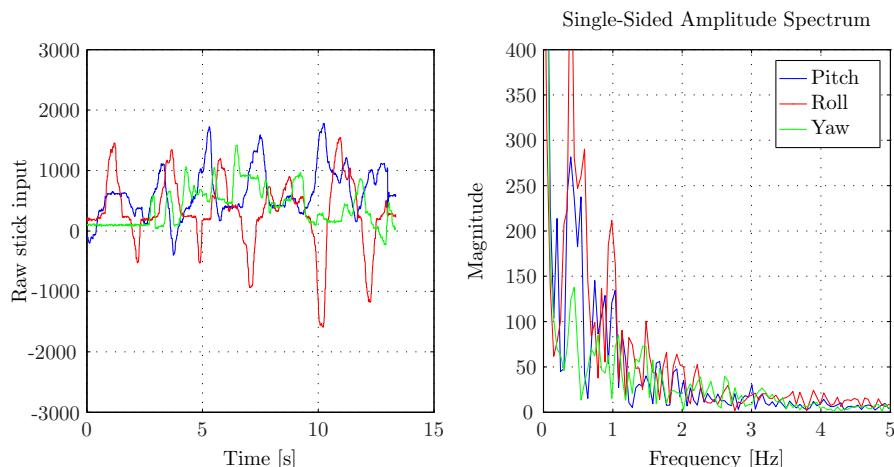


Figure 8-3: Control activity with only rate feedback

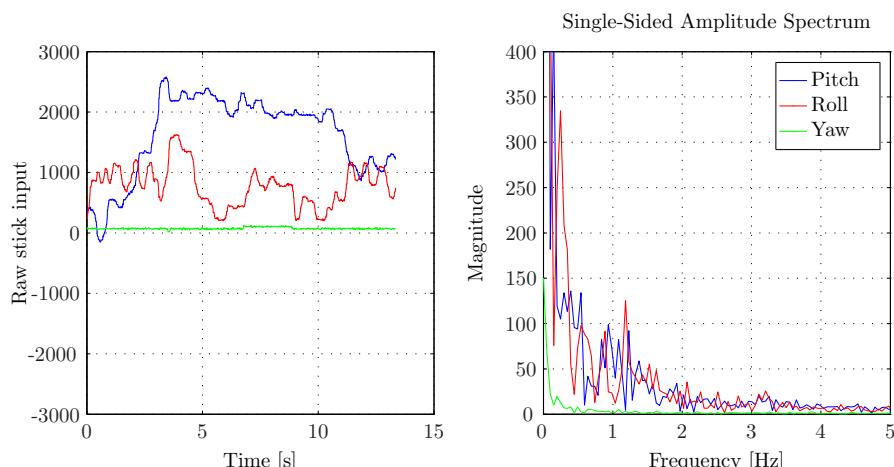


Figure 8-4: Control activity with rate and angle feedback

8-6 Conclusions

To minimise the delay between the measurements of the sensors and the feedback to the controller the motor commands are not calculated offboard on a desktop computer, but filtering and control is implemented onboard the quadrotor. In this way the can be used to perform test flights. Onboard the quadrotor the roll and pitch angles and the biases of the p and q gyroscopes are estimated using a fixed point steady state Kalman filter. The yaw angle is estimated using the X and Y magnetometers and a lookup table.

Because the onboard filtering is optimised for performance and not for quality, the controller is designed to work only for small angles. Flight test show that the control activity of a pilot is reduced significantly when applying rate feedback.

8-7 Recommendations

The calculation power onboard the quadrotor is limited, but in future the following could be implemented.

- Perform position estimation from the IR-sensors and derive the velocities of the vehicle. Next, implement position feedback in the controller.
- Implement onboard correction of the measured values of the magnetometers in order to get a better estimate of the yaw angle with changing throttle settings, see also [9-1-3](#);
- After the correction of the magnetometer for the throttle effect, a Kalman filter can also be implemented similar to the one described for the roll and pitch angles, in order to improve the estimation of the yaw angle.

Chapter 9

Indoor navigation with real data

In this chapter the AHRS Kalman filter and the full Kalman filter from chapter 6 are tested with real data from the quadrotor. First the calibration and preprocessing of the sensor measurements is explained, then the AHRS Kalman filter is tested, following with full Kalman filter.

9-1 Calibration and Preprocessing

The sensors data of the quadrotor is sent to the computer at 100.8065 Hz and logged. This data is binary format and should be converted to the appropriate units in order to be able to use the data in the filter. For the magnetometers and infrared sensors additional preprocessing is necessary in order to be able to use the data. All calibration parameters can be found in appendix C. An example of raw data can be found in appendix C-1, the result after preprocessing and calibration of these measurements is shown in C-2.

9-1-1 Gyroscopes

The measurements of the gyroscopes are in binary format and need to be converted to radians per second. In order to do this, first the bias is removed, which is measured at a stationary measurement. This result is then scaled to get radians per second.

9-1-2 Accelerometers

The measurements of the accelerometers are in binary format and need to be converted to meters per square seconds. Similar to the gyroscope, first the bias is removed, which is measured previously at a stationary measurement. This result is then scaled to get meters per square seconds.

9-1-3 Magnetometers

The magnetometers measures the vector of the gravity field, so the magnetometer does not need to be scaled and no biased need to be removed. Due to use of the larger motors in the Draganfly frame in the final design, an increased effect of the motors on the magnetometer measurement

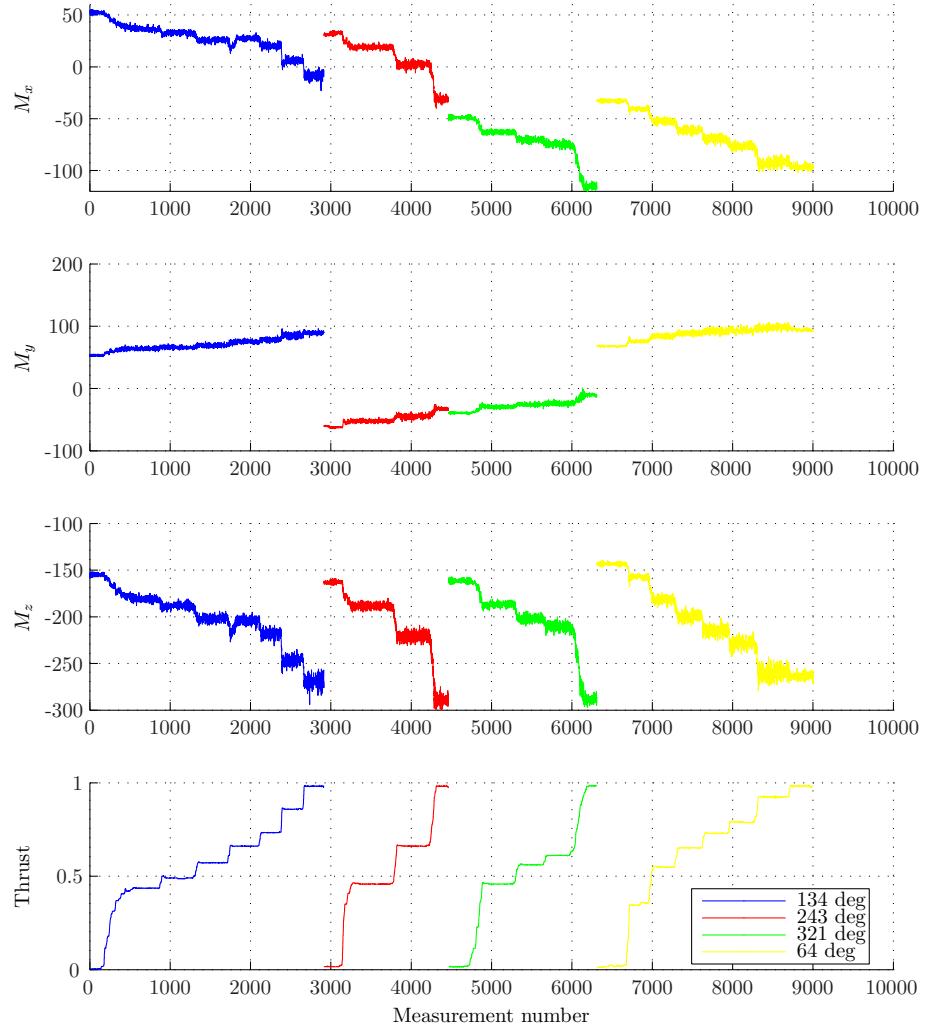


Figure 9-1: Comparison of the output of the accelerometer under 4 different yaw angles with varying throttle settings.

was noticed. The measurements should be corrected for this effect.

In figure 9-2 and 9-1 the effect under four different yaw angles is shown with varying throttle settings. The difference of the measurement in M_x , M_y , M_z without increasing throttle is shown in 9-3. For each of the measurements M_x , M_y , M_z an second order polynomial is fitted. This polynomial is used to correct the measurements for the effect of the motors. In the test the motors were supposed to have equal motor speeds. With roll, pitch and yaw commands the motors this will not be the case. Because the variations of motors speeds due to the commands are small this effect is neglected.

After the correction of the magnetometer, the measured vector is normalised for each timestep to overcome problems with spikes in the measurements.

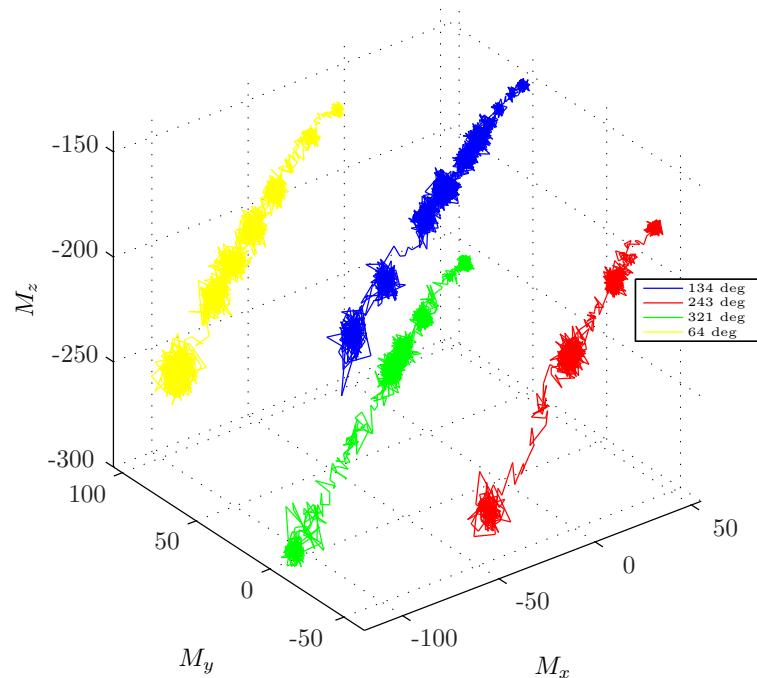


Figure 9-2: 3D - Comparison of the output of the accelerometer under 4 different yaw angles with varying throttle settings

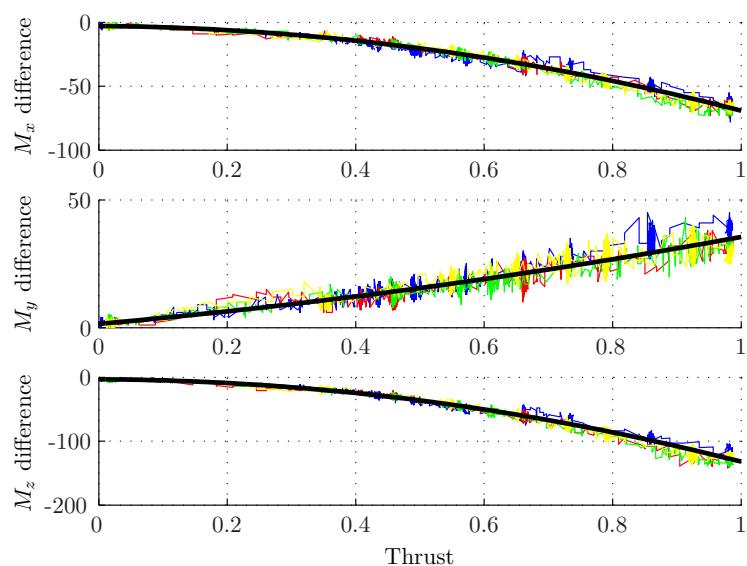


Figure 9-3: Polynomial estimation of the difference in M_x , M_y , M_z due to the setting of the throttle.

9-1-4 Infrared sensors

The measurements of the infrared sensors are disturbed by several factors, as described in 5-4. Not all of the factors can be ignored before using the calibrated infrared signal as input for the Kalman filter, as without the filtering the noise characteristics of the infrared measurement are not ideal and are biased.

In order to remove the measurement interference with the other infrared sensors, for each sample the maximum of the last n measurements is used. After this still unwanted spikes are noticed in the filtered signal, which can be a result of the averaging or difference between the ADC and sensor update rate. In order to overcome those effects a median filter over m number of samples is used.

This preprocessed signal is then calibrated, assuming a linear relationship between the inverse of the measurement and the measurement in meters.

$$IR_m = \frac{1}{a \times IR_{\text{preprocessed}}} + b \quad (9-1)$$

where a and b are coefficients depending on the IR sensor and found in appendix C.

The measurements of two IR sensors in Z-direction, both pointing downwards, are fused in one measurement. For the range below 1 meter, the measurement of the GP2Y0A02YK are taken, above 1 meter those of the GP2Y0A0700. It is assumed that the IR sensors are placed on the axis of the body fixed frame and they are corrected with an estimated distance of the IR over the axis of the body fixed frame to the centre of gravity.

9-2 Results of the AHRS Kalman filter

First the AHRS Kalman filter is tested with three sets of real data. A subsequent movement in roll, pitch and yaw with and without motors on. Next a real flight test is done in testroom.

9-2-1 Roll, pitch, yaw movement - with and without motors

In appendices G-1 and G-2 the results can be seen for the AHRS Kalman filter on real sensor data with and without motors on performing subsequently a roll, pitch and yaw move by hand. The \mathbf{Q} and \mathbf{R} matrices are tuned by hand. It can be clearly seen that in both cases the filter is stable, thus the P-matrix converges to one value and that the measurements can be tracked properly.

9-2-2 Flight test

For the test of the Kalman filters in free space, flight test are performed by an experienced pilot in a prepared testroom (see figure 9-4). The task of the pilot is to keep the quadrotor at an fixed horizontal position. Therefore the pitch and roll angles should move around zero. Specifications of the room can be found in appendix C.

In G-3 the results of the AHRS Kalman filter using data from a flight is shown. The filter is stable (the P-matrix converges) but it is clearly visible that the estimation of the attitude angles, especially the pitch angle θ are biased.



Figure 9-4: Testroom setup

9-3 Results of the full Kalman filter

First the quadrotor is hanged in a pendulum construction and an initial deflection in y direction is given with motors off. The result is shown in appendix I-1. Using a camera fixed on the ceiling the motion of the quadrotor is traced. The traced motion is not accurate enough to be used as an exact reference signal, but gives an indication of the motion. After tuning the Q and R matrices, the full Kalman filter is able to track the motion which shows high correlation to the camera tracing. However the pitch and yaw angle show a bias with the expected values. Next a similar experiment is done with motors at 40% power, just not enough to lift the quadrotor, shown in appendix I-2, again the motion itself is traced correctly, however the yaw angle is not around zero. The experiment video shows that this is also the case during the test. This video is however not accurate enough to trace the yaw angle.

In another test a movement of the quadrotor in the testroom is done by hand, results of this test can be found in appendix I-3. The quadrotor is moved in a square at two different heights. In I-17 this pattern can be clearly recognised. During the test the attitude angles were kept as level as possible. However the estimation of the roll angle ϕ shows an bias of about 5 degrees.

Next the data from the flight test as described in 9-2-2 is filtered with the full Kalman filter using measurements of the IR sensors in five directions. Again the **Q** and **R** matrices are tuned by hand. The results are shown in appendix H. Most standard deviations of the state estimation error show clear convergence, the standard deviations of pitch and yaw do also not diverge but do change over time, which indicate a changed observability over time. After about 15 seconds all the measurements are properly tracked. In I-22 the 3D track after filtering is shown.

9-4 Conclusions

In this chapter the filtering algorithms as derived in 6 are tested on real data. For the AHRS filter is shown that this filter is able to correctly track roll, pitch and yaw movements with and without motors on. In real flight however the filter is disturbed by the bias of the accelerometers.

Therefore it is necessary to estimate the bias of accelerometers as well. This is tested using the full Kalman filter on the real flight data.

The test shows that the full Kalman filter is stable and capable of estimating the biases and tracking the measurements. However in tests using real data, some of the states were not estimated as expected. There could be several reasons for this error:

- First of all in the test with the pendulum motion, an ideal situation was expected, where only the y , v , \dot{v} , ϕ should change. In the test with the pendulum motion, it was tried to get this ideal motion, but off course in reality this motion is not ideal.
- The sensors onboard the quadrotor could be not aligned perfectly, in the simulation this is assumed, however the specifications of MEMS gyroscopes and accelerometers show that there the misalignment internally possible in the order of one degrees. Next to that the placing of the sensors on the printed circuit board (PCB) could not perfect. Also the PCB is tightened on the frame with rubber bands on a piece of foam (for mechanical damping), in order to be able to easily detach the PCB. This does not guarantee that the sensors are placed on exactly the correct axis. It is also assumed that the infrared sensors are placed on one of the axis of the body axis frame, however first of all their position is not exactly on this axis. Next to that it could be that they are not pointing exactly along this axis.
- Placement of accelerometers is assumed in simulation to be in the centre of gravity, on the real quadrotor this is not possible due to placing limitations, however in the state estimation algorithm this is assumed.
- The sensors are send down at 100.8065 Hz. Onboard the quadrotor it is not possible to read the sensors all at the same time, due to the working of the analog-digital converter. So the measurements processed are not of exactly at the same timestep.
- The preprocessing of the IR sensors was necessary in order to get the state estimation working, as the noise of the unprocessed signal was not gaussian. However this preprocessing does introduce a lag in the measurement and could have other non-beneficial effects.
- In the measurements of the magnetometers it was noticed that the motors create their own magnetic field and therefore the measurement should be corrected for this bias. The effect of settings in roll, pitch and yaw were ignored, but differences in the motor speeds due to these settings could also be of an effect.
- The accelerometers and gyroscopes random walk is in reality not an perfect random walk, but is bounded.
- The calibration of the sensors could not be perfect. This could be enhanced for different situations, such as different temperature and vibrations.
- The sensor board damping is tuned by hand. It was found that the signal on the accelerometers and gyroscopes is still very noisy due to the engines. Improved damping could be used to filter out the noise of the engines even more.
- In each of the test the Q and R matrices are tuned by hand, for optimal tracking these need to be tuned even more.

9-5 Recommendations

It is proven that the state estimation filter stabilise in practice and gives a state estimation, but further validation of the state estimation is necessary with a accurate reference states. Next to

that improvements in sensors and calibration are possible. Because initial and noise parameters of the sensors onboard the quadrotor are not well known, a modified recursive maximum likelihood (MRML) adaptive filter can help in determining these parameters.

9-5-1 Validation

Due to time limitations it was not possible to accurately validate the results of the Kalman filtering with an accurate reference state. Now the \mathbf{Q} and \mathbf{R} matrices are tuned by hand. A reference state can be used to not only validate the results but also optimise the \mathbf{Q} and \mathbf{R} matrices to get the best state estimate. Also it can show other if there are other sources of errors. Two accurate measurement methods that could be used to determine the exact movements of the quadrotor:

- Using several infrared cameras in the room to track at least three infrared LEDs attached to the frame of the quadrotor, in order to estimate position and attitude.
- Using a camera onboard the quadrotor, as described in [Lagarde, 2008] to track the state of the quadrotor looking at one or more checkerboard patterns and using the vehicle model. It is also possible to do this without the vehicle model, with the GML MatLab Camera Calibration Toolbox [Vezhnevets, 2008].

9-5-2 Sensor improvement

The quality of the state estimation depends obviously a lot on the quality of the sensors. For now very inexpensive gyroscopes were used in p and q direction, the IDG 300. In r direction the ADXRS300 was used, which is more expensive, but as test has shown has a more stable bias. It would be advisable to use this sensor (or similar, but with even a higher sensitivity such as the ADXRS 150 or ADXRS 401) also for p and q directions. To improve the state estimation it is also advisable to use also accelerometers with a more stable bias. The calibration of the sensors can also be improved. For example it is shown that the magnetometer also measures the magnetic field induced by the motors, this effect can be investigated further. Recently at the department of Control and Simulation a test facility is build to calibrate gyroscopes and accelerometers very accurately. This can be used to make the calibrations of these sensors better. Also the damping of the sensor board could be improved.

9-5-3 Algorithm improvement

Next to better sensors and calibration, the state estimation algorithm can also be improved. A problem in applying the EKF is that not all necessary a priori information may be known, such as the measurement noise intensities of the sensors, as in the case with the quadrotor. Moreover, the EKF is only a first-order minimum variance filter. The higher-order terms neglected in propagation of the error covariances may lead to biased estimates, as noticed in the simulations.

An alternative approach is to compute a likelihood function from the prediction errors generated by a nonaugmented EKF and estimate all unknown parameters such as noise intensities, biases, and scale factors by optimizing this likelihood function with respect to these parameters. This leads to the so-called iterative maximum likelihood method in which the state trajectory is estimated in conjunction with these unknown parameters. Among many schemes, the recursive maximum likelihood (RML) adaptive filter is a typical and well-known example of a general recursive prediction error identification algorithm. The RML algorithm for adaptive filtering is, however, sensitive to initialization errors of the unknown system parameters. Divergence may occur at large values of these initialization errors. In [Chu et al., 1996] a modified recursive maximum likelihood (MRML) adaptive filter for nonlinear state-parameter estimation problems

is developed. To improve the robustness of the RML adaptive filter to parameter initialization errors, the new algorithm revises the conventional RML adaptive filter by including the effect of the parameter estimator in the prediction error vector computation. The results indicate that the MRML adaptive filter, produces estimates that are both more accurate and less sensitive to parameter initialisation errors than those obtained with the conventional RML adaptive filter.

Next to the initialisation errors encountered with the quadrotor sensors the noise intensities are not constant. The noise intensity of the gyroscopes, accelerometers and magnetometers change with changing thrust and the noise intensity of the infrared sensors change with the distance measured. In [Bennis et al., 1999] it is shown that the MRML adaptive filter can also be applied to estimate non-stationary noise intensities. Because of the fact that for the MRML adaptive filter the effect of the parameter estimator is included in the prediction error computation, the possibility exists to introduce time-varying noise statistical uncertainties as additional stochastic variables instead of constant parameters. From numerical experiments it could be seen that the original MRML adaptive filter is not able to give a satisfactory estimation of non-stationary noise statistical parameters. After introduction of noise statistical uncertainties as random walks instead of constant parameters, the MRML adaptive filter is capable of tracking the time-varying noise intensity.

Drawback of the MRML adaptive filter is that these require increased computations with respect to the standard EKF filter. Therefore for application of this filter in control this probably induces to large lag. However for the application in online model parameter estimation it is advised to optimise the state estimation and this optimisation can be applicable.

Part V

Conclusions

Chapter 10

Conclusions

10-1 Conclusions

In this thesis we investigated the feasibility of a filtering approach for autonomous indoor flight with a quadrotor. From variety of simulations and experiments, the following main conclusions can be drawn.

- The selected controller structure, a PID controller for each of the states with a special nested PID controller for the position control is able to successfully stabilise the quadrotor in simulation and using interpolated waypoint commands a smooth trajectory can be flown.
- The gyroscopes, accelerometers, magnetometers and IR sensors can be described by non-linear equations when the parameters of the width, length, height and heading with respect to the magnetic field are known (x_{room} and y_{room} , z_{room} and ψ_{room}).
- The AHRS Kalman filter can estimate the bias of the gyroscopes correctly, but cannot estimate the attitude angles correctly because of the bias of the accelerometers.
- The full Kalman filter designed for indoor navigation using 6 infrared sensors provides a good estimation of the states in simulation using realistic values for the noise standard deviation and can determine the biases of the gyroscopes and accelerometers.
- Observability of the full Kalman filter is guaranteed for systems with the use of a minimum of 4 sensors in any direction. Clear degradation of the result is found when using 4 instead of 5 or 6 measurements, however the system is still observable. With the use of 3 infrared sensors the Kalman filter is unobservable and will not converge to a meaningful solution.
- The observability degree of the Kalman filter using 5 or 6 measurements is very good, in the order of 10^3 .
- Using the work of the Embedded Systems Lab as a basis, a quadrotor UAV is built with three gyroscopes, three accelerometers and three magnetometers and six infrared sensors to test the indoor navigation with data from real sensors.
- Onboard the fixed point processor of the quadrotor filtering and control is implemented, which stabilises the attitude angles of the quadrotor. The attitude feedback allows even inexperienced pilots to fly the quadrotor.

- Actual data from measurements in a test environment show that the state estimation algorithm is stable and able to give an estimates of the states.

10-2 Discussion

The tests with the real data show some unexpected results, therefore some points of discussion on the conclusions:

- Further verification of the state estimation method is required, comparing the results with other already proven state estimation methods, such as 3D camera tracking.
- The calibration and preprocessing of the measurement data is vital for the results. The current calibrations and preprocessing could be improved to enhance results. Also damping of the sensor board could be improved to minimize the noise due to the motors.
- Exact alignment of the sensors should be validated. The assumptions that the accelerometers are placed in the centre of gravity and the IR sensors along the axis of the body frame can influence the result, so their positioning can be implemented in the filter.
- In the Extended Kalman Filter the nonlinear models are linearised every step. However linearisation can produce unexpected results, so an alternative estimator algorithm could be implemented, such as the Iterated Extended Kalman Filter, the Unscented Kalman Filter or the Hybrid Kalman - Minimax filter. In these practical implementations it is very hard to estimate the initial noise parameters or the noise parameters change over time, therefore evaluation of the modified recursive maximum likelihood (MRML) adaptive filter would be interesting, which can estimate these parameters online.
- Drawback of the proposed state estimation algorithm is that the room parameters (x_{room} , y_{room} , z_{room} and ψ_{room}) should be specified in advance. However from a static ground measurement, positioning the quadrotor perpendicular to a wall, these values can be estimated before startup.

10-3 Future work

The quadrotor is a very inexpensive solution to test various state estimation and control strategies in practice, and therefore numerous options for future work are available:

- The development of the quadrotor UAV can continue by improving the current sensors used and adding extra hardware, such as camera's, wireless links and GPS receivers.
- Onboard the quadrotor a position controller can be implemented for autonomous flight. Also instead of the current PID control, other more advanced controllers can be tested in reality, with in mind that the quadrotor has only limited onboard calculation capacity.
- Using the state estimation, an online parameter estimation could be developed, which then can be fed back to for example an NDI controller.
- When combining the infrared sensors with other sensors, such as optical flow sensors, an indoor flying Simultaneous Localization And Mapping (SLAM) system could be developed, able to autonomously fly in known and unknown indoor environments.
- Because the developed quadrotor is very inexpensive but successfull platform several quadrotors could be used in research in UAV swarm behaviour.

Part VI

Appendices

Bibliography

- [Altuğ, 2003] Altuğ, E. (2003). *Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, quadrotor*. PhD thesis, GRASP Lab, University of Pennsylvania.
- [Altuğ et al., 2002] Altuğ, E., Ostrowski, J. P., and Mahony, R. (2002). Control of a quadrotor helicopter using visual feedback. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, volume 1, pages 72–77, Washington, DC. IEEE.
- [Altuğ et al., 2005] Altuğ, E., Ostrowski, J. P., and Taylor, C. J. (2005). Control of a quadrotor helicopter using dual camera visual feedback. *International Journal of Robotics Research*, 24(5):329 – 341.
- [Arras, 2003] Arras, K. O. (2003). *Feature-based robot navigation in known and unknown environments*. PhD thesis, EPFL.
- [Benallegue et al., 2006] Benallegue, A., Mokhtari, A., and Fridman, L. (2006). Feedback linearization and high order sliding mode observer for a quadrotor UAV. In *Proceedings of the 2006 International Workshop on Variable Structure Systems*, Alghero, Italy.
- [Benallegue et al., 2007] Benallegue, A., Mokhtari, A., and Fridman, L. (2007). High-order sliding-mode observer for a quadrotor uav. *International Journal of Robust and Nonlinear Control*.
- [Bennis et al., 1999] Bennis, R., Chu, Q., and Mulder, J. (1999). Non-stationary noise estimation with adaptive filters. In *AIAA Guidance, Navigation, and Control Conference*.
- [Bouabdallah, 2007] Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying*. PhD thesis, EPFL.
- [Bouabdallah et al., 2004a] Bouabdallah, S., Murrieri, P., and Siegwart, R. (2004a). Design and control of an indoor micro quadrotor. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA.
- [Bouabdallah et al., 2005] Bouabdallah, S., Murrieri, P., and Siegwart, R. (2005). Towards autonomous indoor micro VTOL. In *Autonomous Robots*, volume 18, pages 171–183. Springer Science + Business Media, Inc.
- [Bouabdallah et al., 2004b] Bouabdallah, S., Noth, A., and Siegwart, R. (2004b). PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 2451–2456.

- [Bouabdallah and Siegwart, 2005] Bouabdallah, S. and Siegwart, R. (2005). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain.
- [Bouabdallah and Siegwart, 2006] Bouabdallah, S. and Siegwart, R. (2006). Towards intelligent miniature flying robots.
- [Bouadi et al., 2007] Bouadi, H., Bouchoucha, M., and Tadjine, M. (2007). Sliding mode control based on backstepping approach for an UAV type-quadrotor.
- [Brigham Young University, 2008] Brigham Young University (2008). Quadrotor hallway following using optic flow sensors. <http://www.et.byu.edu/groups/magicc/cmsmadesimple>.
- [Buskey et al., 2003] Buskey, G., Roberts, J., Corke, P., and Wyeth, G. (2003). Helicopter automation using a low-cost sensing system. In *Australasian Conference on Robotics and Automation*.
- [Castillo et al., 2004] Castillo, P., Dzul, A., and Lozano, R. (2004). Real-time stabilization and tracking of a four-rotor mini rotorcraft. *IEEE Transactions on Control Systems Technology*, 12(4).
- [Castillo et al., 2005] Castillo, P., Lozano, R., and Dzul, A. (2005). Stabilization of a mini rotorcraft with four rotors. *IEEE Control and Systems Magazine*.
- [Cheeseman and Bennett, 1957] Cheeseman, I. and Bennett, W. (1957). The effect of the ground on a helicopter rotor in forward flight. *Aeronautical Research Council*, 3021.
- [Cho and Park, 2003] Cho, S. Y. and Park, C. G. (2003). Tilt compensation algorithm for 2-axis magnetic compass. *ELECTRONICS LETTERS*, 39(22).
- [Chu, 2006] Chu, Q. (2006). Lecture notes - Modern flight test technologies and system identification. Technical report, Delft University of Technology.
- [Chu et al., 1996] Chu, Q. P., Mulder, J. A., and van Woerkom, P. T. L. M. (1996). Modified recursive maximum likelihood adaptive filter for nonlinear aircraft flight-path reconstruction. *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS*, 19(6):1285–1295.
- [de Hoop, 2007] de Hoop, M. (2007). Development of an embedded system for quad-motor aerial vehicles - research assignment report. Technical report, Software Engineering Research Group, Department of Software Technology, Faculty EEMCS, Delft University of Technology.
- [Dunfied et al., 2004] Dunfied, J., Tarbouchi, M., and Labonte, G. (2004). Neural network based control of a four rotor helicopter. In *IEEE International Conference on Industrial Technology (ICIT)*.
- [Epiney et al., 2006] Epiney, J., Siegwart, R., Bouabdallah, S., and Scaramuzza, D. (2006). Quadrotor helicopter hovering with vision. Master's thesis, EPFL.
- [Escareno et al., 2006] Escareno, J., Salazar-Cruz, S., and Lozano, R. (2006). Embedded control of a four-rotor UAV. In *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA.
- [Fay, 2001] Fay, G. (2001). Derivation of the aerodynamic forces for the mesicopter simulation. Technical report, EPFL.
- [Fowers et al., 2007] Fowers, S. G., Lee, D.-J., Tippetts, B. J., Lillywhite, K. D., Dennis, A. W., and Archibald, J. K. (2007). Vision aided stabilization and the development of a quad-rotor micro uav. *International Symposium on Computational Intelligence in Robotics and Automation*, pages 143–148.

- [Franklin and Emami-Naeini, 2006] Franklin, P. and Emami-Naeini (2006). *Feedback control of Dynamic Systems*. Pearson Prentice Hall, New Jersey.
- [Guenard et al., 2005] Guenard, N., Hamel, T., and Moreau, V. (2005). Dynamic modeling and intuitive control strategy for an X4-flyer. In *ICCA 05 Budapest*.
- [Hoffmann et al., 2004] Hoffmann, G., Rajnarayan, D., Waslander, S., Dostal, D., Jang, J., and Tomlin, C. (2004). The stanford testbed of autonomous rotorcraft for multi agent control (starmac). *Proceedings of the 23rd Digital Avionics Systems Conference*, 2:12.E.4– 121–10.
- [Hoffmann et al., 2007] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J. (2007). Quadrotor helicopter flight dynamics and control theory and experiment.
- [James, 2008] James, D. (2008). Two different approaches to integrated mems.
- [Julier and Uhlmann, 1997] Julier, S. and Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *International Symposium of Aerospace/Defense Sensing, Simulation and Controls*.
- [Julier and Uhlmann, 2001] Julier, S. and Uhlmann, J. K. (2001). *Handbook of Multisensor Data Fusion*. CRC Press.
- [Kemp, 2006] Kemp, C. (2006). *Visual Control of a Miniature Quad-Rotor Helicopter*. PhD thesis, University of Cambridge.
- [Kroo and Prinz, 2001a] Kroo, I. and Prinz, F. (2001a). The mesicopter: A meso-scale flight vehicle - NIAC Phase I final report. Technical report, Stanford University.
- [Kroo and Prinz, 2001b] Kroo, I. and Prinz, F. (2001b). The mesicopter: A meso-scale flight vehicle - NIAC Phase II technical proposal. Technical report, Stanford University.
- [Lagarde, 2008] Lagarde, R. (2008). Vision-based control and localization for MAVs - a complete framework for autonomous control. Master's thesis, Delft University of Technology.
- [Madani and Benallegue, 2006a] Madani, T. and Benallegue, A. (2006a). Backstepping control for a quadrotor helicopter. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China.
- [Madani and Benallegue, 2006b] Madani, T. and Benallegue, A. (2006b). Control of a quadrotor mini-helicopter via full state backstepping technique. In *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA.
- [Martinez, 2007] Martinez, V. (2007). Modelling of the flight dynamics of a quadrotor helicopter. Master's thesis, CRANFIELD UNIVERSITY.
- [MAXIM, 2008] MAXIM (2008). Ultrasonic rangefinders feature custom beam width.
- [McKerrow, 2004] McKerrow, P. (2004). Modelling the draganflyer four-rotor helicopter. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA.
- [Mokhtari and Benallegue, 2004] Mokhtari, A. and Benallegue, A. (2004). Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 3, pages 359–2366, New Orleans, LA. IEEE.
- [Mokhtari et al., 2005] Mokhtari, A., Benallegue, A., and Daachi, B. (2005). Robust feedback linearization and $GH\infty$ controller for a quadrotor unmanned aerial vehicle.

- [Mokhtari et al., 2006] Mokhtari, A., M'Sirdi, N. K., Meghriche, K., and Belaidi, A. (2006). Feed-back linearization and linear observer for a quadrotor unmanned aerial vehicle. *Advanced Robotics*, 20(1):71–91.
- [Mulder, 2006] Mulder, J. (2006). Flight dynamics - Lecture Notes ae3-302. Technical report, TU Delft.
- [Ng et al., 2004] Ng, A., Kim, H., Jordan, M., and Sastry, S. (2004). Autonomous helicopter flight via reinforcement learning. *Advances in Neural Information Processing Systems*.
- [Petersen et al., 2008] Petersen, C. F., Hansen, H., Larsson, S., Madsen, L. B. T., and Rimesstad, M. (2008). Autonomous hovering with a quadrotor helicopter. Technical report, Aalborg Universitet.
- [Pounds et al., 2006] Pounds, P., Mahony, R., and Corke, P. (2006). Modelling and control of a quad-rotor robot. In *In Proceedings of the Australasian Conference on Robotics and Automation*, pages 27–29.
- [Pounds et al., 2004] Pounds, P., Mahony, R., Gresham, J., Corke, P., and Roberts, J. (2004). Towards dynamically-favourable quad-rotor aerial robots. In *Proceedings of the 2004 Australasian Conference on Robotics*.
- [Pounds et al., 2002] Pounds, P., Mahony, R., Hynes, P., and Roberts, J. (2002). Design of a four-rotor aerial robot. In *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, Auckland.
- [Prouty, 2002] Prouty, R. (2002). *Helicopter Performance, Stability and Control*. Krieger Publishing Company. reprint with additions, original edition 1986.
- [Roberts et al., 2007] Roberts, J. F., Stirling, T. S., Zufferey, J.-C., and Floreano, D. (2007). Quadrotor using minimal sensing for autonomous indoor flight. In *3rd US-European Competition and Workshop on Micro Air Vehicle Systems (MAV07) & European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, Toulouse, France.
- [Salazar-Cruz et al., 2005] Salazar-Cruz, S., Palomino, A., and Lozano, R. (2005). Trajectory tracking for a four rotor mini-aircraft. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain.
- [Simon, 2007] Simon, D. (2007). Hybrid Kalman - Minimax filtering. <http://www.innovatia.com/software/papers/hybrid.htm>.
- [Tayebi and McGilvray, 2006] Tayebi, A. and McGilvray, S. (2006). Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571.
- [Thus et al., 2007] Thus, S., Boon, P., Diepeveen, N., Helderweirt, A., Kuiper, B., Moerland, E., Scherders, R., and van Staveren, R. (2007). The Insight - Design Synthesis 2007. Technical report, TU Delft.
- [Tournier et al., 2006] Tournier, G. P., Valenti, M., How, J. P., and Feron, E. (2006). Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
- [van der Merwe, 2004] van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering.
- [Vezhnevets, 2008] Vezhnevets, V. (2008). Gml matlab camera calibration toolbox. <http://research.graphicon.ru/calibration/gml-matlab-camera-calibration-toolbox.html>.

- [Waslander et al., 2005] Waslander, S. L., Hoffmann, G. M., Jang, J. S., and Tomlin, C. J. (2005). Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In *In Proceedings 2005 IEEE/RSJ International Conference Intelligent Robots and Systems*, page 37123717.
- [Welch and Bishop, 2001] Welch, G. and Bishop, G. (2001). An introduction to the Kalman filter.
- [Woodman, 2007] Woodman, O. J. (2007). An introduction to inertial navigation. Technical Report 696, University of Cambridge - Computer Laboratory.
- [Xu and Özgüner, 2006] Xu, R. and Özgüner, U. (2006). Sliding mode control of a quadrotor helicopter. In *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA.

Appendix A

Simulation parameters

A-1 Quadrotor

| | |
|---|--|
| Sampling period [s] | $dt = 0.01$ |
| Gravity constant [m/s^2] | $g = 9.806$ |
| Air density [kg/m^3] | $\rho = 1.293$ |
| Air viscosity at 20 degrees Celsius [$Pa \cdot s$] | $\nu = 1.8 \cdot 10^{-5}$ |
| Number of propellers | $P = 4$ |
| Arm length [m] | $L = 0.232$ |
| Volume [m^3] | $Vol = 3.04 \cdot 10^{-4}$ |
| Total mass [kg] | $m = 0.53$ |
| Vertical distance between CoG and propellers plan [m] | $h = 0.058$ |
| Thrust factor in hover [Ns^2] | $b = 3.13 \cdot 10^{-5}$ |
| Drag factor in hover [Nms^2] | $d = 7.5 \cdot 10^{-7}$ |
| Inertia components [kgm^2] | $I_{xx} = 6.228 \cdot 10^{-3}$ $I_{yy} = 6.228 \cdot 10^{-3}$ $I_{zz} = 1.121 \cdot 10^{-2}$ |
| Reduction ratio | $r = 4$ |
| Rotor inertia [kgm^2] | $J_r = 6.0100 \cdot 10^{-5}$ |
| Slope of the linear curve $\Omega = f(\text{bin})$ | $a_\Omega = 2.7542$ |
| Shift of the linear curve $\Omega = f(\text{bin})$ | $b_\Omega = 3.627$ |
| Number of blades per propeller | $N = 2$ |
| Propeller radius [m] | $R = 0.15$ |
| Propeller disk area [m^2] | $A = \pi \cdot R^2 = 0.0707$ |
| Chord [m] | $c = 0.0394$ |
| Pitch of incidence [rad] | $\theta_0 = 0.2618$ |
| Twist pitch [rad] | $\theta_{tw} = 0.045$ |
| Solidity ratio (rotor fill ratio) [rad^{-1}] | $\sigma = \frac{N \cdot c}{\pi \cdot R} = 0.1672$ |
| Lift slope | $a = 5.7$ |
| Airfoil drag coefficient | $C_d = 0.052$ |
| Helicopter center hub area [m^2] | $A_c = 0.005$ |
| Weight of the quadrotor per propeller [N] | $W_{prop} = \frac{mg}{P} = 1.2993$ |
| Propeller speed at hover [rad/s] | $\Omega_H = \sqrt{\frac{W_{prop}}{b}} = 203.7425$ |
| Longitudinal drag coefficient | $C_x = 1.32$ $C_y = 1.32$ $C_z = 1.32$ |

A-2 Sensors

Noise parameters are estimated from real measurement data, using worst case scenarios. The bias is set to 1 bit of data, which is realistic for an initial error. The random walk variable is set to 10 bit, which results in realistic random walk.

Gyroscope noise standard deviation [rad/s]

$$\begin{aligned}\sigma_{p_m} &= 0.0687 \\ \sigma_{q_m} &= 0.0486 \\ \sigma_{r_m} &= 0.0515\end{aligned}$$

Gyroscope bias [rad/s]

$$\begin{aligned}\lambda_{p_m} &= (720/457) * (\pi/180) \\ \lambda_{q_m} &= (720/437) * (\pi/180) \\ \lambda_{r_m} &= (720/696) * (\pi/180)\end{aligned}$$

Gyroscope random walk bias [rad/s²]

$$\begin{aligned}\sigma_{\lambda_{p_m}} &= 10 * 720 / 457 * (\pi/180) \\ \sigma_{\lambda_{q_m}} &= 10 * 720 / 437 * (\pi/180) \\ \sigma_{\lambda_{r_m}} &= 10 * 720 / 696 * (\pi/180)\end{aligned}$$

Accelerometers noise standard deviation [m/s²]

$$\begin{aligned}\sigma_{Ax_m} &= 1.5247 \\ \sigma_{Ay_m} &= 1.5376 \\ \sigma_{Az_m} &= 1.0\end{aligned}$$

Accelerometers bias [m/s²]

$$\begin{aligned}\lambda_{Ax_m} &= 9.81/100 \\ \lambda_{Ay_m} &= 9.81/100 \\ \lambda_{Az_m} &= 9.81/100\end{aligned}$$

Gyroscope random walk bias [rad/s²]

$$\begin{aligned}\sigma_{\lambda_{Ax_m}} &= 9.81/10 \\ \sigma_{\lambda_{Ay_m}} &= 9.81/10 \\ \sigma_{\lambda_{Az_m}} &= 9.81/10\end{aligned}$$

Magnetometer settings

$$\begin{aligned}\text{Declination} &= 60 \text{ degrees} \\ \text{Inclination} &= 0 \text{ degrees}\end{aligned}$$

Magnetometer noise standard deviation []

$$\begin{aligned}\sigma_{Mx_m} &= 0.0056 \\ \sigma_{My_m} &= 0.0068 \\ \sigma_{Mz_m} &= 0.0024\end{aligned}$$

Room parameters

$$\begin{aligned}x_{room} &= 4 \text{ [m]} \\ y_{room} &= 4 \text{ [m]} \\ z_{room} &= 4 \text{ [m]} \\ \psi_{room} &= 0 \text{ [deg]}\end{aligned}$$

Infrared noise standard deviation [m]

$$\begin{aligned}\sigma_{IRx1_m} &= 0.1 \\ \sigma_{IRx2_m} &= 0.1 \\ \sigma_{IRy1_m} &= 0.1 \\ \sigma_{IRy2_m} &= 0.1 \\ \sigma_{IRz1_m} &= 0.1 \\ \sigma_{IRz2_m} &= 0.1\end{aligned}$$

Appendix B

Simulink Model

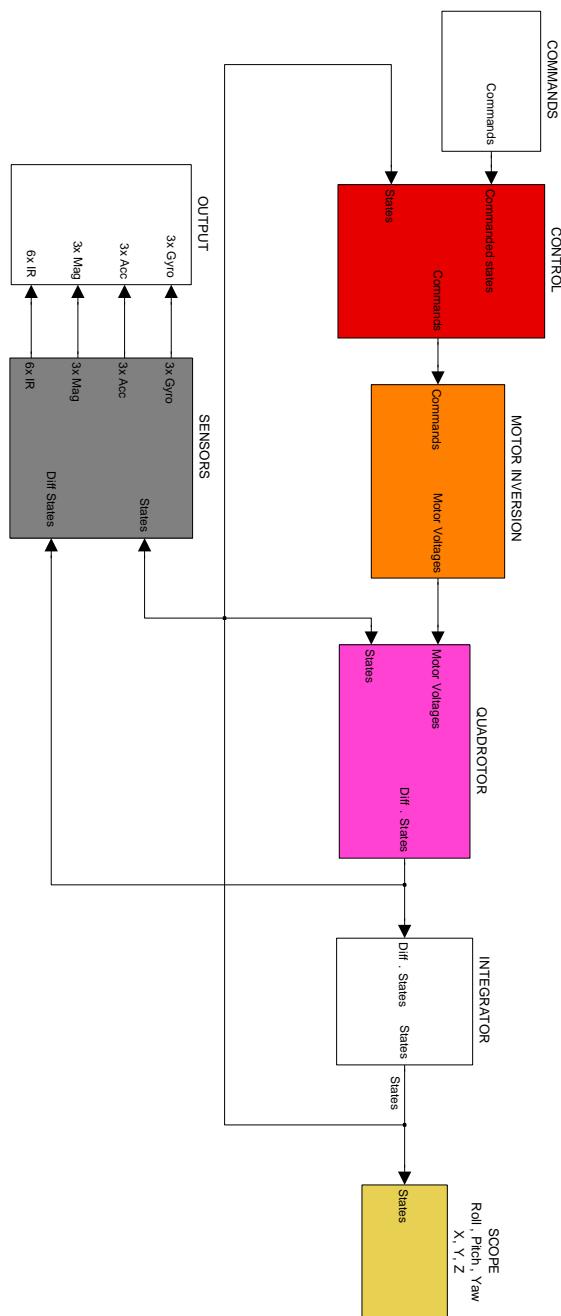


Figure B-1: Simulink model - The implementation of the dynamical model of the quadrotor in Simulink. This set up allows to set initial states and state commands as input. Optionally a joystick block can be connected to give commanded states. The motor inversion block is the implementation of equation 3-44. Instead of the scope with the states a 3D view of the quadrotor can be selected as output. The sensor and filtering blocks allow testing of several state estimation techniques

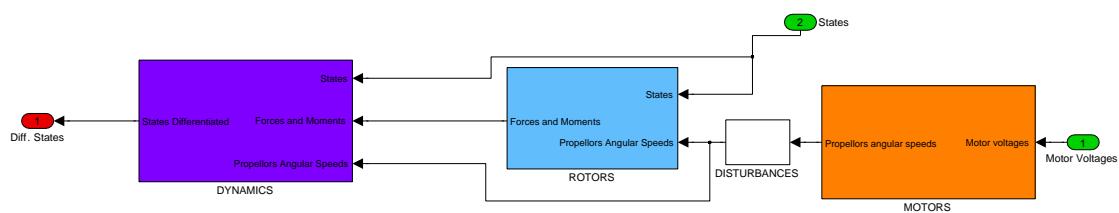


Figure B-2: Simulink model - The quadrotor UAV block consists of three parts. First the motor block, where the angular speeds of the rotors are calculated from the input voltages. Next block calculates the aerodynamic forces and moments on the rotors. In the dynamics block the dynamic equations of the quadrotor are calculated

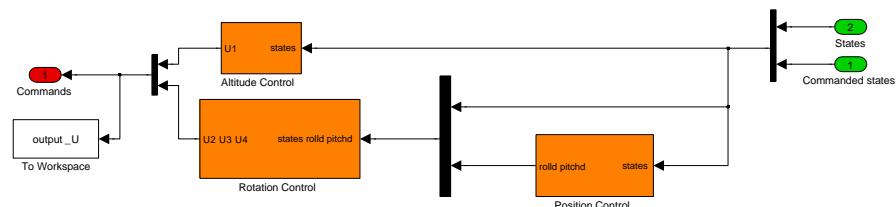


Figure B-3: Simulink model - The control block with the nested PID control approach, the position control sends demanded pitch and roll angles to the rotations control. All control laws are written in Matlab functions.

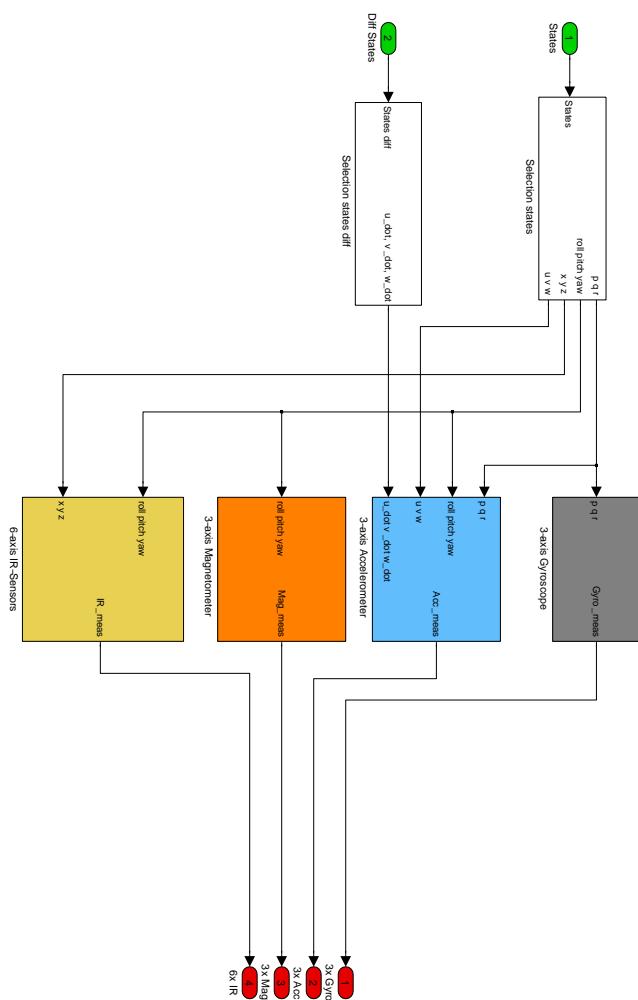


Figure B-4: Simulink model - In the sensor block the measurements of the gyroscopes, accelerometers and magnetometers are generated.

Appendix C

Calibration parameters

Calibration is done in the lab using known motions and/or positions of the sensors.

| | | |
|---|---|--|
| Gyro X calibration factor | = | $(720 * \pi) / (457 * 180)$ [rad/step] |
| Gyro Y calibration factor | = | $(720 * \pi) / (437 * 180)$ [rad/step] |
| Gyro Z calibration factor | = | $(720 * \pi) / (696 * 180)$ [rad/step] |
| Accelerometer X calibration factor | = | 9.8065/100 [m/s ² /step] |
| Accelerometer Y calibration factor | = | 9.8065/100 [m/s ² /step] |
| Accelerometer Z calibration factor | = | 9.8065/100 [m/s ² /step] |
| Magnetometer X thrust polynomial coefficients | = | -62.58 |
| | = | -3.63 |
| | = | -2.90 |
| Magnetometer Y thrust polynomial coefficients | = | 12.09 |
| | = | 21.90 |
| | = | 1.52 |
| Magnetometer Z thrust polynomial coefficients | = | -125.71 |
| | = | -3.07 |
| | = | -3.30 |
| GP2Y0A02YK coefficient a | = | 207.62 |
| GP2Y0A02YK coefficient b | = | -0.0382 |
| GP2Y0A0700 coefficient a | = | 905.65 |
| GP2Y0A0700 coefficient b | = | -0.015 |
| X position horizontal IR sensors | = | 0.04 [m] |
| Y position horizontal IR sensors | = | 0.04 [m] |
| Z position vertical IR sensors | = | 0.01 [m] |
| Test room x_{room} | = | 4.05 [m] |
| Test room y_{room} | = | 3.82 [m] |
| Test room z_{room} | = | 2.56 [m] |
| Test room ψ_{room} | = | 87 [deg] |

C-1 Raw measurements

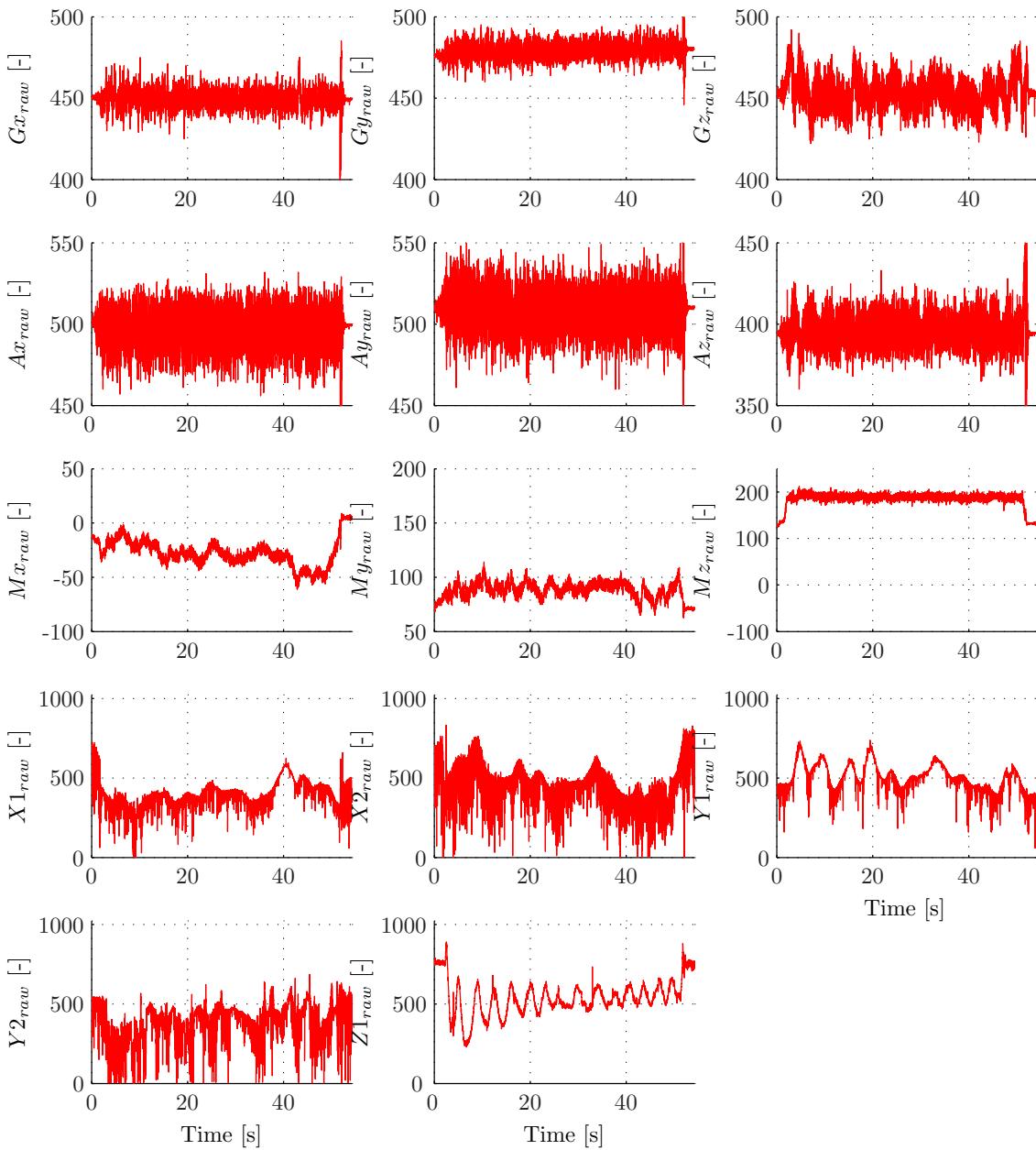


Figure C-1: Raw measurements

C-2 Preprocessed and calibrated measurements

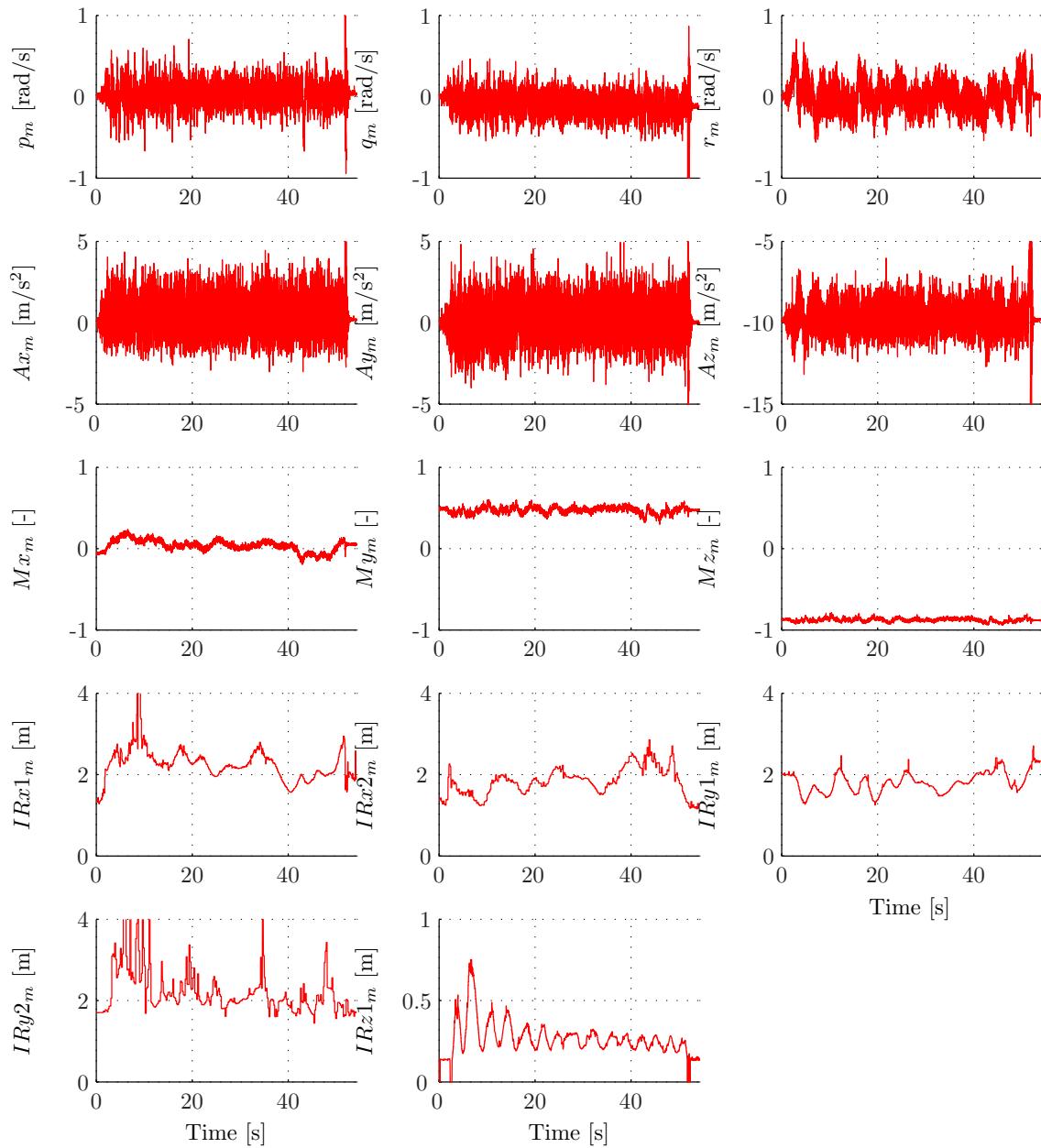


Figure C-2: Preprocessed and calibrated measurements

Appendix D

Schematics

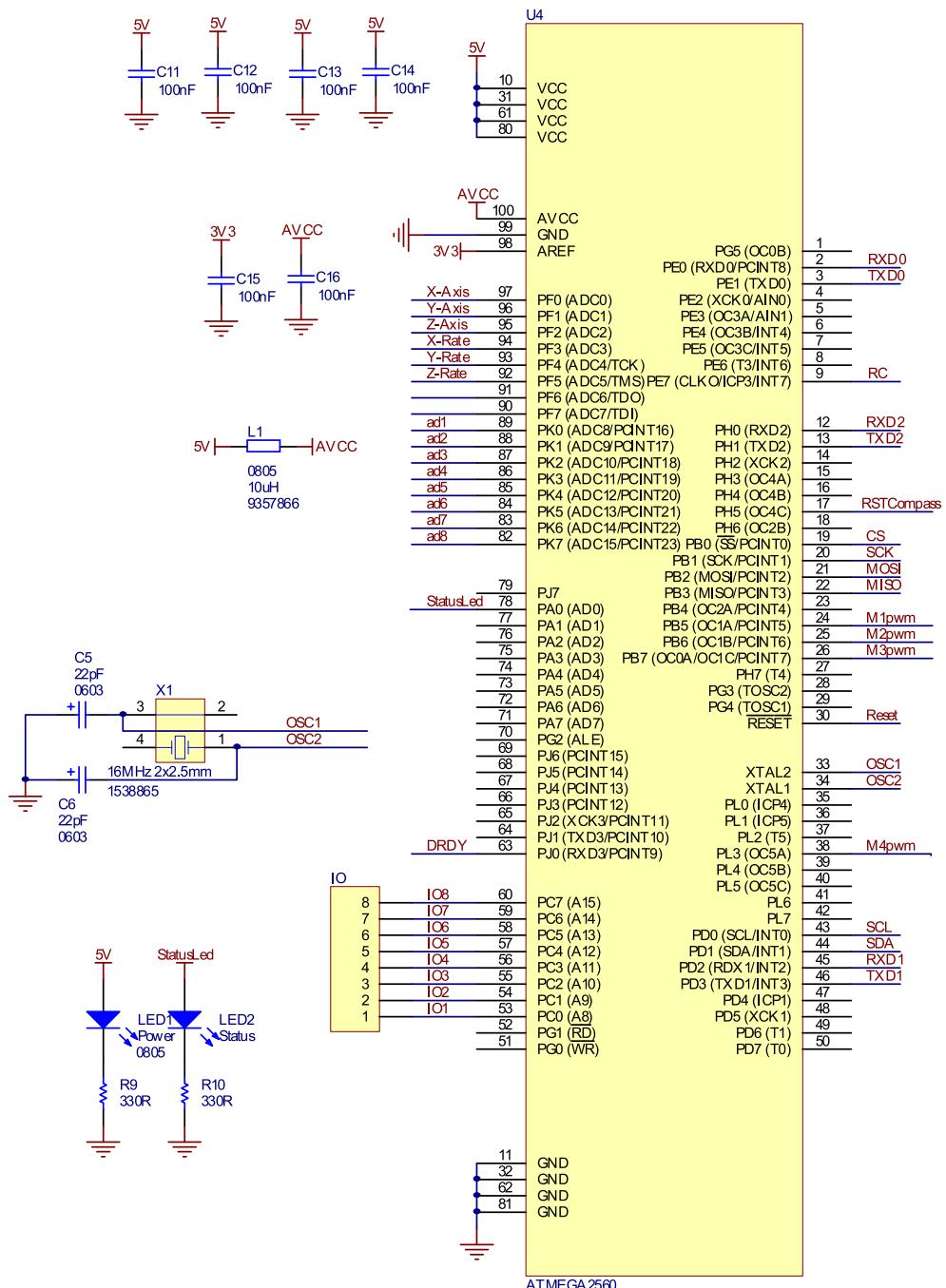


Figure D-1: Schematics of the connections of the Atmega 2560

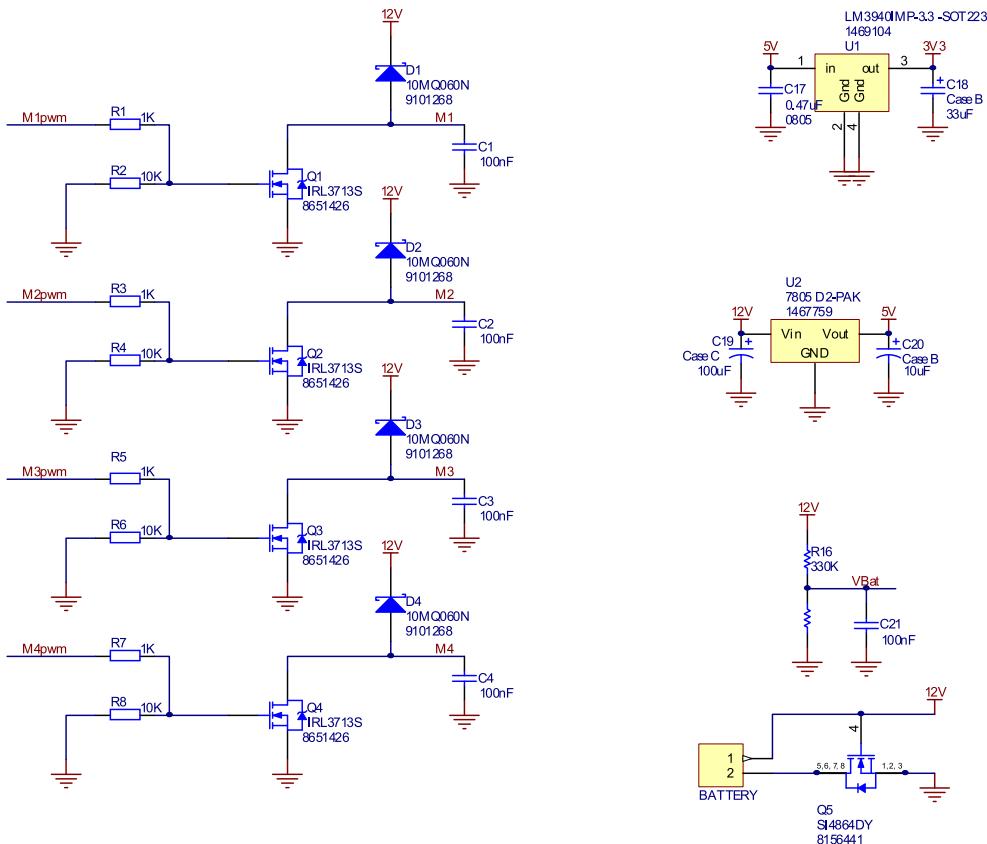


Figure D-2: Schematics of the power group

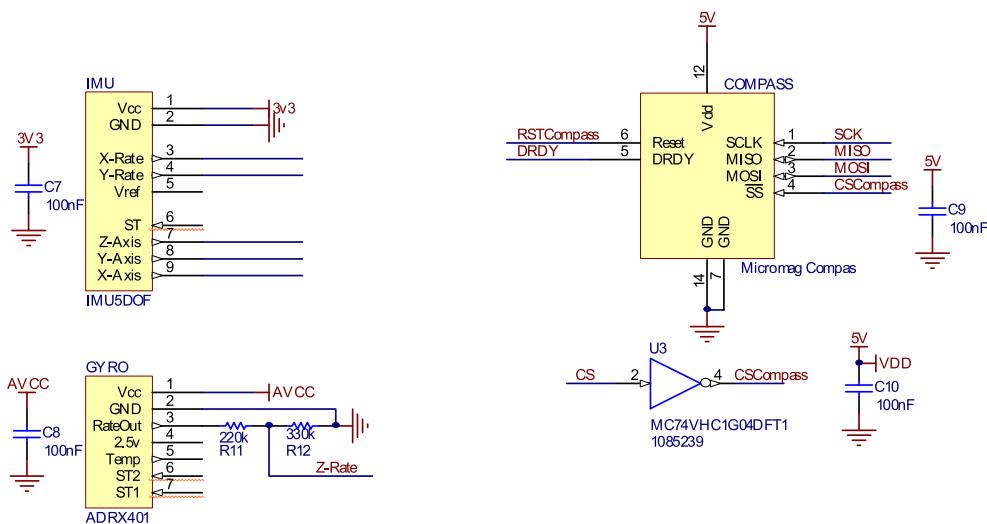


Figure D-3: Schematics of the sensors group

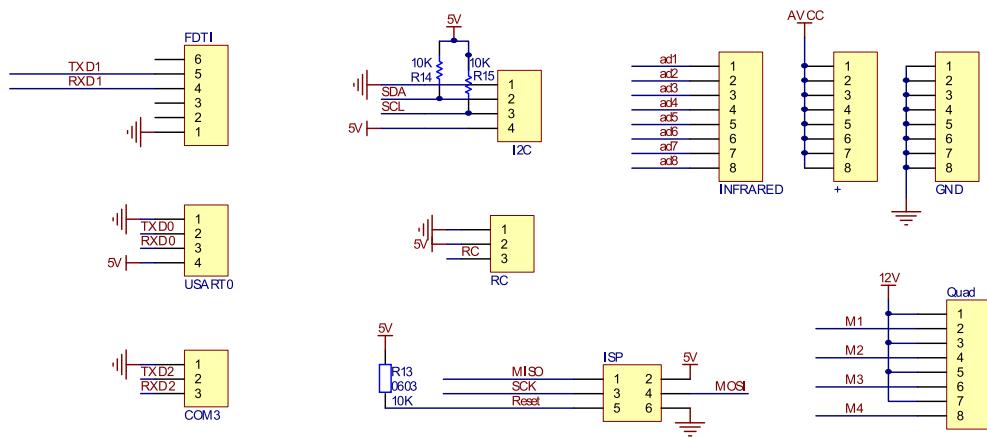


Figure D-4: Schematics of the external connectors

Appendix E

Comparison between complete model and simplified model

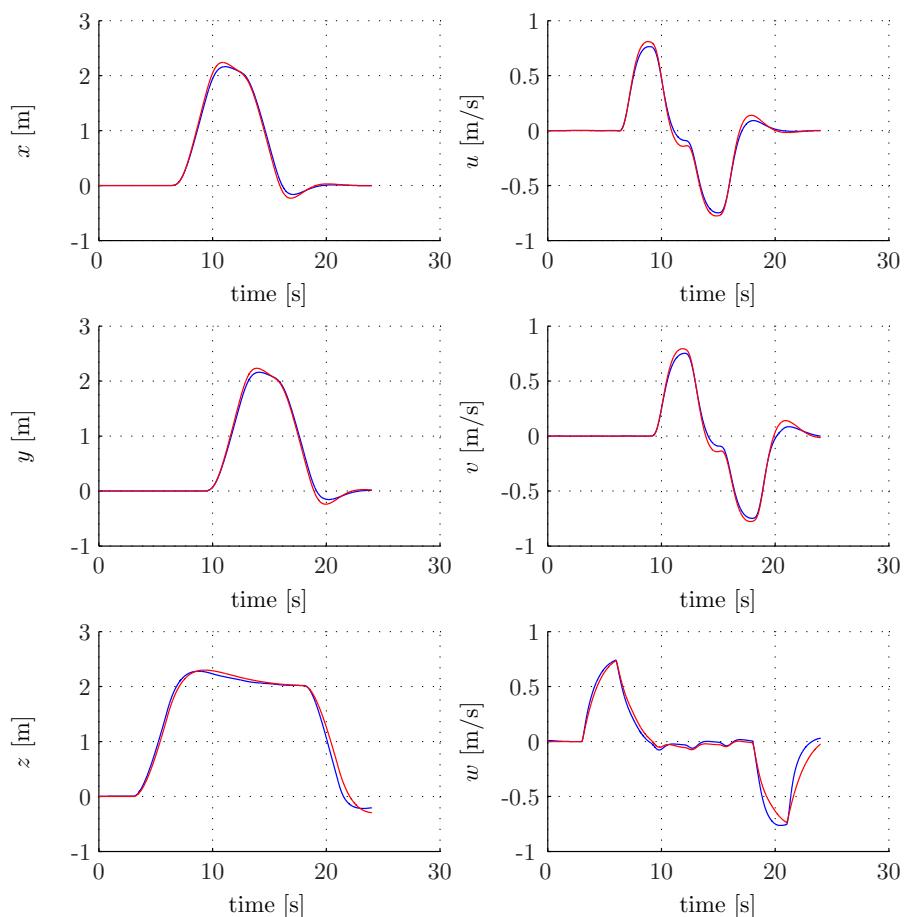


Figure E-1: Comparison of complete model (blue) with simplified model (red) using same simulation parameters.

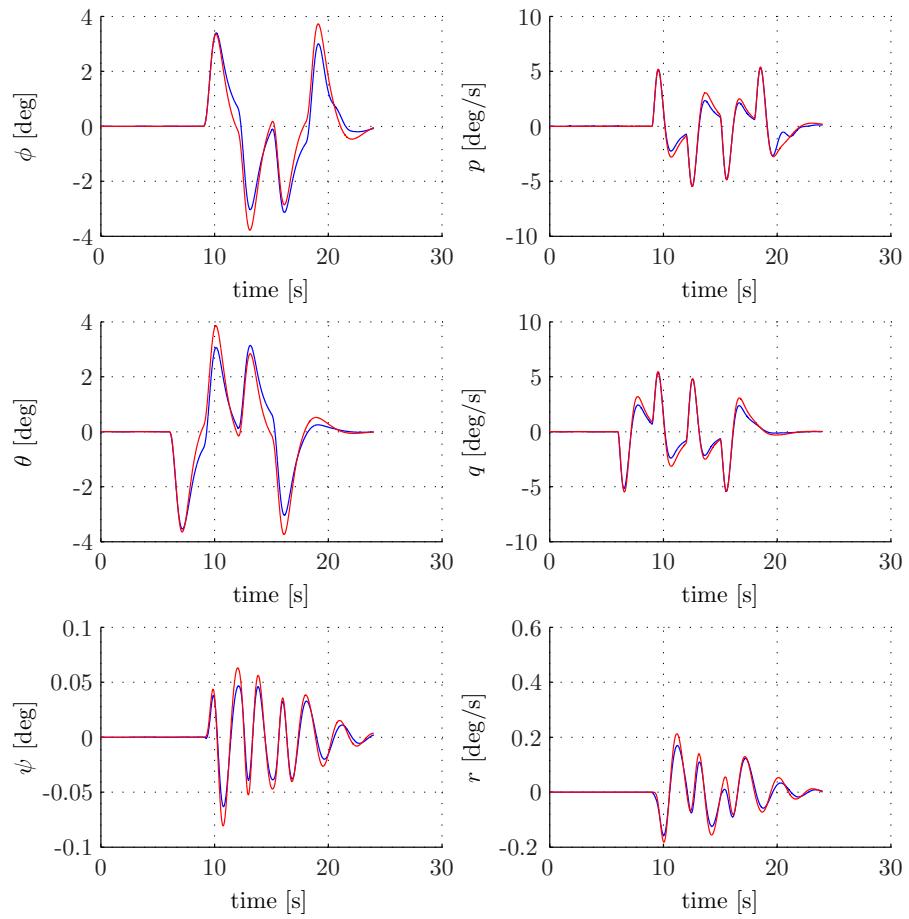


Figure E-2: Comparision of complete model (blue) with simplified model (red) using same simulation parameters.

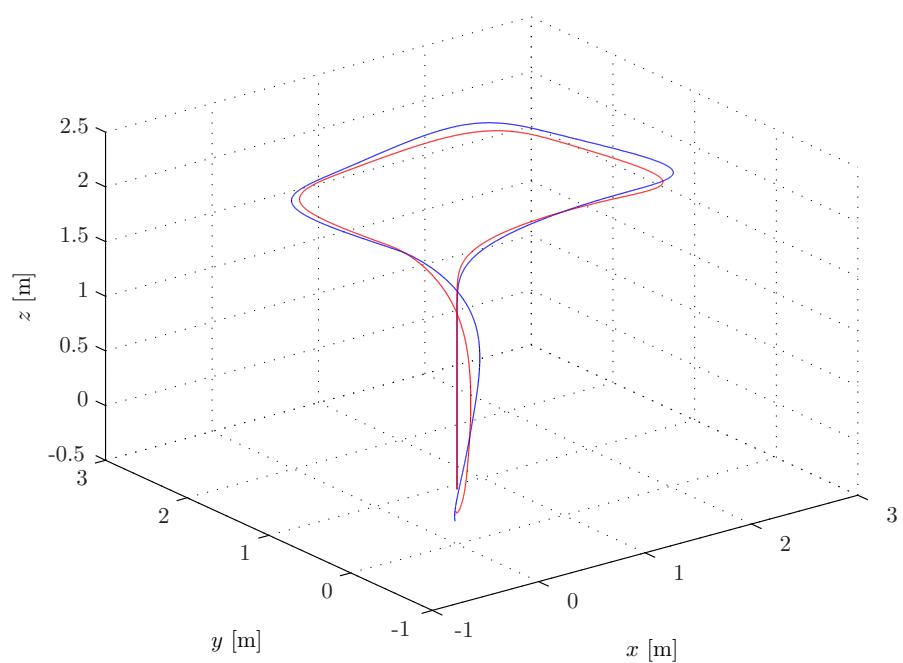


Figure E-3: Comparision of complete model (blue) with simplified model (red) using same simulation parameters.

Appendix F

AHRS Kalman filtering tested with simulated data

F-1 Using realistic sensor parameters

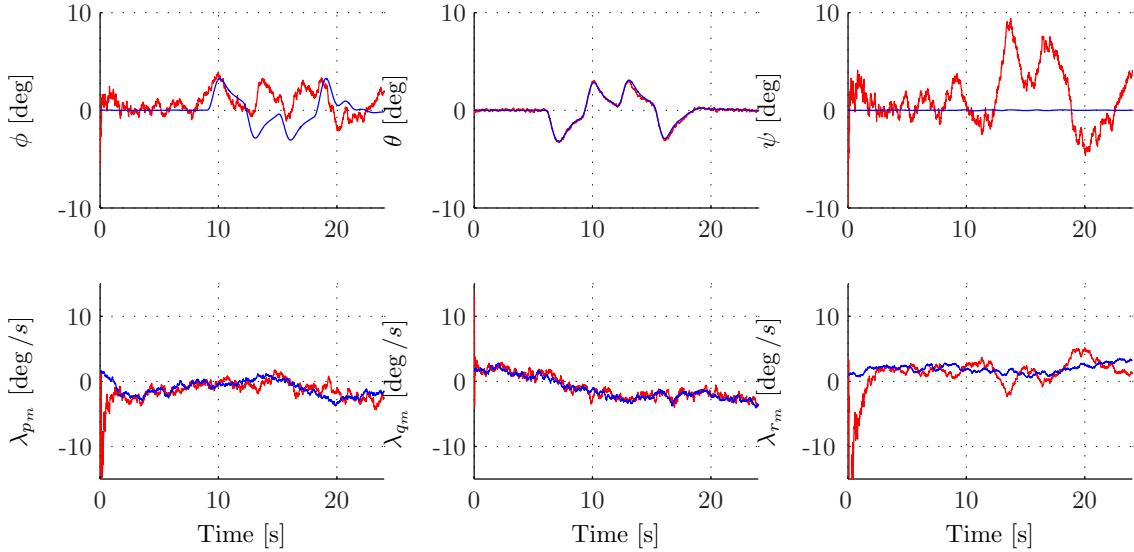


Figure F-1: AHRS filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e-005; 5e-005; 6e-006;])$$

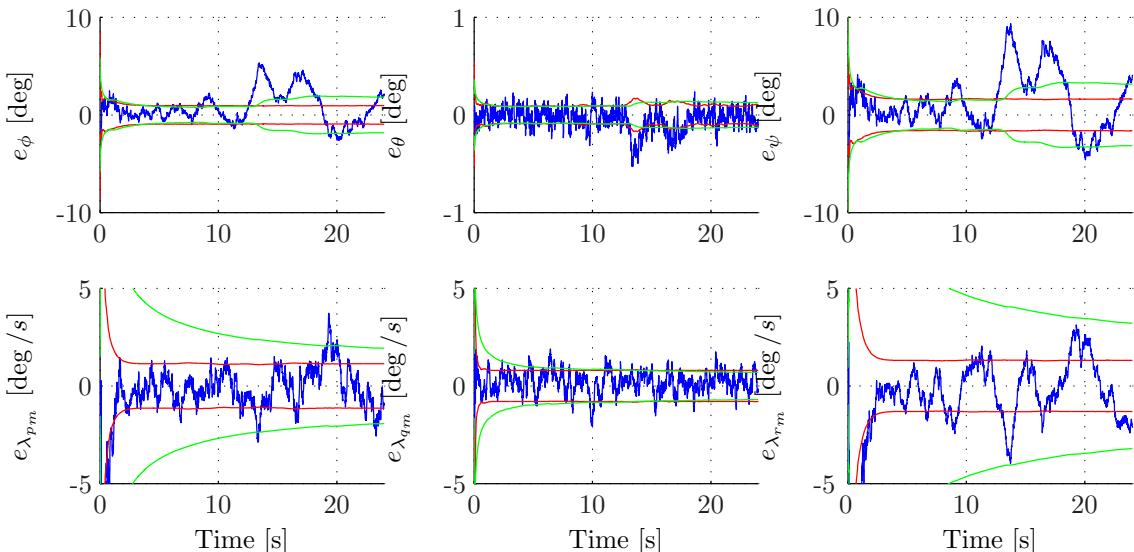


Figure F-2: AHRS filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e-005; 5e-005; 6e-006;])$$

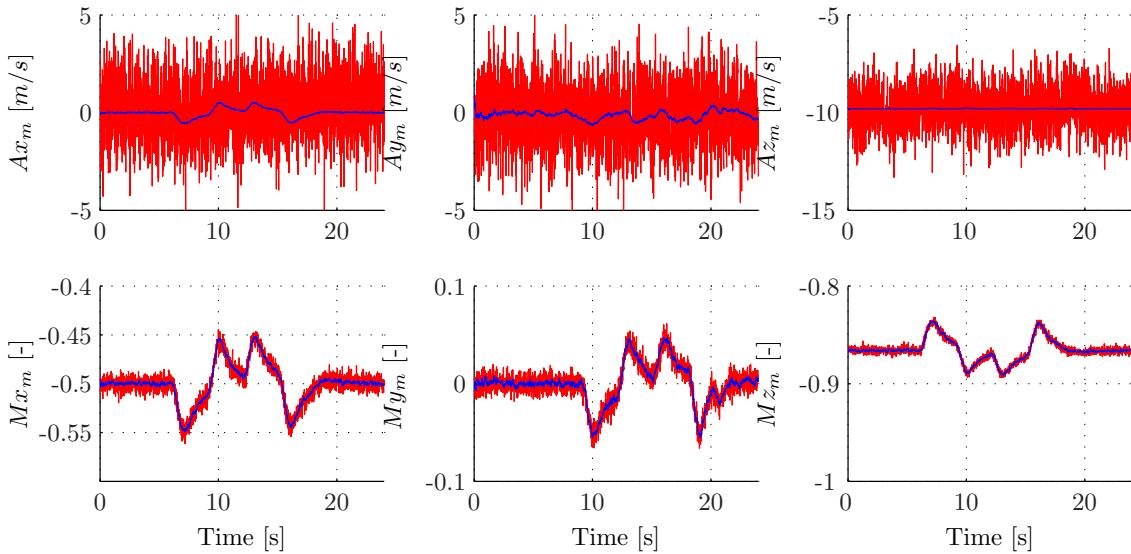


Figure F-3: AHRS filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e - 005; 5e - 005; 6e - 006;])$$

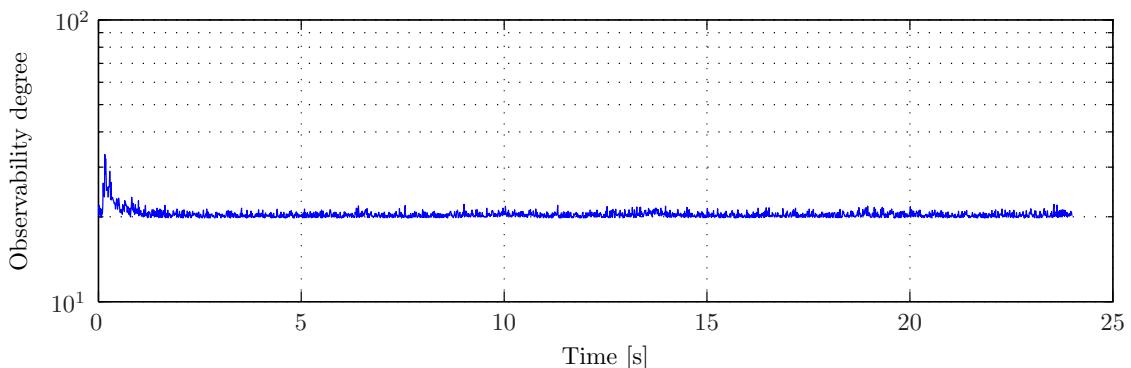


Figure F-4: AHRS filter results - observability degree.

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e - 005; 5e - 005; 6e - 006;])$$

F-2 Using 1/10 random walk on sensors

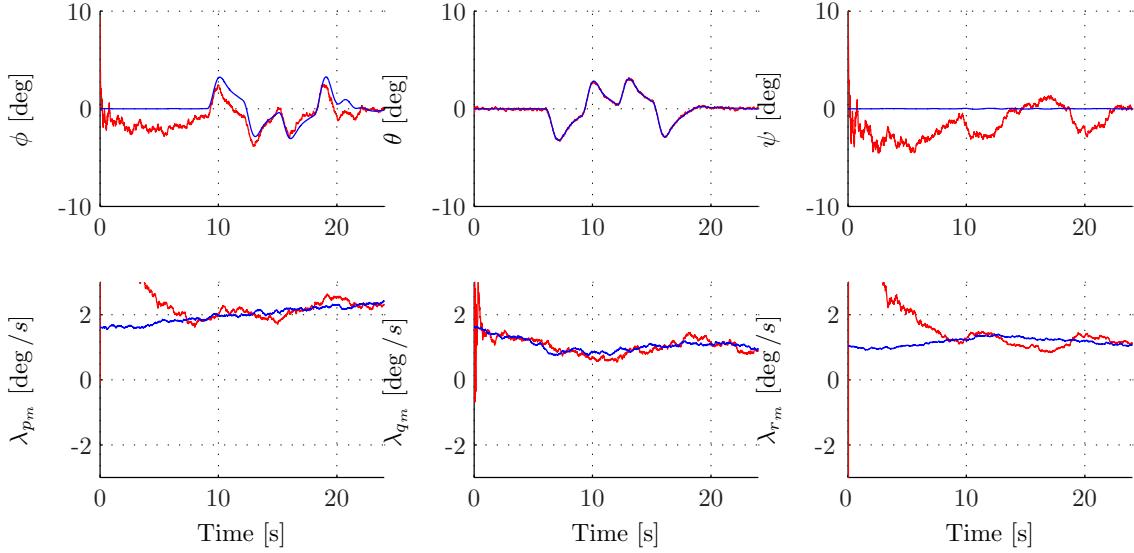


Figure F-5: AHRS filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e-005; 5e-005; 6e-006;])$$

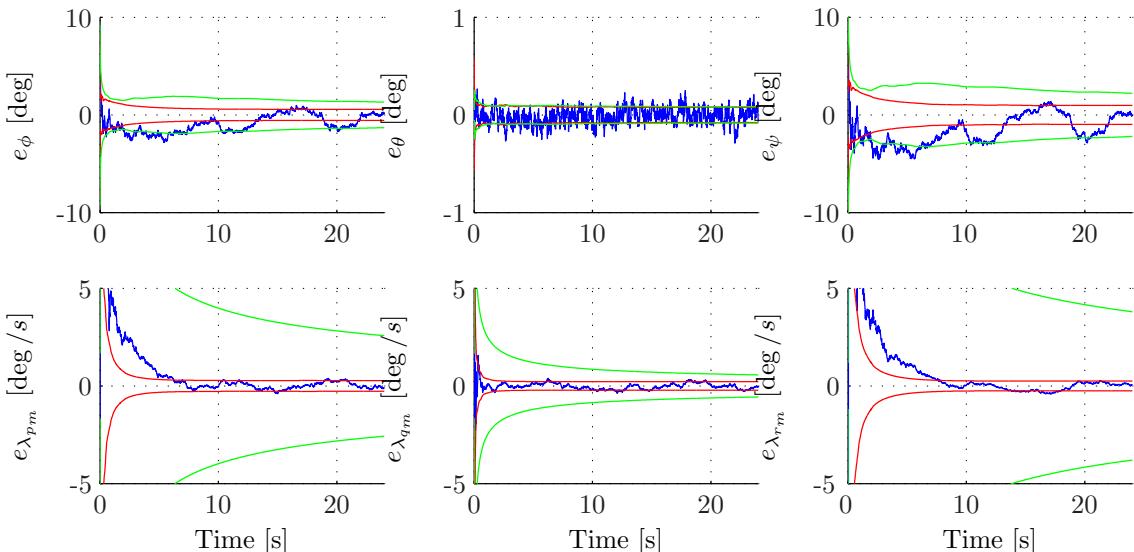


Figure F-6: AHRS filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = diag([2; 2; 1; 3e-005; 5e-005; 6e-006;])$$

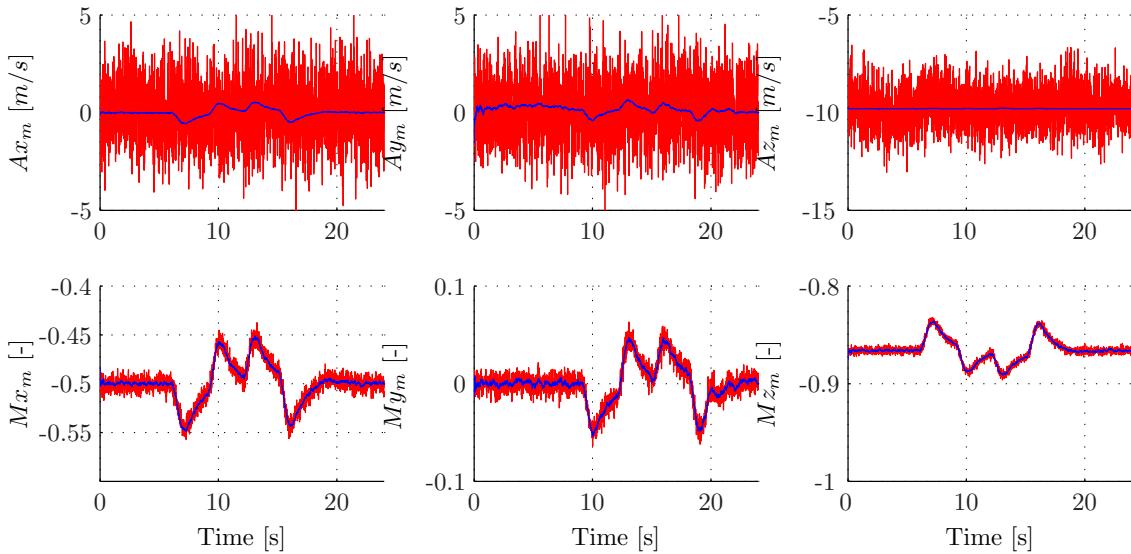


Figure F-7: AHRS filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$

$\mathbf{R} = diag([2; 2; 1; 3e - 005; 5e - 005; 6e - 006;])$

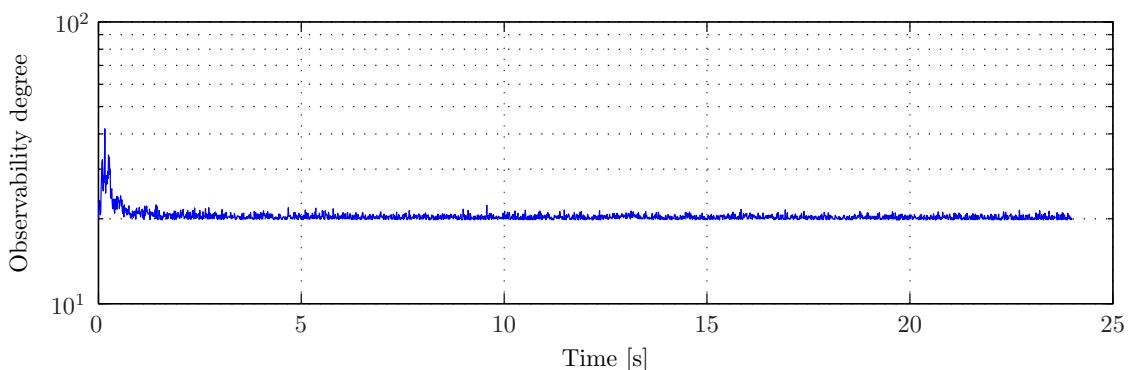


Figure F-8: AHRS filter results - observability degree.

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$

$\mathbf{R} = diag([2; 2; 1; 3e - 005; 5e - 005; 6e - 006;])$

Appendix G

AHRS Kalman filtering tested with real data

G-1 Roll, pitch, yaw movements, with motors off

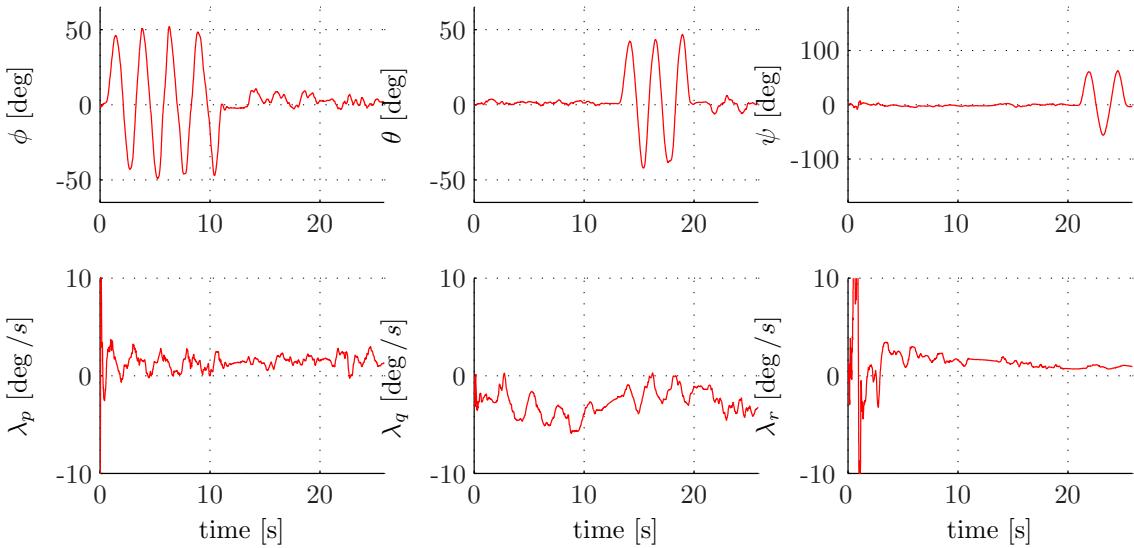


Figure G-1: AHRS filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$

$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.04; 0.04; 0.04;])$

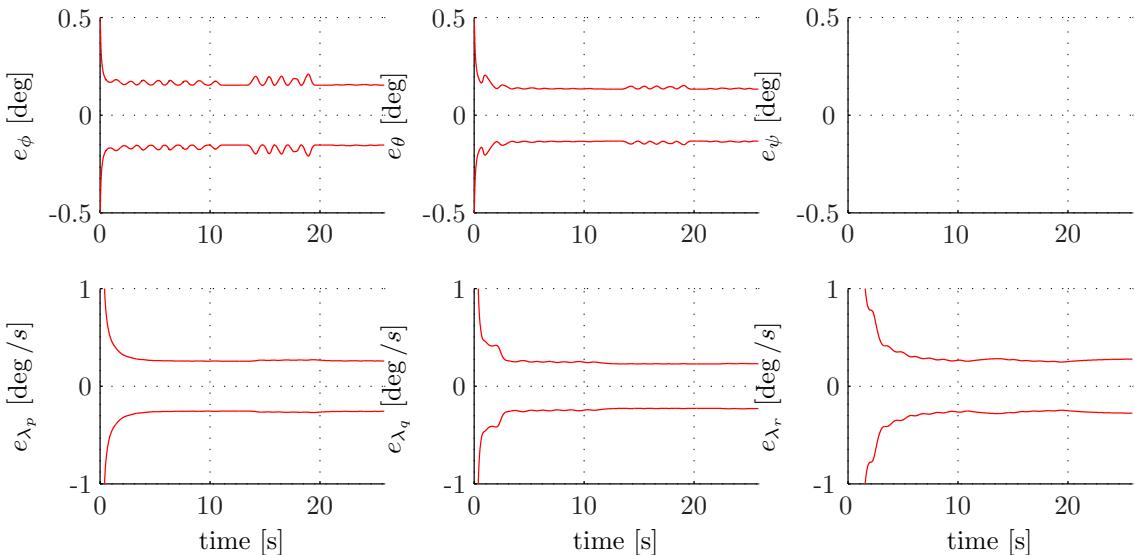


Figure G-2: AHRS filter results - standard deviation (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$

$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.04; 0.04; 0.04;])$

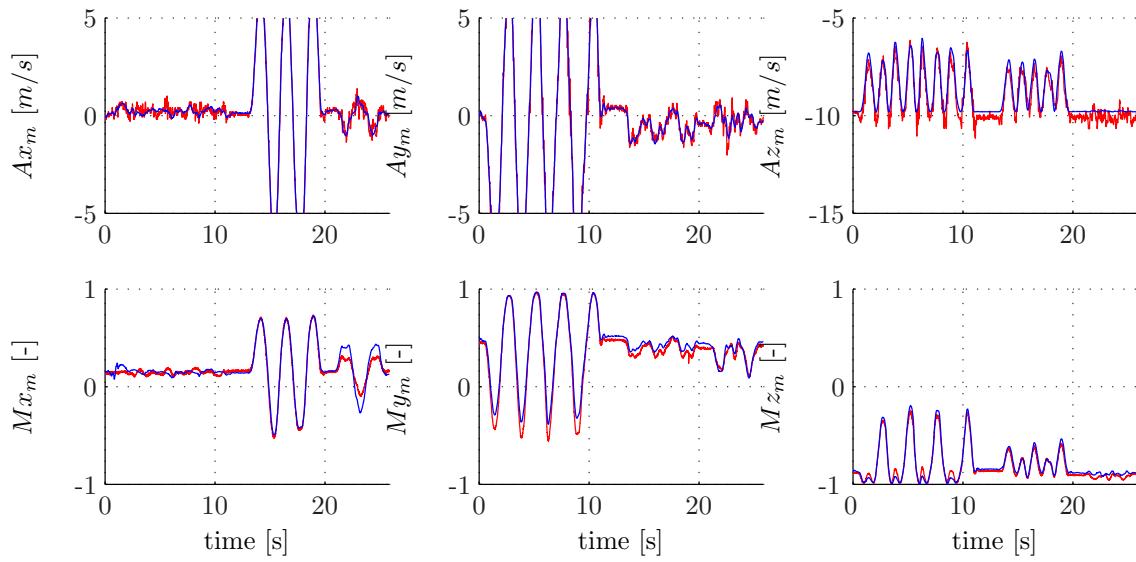


Figure G-3: AHRS filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.04; 0.04; 0.04;])$$

G-2 Roll, pitch, yaw movements, with motors on

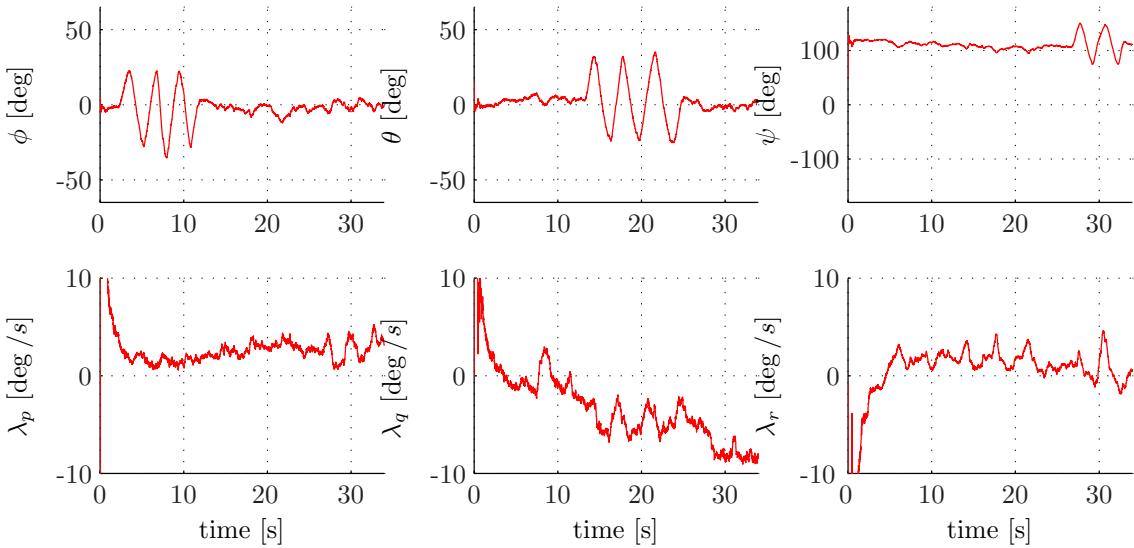


Figure G-4: AHRS filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{\text{select}} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = \text{diag}([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = \text{diag}([0.01; 0.01; 0.04; 0.0001; 0.0001; 0.0001;])$$

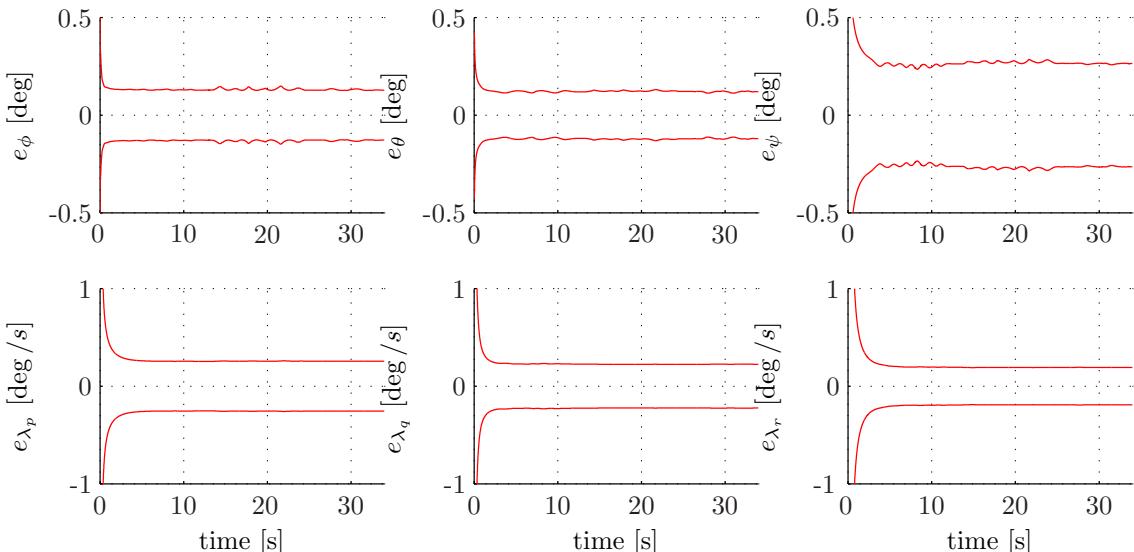


Figure G-5: AHRS filter results - standard deviation (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{\text{select}} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = \text{diag}([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = \text{diag}([0.01; 0.01; 0.04; 0.0001; 0.0001; 0.0001;])$$

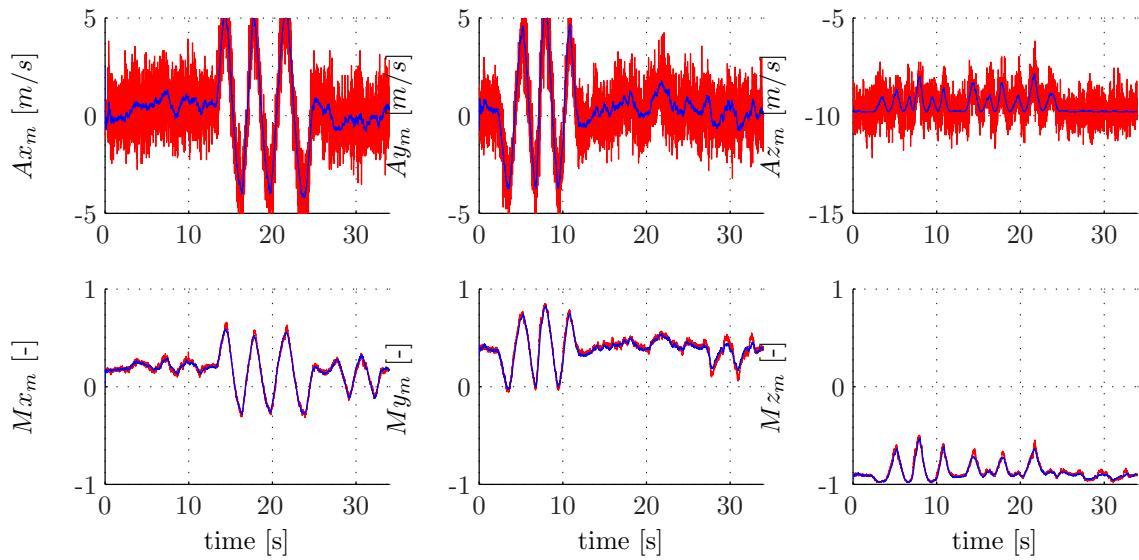


Figure G-6: AHRS filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 6$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003;])$$

$$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.0001; 0.0001; 0.0001;])$$

G-3 Actual flight data

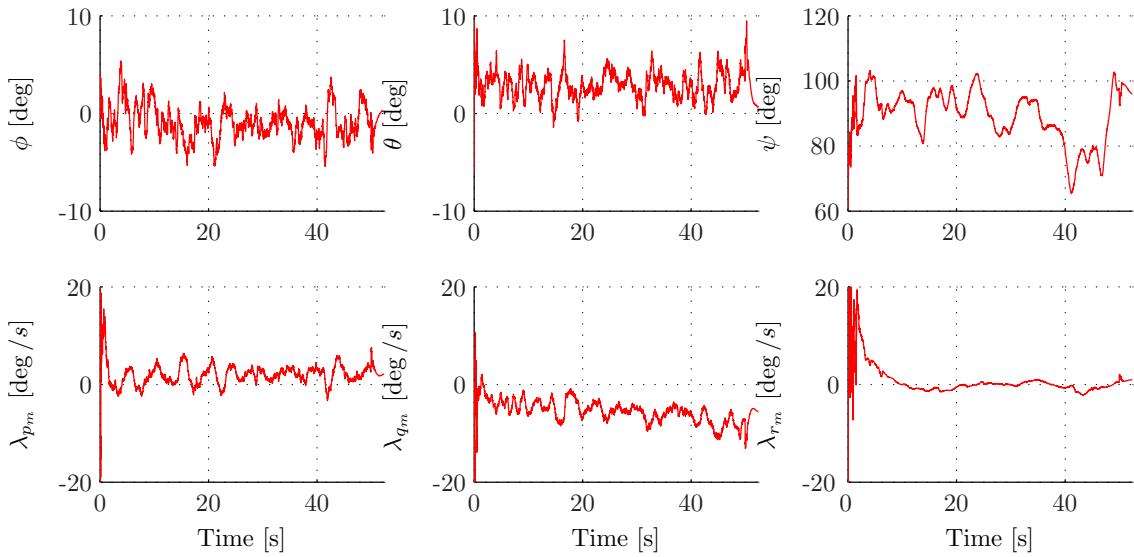


Figure G-7: AHRS filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([1e-006; 1e-006; 1e-006; 0.0004; 0.0004; 0.0004;])$

$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.01; 0.01; 0.01;])$

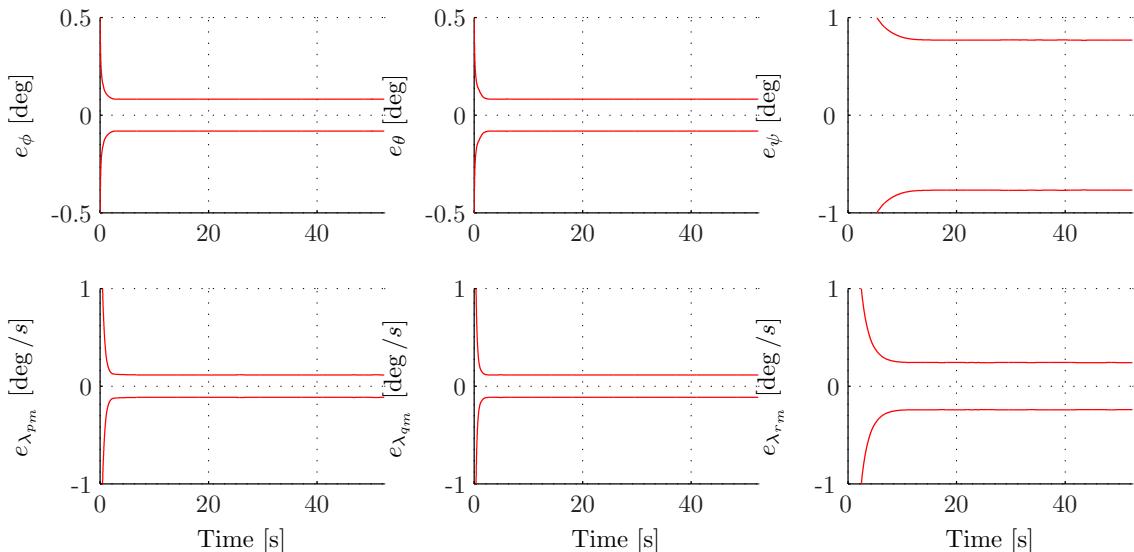


Figure G-8: AHRS filter results - standard deviation of the state estimation error (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([1e-006; 1e-006; 1e-006; 0.0004; 0.0004; 0.0004;])$

$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.01; 0.01; 0.01;])$

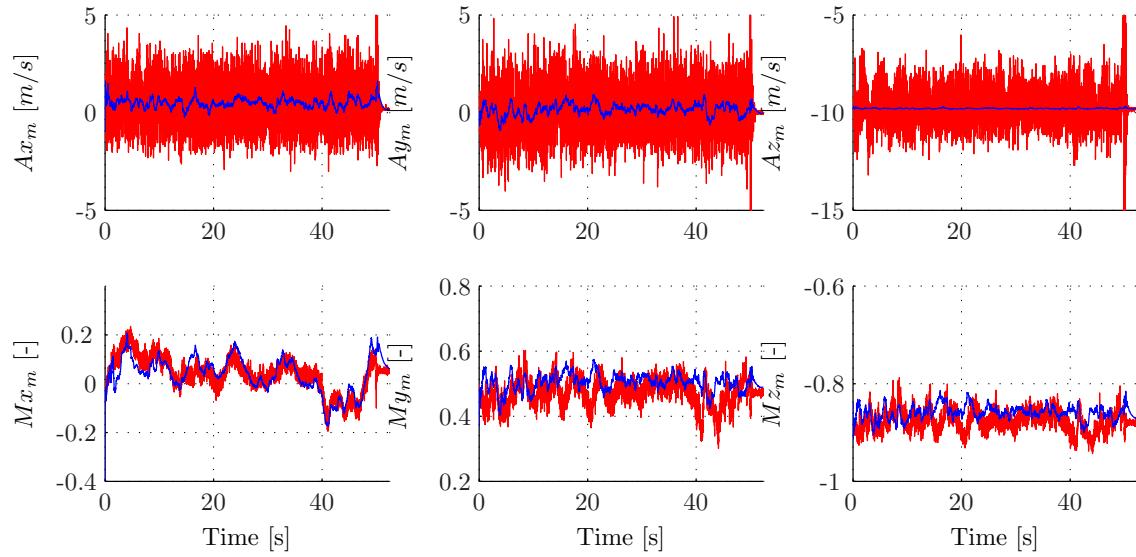


Figure G-9: AHRS filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 6$.

$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1]$

$\mathbf{Q} = diag([1e-006; 1e-006; 1e-006; 0.0004; 0.0004; 0.0004;])$

$\mathbf{R} = diag([0.01; 0.01; 0.04; 0.01; 0.01; 0.01;])$

Appendix H

Full Kalman filtering tested with simulated data

H-1 Using 6 infrared sensors, directions X1-X2-Y1-Y2-Z1-Z2

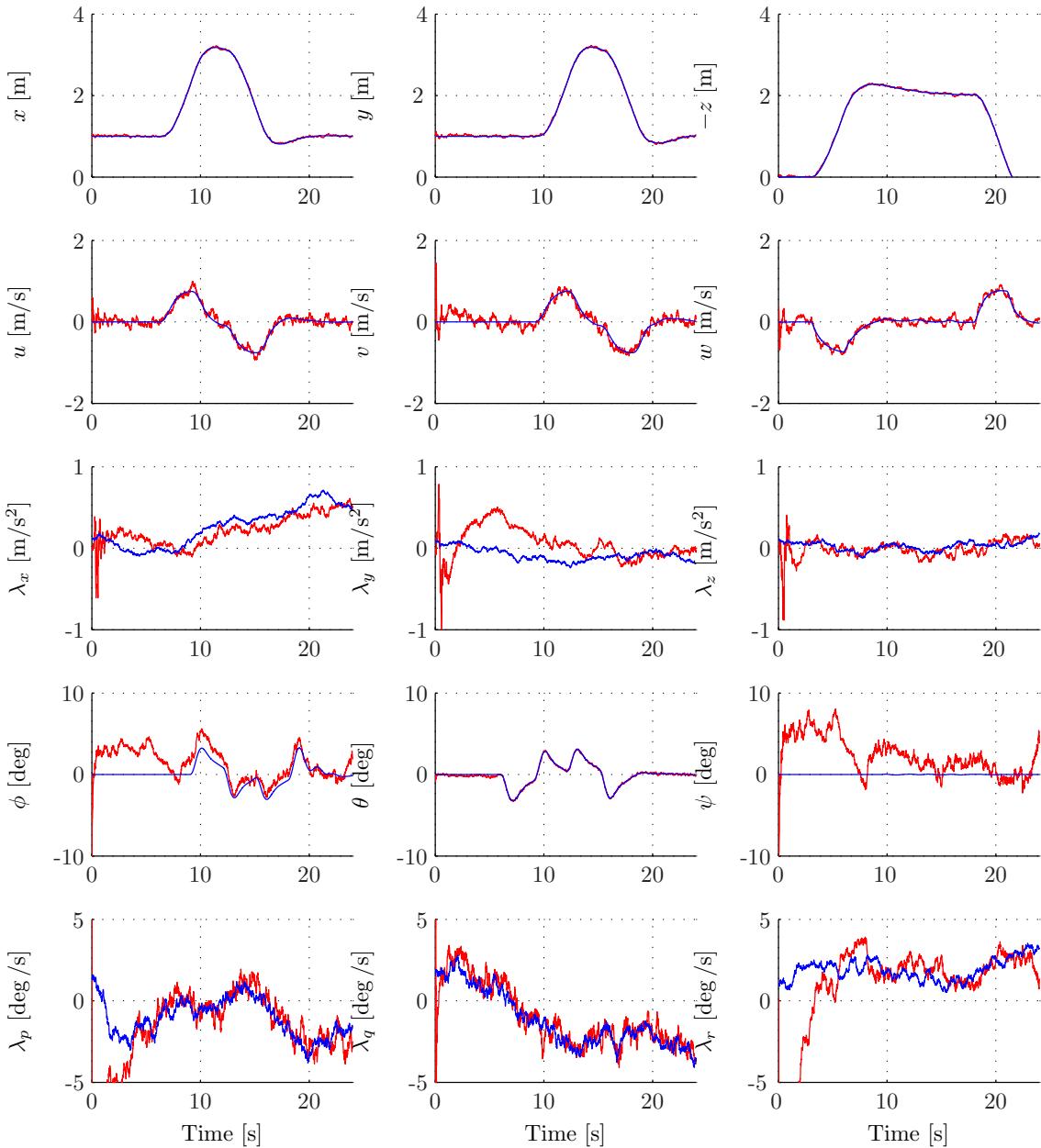


Figure H-1: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

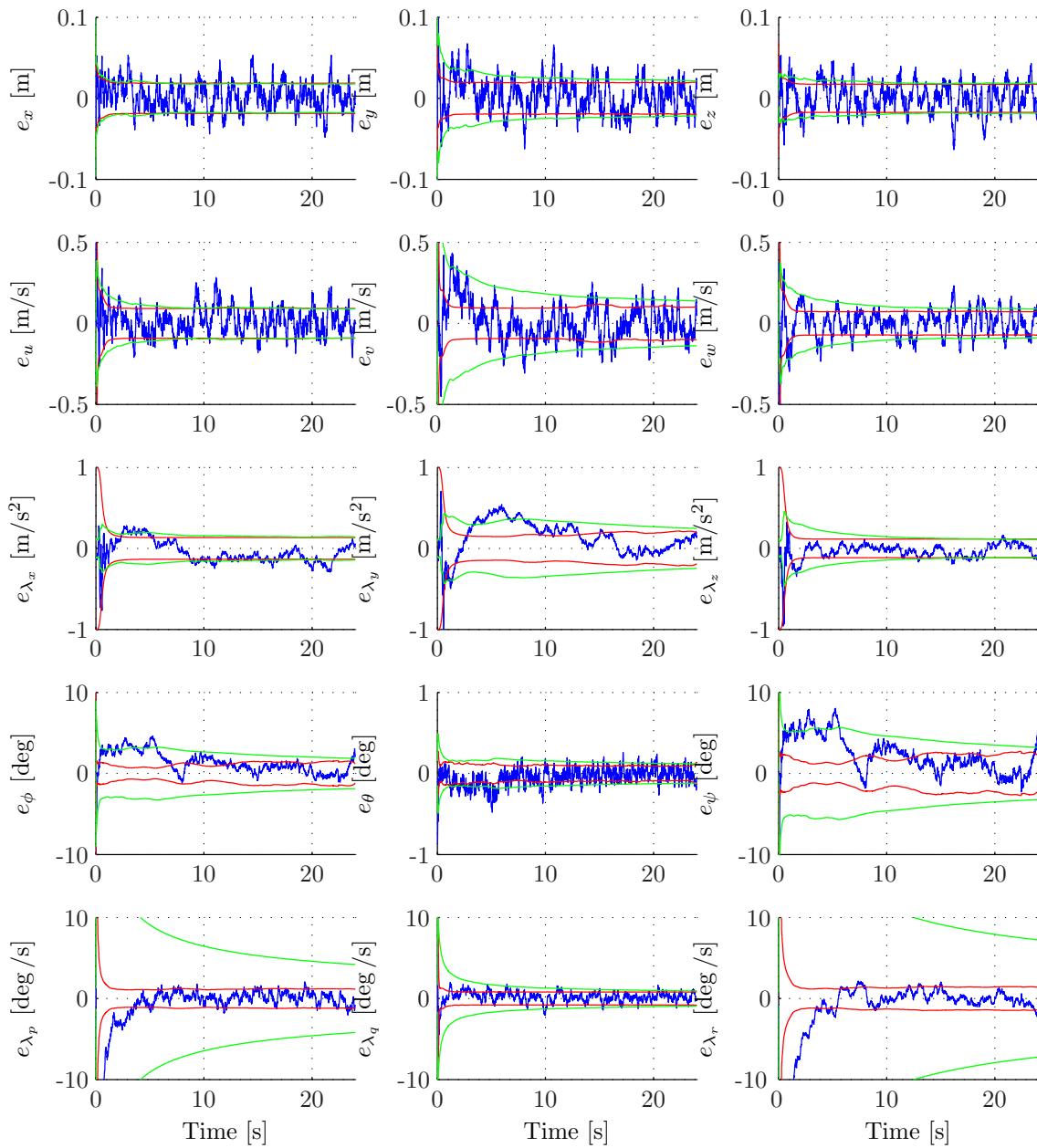


Figure H-2: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

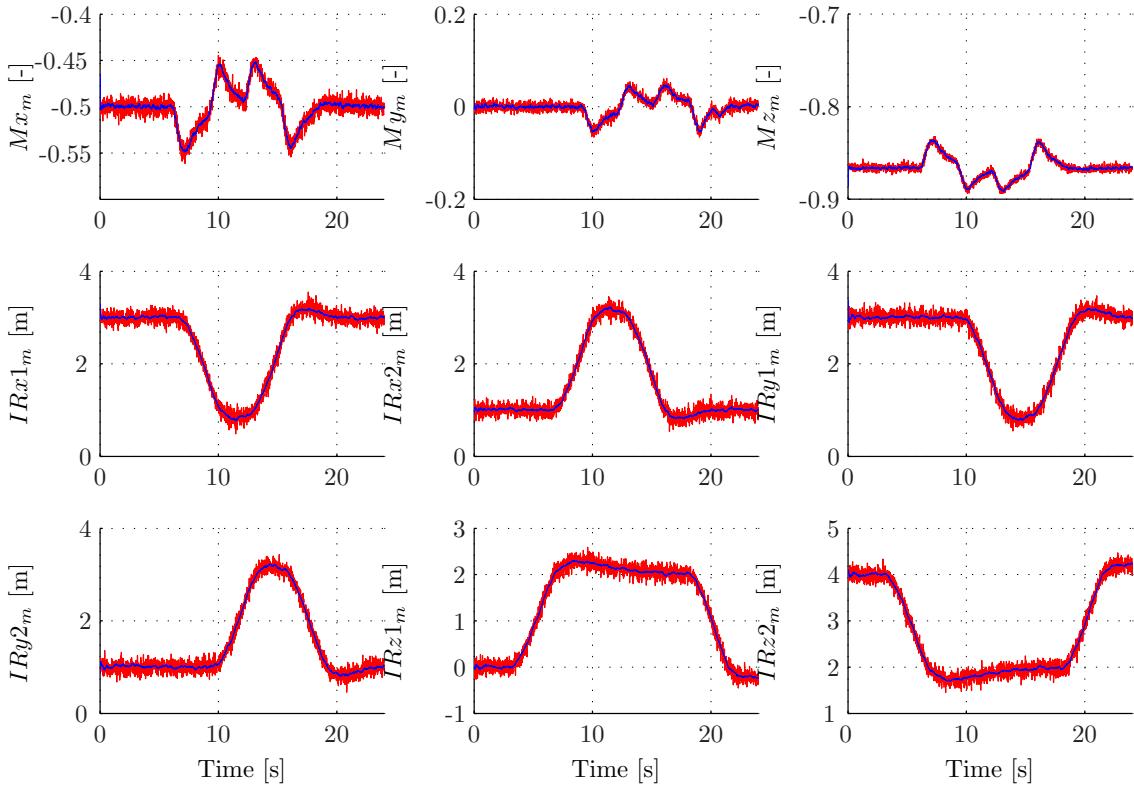


Figure H-3: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

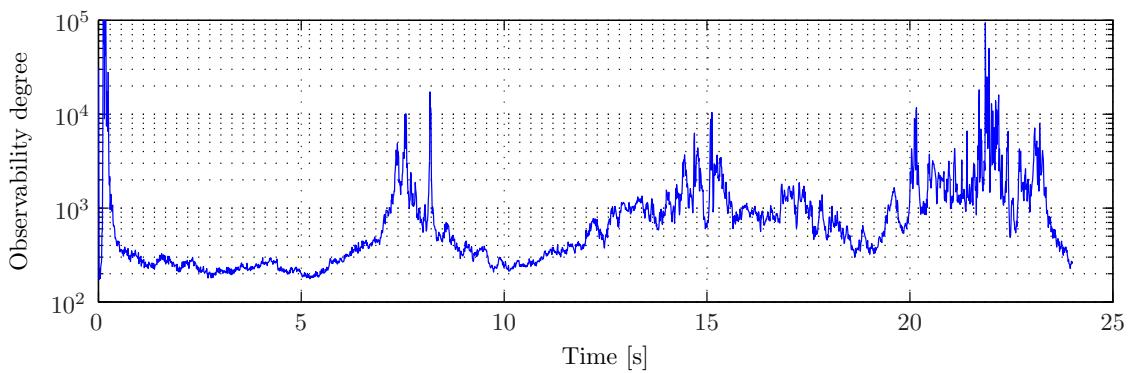


Figure H-4: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

H-2 Using 5 infrared sensors, directions X1-X2-Y1-Y2-Z1

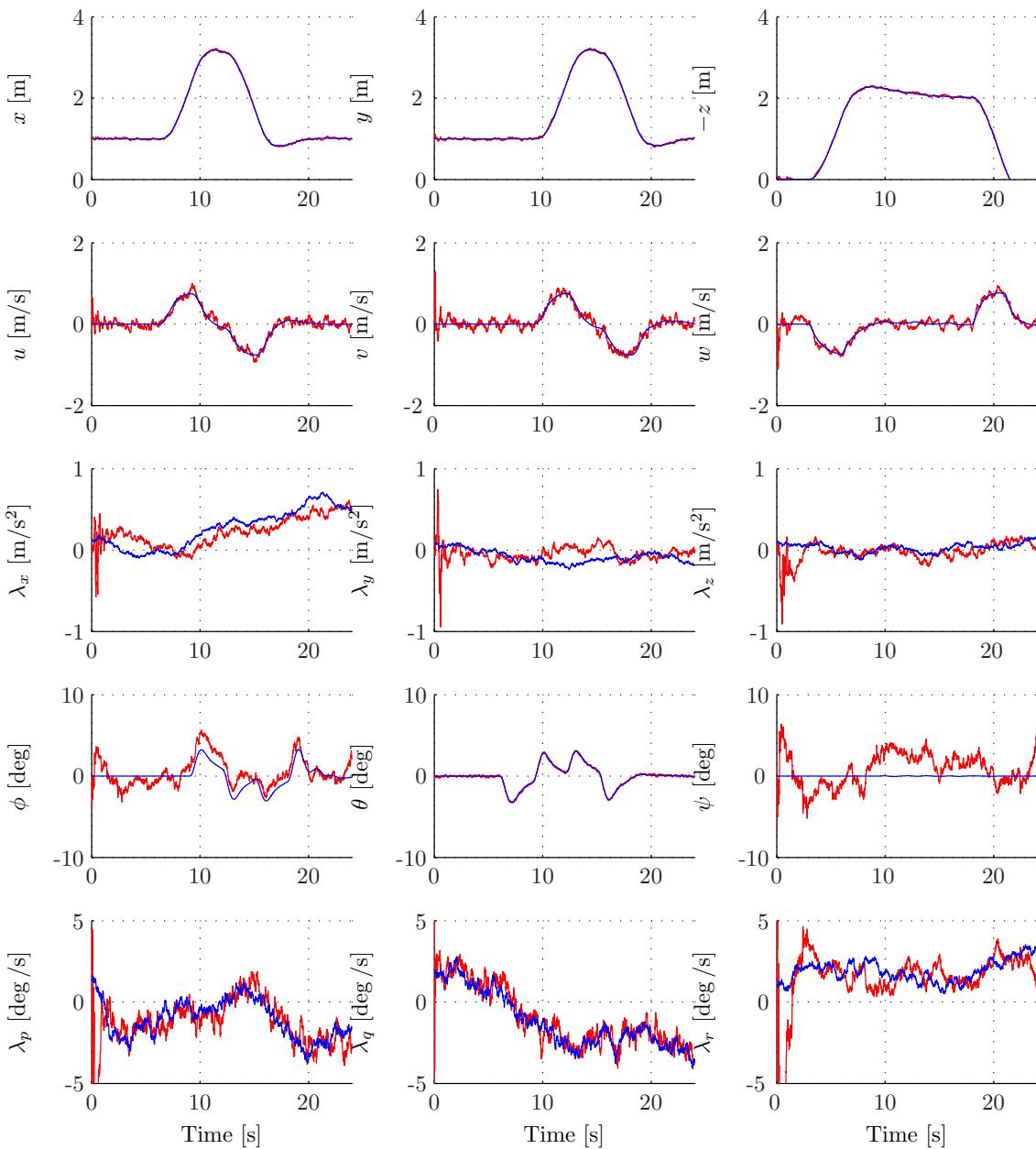


Figure H-5: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

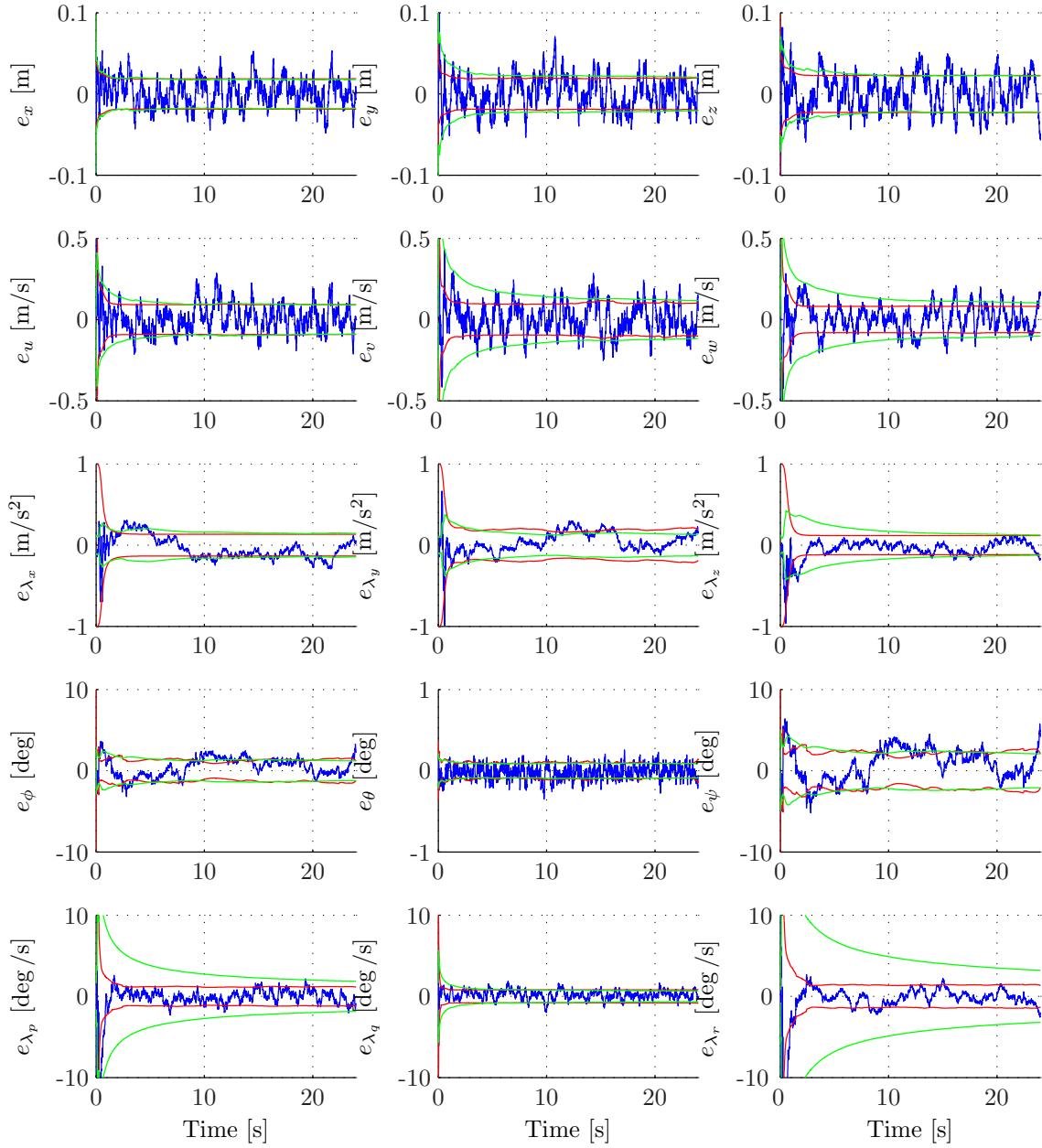


Figure H-6: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

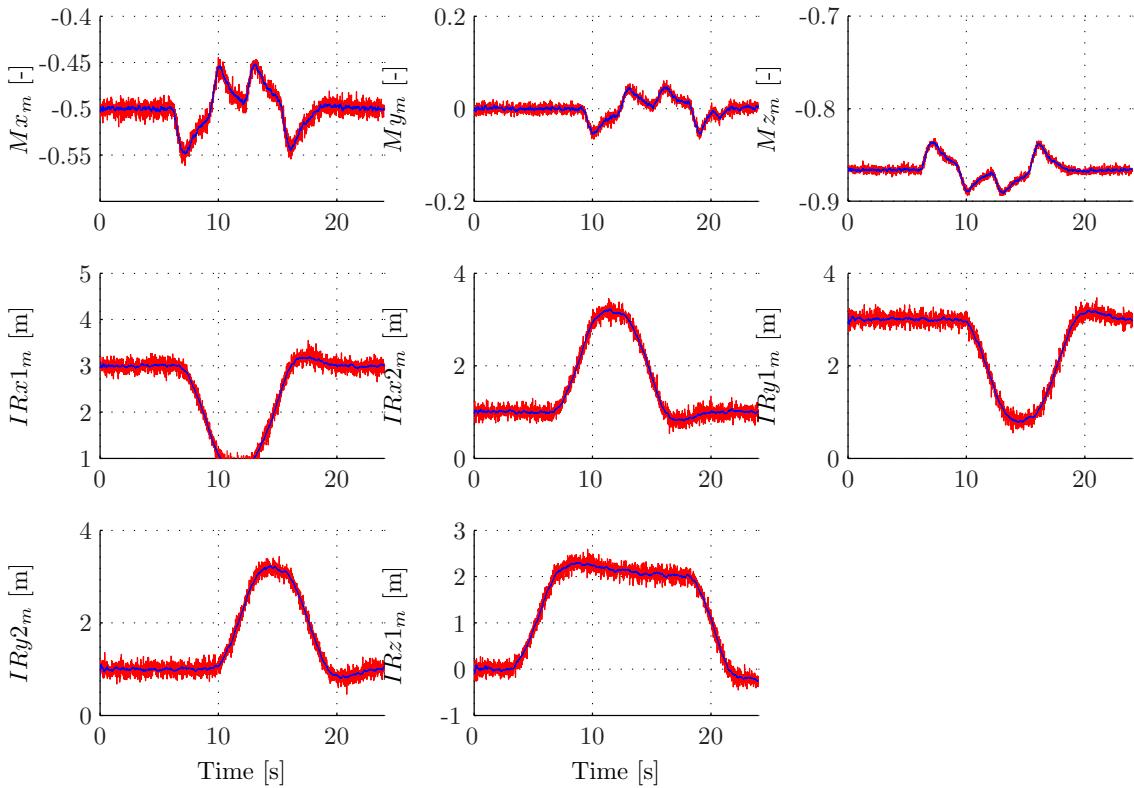


Figure H-7: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

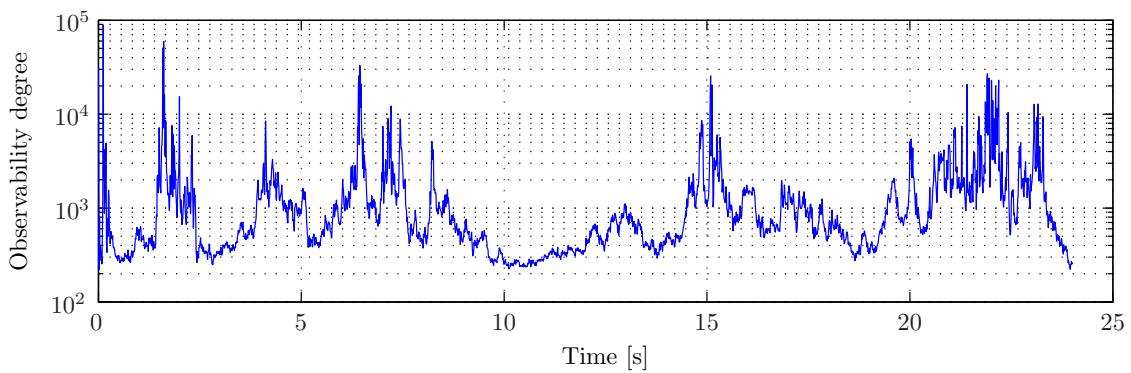


Figure H-8: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

H-3 Using 4 infrared sensors, directions X1-X2-Y1-Z1

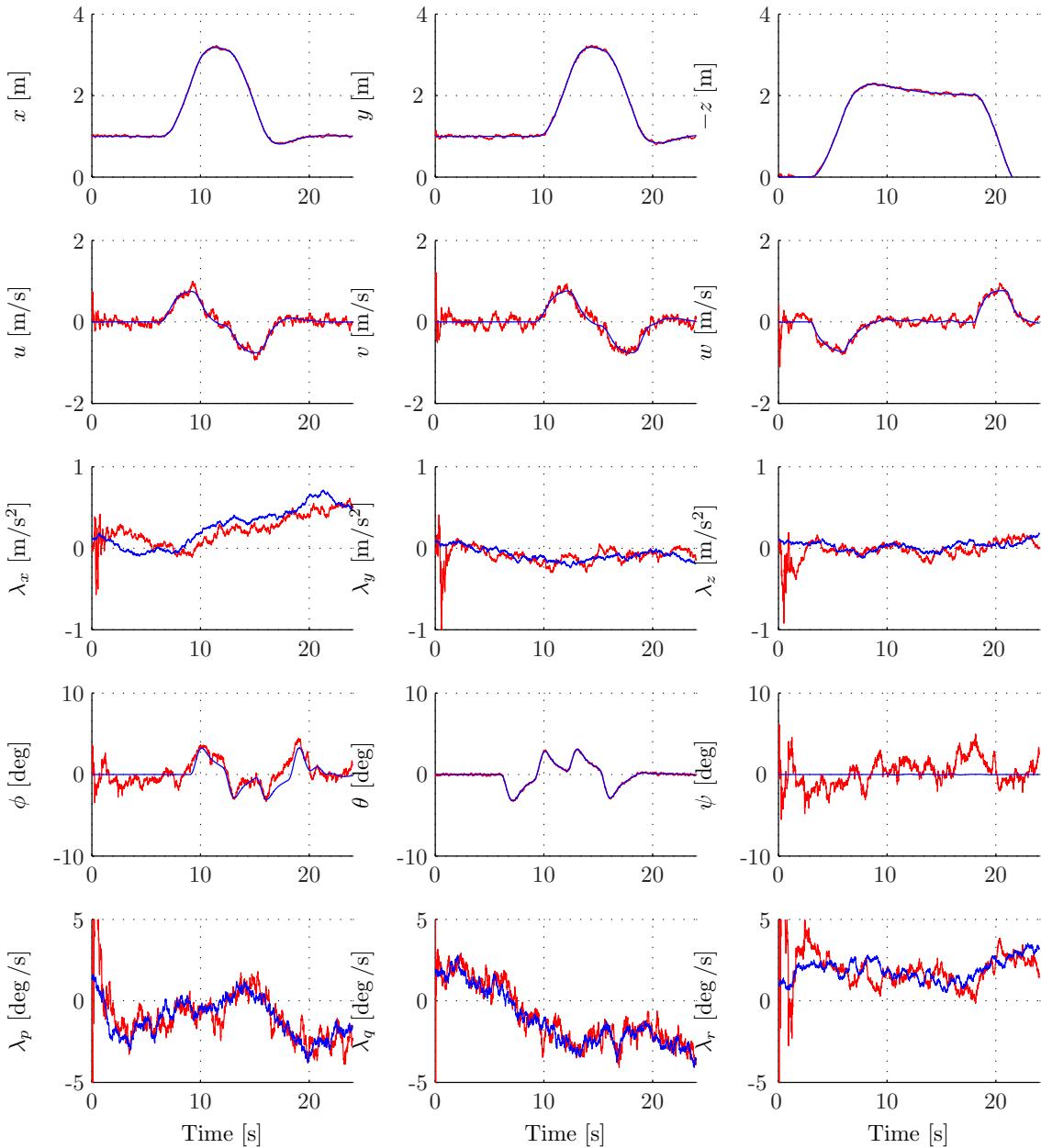


Figure H-9: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

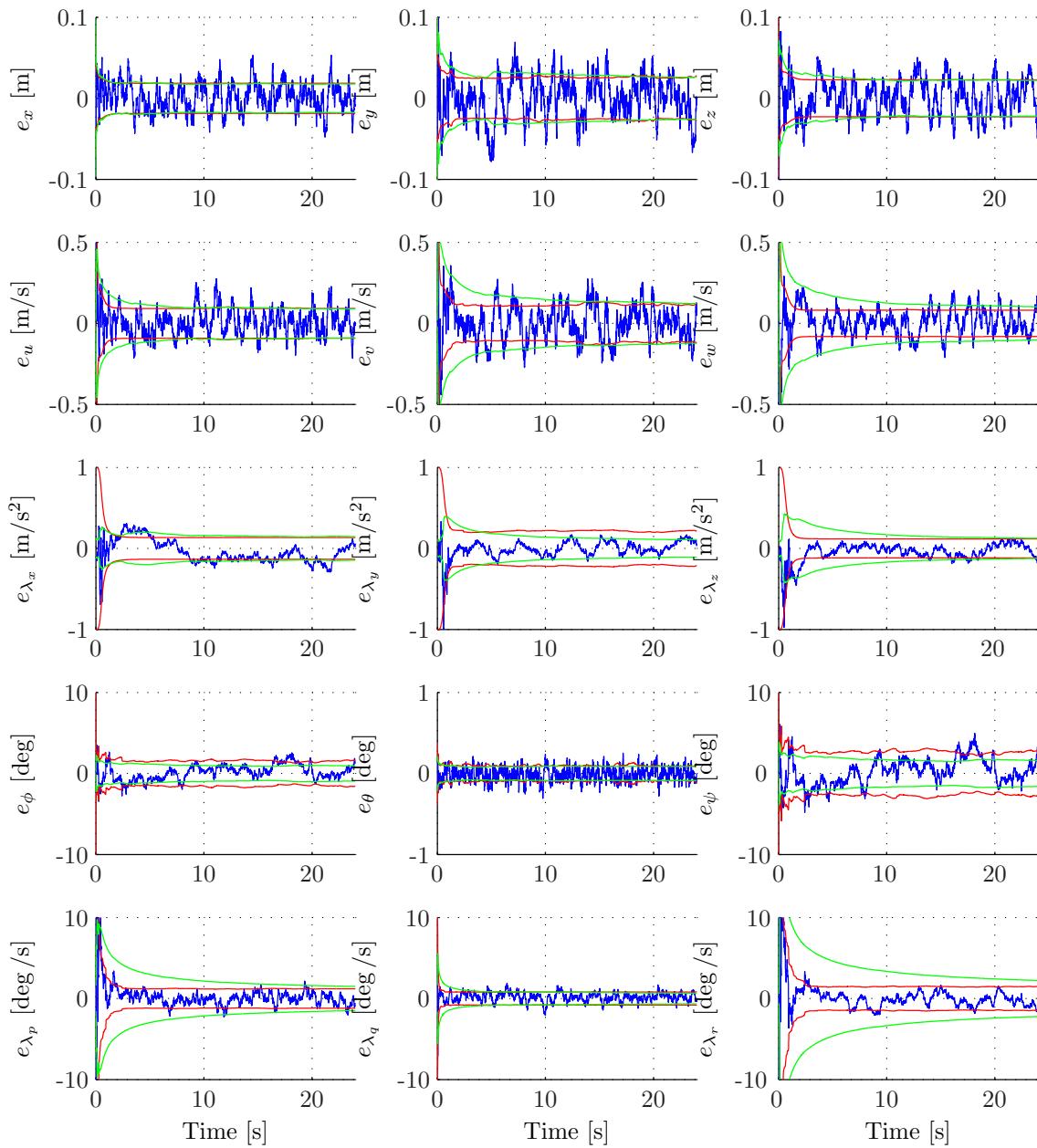


Figure H-10: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

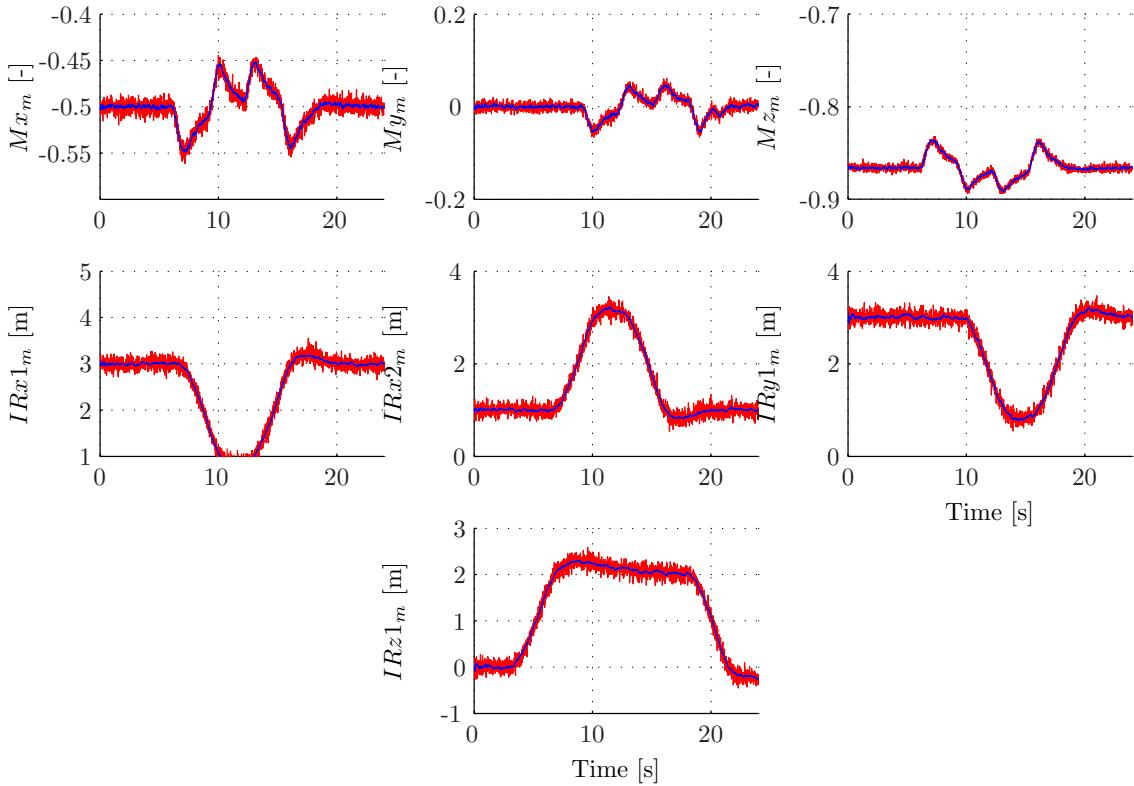


Figure H-11: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

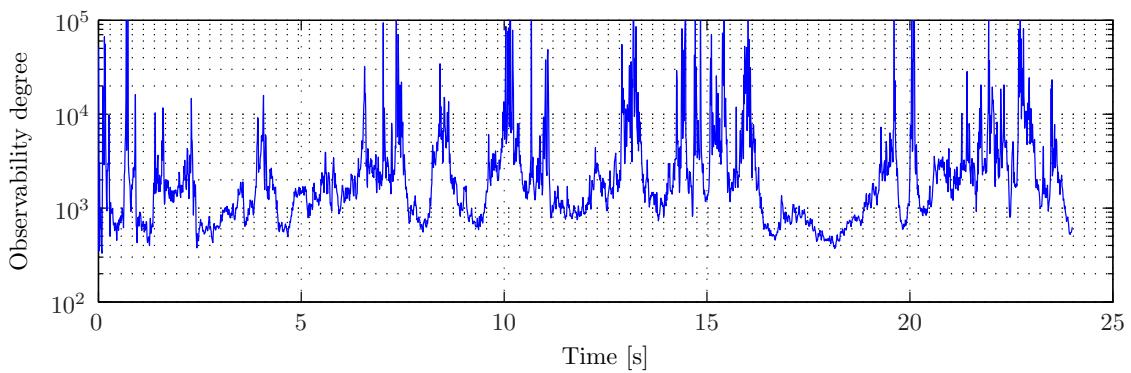


Figure H-12: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

H-4 Using 4 infrared sensors, directions X1-Y1-Y2-Z1

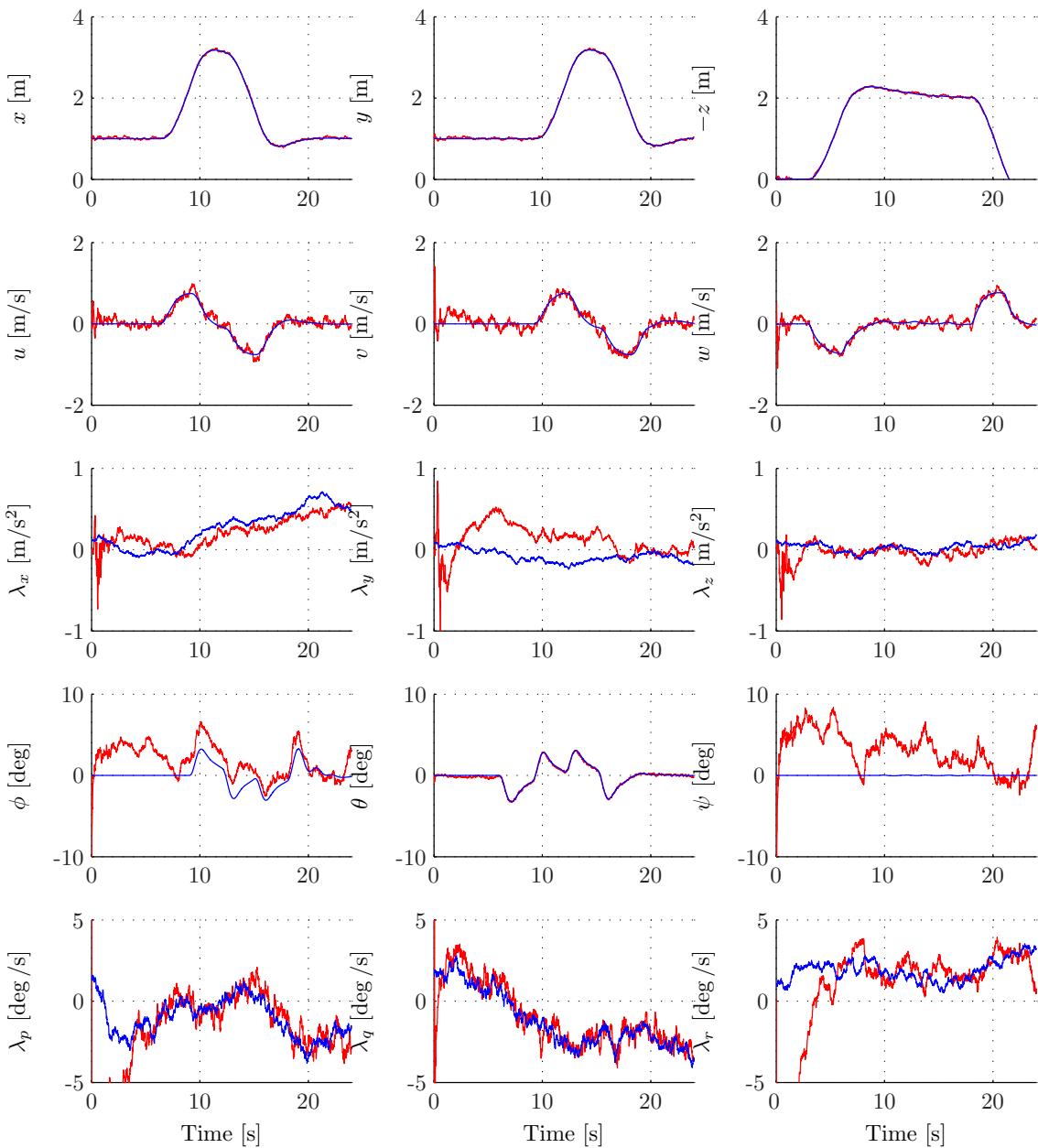


Figure H-13: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = \text{diag}([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

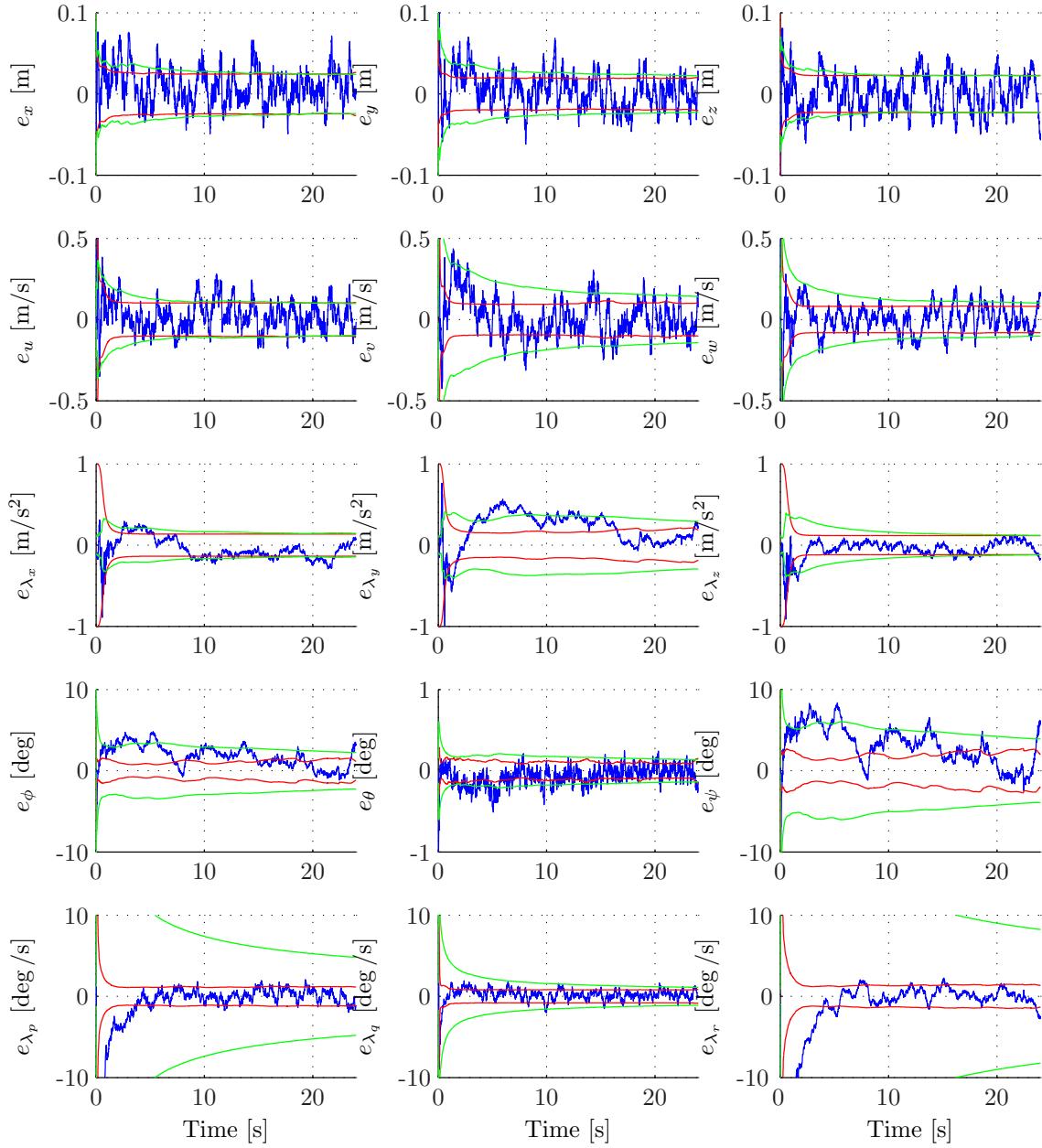


Figure H-14: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

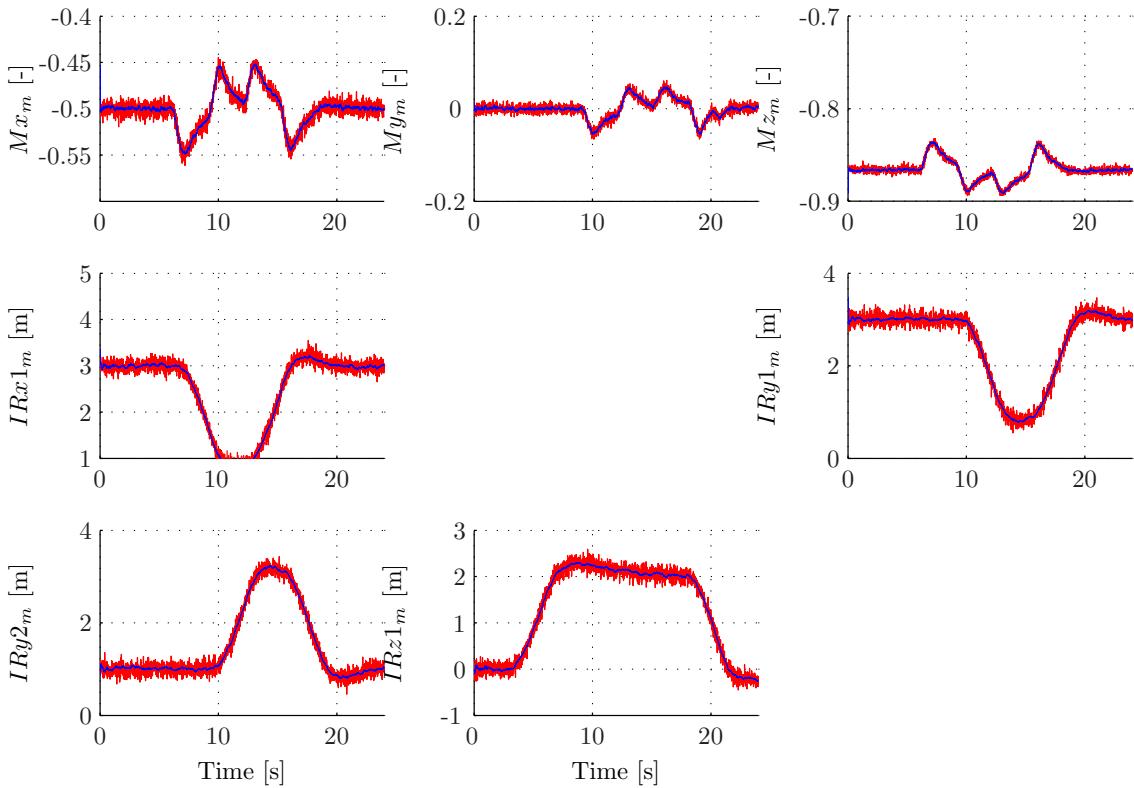


Figure H-15: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

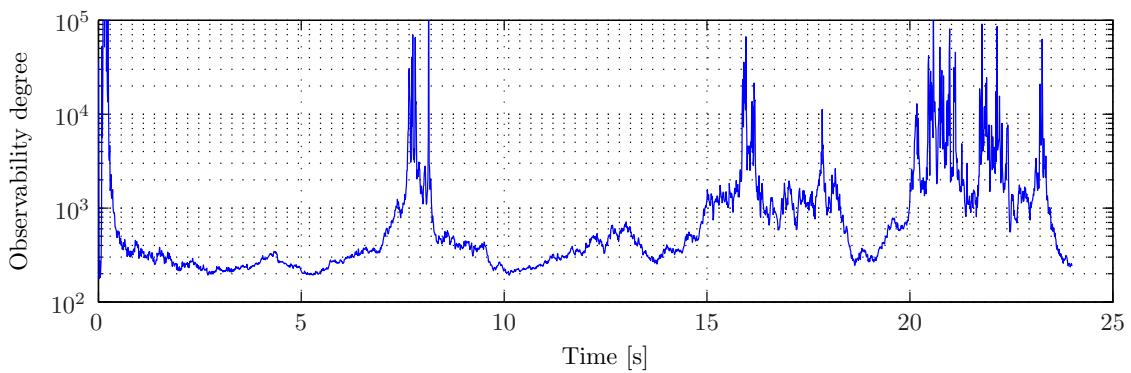


Figure H-16: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

H-5 Using 4 infrared sensors, directions X1-Y1-Z1-Z2

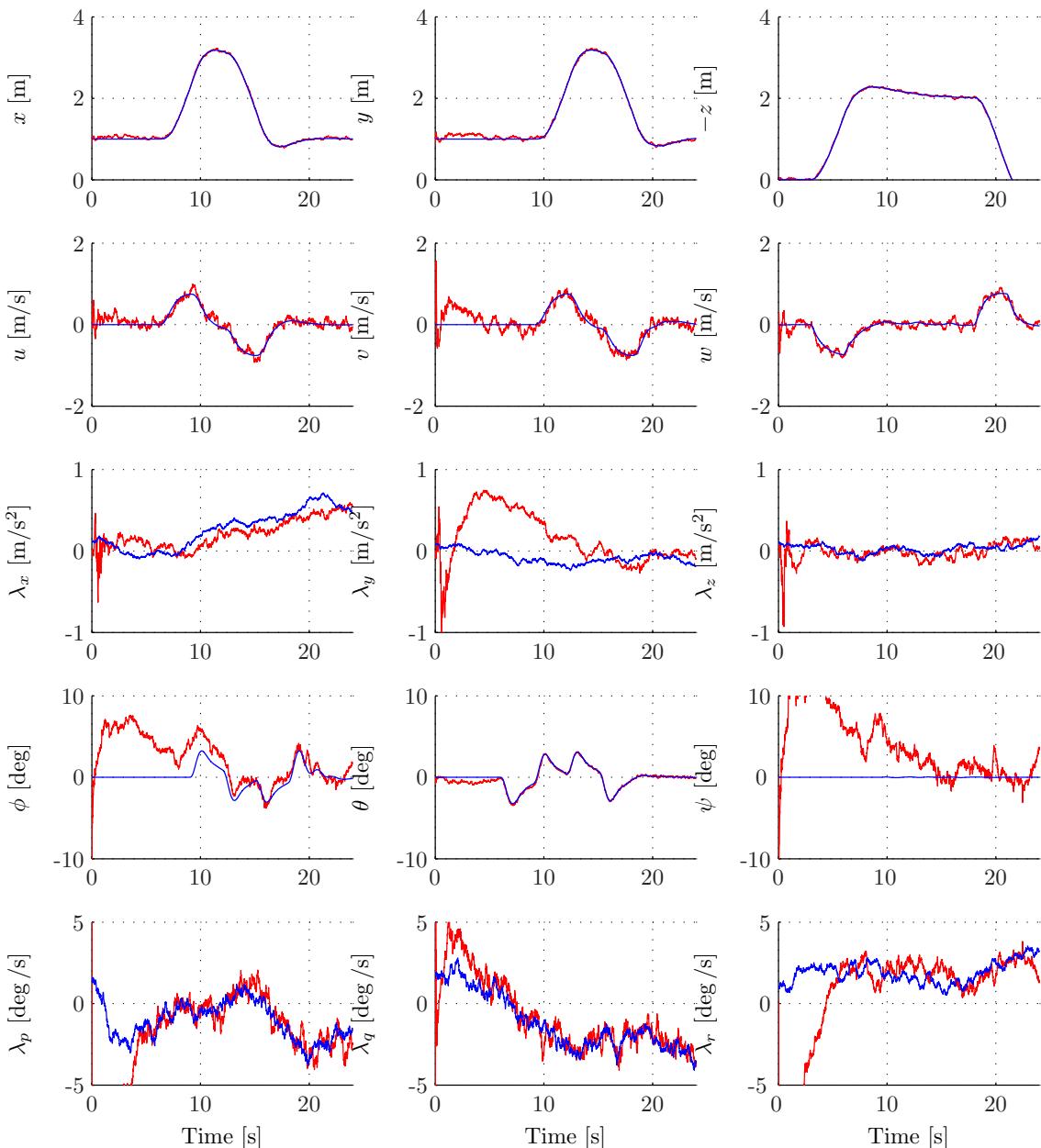


Figure H-17: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

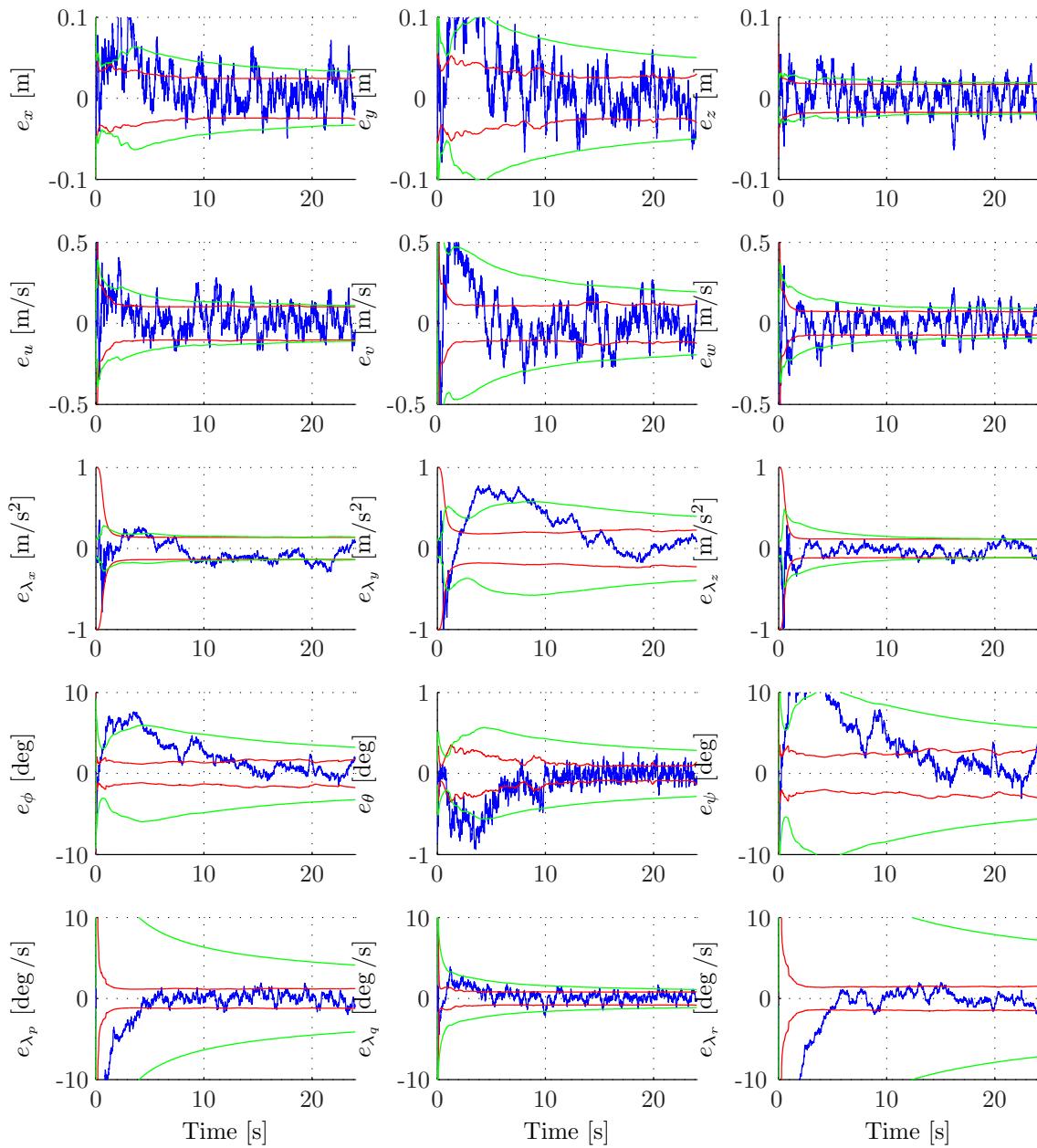


Figure H-18: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

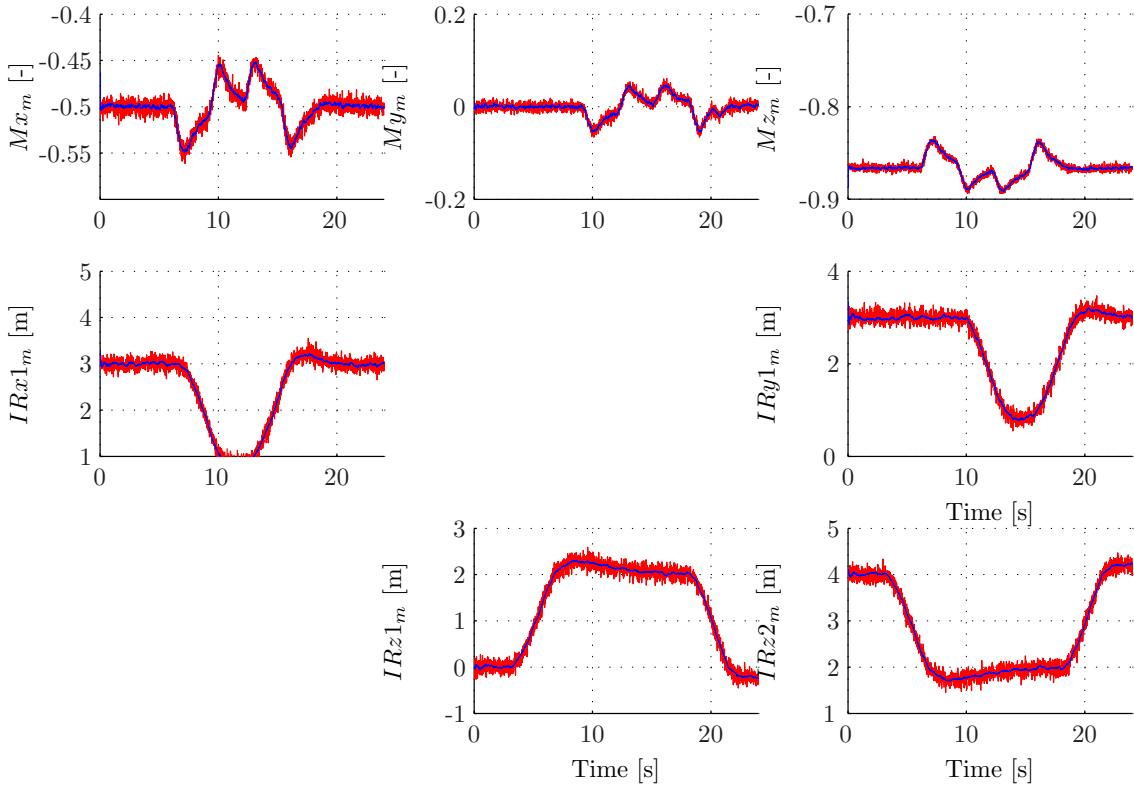


Figure H-19: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

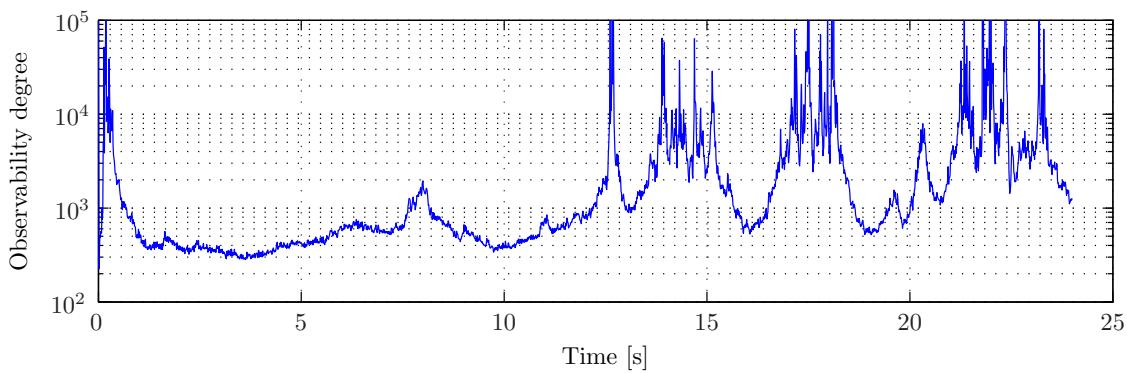


Figure H-20: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

H-6 Using 3 infrared sensors, unobservable, directions X1-Y1-Z1

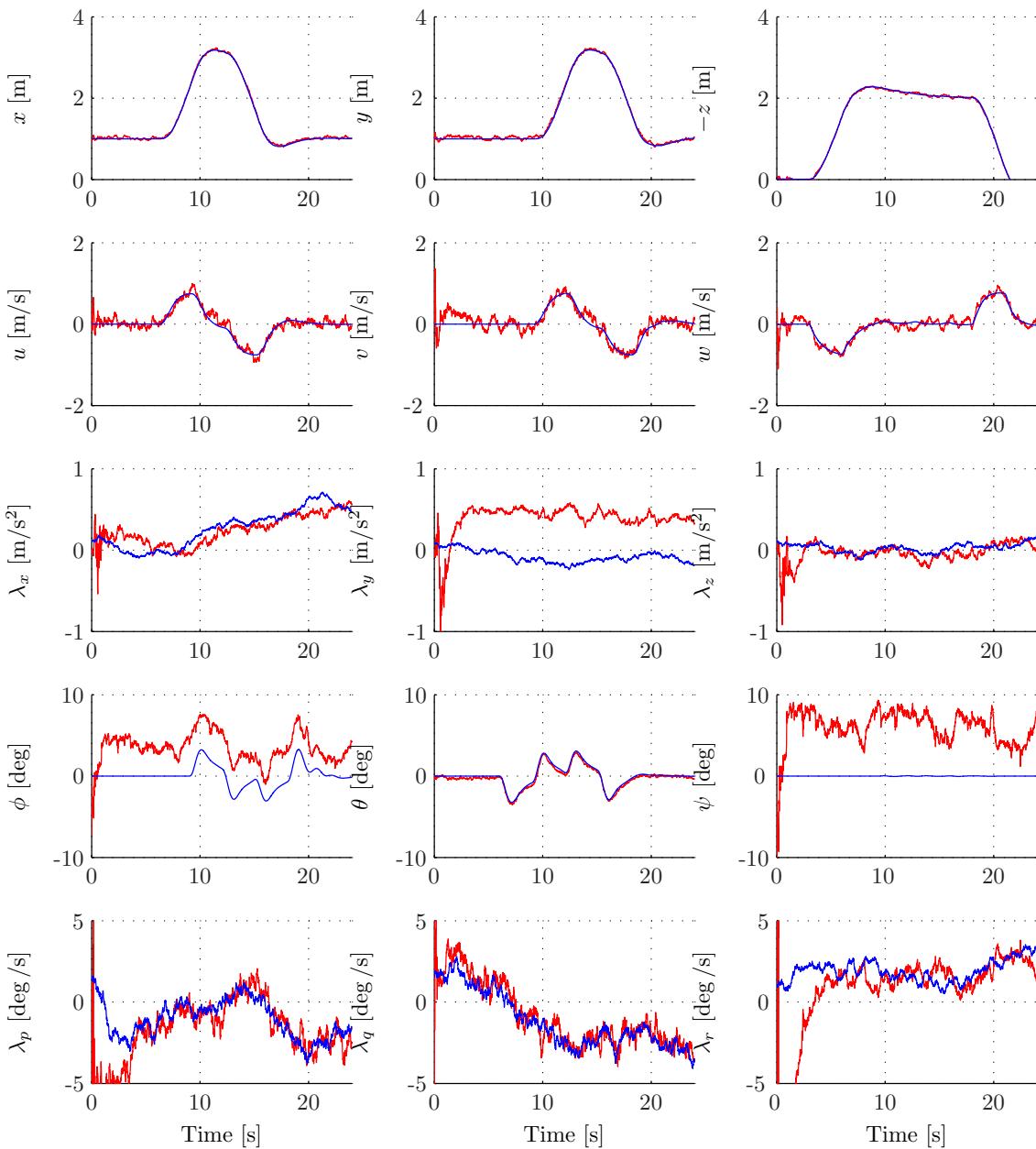


Figure H-21: FULL filter results - original states (blue) and filtered states (red).

Unobservable. Rank $\mathbf{W} = 14$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

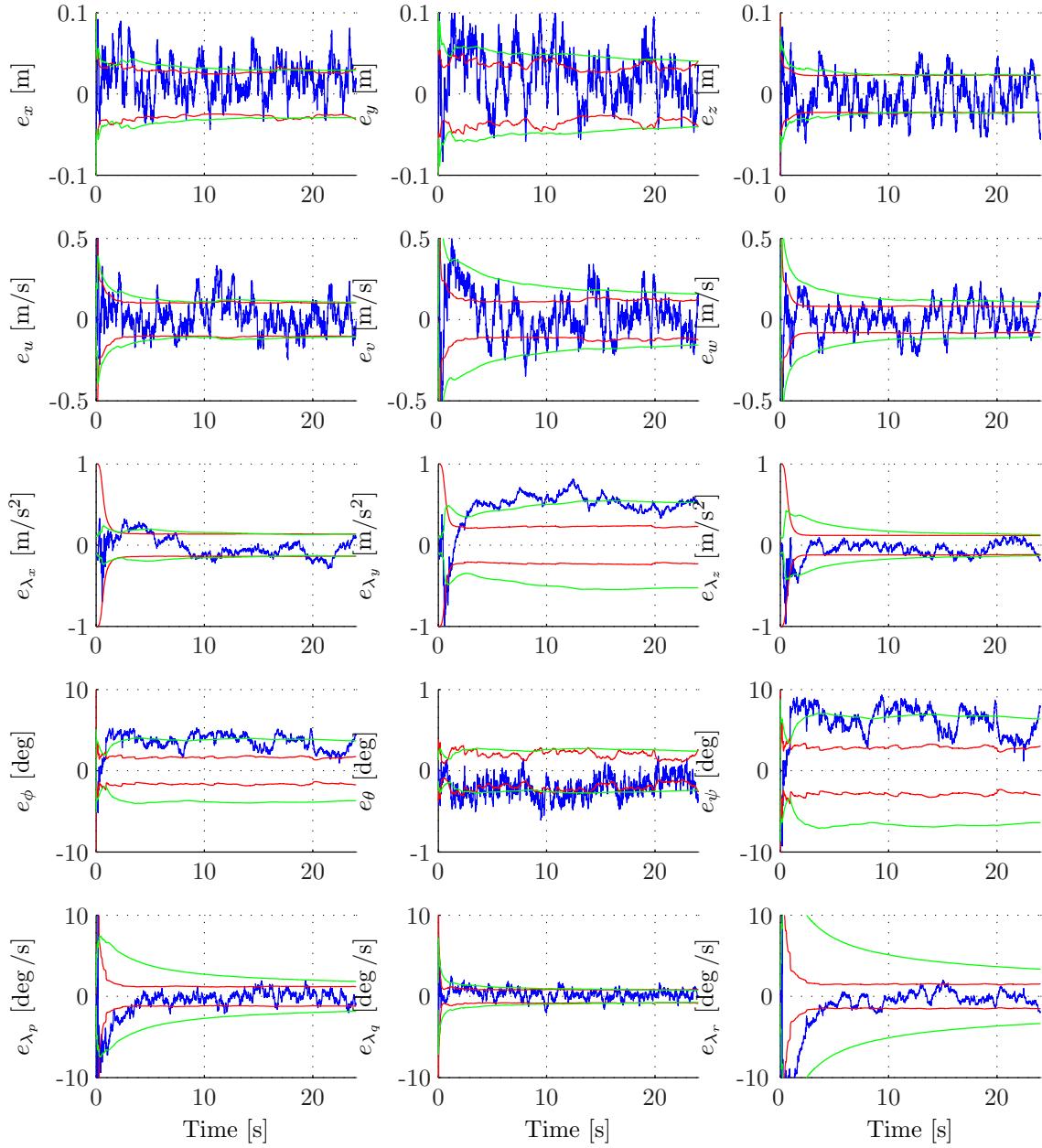


Figure H-22: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Unobservable. Rank $\mathbf{W} = 14$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

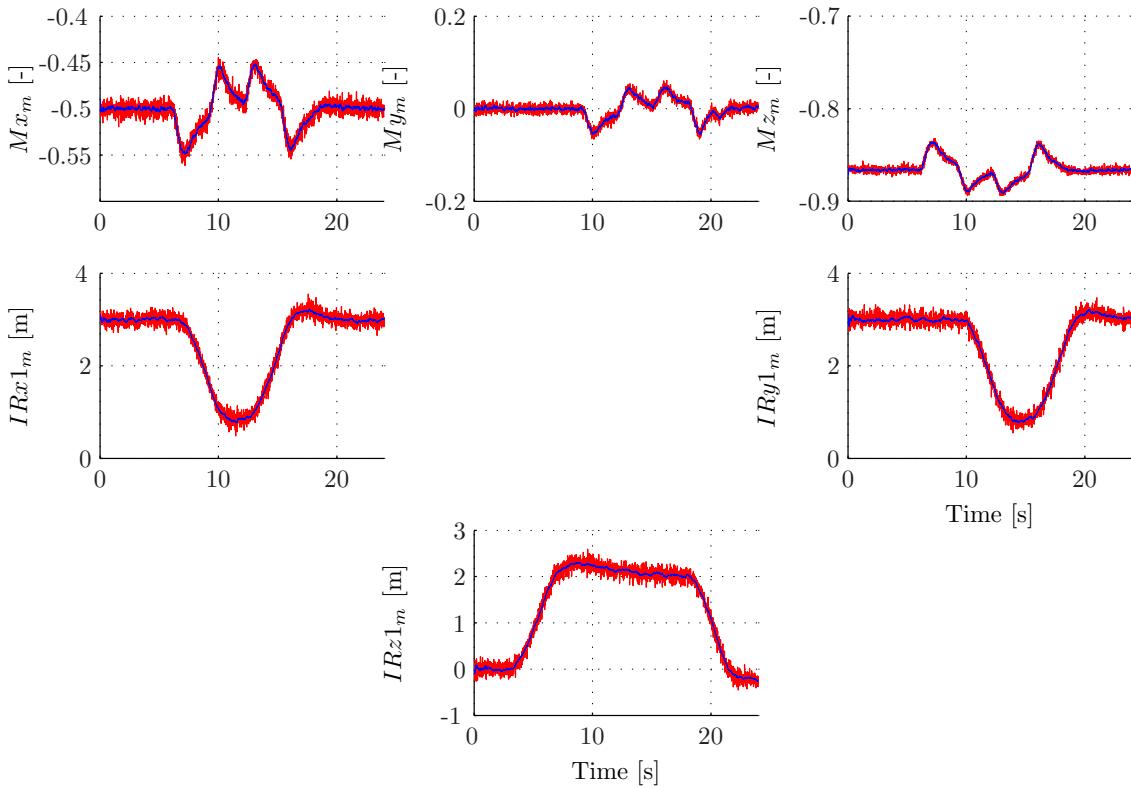


Figure H-23: FULL filter results - Original measurement (blue) and corrected measurement (red).

Unobservable. Rank $\mathbf{W} = 14$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

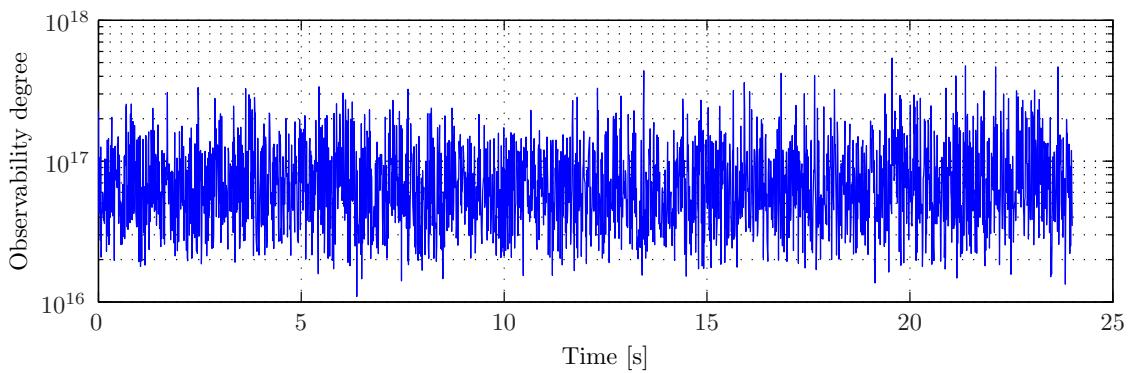


Figure H-24: FULL filter results - observability degree.

Unobservable. Rank $\mathbf{W} = 14$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 0; 1; 0; 1; 0]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.08; 0.08; 0.03; 2; 2; 1; 1; 1; 1])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01])$$

H-7 Using 6 infrared sensors, directions X1-X2-Y1-Y2-Z1-Z2, with 1/10 random walk on sensors

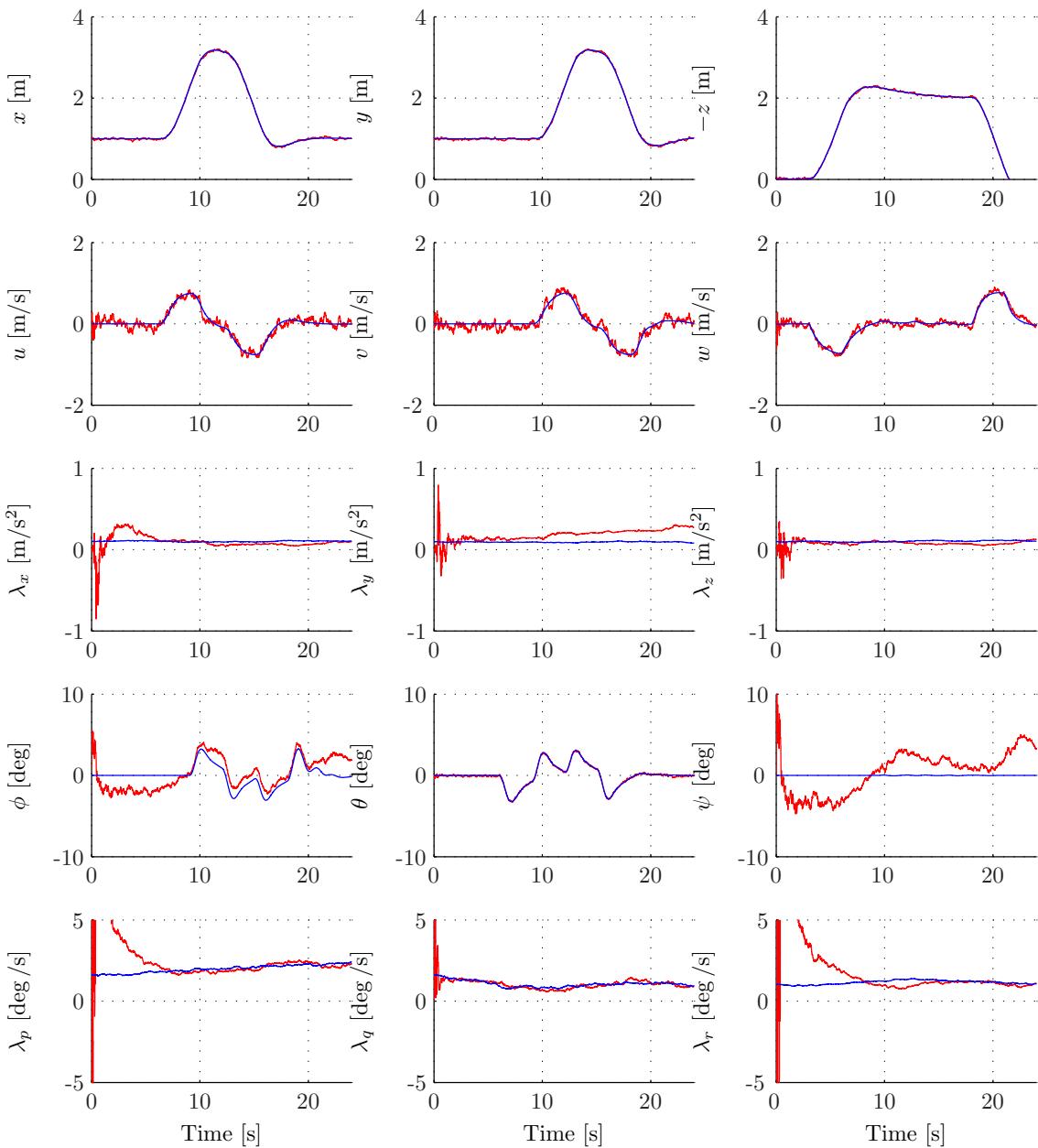


Figure H-25: FULL filter results - original states (blue) and filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003; 2; 2; 1; 0.01; 0.01; 0.01; 0.01;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

H-7 Using 6 infrared sensors and 1/10 random walk, directions X1-X2-Y1-Y2-Z1-Z2

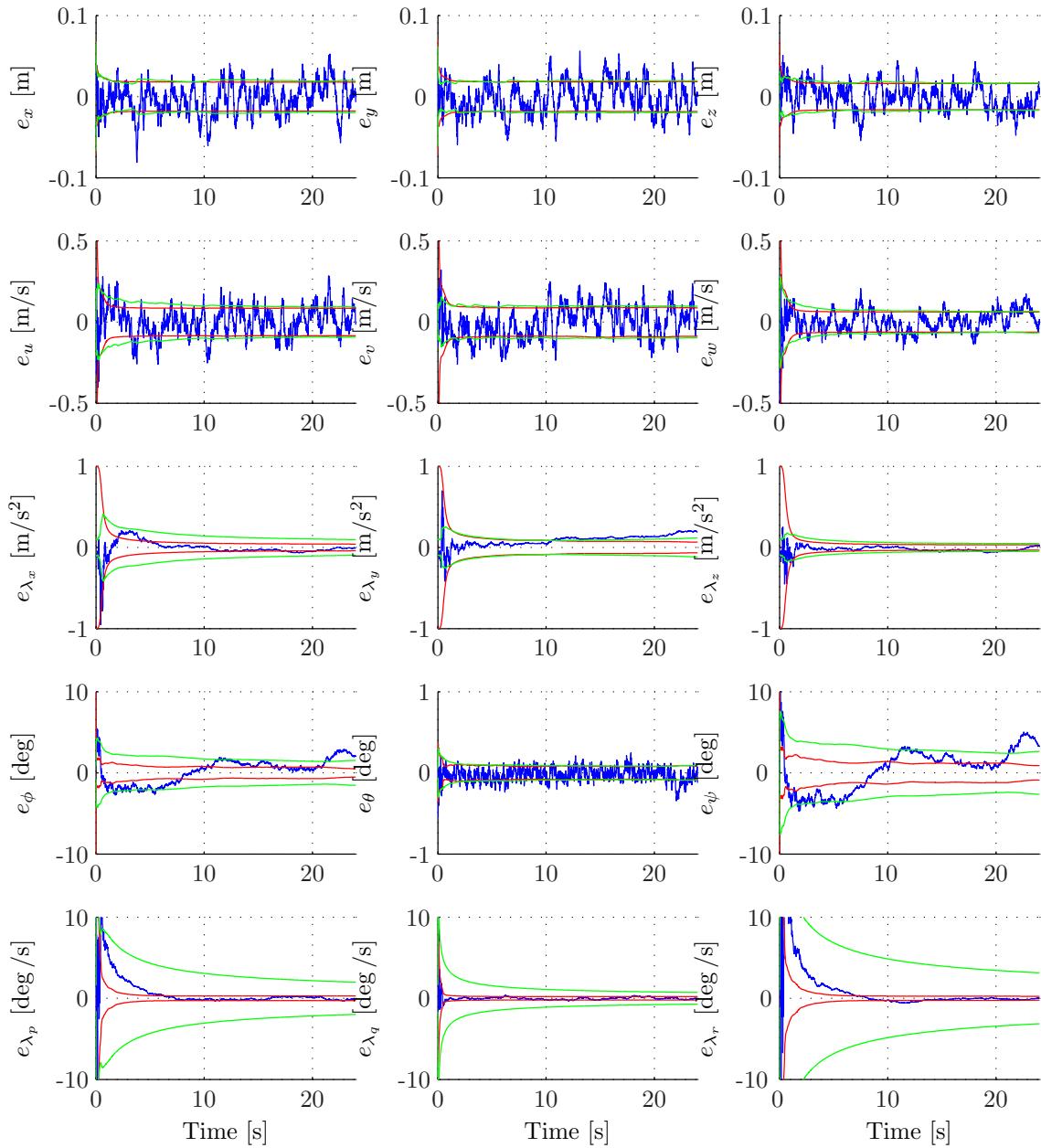


Figure H-26: FULL filter results - Error (blue), standard deviation of the state estimation error (red) and RMSE (green).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003; 2; 2; 1; 0.01; 0.01; 0.01; 0.01;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

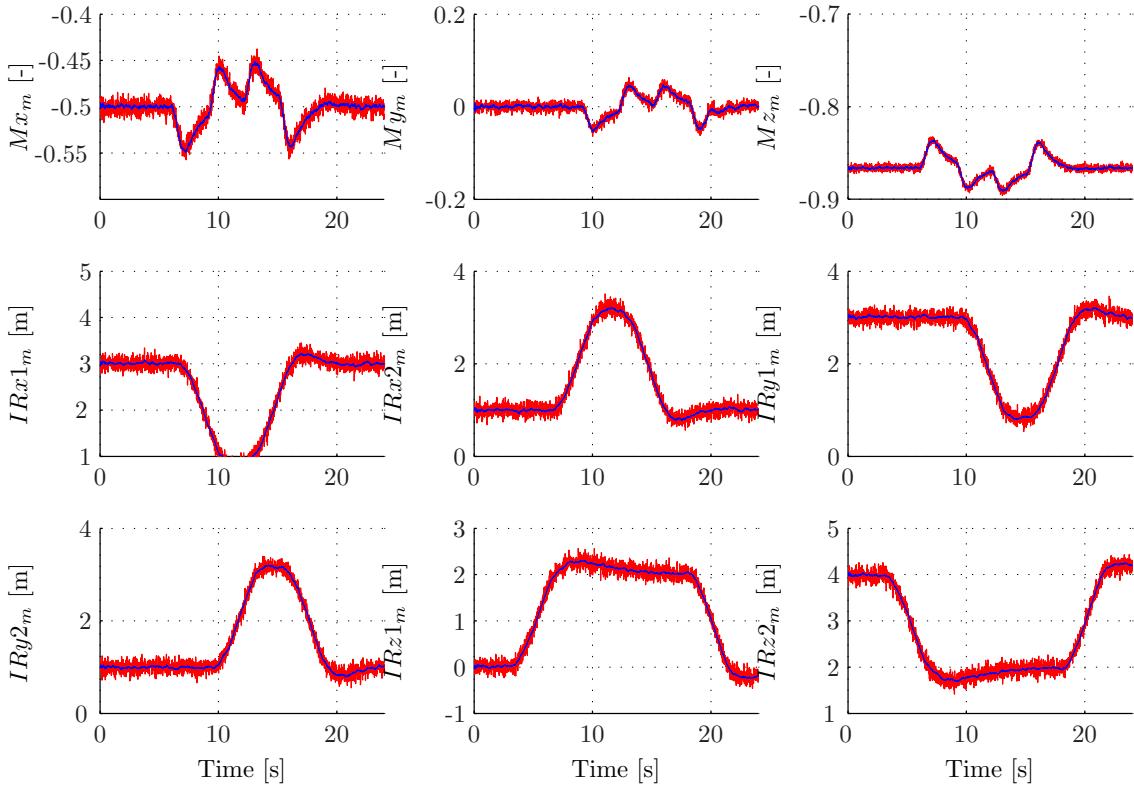


Figure H-27: FULL filter results - Original measurement (blue) and corrected measurement (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003; 2; 2; 1; 0.01; 0.01; 0.01;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

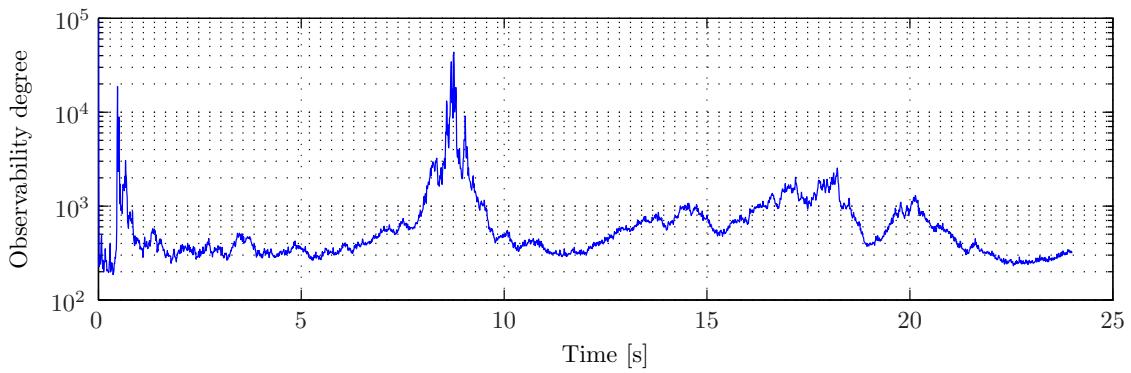


Figure H-28: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 1; 1]$$

$$\mathbf{Q} = diag([0.005; 0.002; 0.003; 0.0008; 0.0008; 0.0003; 2; 2; 1; 0.01; 0.01; 0.01;])$$

$$\mathbf{R} = diag([3e-005; 5e-005; 6e-006; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

Appendix I

Full Kalman filtering tested with real data

I-1 Using 5 infrared sensors - Pendulum movement - Motors off

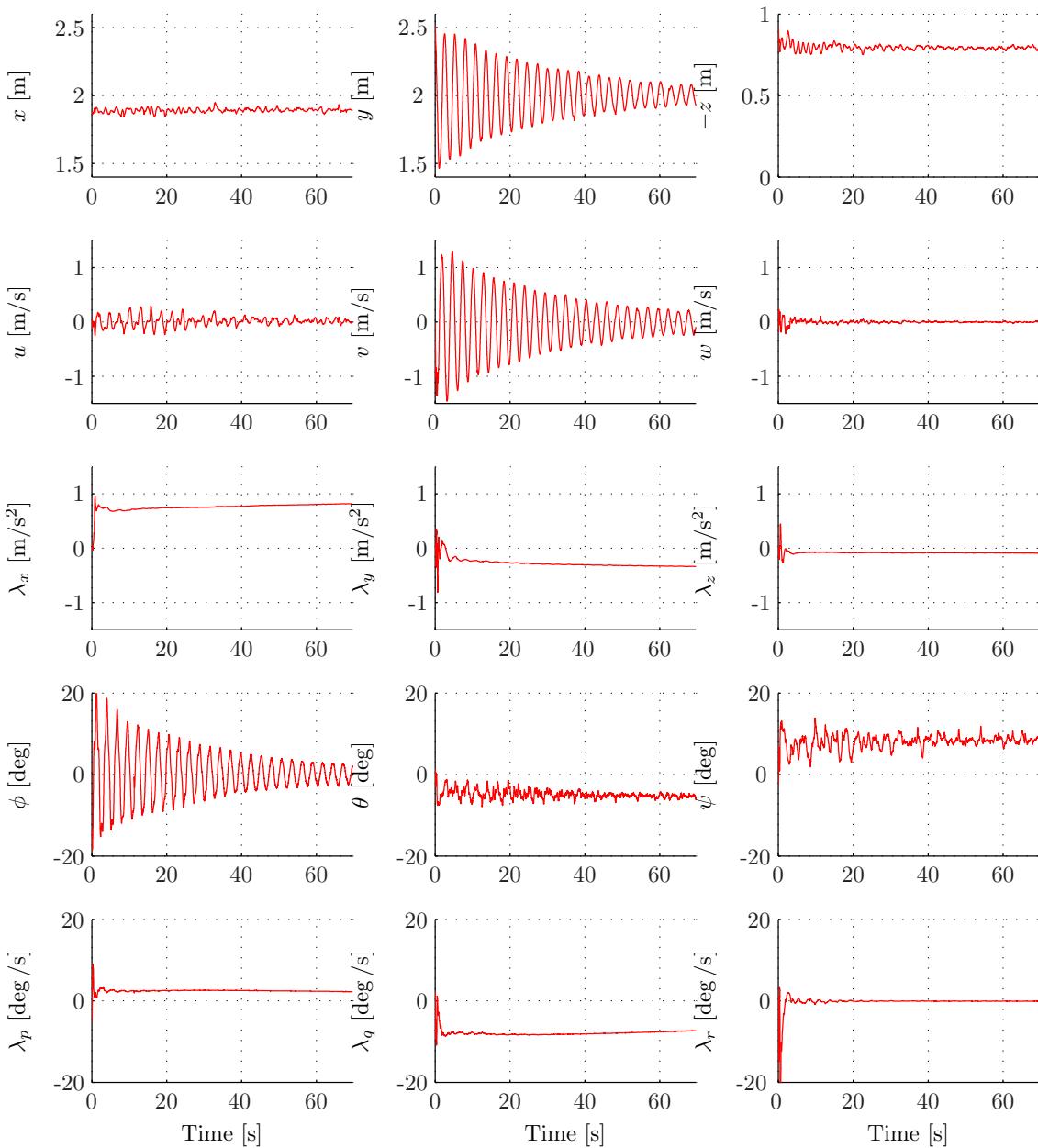


Figure I-1: FULL filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

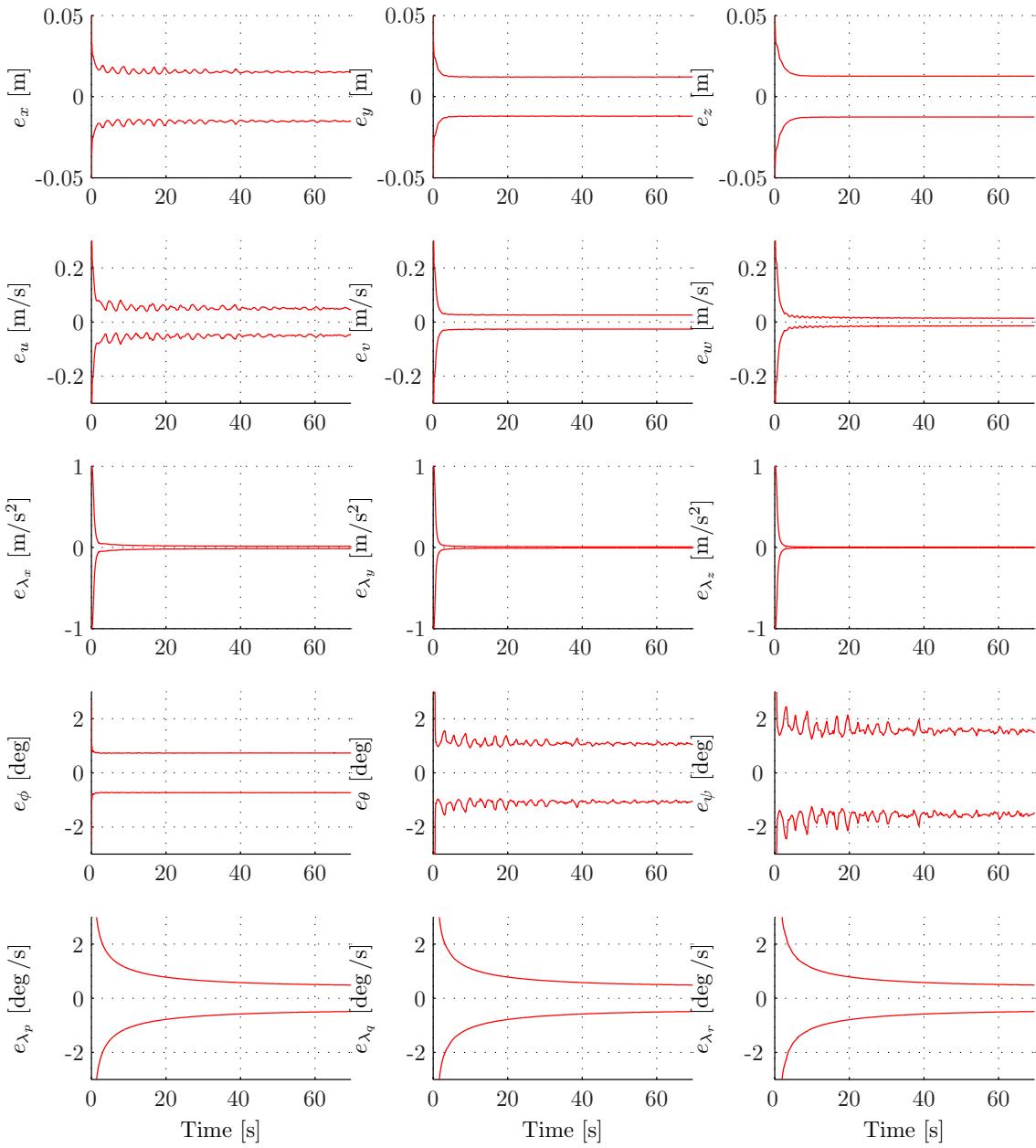


Figure I-2: FULL filter results - standard deviation of the state estimation error (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

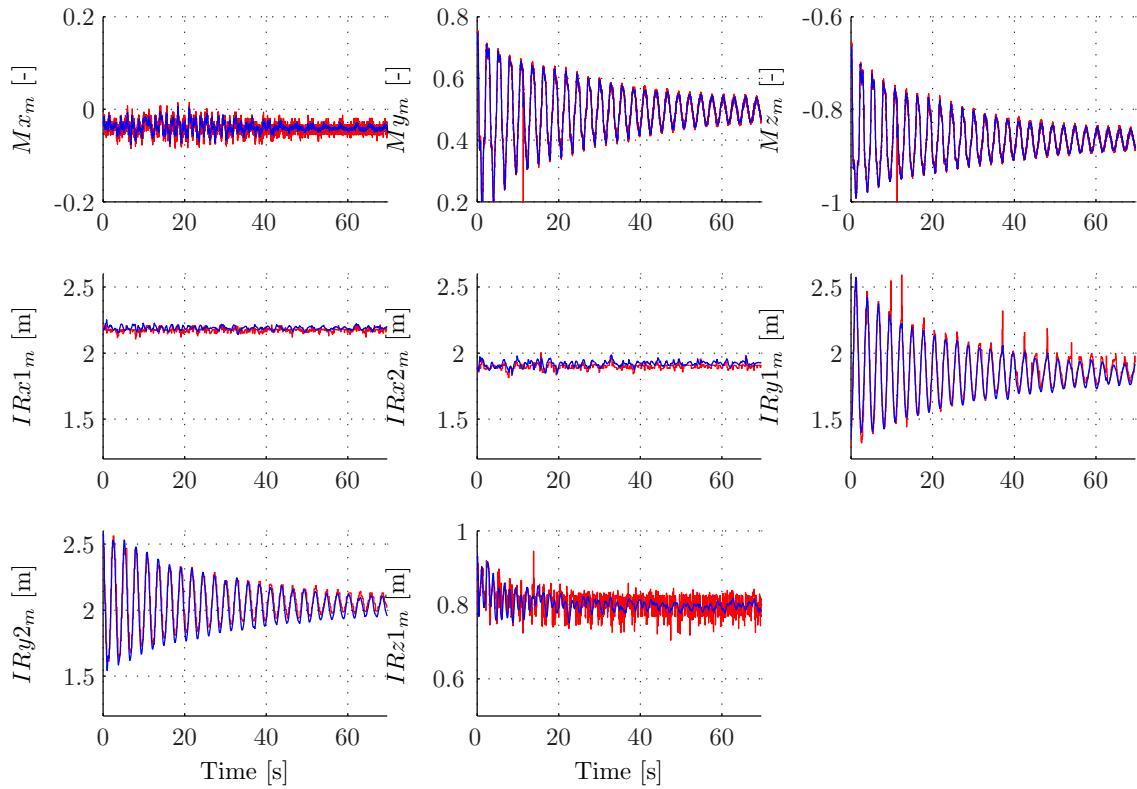


Figure I-3: FULL filter results - Original measurement (blue) and corrected measurement (red).
Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

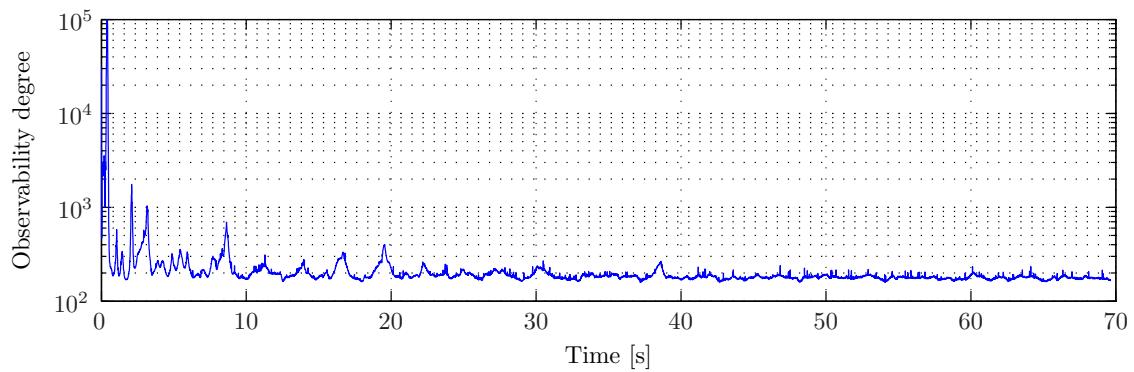


Figure I-4: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

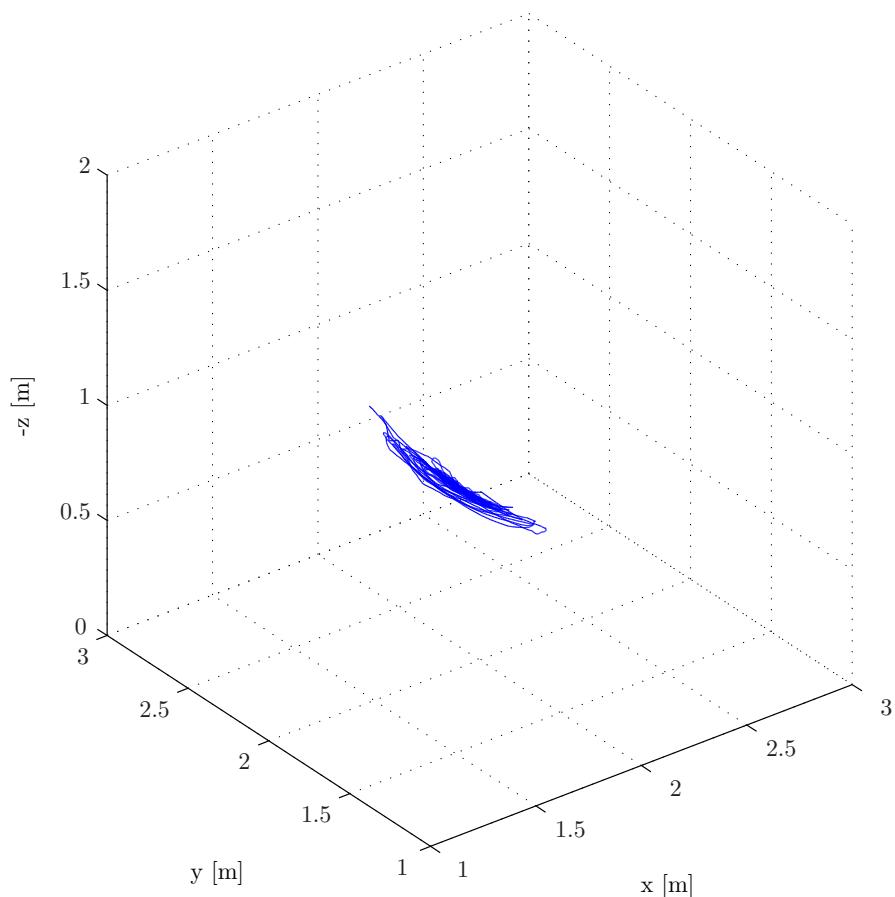


Figure I-5: FULL filter results - 3D plot.

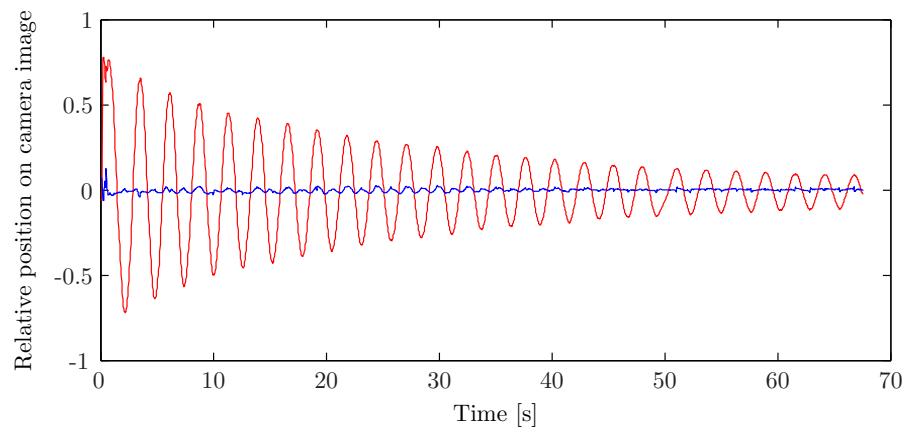


Figure I-6: FULL - Tracing of the relative motion by a camera in X (blue) and Y (red) position.

I-2 Using 5 infrared sensors - Pendulum movement - Motors on

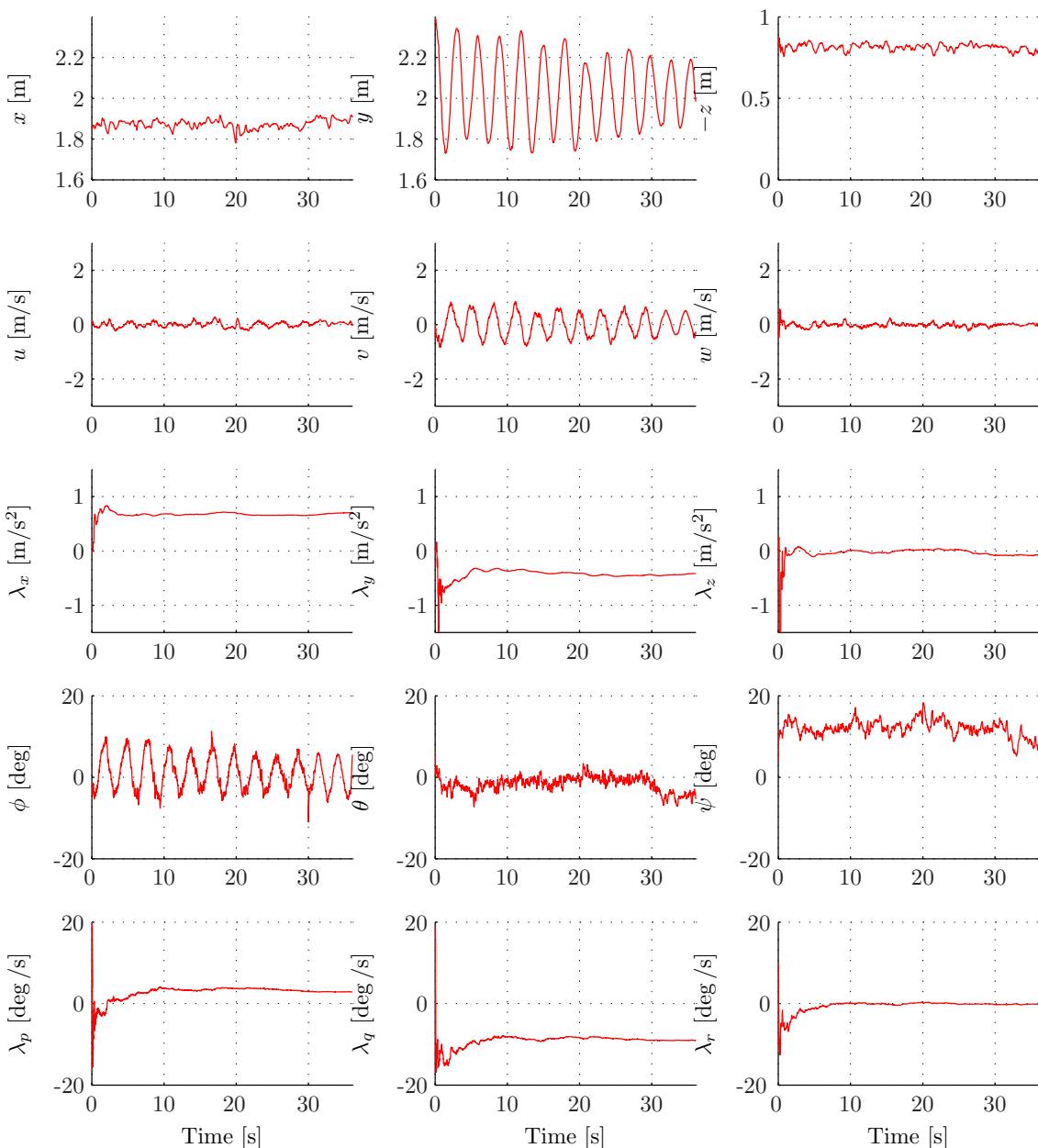


Figure I-7: FULL filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.003; 0.003; 0.003; 0.003; 0.0001; 0.01;])$$

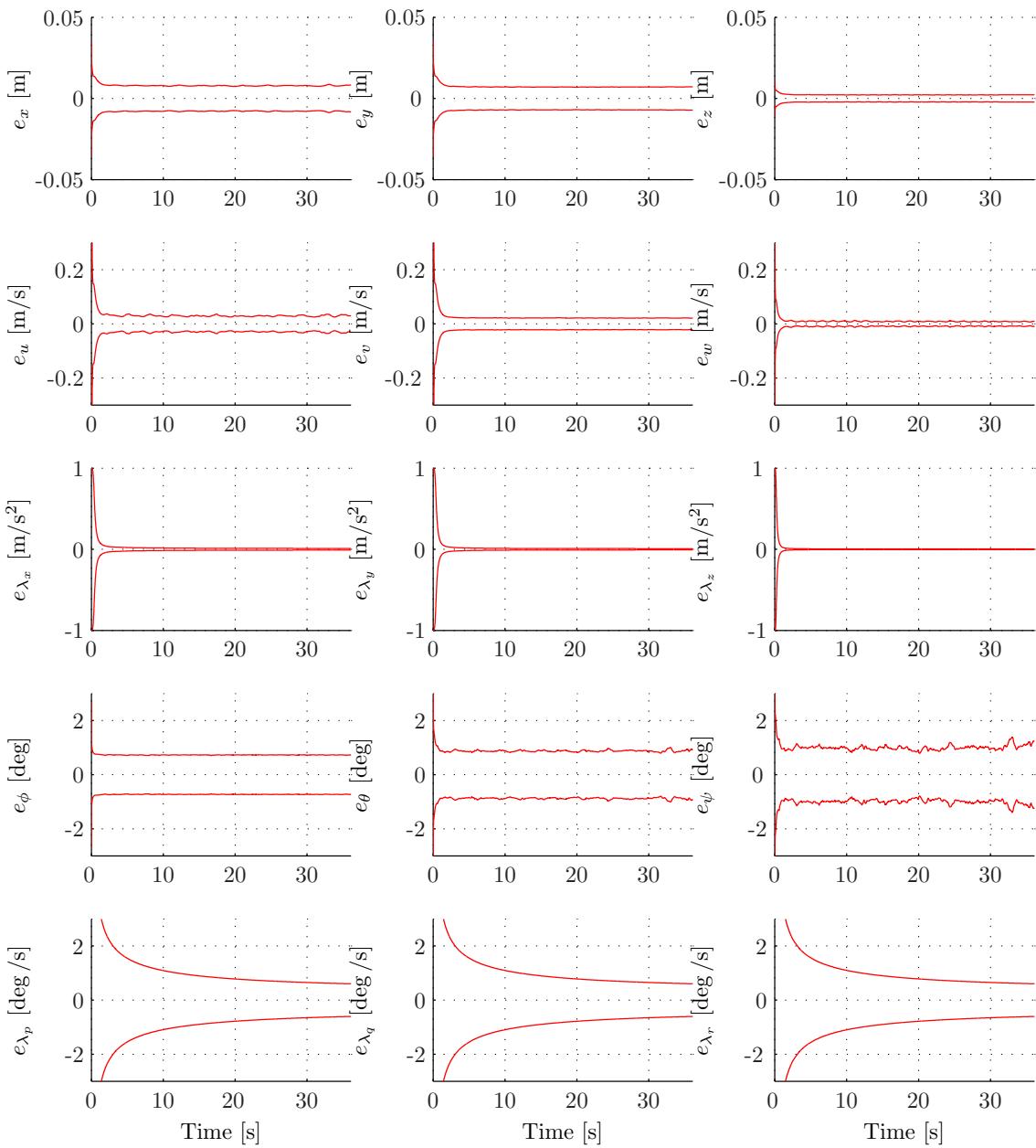


Figure I-8: FULL filter results - standard deviation of the state estimation error (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.003; 0.003; 0.003; 0.003; 0.0001; 0.01; 0])$$

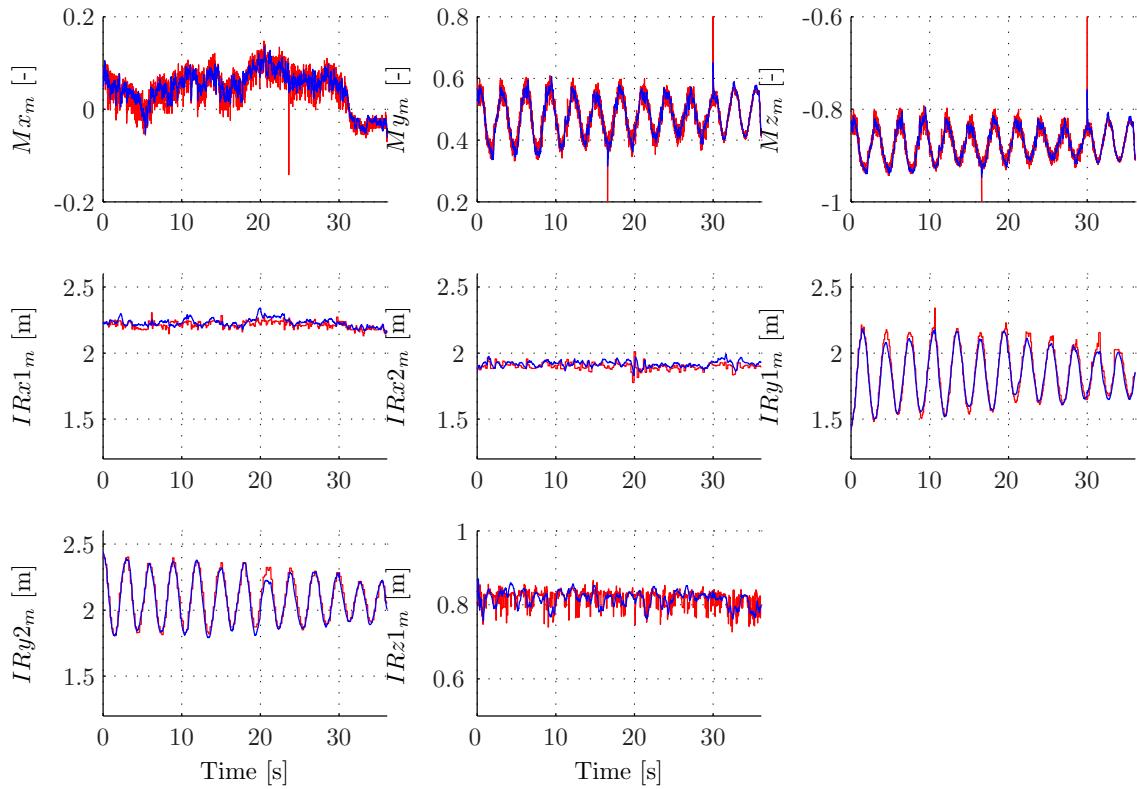


Figure I-9: FULL filter results - Original measurement (blue) and corrected measurement (red).
Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.003; 0.003; 0.003; 0.003; 0.0001; 0.01;])$$

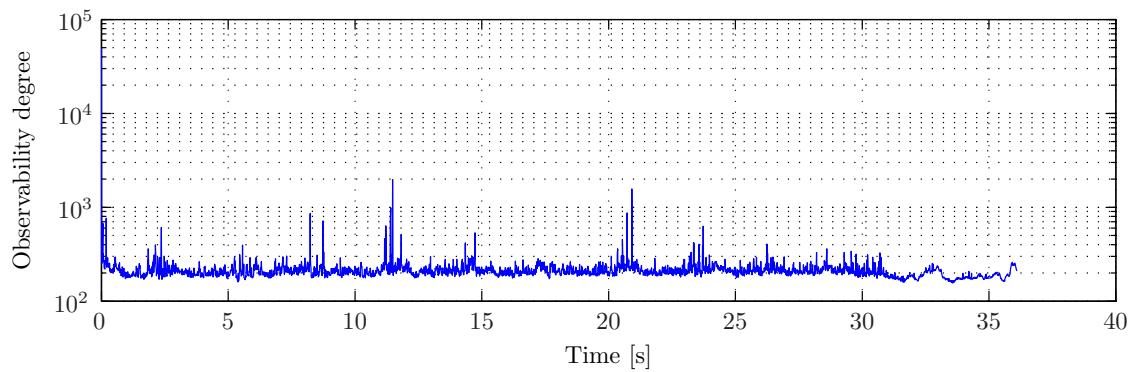


Figure I-10: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.003; 0.003; 0.003; 0.003; 0.0001; 0.01;])$$

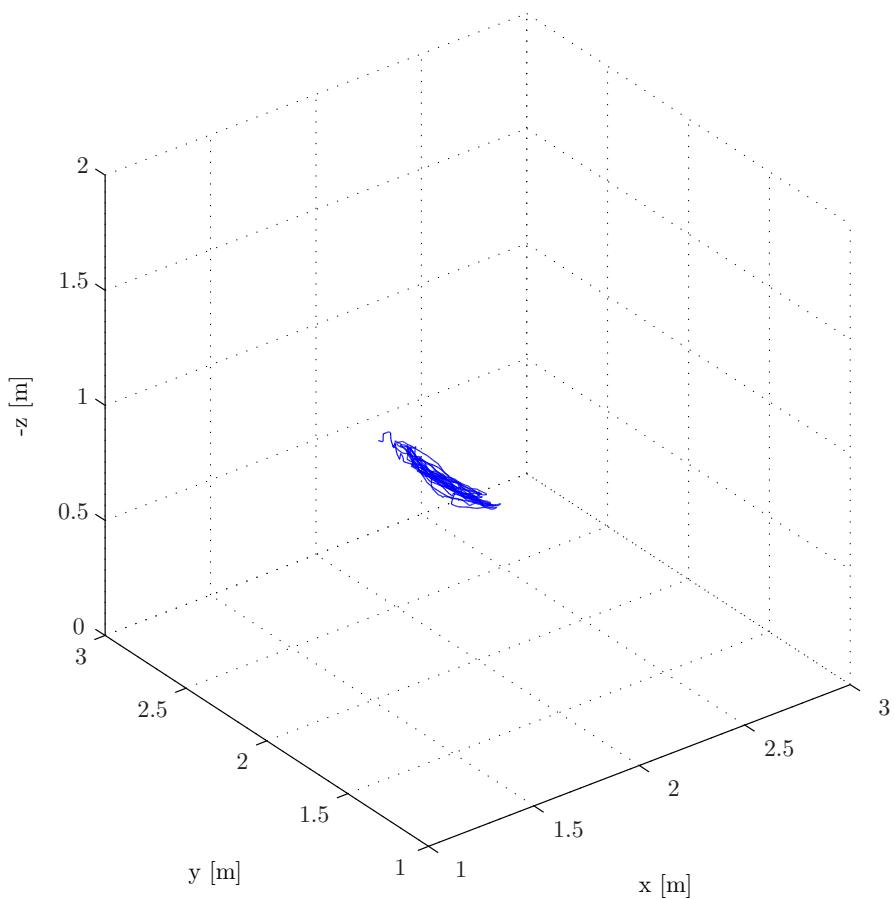


Figure I-11: FULL filter results - 3D plot.

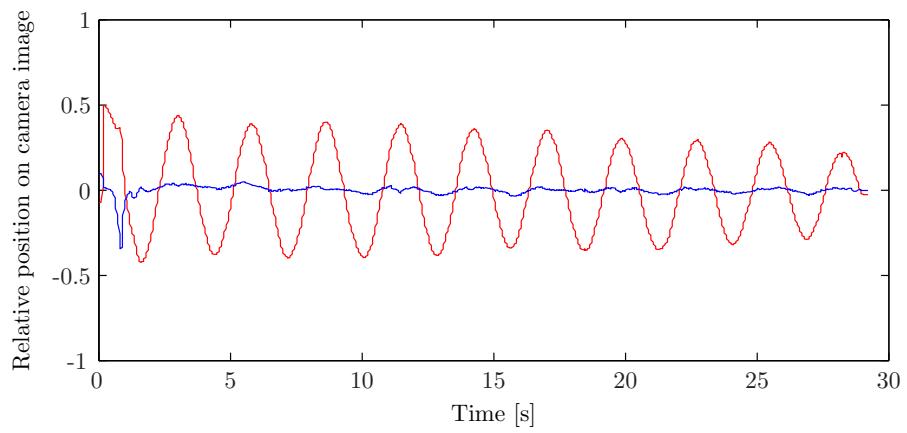


Figure I-12: FULL - Tracing of the relative motion by a camera in X (blue) and Y (red) position.

I-3 Using 5 infrared sensors - Hand movement Motors off

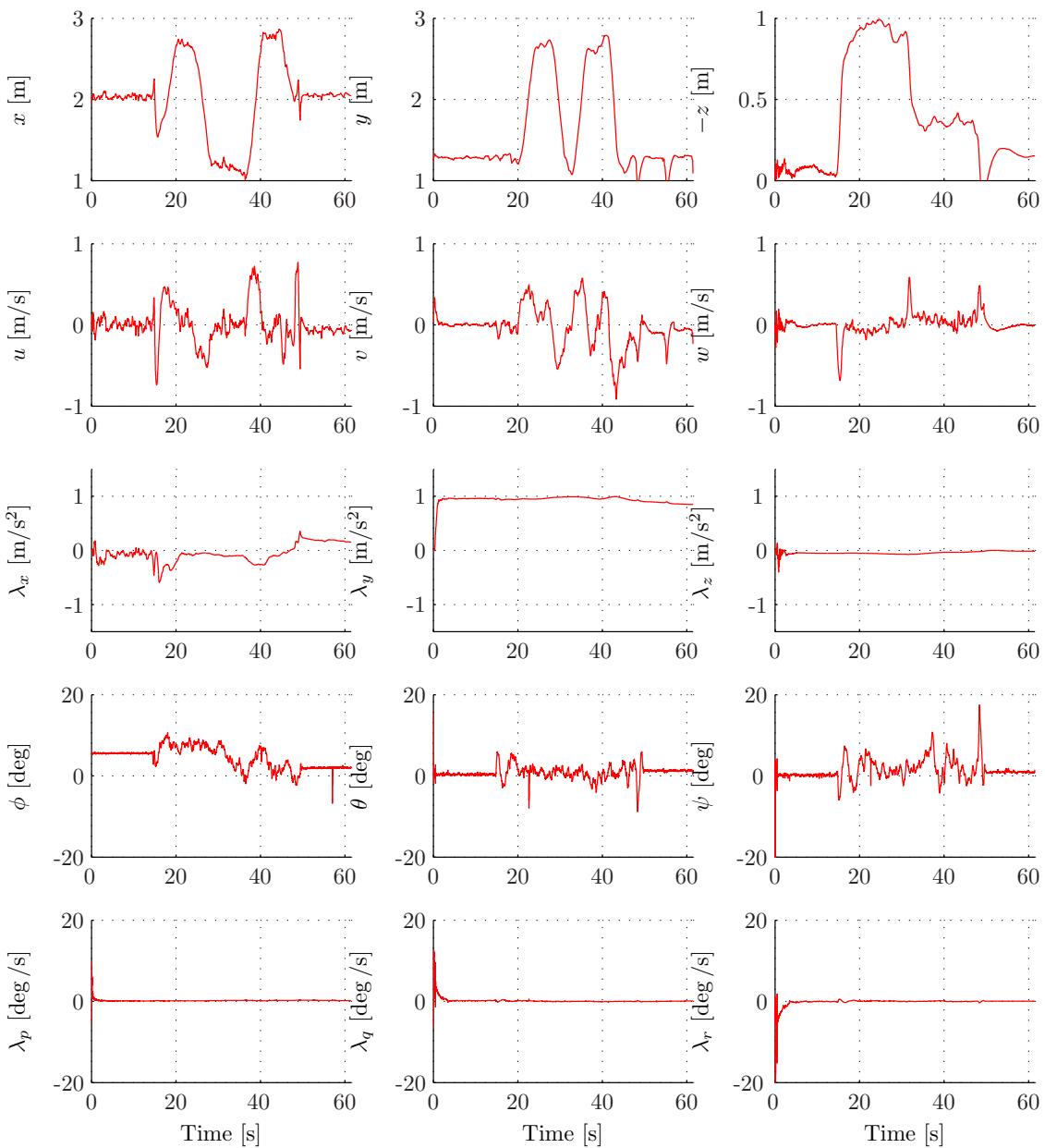


Figure I-13: FULL filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.09; 0.09; 0.09; 0.09; 0.09; 0.09; 0.01;])$$

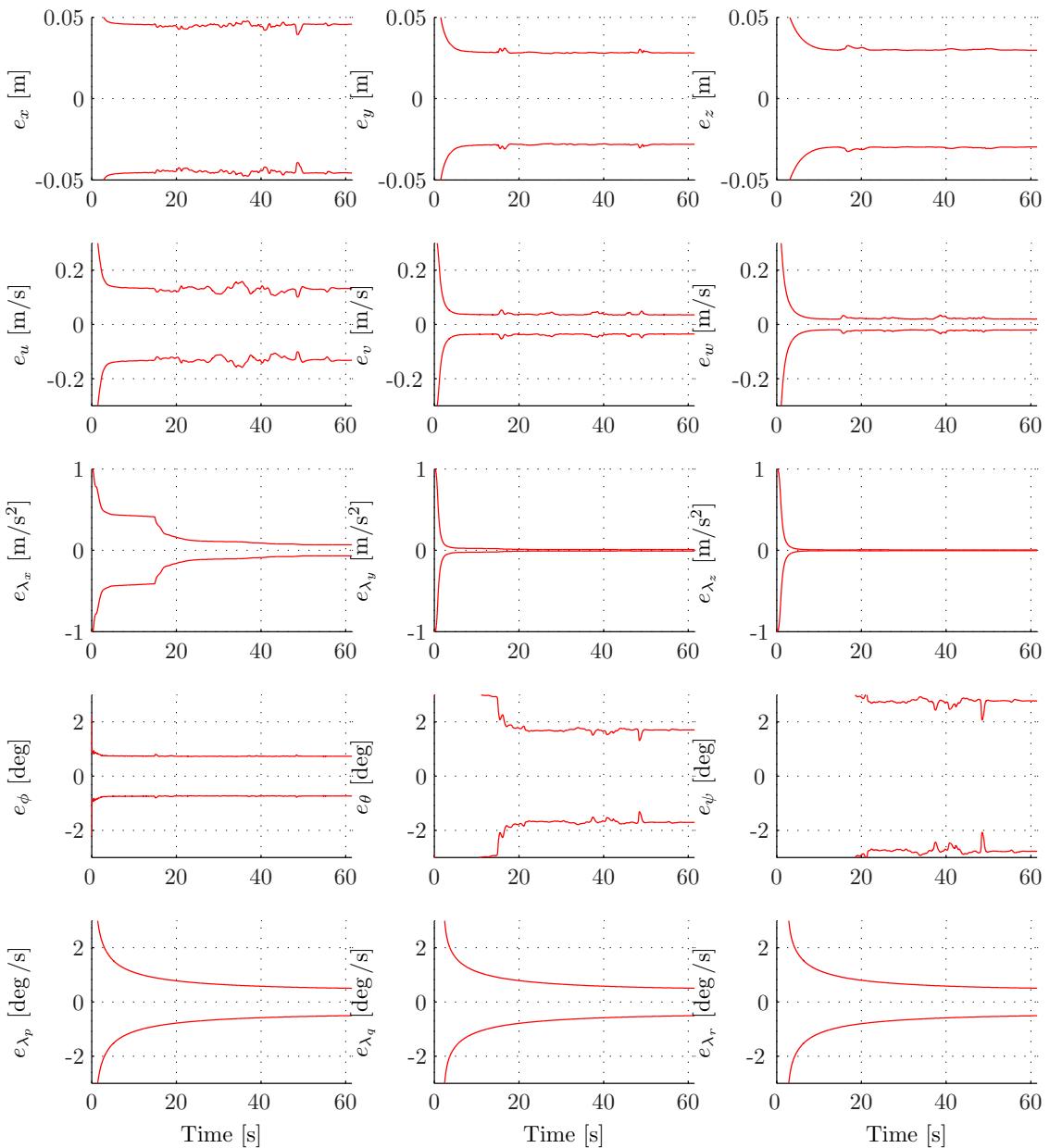


Figure I-14: FULL filter results - standard deviation of the state estimation error (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.09; 0.09; 0.09; 0.09; 0.09; 0.09; 0.01;])$$

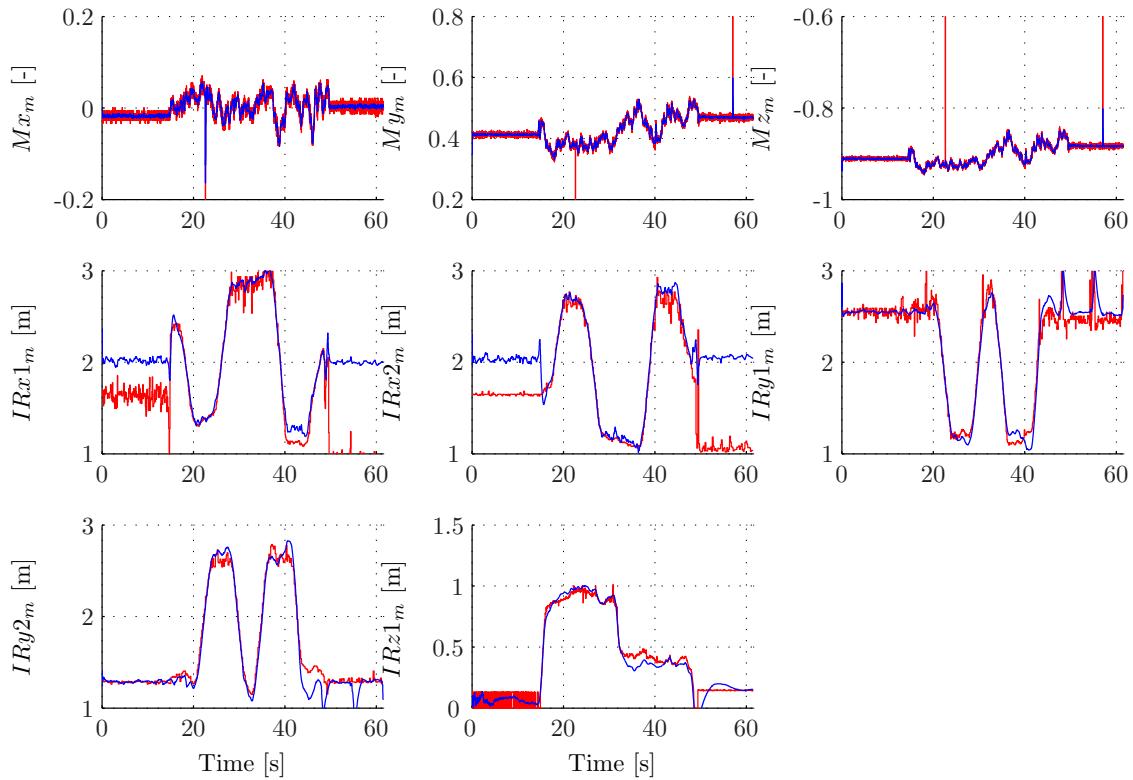


Figure I-15: FULL filter results - Original measurement (blue) and corrected measurement (red). Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.09; 0.09; 0.09; 0.09; 0.09; 0.09; 0.01;])$$

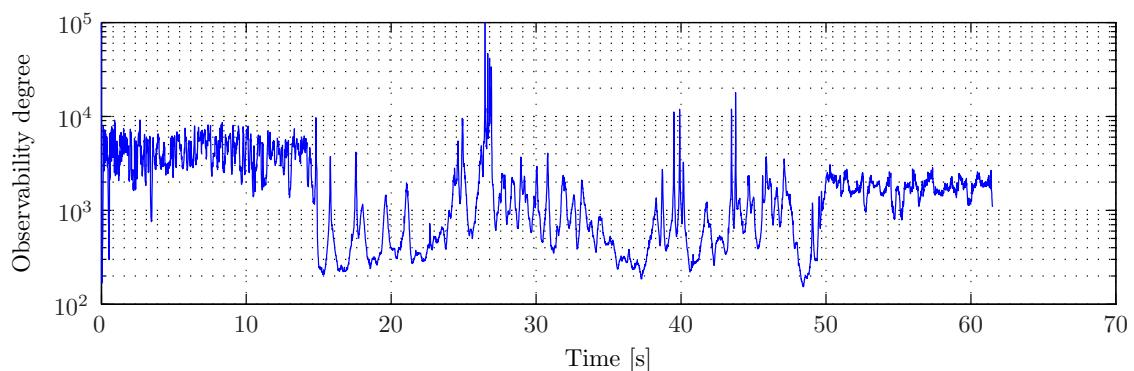


Figure I-16: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.09; 0.09; 0.09; 0.09; 0.09; 0.09; 0.01;])$$

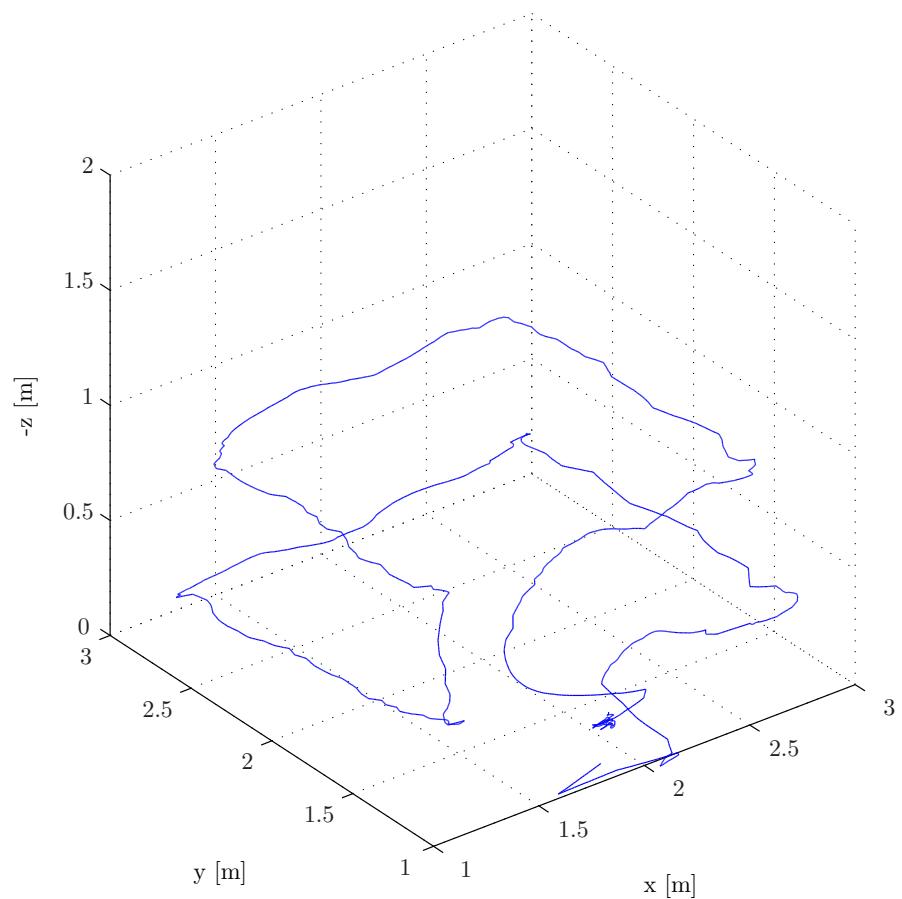


Figure I-17: FULL filter results - 3D plot.

I-4 Using 5 infrared sensors - Actual flight data

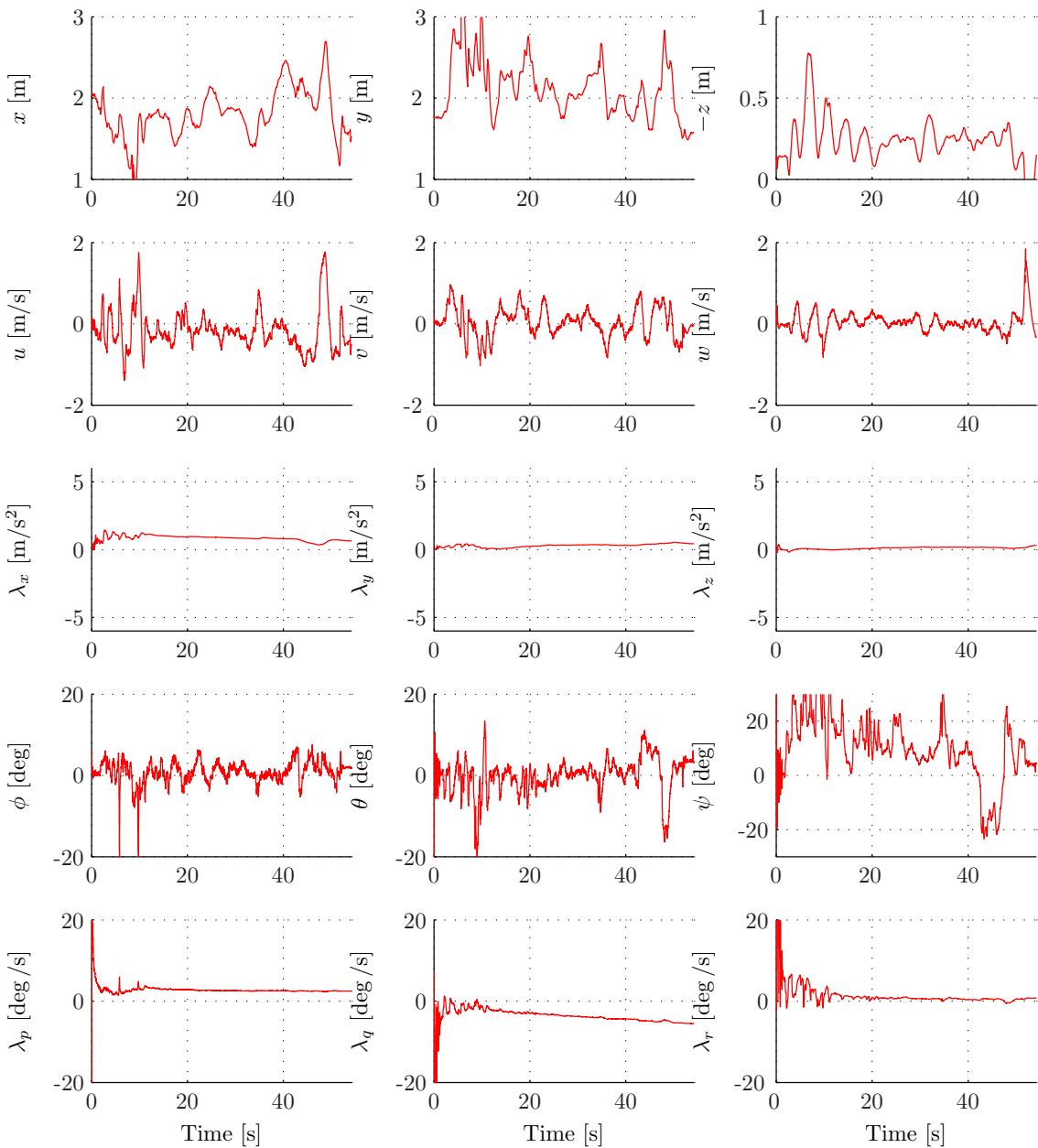


Figure I-18: FULL filter results - filtered states (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

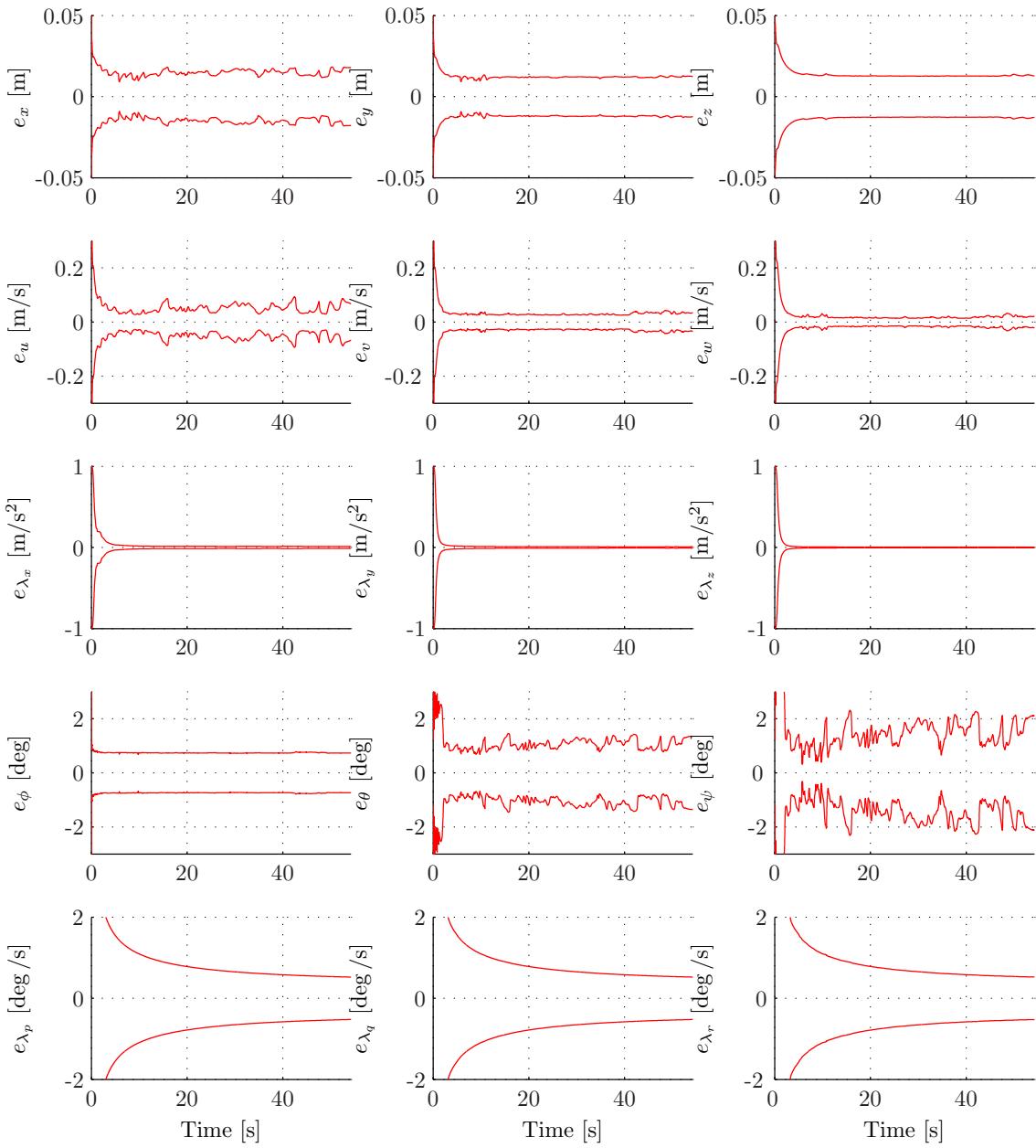


Figure I-19: FULL filter results - standard deviation of the state estimation error (red).

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

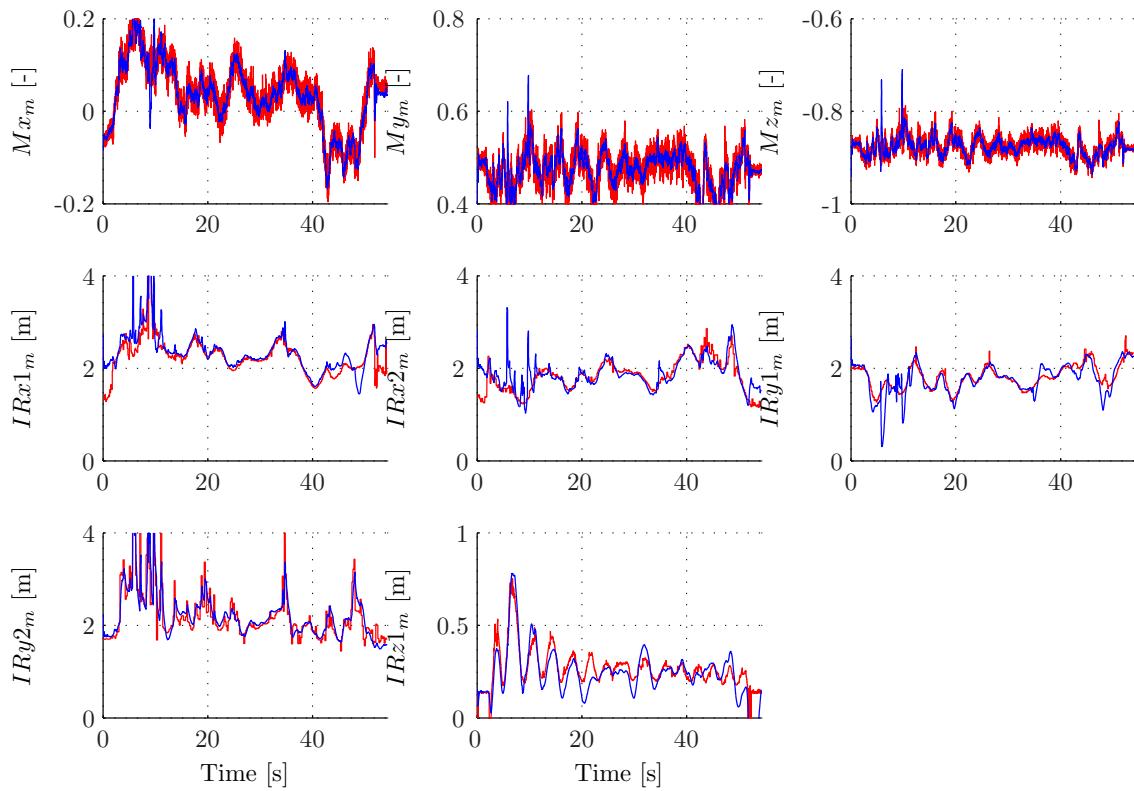


Figure I-20: FULL filter results - Original measurement (blue) and corrected measurement (red). Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

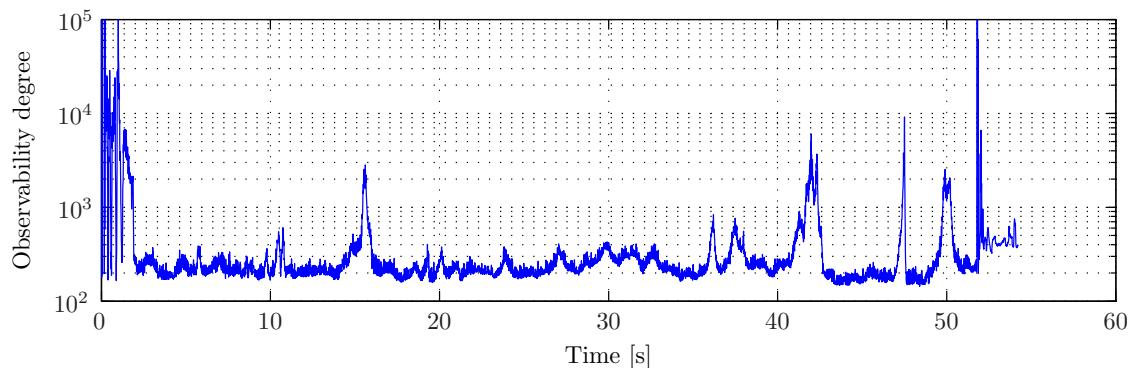


Figure I-21: FULL filter results - observability degree.

Observable. Rank $\mathbf{W} = 15$.

$$\mathbf{H}_{select} = [1; 1; 1; 1; 1; 1; 1; 0]$$

$$\mathbf{Q} = diag([0.4; 0.4; 0.4; 0.0001; 0.0001; 0.0001; 0.01; 0.01; 0.01; 0.0004; 0.0004; 0.0004;])$$

$$\mathbf{R} = diag([0.0009; 0.0009; 0.0009; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01; 0.01;])$$

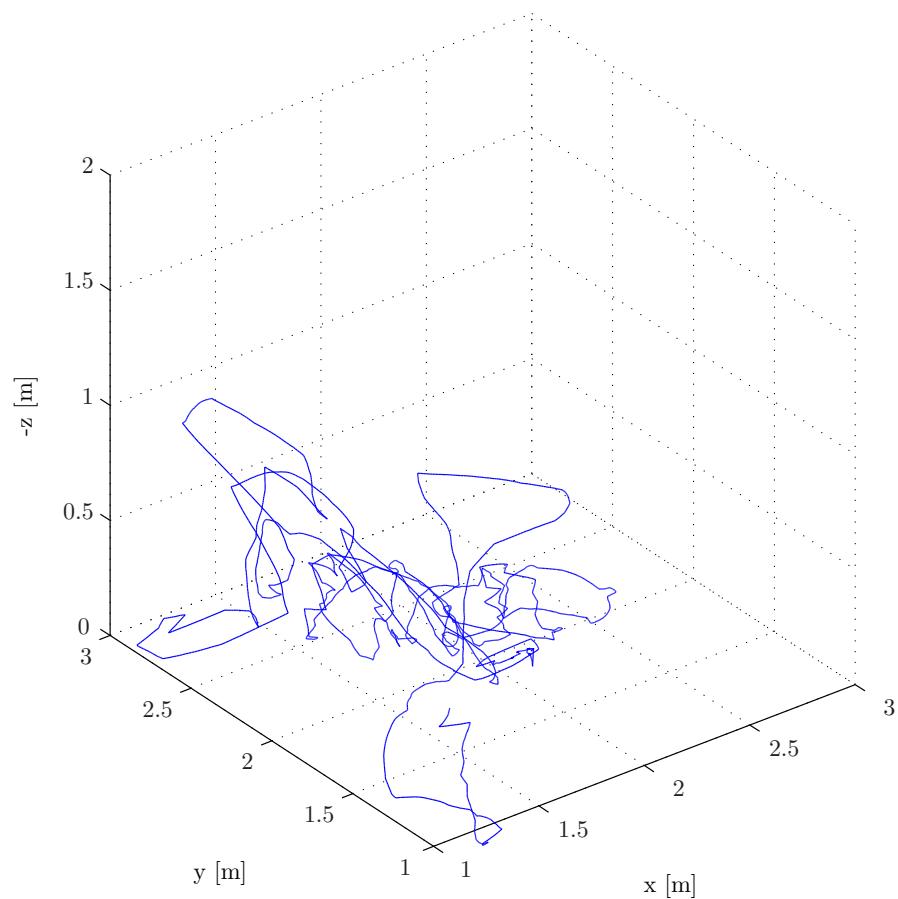


Figure I-22: FULL filter results filtered states positio states.