Fault-tolerant Multirotor Systems

Autumn Term 2011
Abstract

A growing concern among the users of multirotor MAV is the safety and reliability of their systems. This work tries to minimize these risks. First an analysis is performed to identify possible threats. Concepts against the identified risks are presented.

Rotor and motor failures are considered as the risk of highest priority. Based on this analysis a fault-tolerant control allocator is developed that makes full use of the attainable control set. The control allocation problem is stated using a parametric programming formulation and is solved for an explicit solution. Further a controllability study is presented to assess the performance degradation under failures. The optimal rotor-turn arrangements in terms of fault-tolerance is investigated using the attainable control set method.

In a last step the fault-tolerant control system is demonstrated on a real hexacopter system. Experiments that show a good performance of the control allocator under failures are presented. Additionally the reconfiguration delay is identified as an uncritical parameter.
Acknowledgments

First of all I would like to thank my supervisors Samir Bouabdallah and Konrad Rudin at Autonomous Systems Lab for their assistance and letting me work on this interesting topic.

Further my deepest appreciation goes to Guillaume Ducard and Minh Duc Hua for their great support throughout this work and also for giving me the opportunity to work in Nice at the I3S lab at the beginning of the project. Without their valuable input and the discussions this work would not have been possible in this form.

Also I would like to thank Pascal Struppler from Skybotix for his assistance and endurance in helping me fixing the electronics and providing me with all the material needed. I am also grateful to Prof. Siegwart who made this thesis possible at the Autonomous Systems Lab.

Further I would like to thank Thomas Baumgartner and Stefan Bertschi for their great help in electronics and IT issues.

And most importantly I would like to thank my colleagues at the ASL for the great time and inspiring coffee breaks during this thesis.
Contents

Abstract i
Acknowledgments iii
List of Figures viii
List of Tables ix
Symbols xi

1 Introduction 1
  1.1 Motivation ........................................... 1
  1.2 Goals .............................................. 1

2 Risk Analysis and Concepts 3
  2.1 Risk Analysis for a Multirotor System .................. 3
    2.1.1 Functional Block Description ...................... 4
    2.1.2 Nominal Operation Mode ............................ 6
    2.1.3 Failure Mode and Effect Analysis ................ 6
    2.1.4 Identified High Priority Risks .................... 7
  2.2 Concepts for Improving the Fault-tolerance .......... 7
    2.2.1 Propulsion System ................................ 7
    2.2.2 Motor Communication Bus .......................... 8
    2.2.3 Backup Flight Electronics ....................... 12
    2.2.4 Battery ........................................ 13
    2.2.5 Conclusions ..................................... 13

3 Fault Detection for the Propulsion Groups 15
  3.1 Smart Actuators .................................... 16
  3.2 Rotor Efficiency Estimation .......................... 16
    3.2.1 Definition of the Rotor Efficiency ............... 16
    3.2.2 Identification Method ........................... 17
    3.2.3 Simulation Results .............................. 19

4 Controllability of Multirotor MAVs 21
  4.1 Attainable Control Set ................................ 21
  4.2 Controllability under Faults ......................... 23
  4.3 Influence of the Number of Rotors and Arrangement .. 25

5 Fault-Tolerant Control Allocation 33
  5.1 Control Allocation Problem ........................... 34
  5.2 Literature Review - Classical Control Allocation Strategies .... 35
    5.2.1 Pseudo-Inverse Solutions ....................... 35
# List of Figures

2.1 Steps of the risk analysis ............................................. 3  
2.2 Functional block diagram of a multirotor system ......................... 5  
2.3 Overview of control allocation ........................................... 8  
2.4 Standard I²C bus architecture ............................................ 8  
2.5 Concept for the fault-tolerant I²C motor communication bus .............. 11  
2.6 Overview of the fault-tolerance concept .................................. 13  
2.7 Detailed description of the fault-tolerance concept ......................... 14  

3.1 Rotor efficiency estimation - structure .................................. 17  
3.2 Rotor efficiency estimation cycle ....................................... 18  
3.3 Rotor speeds during the estimation ..................................... 19  
3.4 Euler angles during the estimation ..................................... 20  

4.1 Attainable control set for an hexacopter (T=thrust, L=roll-, M=pitch-, N=yaw-torque) ..................................................... 23  
4.2 Quantifying the performance degradation using cuts of the ACS ........... 24  
4.3 Traditional arrangements (a,c) and arrangements for improved fault-tolerance (b,d) ..................................................... 26  

5.1 Control system with control law, control allocator and failure detection 33  
5.2 Cut though the ACS polytope and the PSI subset at $N = 0, T = T_{hover}$ (red: ACS, blue: PSI) ........................................ 36  
5.3 Cut though the ACS polytope and the PSI subset at $N = 0$ ............... 37  

6.1 Structure of the proposed control allocator ................................ 39  
6.2 Geometry definition for multirotor system ................................ 41  
6.3 Explicit solution for an PNPNPN hexacopter with 61 regions (10 dimensional) ..................................................... 43  
6.4 Constrain by projecting onto the surface of the attainable control set 44  

7.1 Overview of the test environment ....................................... 47  
7.2 Used hexacopter system .................................................... 48  
7.3 Overview of the test setup ................................................ 49  
7.4 Coordinate systems used in the position controller ......................... 51  
7.5 Structure of the X-Y position controller ................................ 52  
7.6 Structure of the yaw pose controller ................................... 52  
7.7 Structure of the height controller structure ................................ 53  

8.1 Rotor numbering and torque axis definitions ................................ 56  
8.2 Angle tracking error during failure of motor 3 ................................ 56  
8.3 Position error during failure of motor 3 .................................. 57  
8.4 Tracking error of roll Angle $\phi$ for the two control allocators .......... 58  
8.5 Tracking error of pitch angle $\theta$ for the two control allocators .......... 58  
8.6 Position error for the two control allocators ................................ 59
8.7 Roll angle tracking error with different reconfiguration delays (failure of motor 3) ................................................................. 60
# List of Tables

2.1 Ratings for occurrence, severity and detection ........................... 7  
2.2 Concepts for a fault-tolerant I²C bus ................................. 10  
3.1 Results of the rotor efficiency estimation .............................. 20  
4.1 Number of different fault configurations (including the fault-free nominal case) for different number of rotors .................. 23  
4.2 Fault-tolerance comparison for different multirotor arrangements with up to two rotor failures .......................................................... 27  
4.3 Controllability table for an PNPNPN hexacopter ..................... 28  
4.4 Controllability table for an PPNNPN hexacopter ...................... 29  
4.5 Controllability table for a PNPNPNNP octocopter .................... 30  
4.6 Controllability table for a PPNNPNNN octocopter .................... 31  
4.7 Complexity of the explicit solution ..................................... 43  
6.1 Ratio of the allocatable ACS volume to the ACS volume (see section 5.2.1) for the pseudo-inverse and mpQP allocator ......... 46  
7.1 Parameters of the hexacopter ............................................. 49  
7.2 Range of euler angles ...................................................... 51  
7.3 Parameters for the X-Y controller ....................................... 52  
7.4 Parameters for the yaw controller ....................................... 53  
7.5 Parameters for the height controller .................................... 53  
A.1 System FMEA sheet for multirotor system (top level) .............. 66  
A.2 System FMEA sheet for multirotor system (continuation) .......... 67  
A.3 System FMEA sheet for flight electronics ............................. 68  
A.4 System FMEA sheet for communication modules .................... 69  
A.5 System FMEA sheet for microprocessor and software .............. 70  
A.6 System FMEA sheet for flight electronics (Continuation) ....... 71  
A.7 System FMEA sheet for propulsion groups ......................... 72  
A.8 System FMEA sheet for inertial measurement unit ................. 73  
A.9 FMEA entries with an RPN>30 (High Priority Risks) .............. 75
Symbols

A  Control Effectiveness Matrix: $\vec{\omega}^2 \rightarrow \vec{v}$ (thrust/torque)
B  Control Allocation Matrix: $\vec{v} \rightarrow \vec{\omega}^2$
m  Number of Rotors
n  Number of Control Objectives (=4)
$\vec{v}$  Virtual Controls (thrust, roll, pitch, yaw torque $\rightarrow$ TLMN)
$\vec{\omega}$  Vector containing the Squared Angular Rotor Speeds
$\eta_{LMN}$  Performance Degradation Factor for Roll-Pitch-Yaw Control
$\eta_{LM}$  Performance Degradation Factor for Roll-Pitch Control
$\vec{\eta}$  Rotor Efficiency Vector
$\omega$  Angular Rotor Speed
$\vec{\omega}^2$  Vector containing the Squared Angular Rotor Speeds
$\Phi$  Attainable Control Set
$\Omega$  Motor Constraint Set
$V_{ACS}$  Volume of the Attainable Control Set

Acronyms and Abbreviations

ASL  Autonomous Systems Lab
EMI  Electromagnetic Interference
ETH  Eidgenössische Technische Hochschule
          (Swiss Federal Institute of Technology)
$\text{I}^2\text{C}$  Inter-Integrated Circuit
IMU  Inertial Measurement Unit
LUT  Lookup Table
FMEA  Failure Mode and Effect Analysis
MAV  Micro Aerial Vehicle
MIPS  Mega Instructions per Second
mpLP  Multiparametric Linear Programming
mpQP  Multiparametric Quadratic Programming
MPT  Multiparametric Toolbox [2]
RP  Roll Pitch (without yaw!)
RPY  Roll Pitch Yaw
RPN  Risk Priority Number
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLMN</td>
<td>Thrust Force; Roll, Pitch, Yaw torque</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>UNICE</td>
<td>Université Nice Sophia-Antipolis</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Today’s Micro Aerial Vehicles (MAV) have proven to be developed enough to be used for real world applications which make use of the MAVs as sensor carriers to easily reach spots in rough terrain or to oversee large regions with ease. Advanced measurement systems and the work involved with its integration onboard of these MAV can easily exceed the total cost of the MAV itself. Therefore users are more and more concerned about the safety of their equipment and the surrounding environment. Thus it proves to be beneficial having a system that can tolerate certain faults and failures without endangering itself and its environment. These fault-tolerance capabilities help to minimize the impact of possible faults in the system and prevent damage or even loss of the entire system which would lead to very expensive and time consuming repairs or could possibly even harm people in its surrounding area. This fact justifies the additional costs and complexity in designing the systems towards improved fault-tolerance.

1.2 Goals

Multirotor systems (mainly quadcopter and hexacopter systems) are particularly well suited as sensor carriers due to their property to carry enough payload, combined with their agility and the redundancy offered by their design. The focus of this work will therefore be on multirotor-type Micro Aerial Vehicles.

The primary goal of this work is to first analyze possible faults that threaten the success of a multirotor’s mission and its environment. Using the result of this analysis the most threatening faults are identified and concepts are developed to minimize these risks. Especially failures that compromise the stability of the MAV platform should be addressed. Therefore a special focus of this analysis lies in faults occurring in the actuators. These faults have been identified as a major risk due to the big impact it has on the stability of the vehicle combined with the high likelihood of occurrence during the life of a MAV.

Important to note here is that this thesis originated from the ASL start-up Skybotix. This company develops, produces and sells micro aerial vehicles of the multirotor-type commercially. Therefore this work tries to add a value to their products and decisions are made considering their requirements. Generally this work puts a focus on the usability of the results in practice.
Chapter 1. Introduction

The goals can be summarized with the following tasks:

- Identify possible faults in the mechatronic system
- Classify these faults by using the FMEA method
- Select the most threatening faults
- Developing solutions to minimize the risk of the overall system for the selected faults with a focus on the fault tolerance in the propulsion system
- Implementing the developed strategies to a real hexacopter system
- Testing the developed strategies on the real system
Chapter 2

Risk Analysis and Concepts

The output of the risk analysis is a compilation of the most threatening faults and failures that can compromise the success of the mission, the vehicle itself or its surrounding environment. Further the threats are assessed for occurrence and impact. Based on this analysis one can select the risks which should be addressed with a high priority in order to improve the fault-tolerance with minimal changes to the system. Risks that show a risk priority number (RPN) above a certain limit in the failure mode and effects analysis (FMEA) will be selected and concepts to address those risks are proposed. Therefore this analysis builds the basis for the next steps of this thesis.

2.1 Risk Analysis for a Multirotor System

An overview of the risk analysis process is shown in figure 2.1. Each of these steps will be described along with its output in the following subsections.

![Figure 2.1: Steps of the risk analysis](image)
2.1.1 Functional Block Description

In this step the system’s physical and logical low-level components are abstracted into functional blocks. This helps to simplify and permit the following risk analysis. The obtained functional block diagram (FBD) is shown in figure 2.2. A detailed focus during this analysis was put on the actuators, sensors and all the logical parts of the system such as the control and estimation algorithms. The line connections between the blocks describe the dependence and interfaces between the blocks. The mechanical part is represented by one block because the system can not be recovered from mechanical failures during flight.
2.1. Risk Analysis for a Multirotor System

Figure 2.2: Functional block diagram of a multirotor system
2.1.2 Nominal Operation Mode

In order to assess the risks and its effects for the individual components, it is necessary to define a condition in which the system operates during the occurrence of the described failures. These assumptions are considered very important especially in the next step of finding concepts that minimize the impact of these risks.

This analysis assumes the following operation mode:

- Multirotor is flying in autopilot mode or in manual flight
- Safety pilot is available and in range, in case of an emergency switch to manual mode
- No obstacles in the near surroundings of the MAV
- No or only calm / steady wind

2.1.3 Failure Mode and Effect Analysis

Failure mode and effect analysis is a method widely used in industry to assess a product or system for risks in various stages of the product development. Risk in this context is equal to the product of its occurrence rating and the rating of the impact to the system (severity) on a defined scale. In a first step the system is abstracted into functional blocks. For each of these blocks possible failure modes are listed in a spreadsheet along with the effect of these failure to the system. Therefore each component can have several failure modes and each failure mode can have several effects. Each of those entries is assigned with a risk priority number which is calculated by the product of the three factors: occurrence, severity and detection. The RPN therefore is a measure of the priority in which the failures should be addressed to improve the fault-tolerance capabilities of the product with minimal effort. Detailed information about the failure mode and effect analysis can be found in [3, 4].

The three factors occurrence, severity and detection are assigned in a range of 1-5 in this analysis and the corresponding description can be found in table 2.1. Because the RPN consists of three factors in the range of 1 to 5 the resolution of the RPN towards the maximum of 125 is decreasing. Therefore a RPN above 30 is considered as a high priority risk.

The spreadsheets that contain the failure mode listings can be found in the appendix A.
2.2. Concepts for Improving the Fault-tolerance

Table 2.1: Ratings for occurrence, severity and detection

<table>
<thead>
<tr>
<th>Rating</th>
<th>Occurrence Description</th>
<th>Severity</th>
<th>Detection</th>
<th>Risk Priority Number RPN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Never (so unlikely that it is not expected to happen in the entire lifetime of a system)</td>
<td>No effect</td>
<td>Almost certain</td>
<td>1</td>
<td>no risk</td>
</tr>
<tr>
<td>2</td>
<td>Low (not expected to happen several times in the entire life time of the system)</td>
<td>Minor (affects very little of the system, noticed by average customer)</td>
<td>High</td>
<td>&lt;=16</td>
<td>small risk</td>
</tr>
<tr>
<td>3</td>
<td>Moderate (unlikely to occur to all systems during their life time, occur several times in life)</td>
<td>Moderate (most customers are annoyed)</td>
<td>Moderate</td>
<td>&lt;=30</td>
<td>medium risk</td>
</tr>
<tr>
<td>4</td>
<td>High (occurs to most of the systems during their life time)</td>
<td>High (causes a loss of primary function; customers are dissatisfied)</td>
<td>Low</td>
<td>&lt;=125</td>
<td>high risk</td>
</tr>
<tr>
<td>5</td>
<td>Very high (occurs to all system several times during their lifetime)</td>
<td>Very high and hazardous (product becomes inoperative)</td>
<td>Very remote to absolute uncertainty</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.1.4 Identified High Priority Risks

All failures which were identified as high priority risks (RPN > 30) during the failure mode and effect analysis are summarized in Table A.9 (see appendix A). Most of these risks can be treated in groups, which will be called a "risk reduction group". These groups form a new functional group whose risks will be addressed as if it would only be one high level component. This approach simplifies the analysis and ensures the required actions with minimal effort by providing only the required functionality. The following risk reduction groups have been defined that cover all the identified high priority risks:

1. Propulsion system
2. Motor communication bus
3. Flight electronics
4. Battery

2.2 Concepts for Improving the Fault-tolerance

This section presents concepts to improve the fault-tolerance of the risk reduction groups. The identified high priority risk are each assigned to one group to form the basis and requirements of the corresponding group. The concepts target at reducing the occurrence and the impact that these failures have on the system.

2.2.1 Propulsion System

Multirotor systems offer a certain propulsion redundancy by design. Therefore the existing control system should be extended by the capability of exploiting the available propulsion redundancy in the presence of failures. As the control law itself
should remain unchanged the approach to achieve this fault-tolerance is to use a fault-tolerant control allocator. This control allocator translates the thrust and torque commands calculated by the control law to motor speed setpoints that are sent to the motor controllers. By using such an adaptive mapping between those quantities it is possible to only consider the available actuators in the distribution.

![Fault-tolerant Control Allocation](image)

**Figure 2.3: Overview of control allocation**

The fault-tolerant control system consists of a fault detector module and the fault-tolerant control allocator itself as shown in figure 2.3.

### 2.2.2 Motor Communication Bus

All the I²C slaves (motor controllers) are connected directly to the master device (flight electronics) on one I²C bus. This is very critical in terms of safety. A single fault on a slave (or master) device could compromise the entire bus and make the communication for other healthy slaves impossible. Even a loss of power on one slave device could lead to a bus lockup due to a possible connection of the bus lines to the ground potential depending on the default pin state of the used microcontroller in the power off state. In this application to the control of actuators this would render the system inoperable if only one slave fails even if the actuators are redundant.

![Standard I²C bus architecture](image)

**Figure 2.4: Standard I²C bus architecture**

The requirements for the fault-tolerant bus system can be stated as:

- Failures of one or more slaves should not affect the communication to the remaining healthy slaves on the bus
- There should be the possibility to have redundant master devices that can take control of the bus
• The solution itself should be fail-safe and should not introduce other sources of failure

Different Strategies to Prevent a Bus Lockup

Solutions to prevent that a failure on the side of one of the I²C slaves can affect the entire bus and make the communication among the other slaves impossible are listed in table 2.2. Only failures that affect the entire bus are considered. Therefore a physical cable damage is not considered separately and is treated as a unknown failure in the slave where the cable leads to. Thus the isolation of a faulty slave should preferably be made as far upstream of the slave’s cable as possible.

The following table 2.2 lists different concepts to achieve this fault-tolerance capabilities.
<table>
<thead>
<tr>
<th>Nr</th>
<th>Description</th>
<th>Elec. Parts</th>
<th>Covered Failures</th>
<th>Uncovered Failures</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Segment isolation with only 1 cable for the bus</td>
<td>PCA9548</td>
<td>• Active low on the slave side</td>
<td>• Active low on the master side</td>
<td>• Only 1 component &lt;br&gt; • Special and electrical separation of the switch and the failing component &lt;br&gt; • I2C frequency limited to 400kHz &lt;br&gt; • Active switching logic &lt;br&gt; • Does not allow two redundant masters</td>
</tr>
<tr>
<td>2</td>
<td>Bus isolator on each motor controller</td>
<td>PCA9541 PCA9548</td>
<td>• Active low on the slave side</td>
<td>• Active low on the master side</td>
<td>• No changes to the main mcu &lt;br&gt; • Passive switching logic &lt;br&gt; • I2C frequency limited to 400kHz &lt;br&gt; • Difficult to guarantee the operation of the switch during a fault of the BLDC &lt;br&gt; • 8 components</td>
</tr>
<tr>
<td>3</td>
<td>1 to 8 switch with 2 to 1 switch on between primary and sec. flight elec.</td>
<td>PCA9541 PCA9548</td>
<td>• Active low on the slave side</td>
<td>• -</td>
<td>• Covers all failures &lt;br&gt; • Only 2 components &lt;br&gt; • Allows for two redundant masters &lt;br&gt; • I2C Frequency limited to 400kHz &lt;br&gt; • Complex active switching logic</td>
</tr>
<tr>
<td>4</td>
<td>Parallel bus to I2C converters on all bus ends</td>
<td>8xPCA9564</td>
<td>• Active low on the slave side</td>
<td>• -</td>
<td>• Covers all failures &lt;br&gt; • I2C Frequency limited to 400kHz &lt;br&gt; • A lot of components &lt;br&gt; • Changes on both sides (master and slave pcbs) &lt;br&gt; • Weight &lt;br&gt; • Cables</td>
</tr>
<tr>
<td>5</td>
<td>Two redundant buses with two cables to each device</td>
<td>-</td>
<td>• Active low on the slave side</td>
<td>• -</td>
<td>• The cables would also be redundant &lt;br&gt; • Unlimited I2C frequency &lt;br&gt; • No additional power supply &lt;br&gt; • High weight &lt;br&gt; • Installation work &lt;br&gt; • Complex cabling</td>
</tr>
</tbody>
</table>
2.2. Concepts for Improving the Fault-tolerance

Proposed Solution

The solution using a "2to1 I^2C Master Selector" combined with a "1to8 I^2C Switch" (solution nr. 3 in table 2.2) provides the best solution of the ones investigated. With this configuration it is possible to isolate failures which occurred in:

- Cables to the slaves
- All possible failures on the slave device
- Failures on the master device

Additionally the 1to8 Switch acts as a simple bus driver. This increases the performance and quality of the electrically separated bus lines by reducing the EMI impact and the capacitance of the isolated cable segments. Further this configuration can be used to decouple the two redundant master devices on the bus and can be used to switch between the two masters by providing external signals.

Detailed Description

- Master-Switching
  The two master devices decide between each other which has the connection to the bus and set the switch accordingly by providing interrupt signals to the switch. This allows the two master devices to monitor each other and in case the active master gets unusable the inactive master devices can claim the connection to the bus and take over the control.

- Slave-Switch
  The 1to8 slave switch can be configured using I^2C commands or interrupt signals. All upstream channels can be connected to the "Master-Switch" output independently in arbitrary combinations.
• **Isolation of a Faulty Segment**

The master device will detect a bus lockup by the missing confirmation of the slaves and can initiate a recovery procedure. This could consist of probing all the slaves after each other and thus detecting the faulty segments. After this the segments connected to the master are adjusted to exclude the faulty devices and the master can return to normal operation. This allows the detection of failures in the propulsion groups that affect the I2C bus what should be the case in most of the failures. Further defective cable segments can be identified and isolated.

The following pseudo-code describes a possible bus recovery mechanism for a better understanding of how the switching is intended to work:

```
begin RESET_I2C_BUS
    TRIGGER_RESET_INTERRUPT;
    for k to NUMBER_OF_SLAVES
        DISABLE_ALL_SEGMENTS;
        ENABLE_SEGMENT(k);
        TEST_COMMUNICATION_TO_SLAVE(k);
        if COMUNICATION_TO_K == OK
            LIST_OK = STORE_K_IN_LIST;
        end
    end
    DISABLE_ALL_SEGMENTS;
    ENABLE_SEGMENT(LIST_OK);
end
```

• **Redundancy of the Power Supply**

The power supply of the switching components represent an additional single point of failure and needs to be redundant as well. To address this issue the components will be supplied by both power supplies that also power the master devices. This ensures that power is available as long as there is a master available on the bus.

• **Redundancy of the Switching Circuit**

The switching circuit itself could be made redundant as well. But in practice this would need a lot of components which is infeasible compared to the additional complexity and cost. Also because the components used are designed for safety critical applications and are therefore built for high availability, the need for a redundant switching circuit is greatly reduced.

### 2.2.3 Backup Flight Electronics

Because of economical and weight reasons it does not seem feasible to have a redundant complete second flight electronics. Therefore a redundant module is designed as an add-on to the existing electronics which acts as a backup flight controller if the primary electronics should fail. To keep the system simple and to keep the primary system unchanged there is no communication between the primary and the backup flight controller. This module has its own small IMU sensor set consisting of accelerometers and gyros to stabilize the MAV and allow for a controlled safety landing. A failure of the primary electronics is detected by monitoring the I2C communication between the primary system and the motor controllers. If a failure is detected the backup controller takes control of the motor communication bus over
the 2to1 switch on the I^2C bus and connects itself to the motors and performs the safety landing.

This module itself is not monitored for failures and in case it should fail, the system can operate normally controlled by the primary controller.

2.2.4 Battery
A single battery energy storage describes a safety critical part as it represents a single point of failure in the system. The concept to overcome this issue is to split the battery capacity to two accordingly smaller batteries. This offers redundancy in the energy storage. The following components are necessary to discharge both batteries equally and to monitor and isolate the failed battery:

- Load balancer circuit
- Failure detection circuit
- Battery isolation circuit (in case of a battery failure)

2.2.5 Conclusions
A risk analysis has been performed to identify several high priority risks that threaten the success of a mission or its surroundings. The risk of failures in the propulsion groups has been identified as the risk with the highest priority as the likelihood of occurrence is quite realistic during the life of a MAV and the impact without any intervention can cause the loss of the vehicle, damage to its surroundings or even harm people.

Basic concepts have been developed that address the identified risks and are presented in the previous sections. An overview of the concept is shown in figure 2.6 and in more detail in figure 2.7. Based on the outcome of the risk analysis the decision was made that this thesis should focus on the risk posed by propulsion group failures. A fault-tolerant control system should be developed and demonstrated on a real system that can tolerate failures in the propulsion groups and still continue its mission. From this point on the thesis will solely focus on this aspect to improve the fault-tolerance of the overall system. All other risks addressed in this chapter are only worked out as basic concepts and are therefore open for further work.
Figure 2.7: Detailed description of the fault-tolerance concept
Chapter 3

Fault Detection for the Propulsion Groups

The starting point in designing the fault detection architecture is the risk analysis (see chapter 2) which defines the faults that can affect the propulsion groups and therefore should be detected in order to take counteractions. Literature defines several different fault types that can affect an actuator such as: [15]

- Locked-in-place
- Uncontrolled floating
- Loss of effectiveness
- Total failure

The two first fault modes, locked-in-place and uncontrolled floating, can be neglected in this application as these modes can not occur due to the actuators’s design. A loss of communication might lead to a locked-in-place situation, would cause the actuators to shutdown. This is equivalent to a total failure, it is detected and treated as such.

For the actuators used in multirotor applications the following two types of faults are important:

1. Rotor/motor aging and wearing (→ loss of effectiveness)
2. Total failure of a propulsion group

The fault detection architecture therefore has to provide the capability of detecting these two faults. Furthermore it has to estimate the loss of effectiveness in order to compensate for it. The fault detector is therefore divided into two modules:

1. Smart actuators for total failures detection
2. Rotor efficiency estimation for estimating the loss of effectiveness

These modules are described in more detail in the following sections.
3.1 Smart Actuators

Detecting the rotor and/or motor faults directly using the motor controller offers several advantages listed below: [24]

- Only a local motor/rotor model is necessary  
  \( \rightarrow \) insensitive to payload changes / structural changes

- Fast and reliable detection  
  \( \rightarrow \) local model contains only the necessary system parts (decentralization)

- Computations run on motor controller  
  \( \rightarrow \) no additional load for the flight controller

Such a system can detect rotor failures such as a loss of a rotor or a damaged rotor by e.g. comparing the current and motor speed relationship.

The development of these "Smart Actuators" was the topic of another bachelor thesis performed at the Autonomous Systems Lab. Unfortunately the PCBs of these motor controllers were not available during this work. Therefore this feature is not implemented in this work and rotor failures and detection are only simulated using a predefined delay between failure and detection.

3.2 Rotor Efficiency Estimation

Experience has shown that rotors used in MAV may suffer fast aging and wear. This change in the material properties or geometry over time leads to a deterioration of the aerodynamic efficiency of these rotors. In turn this degrades the performance of the control system as the control effectiveness matrix used for calculating the control action deviates from reality. To overcome this issue it is beneficial to estimate the "loss of effectiveness" for each motor-rotor combination. This method is an extension to the control effectiveness matrix identification method described in [25] from systems with four rotors to a general multirotor systems. Having more than four rotors complicates the identification because of the ambiguity in the relationship between rotor speeds and force/torques.

3.2.1 Definition of the Rotor Efficiency

The control effectiveness matrix \( A \) is extended to include a rotor efficiency parameter \( \eta_i \) for each motor-rotor pair \( i \) to quantify the loss of effectiveness relative to a defined nominal value. This works assumes a linear relationship between the aerodynamic forces and torques which a rotor produces and the square of the rotor speeds. This relationship can be expressed as

\[
F = \mu \cdot \omega^2 \quad (3.1)
\]

\[
T = \kappa \cdot \omega^2 \quad (3.2)
\]

where \( F \) describes the force, \( T \) the axial counter torque produced by the rotor and \( \mu, \kappa \) the nominal aerodynamic coefficients of the rotor. The extended model can be written as

\[
F = \eta \cdot \mu \cdot \omega^2 \quad (3.3)
\]

\[
T = \eta \cdot \kappa \cdot \omega^2 \quad (3.4)
\]
Using this extended model one can obtain the extended control effectiveness matrix using the following relationship

\[ A = A_{\text{nominal}} \cdot \text{diag}(\vec{\eta}) \]  

(3.5)

where \( \vec{\eta} \) is a vector containing the individual rotor efficiencies. A nice property about this formulation is that the efficiencies can express "loss of efficiency" faults as well as a total loss of a rotor with \( \eta = 0 \).

### 3.2.2 Identification Method

As the duration of a mission is quite small compared to the lifetime of a rotor it is assumed that the rotor efficiencies do not change during the execution of a mission. Therefore the identification method is designed as a special flight maneuver which has to be executed intentionally e.g. at the start of a mission or periodically between missions when needed.

The basic idea behind the estimation approach is to get the MAV into a stable hover flight during which the thrust and torques are known. In this flight configuration the commanded thrust is equal to the "weight" force and the three torques are zero

\[ \vec{v}_{\text{hover}} = \begin{bmatrix} T \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} m \cdot g \\ 0 \\ 0 \\ 0 \end{bmatrix} \]  

(3.6)

The thrust is the sum of all motor forces and the torques are caused by the difference between the motor forces. This relationship is expressed with the control effectiveness matrix as follows:

\[ \vec{v}_{\text{hover}} = A (\vec{\eta}) \cdot \vec{\omega}^2 = \text{diag}(\vec{\eta}) \cdot A_{\text{nominal}} \vec{\omega}^2 \]  

(3.7)

This equation is solved for \( \vec{\eta} \) using the following equation

\[ \vec{\eta} = \left[A_{\text{nominal}} \cdot \text{diag}(\vec{\omega}^2)\right]^{-1} \cdot \vec{v}_{\text{hover}} \]  

(3.8)

which only has a unique solution if \( A_{\text{nominal}} \) is nonsingular (which is the case for a meaningful control effectiveness matrix) and if it is a square matrix. The second condition is only valid for the quadrotor case but not for multirotor systems with more than four rotors. This issue arises from the ambiguity between the rotor speeds and the thrust/torque mapping in multirotor systems.
To get a set of equations that uniquely define the rotor efficiencies for multicopters one needs to vary the rotors speeds without leaving the hovering condition. In order to change the rotor speeds without changing the produced thrust/torques we define the set that contains all the rotor speeds that correspond to $ \vec{v}_{\text{hover}}$. This can be achieved by finding one solution using e.g. the pseudo-inverse solution and add the degrees of freedom as free parameters $ \vec{\alpha}$. The directions in which the rotor speeds $ \vec{\omega}_2$ can be changed without changing $ \vec{v}$ correspond to the null space of the control effectiveness matrix $ A$. This can be expressed as

$$\vec{\omega}_{\text{hover}}^2 (\vec{\alpha}) = A^+ \cdot \vec{v}_{\text{hover}} + (I - A^+ \cdot A) \cdot \vec{\alpha}$$ (3.9)

where $N(A)$ is the null space (or kernel) of the matrix $A$ and $A^+$ is the pseudo-inverse of $A$ defined as

$$A^+ = (A^T \cdot A)^{-1} \cdot A^T$$ (3.10)

$\vec{\omega}_{\text{hover}}^2 (\vec{\alpha})$ fulfills the following equation

$$\vec{v}_{\text{hover}} = A \cdot \vec{\omega}_{\text{hover}}^2 (\vec{\alpha})$$ (3.11)

for all $\vec{\alpha} \in \mathbb{R}^{m-n}$ with $m$ being the number of rotors and $n = 4$ the number of control objectives. Using equation 3.9 the rotor speeds can be changed without leaving the hovering condition if $A$ is perfectly known. However as the goal of the estimation is to identify $A$ of which just an initial guess $A_{\text{nominal}}$ exists, $\vec{v}$ will deviate from $\vec{v}_{\text{hover}}$ slightly with an error $\Delta \vec{v}_{\text{hover}}$ which can be expressed as

$$\Delta \vec{v}_{\text{hover}} = A \cdot [ A_{\text{nominal}}^+ \cdot \vec{v}_{\text{hover}} + (I - A_{\text{nominal}}^+ \cdot A_{\text{nominal}}) \cdot \vec{\alpha} ] - \vec{v}_{\text{hover}}$$ (3.12)

From this equation it can be seen that if we know the control effectiveness matrix perfectly ($A_{\text{nominal}} = A$) the error $\Delta \vec{v}_{\text{hover}}$ will be 0.

![Figure 3.2: Rotor efficiency estimation cycle](image)

The heart of the estimation method now lies in the fact that a small deviation from $\vec{v}_{\text{hover}}$ caused by a wrong initial guess of $A$ will cause the MAV to roll or pitch. The angular control system of the MAV will correct this error by adjusting the commanded $\vec{v}_{\text{cmd}}$ and therefore drives the system back to the hovering state. Thus the control system lets the error $\Delta \vec{v}_{\text{hover}}$ converge towards zero until we reach the hovering state again. The rotor speeds $\vec{\omega}_2$ can then be measured. Different measurements can be obtained by varying $\vec{\alpha}$. Using the relationship defined in equation 3.7 an overdetermined system can be obtained that defines the rotor efficiencies $\vec{\eta}$ using...
the measurements. After some $\alpha$ variations the obtained overdetermined system can then be solved using equation 3.8 that describes how the rotor speeds $\omega^2$ relate to the rotor efficiencies $\eta$ in the hover state. The following least-squares approach can be used:

$$
\begin{bmatrix}
A_{\text{nominal}} \cdot \text{diag}(\omega^2_{\text{measured,}i}) \\
\vdots \\
A_{\text{nominal}} \cdot \text{diag}(\omega^2_{\text{measured,}N})
\end{bmatrix} \cdot \eta =
\begin{bmatrix}
\vec{v}_{\text{hover}} \\
\vdots \\
\vec{v}_{\text{hover}}
\end{bmatrix}
\quad (3.13)
$$

where $\omega^2_{\text{measured,}i}$ is the vector that contains the square of the measured rotor speeds for the $i$ measurement and $N$ is the total number of measurements using a different $\alpha$. The rotor efficiency estimate is then

$$
\tilde{\eta} = (B^T \cdot B)^{-1} \cdot B^T \cdot h \quad (3.14)
$$

and the corresponding estimate of the control effectiveness matrix is obtained with

$$
\hat{A} = A_{\text{nominal}} \cdot \text{diag}(\tilde{\eta}) \quad (3.15)
$$

### 3.2.3 Simulation Results

This section shows the results of a rotor efficiency estimation in a Simulink simulation. In figure 3.3 the speed for all rotors are shown during the rotor efficiency estimation procedure. Four different null space excitation are injected which can be seen in figure 3.3 as steps in the rotor speeds. The angular excitation however remains comparatively small which can be seen in figure 3.4. This is due to the fact that the rotor speed excitation is chosen to be in the direction of the system’s null space. The deviation of the initial guess of the rotor efficiencies to the real rotor efficiencies lead to small angle excitations. As described in the previous section the control system then tries to force the system back to the hovering state. This can be observed as small oscillations in rotor speeds just after the excitation has been injected. The rotor speeds are measured and averaged over a short period of time after the system has stabilized itself around the hovering point.

![Figure 3.3: Rotor speeds during the estimation](image-url)
After the four rotor speed variation cycles enough information is collected, to estimate the rotor efficiencies $\hat{\eta}$ using the proposed least-squares approach. The relative estimation error is quite small and below 0.3% for all six rotors despite using a very poor initial guess $\eta_0$ compared to the real rotor efficiencies $\eta$. The results are compared in the following Table 3.1.

<table>
<thead>
<tr>
<th>Rotor</th>
<th>True $\eta$</th>
<th>Initial Guess $\eta_0$</th>
<th>Estimation $\hat{\eta}$</th>
<th>Rel. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.80</td>
<td>1.0</td>
<td>0.8014</td>
<td>0.1727 %</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>0.6</td>
<td>0.9998</td>
<td>-0.0183 %</td>
</tr>
<tr>
<td>3</td>
<td>1.05</td>
<td>1.0</td>
<td>1.0490</td>
<td>-0.0994 %</td>
</tr>
<tr>
<td>4</td>
<td>0.85</td>
<td>0.5</td>
<td>0.8520</td>
<td>0.2317 %</td>
</tr>
<tr>
<td>5</td>
<td>0.93</td>
<td>1.0</td>
<td>0.9311</td>
<td>0.1132 %</td>
</tr>
<tr>
<td>6</td>
<td>1.00</td>
<td>0.6</td>
<td>1.0023</td>
<td>0.2330 %</td>
</tr>
</tbody>
</table>

Table 3.1: Results of the rotor efficiency estimation
Chapter 4

Controllability of Multirotor MAVs

An important step in designing a system towards fault-tolerance is to determine how the expected faults will affect the system. This chapter describes a method to assess the performance degradation and the static controllability of a multirotor MAV with arbitrarily distributed rotors. The method is based on the construction of the attainable control set (ACS). The ACS characterizes the thrust and torque producing capabilities of the MAV as a function of available actuators and their physical limits. The ACS can then be used to investigate the performance degradation and to make predictions about the controllability of the system in different fault configurations.

4.1 Attainable Control Set

The attainable control set $\Phi$ describes a convex subspace in TLMN which contains all of the admissible virtual control vectors $\vec{v}$. In other words it defines the limits in thrust and torque that can be allocated while satisfying the speed constraints of the motors. The attainable control set describes the adaption of the attainable moment set which is used in the design of control systems for aircrafts [6, 7, 14]. The knowledge of this set can be very helpful in developing a control allocation algorithm to ensure that the algorithm is capable of allocating a maximum of the admissible control vectors. All parts of the ACS that can not be allocated by the control allocator represent a loss in performance. This should obviously be avoided.

The space of the ACS can be obtained by mapping the boundary of the m-dimensional motor constraints box $\Omega$ in the control space to the Attainable Control Set $\Phi$ in the virtual control space by using the control effectiveness matrix $A$, which describes the linearized relationship between the squared motor speeds and the produced thrust/torque. The two sets can be defined using the following equations:

$$\Omega = \{ \vec{U} \in \mathbb{R}^m | U_{\text{min}} \leq U \leq U_{\text{max}} \}$$  \hspace{1cm} (4.1)

$$\Phi = \{ \vec{v} \in \mathbb{R}^n | H \cdot \vec{v} \leq K \}$$  \hspace{1cm} (4.2)
Chapter 4. Controllability of Multirotor MAVs

with \( n = 4 \) for a system with 4 control objectives as it is the case in a multirotor system with thrust and three axis torque control. In the definition of the ACS \( H \) is a matrix and \( K \) a vector that define the set as a convex polytope using a system of inequalities. This can be thought of as an intersection of a finite number of half-spaces. The problem in finding the ACS is now to obtain the matrix \( H \) and the vector \( K \).

The method to obtain a definition of the ACS \( \Phi \) (equal to \( H, K \)) uses the following properties of a linear transformation: (proofs in [7])

- Convexity of a set will be preserved under a linear transformation
- Points on \( \partial \Phi \) map to unique points on \( \partial \Omega \)
- Vectors remain parallel under linear transformation

First the extremal vertices \( P_\Omega \) of the polytope \( \Omega \) are obtained

\[
P_\Omega = V(\Omega)
\]  

(4.3)

where \( V(\cdot) \) defines an operator that solves the vertex enumeration problem for a convex polytope. Several libraries exist that implement this operator e.g. the CDD library which is used in this work. [13]

The extremal points \( P_\Omega \) are then mapped to the virtual control space using the control effectiveness matrix \( B \) to obtain a list of vertices \( P'_\Phi \).

\[
P'_\Phi = B \cdot P_\Omega
\]  

(4.4)

The obtained points \( P'_\Phi \) either lie on the boundary \( \partial \Phi \) or in the interior of \( \Phi \). Therefore the list contains the extremal vertices \( P_\Phi \) of the ACS \( \Phi \) together with some points that are not extremal points of \( \Phi \) but are contained in \( \Phi \). Further it is possible that more than one point in \( \Omega \) maps to the same point in \( \Phi \) as \( B \) describes as projection from a higher dimensional space onto a subspace of lower dimension. Because it is known from the properties of linear transformations that \( \Phi \) also describes a convex polytope, a convex hull problem over the mapped vertices \( P'_\Phi \) can be solved to obtain a list of the extremal vertices \( P_\Phi \). The parameters \( H \) and \( K \) that describe the polytope \( \Phi \) using a set of linear inequalities as defined in equation 4.2 can also be obtained this way.

An example of such a mapping is illustrated in the following figure 4.1. As it is hard to visualize polytopes of a dimension higher than three the figure shows cuts through the four dimensional polytope with the 4th coordinate (yaw torque) equal to zero.
4.2 Controllability under Faults

To compare the performance under faults to the nominal case it is important to quantify the performance degradation for different fault cases. Predictions over the controllability can be made based on the performance degradation evaluation. First the number of different fault configurations that exist for a given multirotor system is calculated. Assuming that the order of the fault occurrence does not matter and that a minimum of 4 rotor have to run. The number of possible fault configurations can be calculated with the following equation

\[ P = \sum_{i=0}^{N-4} \binom{N}{i} = \sum_{i=0}^{N-4} \frac{i!}{i! \cdot (N-i)!} \]  

(4.5)

where \( P \) is the number of fault configurations (including the fault-free case) and \( N \) the total number of rotors.

<table>
<thead>
<tr>
<th>Number of Rotors</th>
<th>Number of Fault Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>163</td>
</tr>
<tr>
<td>10</td>
<td>848</td>
</tr>
</tbody>
</table>

Table 4.1: Number of different fault configurations (including the fault-free nominal case) for different number of rotors

Figure 4.1: Attainable control set for an hexacopter (T=thrust, L=roll-, M=pitch-, N=yaw-torque)
Only a part of those fault configurations are controllable and allow for a reduced thrust and torque (TLMN) control. To quantify the performance degradation and the controllability of the fault configurations, a parameter $\eta_{LMN}$ can be calculated using the ACS. To calculate $\eta_{LM}$ the ACS is obtained for the nominal case and also for the faulty configuration that is investigated. The two 4-dimensional polytopes are then cut at the nominal flying condition where $T = m_{MAV} \cdot g$ and $N = 0$ to obtain two polygons on the $L - M$ plane. These polygons describe the set of all admissible pitch and roll torque setpoints. This set has to be sufficiently large to allow a stable flight under disturbances. The following figure 4.2 shows an example of such a cut where the two polygons are plotted above each other. The red polygon shows the ACS for the nominal case of an hexacopter whereas the blue polygon shows the ACS for the case in which the rotor 1 has failed.

![Figure 4.2: Quantifying the performance degradation using cuts of the ACS](image)

The performance degradation factor $\eta_{LMN}$ can now be calculated as the ratio of the two biggest circles that fit in the polygons with the center in the origin ($L = 0, M = 0$).

$$\eta_{LMN} = \frac{R_{\text{fault}}}{R_{\text{nominal}}}(4.6)$$

The radius of the biggest enclosed circle expresses the minimal torque that can be produced in all directions which is the important factor to ensure stability for arbitrary disturbances.

Further the controllability can be investigated by defining a minimal torque $R_{\text{min}}$ that is needed in order for the MAV to operate reliably. $R_{\text{min}}$ has to be found
25  4.3  Influence of the Number of Rotors and Arrangement

experimentally or can be guessed based on experience. The following condition
\( R_{\text{fault}} \geq R_{\text{min}} \) with \( R_{\text{min}} > 0 \) then determines over the controllability of a certain
fault configuration. From this condition it can be seen that the vehicle is not con-
trollable if the origin of the L-M plane is not contained in the polygon which would
be equal to \( R_{\text{fault}} = 0 \).

If a given fault configuration is not controllable using full thrust and three torque
control, it might be possible to switch to a reduced thrust, roll and pitch control
only without any yaw control. To assess the controllability in this case the two 4
dimensional polytopes can be projected onto the T-L-M hyperplane (which is or-
thogonal to the N coordinate) and the controllability can be assessed using the same
method as described for the full TLMN control mode. In this case the performance
degradation factor is called as \( \eta_{LM} \) instead of \( \eta_{LMN} \) as in the case with yaw control.
In practice however the vehicle needs to be almost perfectly symmetrical and needs
a very precise attitude estimation system to allow for a stable flight without yaw
control which leads to high yaw rates. Therefore losing the yaw control is only an
option for a short period of time or as a last option in emergency situations.

4.3 Influence of the Number of Rotors and Arrangement

The number of rotors and arrangements have an important influence on the fault-
tolerance capabilities of the propulsion system. The different arrangements can be
compared against each other for its fault-tolerance capabilities and performance
degradation under faults using the method described in the previous chapter. This
section compares the standard (alternating) arrangement of a symmetrical hexa-
copter and octocopter system against the best found rotor arrangement in terms of
fault-tolerance.

The \( \eta_{LMN} \) and \( \eta_{LM} \) values for all possible fault configurations up to a maximum
of two rotors failing at the same time have been calculated to compare the differ-
ent multirotor systems against each other. Further the best possible control mode
that is achievable is determined. In the following tables a three torque control is
abbreviated as RPY and two torque control without yaw control as RP. If there is
no control mode that allows for even a RP control, this is marked as “-”.

The following four multirotor arrangements are investigated whereas figure 4.3(a)
and 4.3(c) describe the most-used rotor arrangements for hexacopters and octo-
copters with alternating rotor-turn directions. The other two configurations shown
in figure 4.3(b) and 4.3(d) are the arrangements that offer the best fault-tolerance
capabilities among the studied multirotor systems.
Figure 4.3: Traditional arrangements (a,c) and arrangements for improved fault-tolerance (b,d)

The results of the controllability evaluation for the four different rotor arrangements (see figure 4.3) are shown in the tables 4.3-4.6 on the following pages. These tables were generated for a maximum of two rotor failures at a time for a good overview. However tables for a higher number of simultaneous rotor failures can easily be generated if needed. Information about the performance degradation and the controllability for all fault configurations can be extracted from these tables. Further it is possible to quantify the fault-tolerance capability of a given multirotor arrangement by comparing how many fault configurations are RPY $N_{RPY}$ or RP $N_{RP}$ controllable compared to the total number of possible fault configurations $N_{tot}$.

$$f_{RPY/RP} = \frac{N_{RPY/RP}}{N_{tot}}$$

These $f_{RPY/RP}$ values were calculated for the four multirotor arrangements and are listed in the following table 4.2. This data shows that the fault-tolerance capabilities of the four-rotor arrangements increase from the arrangement (a) up to (d). Interesting to note is that the traditional PNPNPN hexacopter arrangement (figure 4.3(a)) looses the capability of RPY control and degrades to RP control for even one rotor failure, whereas the PPNNPN configuration (figure 4.3(b)) can still maintain RPY control for 33% of all fault configurations up to two random rotor failures and even 71% of these fault configurations are RP controllable. In the case of an octocopter in the traditional PNPNPNPN configuration 78% of all fault configurations for two random failures are still RPY controllable and all fault configurations are at least RP controllable. Using the optimal PPNNPPNN arrangement
the success of maintaining RPY control increases to 89%.

Further it is important to note that the use of e.g. a PPNNPN instead of a PNPNPN configuration improves the fault-tolerance capabilities but also influences the geometry of the attainable control set. In the investigated cases this led to a decrease of the ACS volume $V_{ACS}$ which corresponds to a decreased torque-producing capability. Therefore changing the rotor arrangement is always a trade off between fault-tolerance and performance although the performance loss is quite small.

<table>
<thead>
<tr>
<th>Arrangement</th>
<th>$i_{RPY}$</th>
<th>$i_{RP}$</th>
<th>$V_{ACS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexacopter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) PNPNPN</td>
<td>0%</td>
<td>57%</td>
<td>145</td>
</tr>
<tr>
<td>(b) PPNNPN</td>
<td>33%</td>
<td>71%</td>
<td>120</td>
</tr>
<tr>
<td>Octocopter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) PNPNPNPN</td>
<td>78%</td>
<td>100%</td>
<td>561</td>
</tr>
<tr>
<td>(d) PPNNPPNN</td>
<td>89%</td>
<td>100%</td>
<td>497</td>
</tr>
</tbody>
</table>

Table 4.2: Fault-tolerance comparison for different multirotor arrangements with up to two rotor failures.
<table>
<thead>
<tr>
<th>Rotor Failures</th>
<th>Rotor 1</th>
<th>Rotor 2</th>
<th>Rotor 3</th>
<th>Rotor 4</th>
<th>Rotor 5</th>
<th>Rotor 6</th>
<th>( R_{LMN} ) [Nm]</th>
<th>( R_{LM} ) [Nm]</th>
<th>( \eta_{LMN} ) [-]</th>
<th>( \eta_{LM} ) [-]</th>
<th>Control Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.67</td>
<td>2.86</td>
<td>1.00</td>
<td>1.07</td>
<td>RPY</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td>RP</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td>RP</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td>RP</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td>RP</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>X X X X</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>0.26</td>
<td>RP</td>
</tr>
</tbody>
</table>

Table 4.3: Controllability table for an PNPNPNP hexacopter
### 4.3. Influence of the Number of Rotors and Arrangement

<table>
<thead>
<tr>
<th>Rotor Failures</th>
<th>Radii $R_{LMN}$ [Nm]</th>
<th>Radii $R_{LM}$ [Nm]</th>
<th>$\eta_{LMN}$ [-]</th>
<th>$\eta_{LM}$ [-]</th>
<th>Control Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X X X X X</td>
<td>1.71 2.86 1.00 1.68</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>1.38 1.54 0.81 0.90</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>X X X X X X</td>
<td>0.89 1.54 0.52 0.90</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>0.89 1.54 0.52 0.90</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>X X X X X X</td>
<td>1.38 1.54 0.81 0.90</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>1.38 1.54 0.81 0.90</td>
<td>RPY 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
<td>X X X X X X</td>
<td>0.00 1.54 0.00 0.90</td>
<td>RP 0.00 0.00 0.00 0.00</td>
</tr>
</tbody>
</table>

Table 4.4: Controllability table for an PPNNPN hexacopter
Table 4.5: Controllability table for a PNPNPNPN octocopter
## 4.3. Influence of the Number of Rotors and Arrangement

<table>
<thead>
<tr>
<th>Rotor Failures</th>
<th>Radii</th>
<th>h</th>
<th>Control Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R_{\text{MIN}}$ [Nm]</td>
<td>$R_{\text{LM}}$ [Nm]</td>
<td>$h_{\text{MIN}}$ [-]</td>
</tr>
<tr>
<td>Rotor 1</td>
<td>3.43</td>
<td>3.94</td>
<td>1.00</td>
</tr>
<tr>
<td>Rotor 2</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 3</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 4</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 5</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 6</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 7</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>Rotor 8</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.18</td>
<td>1.68</td>
<td>0.34</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>1.87</td>
<td>2.52</td>
<td>0.54</td>
</tr>
<tr>
<td>X X</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4.6: Controllability table for a PPNPNN octocopter
Chapter 5

Fault-Tolerant Control Allocation

In general it is a difficult task to design a control law for a system which has more actuators than control objectives (overactuated system) and still satisfies the actuator constraints. To simplify this control system design the task can be split into a control law and a control allocation part. A good examples where this breakup simplifies the control design significantly is the control of a rigid body using force/torque effectors with nonlinear dynamic inversion and backstepping control laws which are e.g. used in modern flight control systems [11, 12, 17]. These control laws usually provide the desired forces and torques to control the system and therefore lead to the need for a method to translate these desired virtual controls to actuator setpoints. Finding such a transformation is known as the control allocation problem. Therefore control allocation can be described as a method to distribute a given virtual control demand commanded by a control law to all available actuators of the system.

![Control System Diagram](image)

Figure 5.1: Control system with control law, control allocator and failure detection

If the system has redundant actuators (overactuated system) certain thrust/torques can be achieved using different actuator setpoints which leads to the same system response. This degree of freedom can be exploited by using a control allocator to account for actuator faults in the system by redistributing the control demand only to the available actuators without the need of adapting the control law itself. This helps to simplify the design of a fault-tolerant control system by splitting the task into the two simpler subtasks of designing a control law and a control allocator. In the fault free case this degree of freedom can also be used to minimize some criterion, such as minimal energy consumption.
Another extension to basic control allocators is to design a dynamic control allocator which accounts for the different bandwidths among the actuators and distributes the demanded controls in the frequency domain to the best suited actuators. This ensures that the rate limits of the actuators are satisfied by using fast actuators for fast control demands. [17]

Typical goals of a control allocator can be summarized as:

- Map the control objectives to actuator setpoints while satisfying the actuator’s constraints
- Achieve fault-tolerance by only considering available actuators in the allocation
- Allow the system to use a maximum of the attainable control set
- The degree of freedom in the distribution of the control objective can be used to optimize for a defined criterion (e.g. minimal energy consumption)

### 5.1 Control Allocation Problem

A general nonlinear system can be divided into two subsystems for the actuator dynamics and the systems dynamics. This can be written as

\[
\begin{align*}
\dot{x} &= f(x, y) \\
\dot{y} &= g(y, u)
\end{align*}
\]  

(5.1)

(5.2)

where \(x\) is the system’s state, \(y\) the actuator’s state and \(u\) the actuator setpoints. Further the actuator state is constrained with

\[ y_{\text{min}}(t) \leq y(t) \leq y_{\text{max}}(t) \]  

(5.3)

where \(y_{\text{min}}, y_{\text{max}}\) are the lower/upper state limits and the actuator rate limit are

\[ |\dot{y}| \leq \dot{y}_{\text{max}} \]  

(5.4)

with \(\dot{y}_{\text{max}}\) as the rate limit of the actuators.

Assuming that the dynamics of the actuators are fast with respect to the system dynamics \((y \approx u)\) and by knowing \(\tau(x, u)\) which is the relationship between the forces/torques \(\tau\) caused by the actuators and the commanded actuator setpoint \(u\), the nonlinear system equations can be written as

\[ \dot{x} = f_\tau(x, \tau(x, u)) \]  

(5.5)

Using this formulation a control law \(C\) can be designed which commands the desired forces and torques acting on the system to achieve the desired closed loop dynamics. The relationship can be written as

\[ \tau(x, u) = C(r, x) \]  

(5.6)

where \(r\) is the user input to the system. To further simplify the design of the control law and allocator we assume that the relationship between the forces and torques \(\tau\) and the actuator setpoints can be modeled by a linear function as

\[ \tau(x, u) = A(t) \cdot u(t) = C(r, x) \]  

(5.7)
by using that
\[ v(t) = C(r, x) \] (5.8)
we can write
\[ A(t) \cdot u(t) = v(t) \] (5.9)
in which \( A \) is the control effectiveness matrix, \( v \) the virtual control commands and \( u \) the actuator setpoints. In a digital system the rate constraints can be reformulated as time varying actuator setpoint constraints. This can be achieved by setting the setpoint limit in the next timestep to the value which could be reached using the maximal actuator rate during the entire timestep \( \Delta T \). The time-varying position limits \( u_{k+1}^{\text{min}}(k) \) and \( u_{k+1}^{\text{max}}(k) \) can then be expressed as
\[ u_{k+1}^{\text{min}}(k) \leq u(k) \leq u_{k+1}^{\text{max}}(k) \] (5.10)
\[ u_{k+1}^{\text{min}}(k) = \max(y_{\text{min}}, u_k - \Delta T \cdot \dot{y}_{\text{max}}) \] (5.11)
\[ u_{k+1}^{\text{max}}(k) = \min(y_{\text{max}}, u_k + \Delta T \cdot \dot{y}_{\text{max}}) \]
A control allocator can now be developed whose task is to solve the equation 5.9 for the actuator setpoints \( u \) given the virtual control \( v \) while satisfying the constraints 5.11. Important to note here is that generally control allocators are designed for overactuated systems which is also the case for multirotor systems. This introduces some degrees of freedom in the mapping of the virtual controls \( v \) to the controls \( u \). Thus different mappings can be obtained depending on the control allocation strategy.

## 5.2 Literature Review - Classical Control Allocation Strategies

This section briefly describes the control allocation strategies found in literature that are related to this work. For more information about classical control allocation strategies one can find very comprehensive collections with detailed descriptions in the following works [5, 6, 7, 10, 16, 17, 18, 23].

### 5.2.1 Pseudo-Inverse Solutions

The Moore-Penrose pseudo-inverse is often used to solve (unconstrained) linear control allocation problems. The pseudo-inverse solution has the nice property to have minimal \( l^2 \)-norm which establishes an equal distribution of the virtual controls among all actuators. The pseudo-inverse \( A^+ \) can be calculated with
\[ A^+ = (A^T \cdot A)^{-1} \cdot A^T \] (5.12)
where \( A \) is the control effectiveness matrix. The controls are then allocated using the following equation
\[ \vec{u} = A^+ \cdot \vec{v} \] (5.13)
where \( v \) is the desired virtual control. As the constraints are not considered in this solution an inadmissible \( \vec{v} \) will lead to controls \( \vec{u} \) that violate the constraints. However every pseudo-inverse (or in general any generalized inverse) used for control allocation of an over actuated system is not able to allocate controls that satisfy the constraints even if the desired virtual control is admissible.
Chapter 5. Fault-Tolerant Control Allocation

For the following control allocator

\[ \vec{u} = A^+ \cdot \vec{v} \]
\[ \vec{u}_{\min} \leq \vec{u} \leq \vec{u}_{\max} \]  

(5.14)

the subset of all virtual controls that can be allocated by the pseudo-inverse can be obtained by defining the following polytope

\[ \vec{u}_{\min} \leq A^+ \cdot \vec{v} \leq \vec{u}_{\max} \]  

(5.15)

or reformulated as

\[ \begin{bmatrix} A^+ \\ -A^+ \end{bmatrix} \vec{v} \leq \begin{bmatrix} \vec{u}_{\max} \\ \vec{u}_{\min} \end{bmatrix} \]

(5.16)

This form \( H \cdot \vec{v} \leq k \) describes a convex polytope in the half-space representation. A comparison between the attainable control set and the subset of virtual controls that the pseudo-inverse can allocate while still satisfying the motor constraints is shown in the following figure 5.3 for an hexacopter (PNPNPNN configuration).

![Diagram showing the ACS polytope and the PSI subset](image)

Figure 5.2: Cut though the ACS polytope and the PSI subset at \( N = 0, T = T_{\text{hover}} \) (red: ACS, blue: PSI)
5.2. Literature Review - Classical Control Allocation Strategies

A cut through the ACS polytope at hover thrust and zero yaw torque is shown in red. The blue area inside the ACS shows the subspace of the virtual controls that can be allocated using the Moore-Penrose pseudo-inverse. It can be seen that a big part is not allocatable using the pseudo-inverse which leads to a loss of performance or even wrong allocated thrust and torques due to a naive saturation of the control allocator output. As a measure of performance for the control allocator the ratio between the volume of the ACS and the volume of the allocatable subspace is defined as

\[ r = \frac{V_{PSI}}{V_{ACS}} \]  

In the example above the ratio is \( r = 0.5 \). This means only 50% of the attainable control vectors (in terms of volume) can be allocated using the Moore-Penrose pseudo-inverse allocator.

### 5.2.2 Cascading Generalized Inverses

Cascading generalized inverses (CGI) is a control allocation method that can extend the volume of admissible controls produced by generalized inverses significantly. This is achieved by using a generalized inverse (or pseudo-inverse) to allocate the controls \( \vec{u} \). The following formulas use the Moore-Penrose pseudo-inverse of the control effectiveness matrix \( A \).

\[ \vec{u}_0 = B_0 \cdot \vec{v} \]  
\[ B_0 = A_0^+ = (A_0 \cdot A_0^T)^{-1} \cdot A_0^T \]

Up to here the method is the same as using the pseudo-inverse. However the volume of admissible controls is enlarged with the following steps. If one of the controls \( u_0 \) is saturated then the desired virtual control vector \( \vec{v} \) lies outside of the admissible region of the generalized inverse \( B_0 \). If at least one control exceeded their limits we proceed by saturating the control of the corresponding actuator to its limit and
deleting the corresponding row from the control effectiveness matrix $A_0$ to get the control effectiveness matrix $A_1$ and $\vec{u}_0^*$ with the saturated controls.

$$\vec{u}_0^* = \text{sat}(\vec{u}_0, \vec{u}_{min}, \vec{u}_{max})$$  \hspace{1cm} (5.20)

From $A_1$ a new pseudo-inverse $B_1$ can be obtained for the reduced problem.

$$B_1 = A_1^+ = (A_1 \cdot A_1^T)^{-1} \cdot A_1^T$$  \hspace{1cm} (5.21)

Because $\vec{u}_0^*$ contains saturated controls we achieve a virtual control command $\vec{v}_1$ which differs from the desired virtual control $\vec{v}$ by the error $\vec{e}_1$.

$$\vec{v}_1 = A_0 \cdot \vec{u}_0^*$$  \hspace{1cm} (5.22)

$$\vec{e}_1 = \vec{v} - \vec{v}_1$$  \hspace{1cm} (5.23)

In a next allocation step the remaining portion $\vec{e}_1$ is allocated using the new pseudo-inverse $B_1$ of the reduced problem.

$$\vec{u}_1 = \vec{u}_0^* + B_1 \cdot (\vec{v} - \vec{v}_1)$$  \hspace{1cm} (5.24)

This iteration loop is repeated until one of the following abort conditions is met:

- No controls $\vec{u}_i$ exceed its limit
- All control $\vec{u}_i$ are saturated ($\rightarrow$ desired virtual control command is not admissible)
- $B_1$ has less than $n$ columns (with $n = \text{dim}(\vec{v})$)

This method can greatly increase the volume of admissible virtual controls using a given generalized inverse using very simple method. A drawback of this method however is that:

- It is not guaranteed that the obtained controls solve the initial equation $\vec{u}_0 = B_0 \cdot \vec{v}$
- Even though a virtual control is admissible, the CGI solution could allocate controls that do not correspond to the desired virtual controls. This happens especially near the surface of the ACS.

More information about different forms and implementation of the CGI method as well as its properties and problems can be found in [7, 16, 23].
Chapter 6

Control Allocation using Parametric Programming

This chapter describes the developed control allocation strategy using parametric programming which has been developed during this thesis. First the overall architecture of the fault-tolerant control system is presented. In the following sections details about the offline calculation of the explicit solution are described as well as the ”constraint handler” module to ensure polytopic constraints on the TLMN vectors. At last the properties and performance of the new control allocator are discussed.

6.1 Architecture of the Control Allocator

The control allocator lies between the control law and the motor controllers. Its task is to map the commanded thrust and torque vectors to motor speeds without violating the motor constraints in speed. Further the control allocator considers the availability of the motors which is determined by the fault detection module. The structure of the proposed control allocator is shown in the following figure 6.1.

Due to the formulation used in the parametric program (see 6.2) it is necessary to ensure that the desired TLMN vectors $\vec{v}_{des}$ are admissible in order for a correct evaluation of the explicit mpQP solution. The control allocator is therefore split into two subproblems to guarantee that $\vec{v}_{proj}$ is always lies inside or on the surface of the attainable control set:
1. **Constraint handler:** Constrains the desired TLMN vector \( \vec{v}_{\text{des}} \) to obtain \( \vec{v}_{\text{proj}} \) which is either inside the ACS or on its surface.

2. **Explicit mpQP solution:** Uses the precomputed solution of a parametric program to solve the optimal control allocation problem. The solution only exists inside the attainable control set. Therefore this module is reliant on that the input TLMN vector is admissible.

Detailed descriptions of these two modules can be found in the following sections of this chapter.

### 6.2 Formulation of the Parametric Program

The control allocation problem (see chapter 5) is formulated as a parametric quadratic program. The following control allocation problem as described in chapter 5 forms the basis of the problem:

\[
A \cdot \vec{u} = \vec{v} \\
\vec{u}_{\text{min}} \leq \vec{u} \leq \vec{u}_{\text{max}}
\]  

where \( A \) is the control effectiveness matrix as defined in the next section 6.2.1, \( \vec{v} \) the desired TLMN vector, \( \vec{u} = \vec{\omega}^2 \) the vector containing the cubes of the rotor speeds, \( \vec{u}_{\text{min}} \) and \( \vec{u}_{\text{max}} \) the lower and upper speed constraints of the motors.

The available degree of freedom in the control allocation problem is used to minimize the sum of the cubes of all rotor speeds which scales with the energy consumption of the system. Further hard equality constraints are used in the formulation to ensure that the desired TLMN vector always corresponds to the allocated TLMN vector. This is to avoid a change in direction of the command torques which could lead to a decrease in performance of the system. Using these definitions the control allocation problem can be written as a parametric program:

\[
\vec{u}_{\text{CA}}(\theta) = \min_{\vec{u}} \vec{u}^T \cdot \vec{u} \quad \text{(6.2)}
\]

\[\text{s.t.} \quad A \cdot \vec{u} = \vec{v} \]
\[\vec{u}_{\text{min}} \leq \vec{u} \leq \vec{u}_{\text{max}}\]

where \( \vec{u}_{\text{CA}}(\theta) \) describes the solution to the parametric program as a function of the parameter vector \( \theta \) which is equal to the TLMN vector as

\[
\theta = \vec{v} \quad \text{(6.3)}
\]

Additionally the availability of the motors is included into the parameter vector \( \theta \) to account for motor failures. The extended parameter vector \( \theta^* \) can be written as

\[
\theta^* = \begin{bmatrix} \vec{v} & \vec{\eta} \end{bmatrix}^T \quad \text{(6.4)}
\]

with

\[
\vec{\eta} \in B^N \\
B \in (0, 1) \quad \text{(6.5)}
\]

where \( \vec{\eta} \) is the vector that contains the motor availability as binary variables and \( N \) the number of motors. The motor speed constraints can be adapted to include this availability vector as follows.
\[ \bar{\eta}_{\max} = \text{diag}(\bar{\eta}) \cdot \bar{u}_{\max} \quad (6.6) \]

In this way a failed motor has maximal rotor speed of 0 which disables the motor in the optimization. Further it is assumed that \( u_{\min} = 0 \). The new parametric program using the extended parameter vector \( \theta^* \) can now be written as:

\[ \bar{u}_{CA}(\bar{v}, \bar{\eta}) = \min_{\bar{u}} \bar{u}^T \cdot \bar{u} \quad (6.7) \]

s.t.
\[ A \cdot \bar{u} = \bar{v} \]
\[ \bar{0} \leq \bar{u} \leq \text{diag}(\bar{\eta}) \cdot \bar{u}_{\max} \]

This is the final formulation of the parametric program that is used in the following sections of this chapter.

### 6.2.1 Control Effectiveness Matrix

The relationship between the squared rotor speed \( \omega^2 \) and the thrust/counter-torque assuming a linear relationship can be written as

\[ F_i = \eta_i \cdot \mu \cdot \omega_i^2 \quad (6.8) \]
\[ T_i = -\eta_i \cdot \text{sign} (\omega_i) \cdot \kappa \cdot \omega^2 \quad (6.9) \]

where \( F_i \) is the thrust of the \( i \)-th rotor, \( T_i \) the counter-torque of the \( i \)-th rotor and \( \mu, \kappa \) are the aerodynamic coefficients of the rotor. \( \eta_i \) describes the rotor’s efficiency and can be used to account for rotor wear or for a failure if \( \eta = 0 \).

![Figure 6.2: Geometry definition for multirotor system](image)

Knowing the geometry of the system the control effectiveness matrix \( A \) can be obtained using the following relationship...
Chapter 6. Control Allocation using Parametric Programming

\[
\begin{bmatrix}
T \\
L \\
M \\
N \\
\vec{v}
\end{bmatrix} = A \cdot \begin{bmatrix}
\omega_1^2 \\
\vdots \\
\omega_N^2
\end{bmatrix}
\]  
(6.10)

where the control effectiveness \(A\) matrix in a parametrized form is

\[
A = \begin{bmatrix}
t_1 & \cdots & t_N \\
l_1 & \cdots & l_N \\
m_1 & \cdots & m_N \\
n_1 & \cdots & n_N
\end{bmatrix}
\]  
(6.11)

with the parameters defined as

\[
t_i = \eta_i \cdot \mu \\
l_i = \eta_i \cdot \mu \cdot r_i \cdot \sin(\phi_i) \\
m_i = \eta_i \cdot \mu \cdot r_i \cdot \cos(\phi_i) \\
n_i = \eta_i \cdot d \cdot \kappa
\]  

and \((r_i, \phi_i)\) describe the position of the \(i\)-th rotor in the horizontal plane relative to the center of gravity in a polar coordinate system and \(d_i \in (-1, 1)\) describes the turn direction of the \(i\)-th rotor.

### 6.3 Explicit Solution and Binary Search Tree

It is infeasible to solve the quadratic program (equation 6.7) in realtime due to the very limited performance available onboard the MAV. To overcome this issue this work uses an explicit solution that is computed offline and can be evaluated online to obtain the optimal solution of the control allocation problem. This way it is possible to trade computational complexity against memory which usually is available on the MAV. Another advantage offered by an explicit solution is the deterministic solution which is an important aspect in safety critical systems such as flying vehicles. This can be used to guarantee the stability of the control system. In contrast an optimization solution does not have to be deterministic and therefore it is hard to predict the stability and behavior beforehand.

The optimal solution to the parametric program (see equation 6.7) is described by a set of piecewise-affine functions where each of it is valid in a defined polytopic region inside the attainable control set. Therefore it is possible to precompute all these PWA functions and the corresponding region of validity. In the end the entire attainable control set is divided into smaller polytopic regions for each of which the optimal piecewise-affine relationship between the virtual controls \(\vec{v}\) and the rotor speed \(\omega^2\) is known. In figure 6.3 these regions are shown in different colors for a PNPVPN hexacopter. Note that the outer hull around all these regions corresponds to the attainable control set.
The complexity of the explicit solution increases drastically with the dimension of the parameter vector $\theta$ and number of constraints. Therefore the complexity scales directly with the number of rotors. In table 6.1 the number of regions together with the memory required to store the solution is shown for four different types of multirotor systems.

<table>
<thead>
<tr>
<th>Arrangement</th>
<th>Regions</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexacopter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNPNPN</td>
<td>61</td>
<td>40 kb</td>
</tr>
<tr>
<td>PPNNPN</td>
<td>61</td>
<td>49 kb</td>
</tr>
<tr>
<td>Octocopter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNPNPNPN</td>
<td>1049</td>
<td>1166 kb</td>
</tr>
<tr>
<td>PPNNPPNN</td>
<td>989</td>
<td>1153 kb</td>
</tr>
</tbody>
</table>

Table 6.1: Complexity of the explicit solution

The control allocation problem can now be solved online in two steps using this explicit solution. First the active region has to be determined and second the corresponding piecewise-affine function has to be evaluated to obtain the solution. The problem of finding the active region can be solved efficiently by using a binary search tree that successively divides the attainable control set into two smaller volumes using hyperplanes. This division is performed so far that all end nodes of the tree correspond to one region of the explicit solution. The used method on how to compute the search tree and the hyperplanes is presented in [19, 20].

Using the explicit solution together with the binary search tree now allows for a computationally very efficient solution of the control allocation problem. The active region is determined by going trough the search tree. Each branch of the tree has a hyperplane associated with it and the next node is determined according to on which side of the hyperplane the demanded virtual control point $\vec{v}$ lies. The active region is found if the half-space defined by the hyperplane only contains one region of the explicit solution. The piecewise-affine function corresponding to the active
region can then be evaluated to find the solution of the control allocation problem.

This work uses the "Multi-Parametric Toolbox (MPT)" [2] to obtain the explicit solution of the parametric program and to calculate the binary search tree. Detailed mathematical descriptions of the calculation process can be found in [1, 2].

### 6.4 Constraint Handler

The parametric program (equation 6.7) has no solution if the desired virtual control setpoint \( \vec{v} \) lies outside the attainable control set. This is due to the use of hard constraints in the parametric program formulation. As the control law does not directly consider the constraint on the rotor speeds it is likely that the desired virtual control command is infeasible at some point. Especially in the presence of rotor failure the commands can easily exceed the attainable control set due to the performance degradation and because the control law is not reconfigured after failures. As no solution exists for the parametric program for unattainable TLMN commands, it has to be guaranteed that no such commands are used as an input to the control allocator. An evaluation of the explicit solution using inadmissible controls leads to wrong results and exceeds the motor constraints.

To overcome this issue a "constraint handler" module is introduced between the control law and the control allocator as it can be seen in figure 6.1. The task of this module is to constrain the desired TLMN vector \( \vec{v} \) commanded by the control law before it is used in the control allocator. This is achieved by projecting an inadmissible \( \vec{v}_{des} \) (which lies outside of the ACS) onto the surface of the attainable control set to obtain \( \vec{v}_{proj} \) as shown in figure 6.4(b).

Using such a projection ensures that the direction of the resulting torque \( \vec{v}_{proj} \) is preserved and therefore the desired control command is only scaled in magnitude. In order to be able to project an inadmissible \( \vec{v}_{des} \) onto the surface for all possible fault configurations, the attainable control set has to be precomputed offline and stored on the MAV. The attainable control sets \( \Phi_i \) are stored in its half-space representation as a set of inequalities for each fault configuration.

Figure 6.4: Constrain by projecting onto the surface of the attainable control set
\[ \Phi_i = \{ \vec{v} \in \mathbb{R}^n | H_i \cdot \vec{v} \leq K_i \} \] (6.16)

\[ H_i = \begin{bmatrix} \vec{h}_1^T \\ \vdots \\ \vec{h}_M^T \end{bmatrix}, \quad K_i = \begin{bmatrix} k_1 \\ \vdots \\ k_M \end{bmatrix} \] (6.17)

where \( H_i \) and \( K_i \) define the ACS for the \( i \)-th fault configuration, \( M \) is the number of half-spaces that define the polytope, \( h_i \) the normal vector corresponding to the \( i \)-th half-space and \( k_i \) the position of the half-space defined as

\[ h_i \cdot \vec{v} \leq k_i \] (6.18)

The projection \( \vec{v}_{\text{proj}} \) can then be calculated using the following steps. First the intersection of the half-line \( C \) to \( \vec{v}_{\text{des}} \) is calculated for all hyperplanes that enclose the ACS. For the \( i \)-th hyperplane this can be achieved using the following equation:

\[ t_i = \frac{k_i - h_i^T \cdot C}{h_i^T \cdot (\vec{v}_{\text{des}} - C)} \] (6.19)

The intersection point \( I_i \) with the \( i \)-th hyperplane is obtained with

\[ I_i = C + (\vec{v}_{\text{des}} - C) \cdot t_i \] (6.20)

Two conditions are used to find \( \vec{v}_{\text{proj}} \) among all the intersection points. First \( t \) has to be positive as we are looking for an intersection of a hyperplane and a half-line with an associated direction. Further the distance between \( C \) and \( \vec{v}_{\text{proj}} \) has to be minimal which is equal to the smallest \( t \) among all the intersection points. This can be written as

\[ t_{\text{proj}} = \min (\begin{bmatrix} t_1 & \cdots & t_M \end{bmatrix}), \quad t \geq 0 \] (6.21)

the projection of \( \vec{v}_{\text{des}} \) is then obtained with

\[ \vec{v}_{\text{proj}} = C + (\vec{v}_{\text{des}} - C) \cdot t_{\text{proj}} \] (6.22)

The steps above use the following point \( C \) as the projection center which leaves the thrust unchanged and scales only the torque subvector in length.

\[ C = \begin{bmatrix} T_{\text{des}} & 0 & 0 \end{bmatrix}^T \] (6.23)

### 6.5 Properties of the Control Allocator

As already mentioned in section 5.2.1 a pseudo-inverse allocator is not able to allocate controls for the entire attainable control set. A big advantage of the new control allocator is the capability of using the entire attainable control set for control allocation. This is possible because the ACS is divided into regions for each of which a specific relationship between \( \vec{v} \) and \( \vec{\omega}_2 \) exists (similar to using various generalized inverses). This leads to an increase in control performance as higher torques can be achieved using the same physical system. A good measure to compare two control allocators in terms of the allocatable space is the ratio between the volume of the set that the control allocator is capable of allocating and the attainable control set as defined in equation 5.17. The following table lists this ratio for four types of multirotor systems:
Further properties can be summarized as:

- continuous rotor speed solution across all regions
- 50-66% improvement in terms of ACS volume that is allocatable compared to the pseudo-inverse solution
- capable of running at 1kHz on a Cortex-M4 MCU using a floating point implementation (side by side with the rest of the flight control system)

<table>
<thead>
<tr>
<th>Arrangement</th>
<th>$r_{\text{PSI}}$</th>
<th>$r_{\text{mpQP}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexacopter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNPFPNP</td>
<td>50 %</td>
<td>100 %</td>
</tr>
<tr>
<td>PPNNPN</td>
<td>52 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Octocopter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNPFPNP</td>
<td>34 %</td>
<td></td>
</tr>
<tr>
<td>PPNNPPNN</td>
<td>43 %</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Ratio of the allocatable ACS volume to the ACS volume (see section 5.2.1) for the pseudo-inverse and mpQP allocator
A test environment has been created to test and compare the behavior and performance of the designed control allocator on a real hexacopter. The setup consists of the following components which will be described in more detail in the following sections.

- Hexacopter System
- Vicon Tracking System
- Ground Station
- Remote Control

Figure 7.1: Overview of the test environment
7.1 Vicon Tracking System

A visual motion tracking system from Vicon company is used to obtain position and angle information. The system consists of several cameras that track the position of reflective markers which are attached to the hexacopter. A software processes the position of each marker and by the known geometry of the markers on the hexacopter the position and the Euler angles of the MAV are calculated at rate of 200 Hz. This information is streamed over a TCP/IP network to the ground station where it is used to control the position and the pose of the vehicle. Further the data is logged on the ground station to use the data in the evaluation of the control allocator.

7.2 Hexacopter System

![Hexacopter System](image)

Figure 7.2: Used hexacopter system

The used MAV is of the hexacopter type. The rotors are distributed on a circle around the center of gravity using a the Mikrokopter [8] frame. The rotors are setup to form a PPNNPN rotor arrangement as shown in figure 7.2(b). The motors are controlled with the Mikrokopter BL-Ctrl V1.2 brushless controllers. To ensure the communication with the ground the system uses the following three communication links:

1. **XBee**
   Up-/Downlink, used for position control and to send "rotor failures commands"

2. **Bluetooth**
   Downlink for Data Logging in the Ground Station

3. **35Mhz Remote Control**
   For manual control and control mode switching

The following table 7.1 lists several parameters which were used during this thesis e.g. to obtain the control allocator, in the controllability study and also in the simulations.
### 7.3 Ground Station

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>$m = 1.4 kg$</td>
</tr>
<tr>
<td>Radius</td>
<td>$r = 0.275 m$</td>
</tr>
<tr>
<td>Aerody. Thrust Coeff. [9]</td>
<td>$\mu = 1.39646 \cdot 10^{-5} \frac{N}{(rad/s)^2}$</td>
</tr>
<tr>
<td>Aerody. Counter-Torque Coeff. [9]</td>
<td>$\kappa = 2.45055 \cdot 10^{-7} \frac{Nm}{(rad/s)^2}$</td>
</tr>
<tr>
<td>Max. Rotor Speed</td>
<td>$\omega_{max} = 6500$ RPM</td>
</tr>
<tr>
<td>Rotors</td>
<td>APC 10x4.7</td>
</tr>
</tbody>
</table>

Table 7.1: Parameters of the hexacopter

The following list gives a brief overview of the main components used in the flight electronics from Skybotix that are used to control the hexacopter:

- ARM Cortex-M4 microprocessor from STmicroelectronics (STM32F407VG)
- I²C bus for communication with the motor controllers
- Bluetooth and Xbee module for data communication with the ground station
- Internal power supply that connects to the battery
- Inertial Measurement Unit with
  - 3-axis Accelerometer
  - 3-axis Gyroscope
  - 3-axis Magnetometer
  - Pressure Sensor
- RC receiver for manual control

**7.3 Ground Station**

The ground station software is built as a Simulink model which runs on a laptop in real time. The software consists of the following modules:

- Position and pose controller

![Figure 7.3: Overview of the test setup](image-url)
• Remote motor deactivation for failure simulation
• Communication link with the MAV
• Data logging

The ground station software receives Euler angle and position information from the Vicon system over a TCP/IP connection. This information is used in the position controller to calculate the desired thrust and three torque setpoints which are then sent over a Xbee wireless link directly to the MAV. Further the ground station receives the following data from the MAV in real time for logging:

• Commanded motor speeds
• Estimated roll, pitch and yaw angles
• Allocated thrust and torques
• Echo of the roll and pitch setpoints

The roll and pitch setpoints are sent back to the ground station after they were received by the MAV so that the communication delay can be determined. Additionally it is possible to send commands to the MAV to switch off motors to simulate motor failures. A separate command is used to simulate the failure detection so this way it is possible to have a delay between the failure and its detection which triggers the control allocator reconfiguration.

7.3.1 Position and Pose Controller

For testing the control allocator it is convenient and beneficial to use a position controller that keeps the MAV at a defined position during the experiments. By using such a controller it is possible to exclude the influence of the pilot on the measurements, keeps the measurements reproducible and allows for comparison among different measurements.

To keep the interference between the cascading control loops of the attitude control system and the position controller minimal, a position controller with a slow response and a simple structure is chosen. The position controller is divided into three independent sub controllers all of which can be independently activated using the remote control:

• X-Y position (horizontal)
• Yaw pose
• Height
## 7.3. Ground Station

### Coordinate Systems and Euler Angles

Figure 7.4: Coordinate systems used in the position controller

The following three rotations and a translation vector $P$ are used to express the attitude and position of the vehicle frame $B$ relative to the navigation frame $N$:

1. Yaw angle $\psi$, rotation around the z-axis
2. Pitch angle $\theta$, rotation around the new y-axis
3. Roll angle $\phi$, rotation around the new x-axis
4. Translation $P$, Vector from the origin of navigation frame $N$ to the origin of the body fixed frame $B$

The rotation can be expressed with in matrix form as

$$T_{BN} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{pmatrix} \cdot \begin{pmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{pmatrix} \cdot \begin{pmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} C_\theta C_\psi & C_\phi S_\psi & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi C_\theta \\ C_\phi S_\theta C_\psi + S_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta \end{pmatrix} \quad (7.1)$$

where $T_{BN}$ is the transformation matrix from the navigation frame $N$ to the body fixed frame $B$, $S_{\phi,\theta,\psi}$ is the sine and $C_{\phi,\theta,\psi}$ is the cosine of the corresponding angle.

The transformation $T_{NB}$ from the body-fixed frame $B$ to the navigation frame $N$ is the inverse of the matrix (7.1) which is equal the transposed matrix. Due to singular points in this representation the angles should not reach the critical points (see table 7.2) since the inverse at this singular points is ambiguous.

<table>
<thead>
<tr>
<th>axis</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi$</td>
<td>$-\pi &lt; \psi &lt; \pi$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>$-\pi/2 &lt; \theta &lt; \pi/2$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>$-\pi &lt; \phi &lt; \pi$</td>
</tr>
</tbody>
</table>

Table 7.2: Range of euler angles
**X-Y Position Controller**

![Diagram of X-Y Position Controller]

The horizontal position is controlled using the position information \( \hat{x}_N \) obtained by the Vicon system. First the error \( e_{xy}^N \) is calculated and transformed to the body fixed frame \( \hat{B} \) using the Euler angle information from the Vicon system. The error is then used as the input signal to a PID controller which outputs the setpoints of the roll \( \phi \) and pitch \( \theta \) angles. These setpoints are then sent to the MAV over the XBee wireless link.

An integrator part is needed in this control loop to compensate the bias between the on board attitude estimation and the Euler angle information obtained by the Vicon system which is used for the position controller. This I-part also influences the performance during rotor failures as undetected rotor failures usually lead to a steady state error in the roll and pitch angles. This error gets compensated by the position controller and the good performance after a rotor failure could be misleading as the position accuracy of the Vicon system is far better than any position sensor used in real-world outdoor applications. Therefore the integrator saturation should be as small as possible.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>0.1429 deg/m</td>
</tr>
<tr>
<td>( I )</td>
<td>0.0214 deg·m/s</td>
</tr>
<tr>
<td>( D )</td>
<td>0.2714 deg/s</td>
</tr>
<tr>
<td>( f )</td>
<td>80 Hz</td>
</tr>
<tr>
<td>( \phi_{max}, \theta_{max} )</td>
<td>±12 deg</td>
</tr>
</tbody>
</table>

**Table 7.3: Parameters for the X-Y controller**

**Yaw Angle Controller**

![Diagram of Yaw Angle Controller]

The yaw pose is controlled by a PI controller using the angle information obtained by the Vicon system. The yaw rate output of the PI controller is sent to the MAV over the wireless XBee link.
Table 7.4: Parameters for the yaw controller

<table>
<thead>
<tr>
<th>value</th>
<th>description</th>
<th>unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$9 , \text{deg/s}$</td>
<td>$\text{deg}$</td>
</tr>
<tr>
<td>$I$</td>
<td>$0.01 , \text{deg}$</td>
<td>$\text{s}$</td>
</tr>
<tr>
<td>$f$</td>
<td>$80 , \text{Hz}$</td>
<td></td>
</tr>
<tr>
<td>$\Psi_{\max}$</td>
<td>$50 , \text{deg}$</td>
<td>$\text{s}$</td>
</tr>
</tbody>
</table>

Height Controller

Figure 7.7: Structure of the height controller structure

For safety reasons the height controller can only command thrust differences $\Delta T$ in a very limited band which are added to the thrust commanded by the remote control $T_{RC}$. This way it is possible to override the height controller using the remote control. In the case that the link between the ground station and the MAV should break, the MAV can easily be controlled using the manual control. The thrust difference $\Delta T$ is calculated using a PID controller and added to the thrust commanded by the remote control after saturation.

Table 7.5: Parameters for the height controller

<table>
<thead>
<tr>
<th>value</th>
<th>description</th>
<th>unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$13 , \text{N}$</td>
<td>$\text{mm}$</td>
</tr>
<tr>
<td>$I$</td>
<td>$2 , \text{N} \cdot \text{s}$</td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td>$4 , \text{N} \cdot \text{s}$</td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td>$80 , \text{Hz}$</td>
<td></td>
</tr>
<tr>
<td>$\Delta T_{\max}$</td>
<td>$\pm 10 , \text{N}$</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8

Experimental Results

This chapter describes and discusses the experiments that have been performed in order to assess the behavior and performance of the proposed control allocator using the parametric programming formulation. All the tests have been performed in the environment described in the previous chapter. Further the position and altitude controller were always active during the tests. The following experiments have been performed:

1. Performance under failures
2. Comparison: pseudo-inverse and mpQP control allocator
3. Influence of the reconfiguration delay

The results of these tests are presented and discussed in the following sections.

8.1 Motor Failures with Proposed Control Allocator

This experiment presents the behavior of the MAV after a motor failure has occurred. At the beginning the hexacopter is brought to a stable hovering position using the position and height controller. The motor 3 is then turned off by disabling the output of the setpoints over the I2C bus. After a reconfiguration delay of 750ms the control allocator is reconfigured to account for the failure. The rotor numbering and the definitions of the torque axis are shown in the following figure 8.1.
In the following figure 8.2 the tracking error of all three Euler angles are shown. The tracking error expresses the difference between the commanded angles by the position controller and the actual angles. The magenta line marks the failure occurrence and the green line the time at which the control allocator is reconfigured. It can be observed that the tracking error quickly grows after the failure has occurred with the control allocator not being reconfigured yet. However the error quickly decays back towards zero when the reconfiguration of the control allocator is performed after the delay of 750\,ms.

Figure 8.2: Angle tracking error during failure of motor 3
The next figure 8.3 shows the position error during the experiment. The horizontal position error stays below 20cm during the entire recovery process despite the quite high reconfiguration delay of 750ms.

Figure 8.3: Position error during failure of motor 3

8.2 Comparison: mpQP/Pseudo-Inverse Control Allocator

The performance of the old pseudo-inverse allocator is compared against the new multiparametric control allocator using the results of these experiments. Again the hexacopter is brought to a stable hovering position using the position and height controller. In this position the motor 3 is turned off and the control allocator is reconfigured after a delay of 500ms. The angle tracking error is defined as the difference between the actual angle of the hexacopter and the commanded angle by the position controller. This quantity is investigated as a measure of performance as the influence of the position controller on the angles can be eliminated in this way. In the following figures 8.4 and 8.5 the error for the roll and pitch angle are shown.
Looking at the roll tracking error it can be seen that the error quickly grows after the motor has failed at $t = 2\text{s}$. After a delay of $500\text{ms}$ the control allocator is reconfigured to account for the failure and it can be observed that the multiparametric control allocator is able to bring back the error to zero during a time of approximately $500\text{ms}$ after the reconfiguration. Using the pseudo-inverse allocator however leads to a steady-state error of around $5\text{deg}$ in the roll angle and is therefore not able to recover from the fault completely. Although the angles remain bounded the introduced offset leads to high position errors very quickly, especially in situations with poor position information where these offsets can only be estimated poorly.

The next figure 8.6 shows the position error for the two control allocators in the same experiment. It can be seen that the position error is bounded as well and remains quite small. This is because the position information used for the position controller is of very high quality (200 Hz with an RMS of around $2\text{mm}$). In real world applications where onboard position sensor such as GPS systems have to be used, the error can be far larger depending on the quality and update rate of the available position information.
8.3 Influence of the Reconfiguration Delay

In this experiment series the effect of the reconfiguration delay on the stability of the MAV should be investigated. The reconfiguration delay describes the time between the actual motor failure to the point at which the control allocator is reconfigured. During the tests the hexacopter is brought to stable hover flight at a defined position in the room using the position and height controller. Once this state is reached the motor 3 is turned off using the interface on the ground station. After the reconfiguration delay has passed the control allocator is reconfigured to account for the failure. Several reconfiguration delays from zero up to 2s have been tested.

To assess the performance of the control allocator, the angle tracking error is investigated which is the error between the commanded angles by the position controller and the actual angle. This has the benefit to only observe the actual error and not the control action that the position controller demands. The roll tracking error is shown in the following figure 8.7 for three different reconfiguration delays.

Figure 8.6: Position error for the two control allocators
Figure 8.7: Roll angle tracking error with different reconfiguration delays (failure of motor 3)

It can be seen in the figure 8.7 that the angle tracking error remains rather small despite the quite long failure detection delays. Further it can be observed that the tracking error quickly decays to zero after the control allocator considers the failure. The fact that the reconfiguration delay is an uncritical parameter which has only a small influence on the angles during failures, allows for a more robust design of the failure detector. This improves the reliability of the detection and helps to avoid false failure detection.

8.4 Conclusions

The performed experiments showed a good behavior of the parametric programming control allocator during failures. Further it is shown that the new control allocator is superior to the old pseudo-inverse solution. The steady-state error that is observed during failures using the pseudo-inverse allocator can be eliminated. Additionally the post failure performance is very comparable to the nominal flight condition after reconfiguration of the allocator for one rotor failure. This might be an effect of the enlarged TLMN domain that is available with the new allocator. Further it is shown that the reconfiguration delay is an uncritical parameter which has been tested up to a delay of 2s. Even with this delay the position and angle error remain in a safe range which leaves the fault detection module enough time to reliably detect the failure and initiate the reconfiguration.

However several problems have been identified while performing the experiments that should be solved for a real application:
1. **Battery voltage:** The battery voltage changes while discharging and causes a change in the mapping of the motor setpoints to the achieved rotor speeds. This is caused by the motor controllers that directly translate the motor setpoint to a duty cycle that drives the power MOSFETs. Therefore only a feedforward control of the motor speed is available and leaves the relationship of duty cycle and motor speed heavily dependent on the battery voltage. This influence of the battery voltage has not been considered in the modeling of the problem and therefore decreases the performance of the system.

2. **Motor speed setpoints:** Another problem that has been identified also relates to the used motor controllers. It has been observed that the motor setpoint to motor speed relationship follows a nonlinear function and therefore is only poorly approximated by the used linear approximation. This problem can be solved by using motor controllers that implement a speed control with the possibility of commanding speeds directly.
Chapter 9

Conclusions and Future Work

This thesis consists of four major parts

1. Risk analysis of multirotor systems
2. Controllability study for multirotor systems under faults
3. Developing a fault tolerant control system (fault detection + control allocator)
4. Testing on a real system

In a first step a risk analysis was performed using the well known failure mode and effect analysis (FMEA) to identify possible risks in multirotor systems. Several risks have been identified as being of high priority such as: failures of the propulsion system, motor communication bus, flight electronics and the battery. Concepts to lower the impact and occurrence of these threats have been proposed and further the decision was made that the risk of a failure in the propulsion system should be considered of highest priority. Upon this decision the focus of this thesis was set to lie on the development of a fault-tolerant control system consisting of a failure detection and a fault-tolerant control allocation module.

The proposed failure detection architecture uses ”Smart Actuators” to detect failures locally in the motor controller units. This decentralized approach offers several benefits such as only requiring simple motor/rotor models as well as lowering the computational effort on the main controller unit. Additionally a rotor efficiency estimation algorithm is proposed to account for the wear and aging of the rotors which has a quite dramatic effect on the performance of the flight control system as shown in the following thesis. [25]

A method has been proposed to assess the controllability of multirotor systems under faults based on the construction of the attainable control set. Using this method a study has been performed that investigates the controllability under rotor failures. This study has shown that the turn direction arrangement of the rotors play a big role in the fault-tolerance capabilities a multirotor system offers. Further by automatically evaluating all fault configurations for the most common hexacopter and octocopter rotor arrangements, the best rotor-turn direction arrangement in terms of fault-tolerance has been identified.
A fault-tolerant control allocator based on the theory of parametric programming has been proposed and an explicit solution together with a binary search tree is obtained using the "Multi-Parametric Toolbox" [2]. Further a constraint handler module was developed that scales inadmissible torques commanded by the control law by projecting the control vector onto the surface of the attainable control set. This preserves the direction of the torques and guarantees a solution without violating the physical motor constraints. Further the volume of admissible virtual controls could be increased by around 50% to 66% using the new control allocator. This corresponds to the ability of producing higher torques using the same physical system.

In a last step the proposed control allocator was implemented on a real hexacopter system. Several tests were performed by injecting simulated failures during flight. The MAV was controlled by a position controller using state information obtained by a Vicon motion tracking system. This excludes the effects of a pilot and allows for reproducible measurements. The results have shown a good performance of the multiparametric control allocator under failures. Further the reconfiguration delay, the time between the fault occurrence and its detection, has been investigated. It has been shown that this reconfiguration delay is an uncritical factor and can easily be as high as 1s without negative effects. This can be of great help while designing a reliable failure detector.

Possible future work might consist of:

- **Inversion of the rotor-turn direction**: An idea that came up during this thesis is to create the possibility to adaptively change the rotor-turn direction of the rotor. This could be used to reconfigure the hexacopter to the optimal rotor arrangement possible with the current rotor availability. The potential for improving fault-tolerance could be huge as the rotor arrangement has a big influence on the controllability under faults as it was shown in this thesis.

- **Evaluate the performance in dynamic flight**: All the tests in this thesis were performed in hovering flight as the available space was very limited. Near the hovering point it is hard to assess the performance of the control allocator because the additional allocatable TLMN space compared to the pseudo-inverse allocator is not fully used. It would be interesting to investigate how the two control allocators compete during dynamic flights.

- **Experiments with degraded control**: The control allocator is capable of performing a degraded control of only thrust, roll and pitch torques with neglecting the yaw control. No experiments were performed using this control mode during this thesis. It would be interesting to see how the vehicle behaves in such a condition as in certain fault configurations this is the only possible control mode.

- **Complexity reduction of the explicit solution**: It has been shown in chapter 6 that the explicit solution of the parametric program for an octocopter already consists of around 1000 regions. This require a lot of memory for storage, leads to a huge search tree and therefore increases the runtime of the control allocation. For an implementation on an octocopter system it could therefore become necessary to perform some kind of complexity reduction. Several methods to reduce the region count of the explicit solution can be found in literature. [21, 22]
Appendix A

Risk Analysis Details

A.1 Failure Mode and Effect Analysis

This appendix contains all the spreadsheets used in the failure mode and effect analysis. (see chapter 2.1.3)
## System FMEA

### Subsystem #: A
### System-Mode: Autopilot Operation - inflight
#### Description: Top Level

<table>
<thead>
<tr>
<th>Part Interface Process</th>
<th>Function</th>
<th>Failure Mode</th>
<th>Effects</th>
<th>Cause(s)</th>
<th>Current controls</th>
<th>Occurrence</th>
<th>Severity</th>
<th>Detection</th>
<th>RPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.101</td>
<td>Mechanical Integrity</td>
<td>holds together all of the system components and ensures their stable orientation and position</td>
<td>Frame structural damage</td>
<td>misalignment of actuators or sensors, decreased performance of the control or even destruction of the system</td>
<td>visual inspection before flight</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>A.202</td>
<td>Flight Control Electronics</td>
<td>holds the logical parts of the system and the interfaces between logical and physical system</td>
<td>Power Supply failures</td>
<td>disables the entire functionality of the flight control, the system gets inoperable</td>
<td>none</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.303</td>
<td>Flight Control Electronics</td>
<td>holds the logical parts of the system and the interfaces between logical and physical system</td>
<td>Communication Loss</td>
<td>weather has no influence on the mission anymore</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>A.404</td>
<td>Flight Control Electronics</td>
<td>holds the logical parts of the system and the interfaces between logical and physical system</td>
<td>IMU failure (part/total)</td>
<td>A failure of the IMU group leads to a worse estimation or no attitude information, this could make the control impossible</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.505</td>
<td>Flight Control Electronics</td>
<td>holds the logical parts of the system and the interfaces between logical and physical system</td>
<td>Actuator setpoint output failure</td>
<td>sensor malfunction, rapid deviation of the actuator position</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.606</td>
<td>Flight Control Electronics</td>
<td>holds the logical parts of the system and the interfaces between logical and physical system</td>
<td>misalignment of the PCB</td>
<td>sensors depend on defined alignment relative to the actuators, a wrong alignment may lead to actuator malfunction</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.707</td>
<td>Battery</td>
<td>Storing Energy</td>
<td>discharging under the required operating voltage</td>
<td>loss of power due to malfunction of the battery, in case of damage the system or even the platform is in an event of crash</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.808</td>
<td>Battery</td>
<td>Storing Energy</td>
<td>sudden failure of the battery</td>
<td>loss of power due to malfunction of the battery, in case of damage the system or even the platform is in an event of crash</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.909</td>
<td>Battery</td>
<td>Storing Energy</td>
<td>overload of the battery</td>
<td>overheating which could lead to an actual failure of the battery</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.100</td>
<td>Battery</td>
<td>Storing Energy</td>
<td>undercharging of the battery</td>
<td>failing to maintain correct voltage</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.111</td>
<td>Battery</td>
<td>Storing Energy</td>
<td>failure of the electrical connection</td>
<td>loss of power due to malfunction of the battery, in case of damage the system or even the platform is in an event of crash</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.201</td>
<td>Propulsion System</td>
<td>generating thrust and torque instability, control and lift the system, consist of several Propulsion Units</td>
<td>generating strong vibrations</td>
<td>possible loss of control, damaging the structure and weakening the performance of the MAV thus increasing the risk of further failures</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.302</td>
<td>Propulsion System</td>
<td>generating thrust and torque instability, control and lift the system, consist of several Propulsion Units</td>
<td>deviation from desired torque</td>
<td>possible loss of controllability of the attitude subsytem</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.403</td>
<td>Propulsion System</td>
<td>generating thrust and torque instability, control and lift the system, consist of several Propulsion Units</td>
<td>deviation from desired thrust</td>
<td>possible loss of controllability of the attitude subsytem</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.504</td>
<td>Propulsion System</td>
<td>generating thrust and torque instability, control and lift the system, consist of several Propulsion Units</td>
<td>exceeding mass continuous power of the battery</td>
<td>heat up of the battery or motor and to a controller, can cause failure or failure of the motor</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>A.605</td>
<td>Propulsion System</td>
<td>generating thrust and torque instability, control and lift the system, consist of several Propulsion Units</td>
<td>failure of propulsion unit</td>
<td>in case of propulsion unit and failure of the controller is no warning to the pilots about the failure</td>
<td>software error, electrical failure, power failure, mechanical failure</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

---

**Table A.1: System FMEA sheet for multirotor system (top level)**
<table>
<thead>
<tr>
<th>#</th>
<th>Part/Interface</th>
<th>Function</th>
<th>Failure Mode</th>
<th>Effects</th>
<th>Cause(s)</th>
<th>Current controls</th>
<th>Occurrence</th>
<th>Severity</th>
<th>Detection</th>
<th>RPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A501</td>
<td>Actuator Signal Cabling</td>
<td>Cables of the signal bus, used for the communication from the flight PCB to the flight controller</td>
<td>Signal cable physically broken</td>
<td>Depending on where the signal is broken it can disable all actuators or just single ones</td>
<td>Physical damage</td>
<td>Visual inspection, BDC controller test, connection at startup, during flight: none</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>A502</td>
<td>Actuator Signal Cabling</td>
<td>Cables of the signal bus, used for the communication from the flight PCB to the flight controller</td>
<td>Short circuit on the signal bus</td>
<td>Enables the unintended communication bus, leads to loss of control and crash</td>
<td>Problems at the BDC, physical connection between the lines</td>
<td>Visual inspection, BDC controller test, connection at startup, during flight: none</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>A503</td>
<td>Actuator Signal Cabling</td>
<td>Cables of the signal bus, used for the communication from the flight PCB to the flight controller</td>
<td>Transmission errors</td>
<td>Communication packet is not received or a wrong value is received, depending on the error rate the control can get unstable</td>
<td>Noise, crosstalk, etc.</td>
<td>Visual inspection of the cable &amp; connections</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A601</td>
<td>Power Cabling</td>
<td>Distribute the energy from the battery to the consumers</td>
<td>Shorts circuit</td>
<td>Loss of power, disables all electrical components, could damage the battery</td>
<td>Heat up of the BDC, software error in the BDC</td>
<td>Visual inspection, BDC controller test, connection at startup, during flight: none</td>
<td>1</td>
<td>1</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>A602</td>
<td>Power Cabling</td>
<td>Distribute the energy from the battery to the consumers</td>
<td>Propulsion units powered off</td>
<td>No or partial actuation</td>
<td>Power cable broken, thermal desoldering of the power cable at the BDC, photo of a short of the unit</td>
<td>Visual inspection of the cable &amp; connections</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A603</td>
<td>Power Cabling</td>
<td>Distribute the energy from the battery to the consumers</td>
<td>Flight PCB not powered</td>
<td>Disables the flight PCB</td>
<td>Power cable broken</td>
<td>None</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A701</td>
<td>Mechanical Integrity</td>
<td>Mechanically connect different parts</td>
<td>Loose screws/attachments</td>
<td>Vibration, control problems to instability</td>
<td>Loose screws/attachments</td>
<td>Visual inspection</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>A702</td>
<td>Mechanical Integrity</td>
<td>Mechanically connect different parts</td>
<td>Mechanical failure, structural damage</td>
<td>Weakens, flight performance or makes flight impossible</td>
<td>Visual inspection</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem</td>
<td>Part Interface</td>
<td>Process</td>
<td>Function</td>
<td>Failure Mode</td>
<td>Effects</td>
<td>Cause(s)</td>
<td>Current controls</td>
<td>Occurrence</td>
<td>Severity</td>
<td>Detection</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>---------</td>
<td>----------</td>
<td>--------------</td>
<td>---------</td>
<td>----------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>C.101</td>
<td>Power Supply</td>
<td>Provides the voltages needed by the consumers on the Flight PCB</td>
<td>low/high/no output</td>
<td>disables/damage parts on the entire external components on the flight PCB</td>
<td>low battery, voltage, damage of the power supply</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>C.201</td>
<td>Communication Modules</td>
<td>Communication with user on the ground</td>
<td>no connection</td>
<td>no external control/influence of the mav</td>
<td>weak signal, out of range, damage of the receiver or sender</td>
<td>none</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>C.202</td>
<td>Communication Modules</td>
<td>Communication with user on the ground</td>
<td>corrupted data</td>
<td>wrong commands get executed</td>
<td>weak signal, disturbance</td>
<td>checksums to ensure data validity</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>C.301</td>
<td>MCU</td>
<td>Runs the software, reads the sensors, commands the motors...</td>
<td>mcu reset</td>
<td>disables the whole logical part of the system, immediate destabilization of the platform</td>
<td>software bugs, hardware fluctuations</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>C.302</td>
<td>MCU</td>
<td>Runs the software, reads the sensors, commands the motors...</td>
<td>mcu freeze</td>
<td>disables the whole logical part of the system, immediate destabilization of the platform</td>
<td>software bugs</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>C.303</td>
<td>MCU</td>
<td>Runs the software, reads the sensors, commands the motors...</td>
<td>mcu damage</td>
<td>disables the whole logical part of the system, immediate destabilization of the platform</td>
<td>voltage spikes, humidity, age...</td>
<td>none</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Subsystem #:</td>
<td>Description: Communication Modules</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No.</th>
<th>Part Interface</th>
<th>Process</th>
<th>Function</th>
<th>Failure Mode</th>
<th>Effects</th>
<th>Cause(s)</th>
<th>Current controls</th>
<th>Occurrence</th>
<th>Severity</th>
<th>Detection</th>
<th>RPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.101</td>
<td>Data Channel (BLE)</td>
<td>communication between the mav and the ground station, primarily used to communicate mission or switch parameters on the mav</td>
<td>no connection</td>
<td>communication unavailable, UAV continues mission</td>
<td>out of range</td>
<td>none</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>D.102</td>
<td>Data Channel (BLE)</td>
<td>communication between the mav and the ground station, primarily used to communicate mission or switch parameters on the mav</td>
<td>corrupted data</td>
<td>corrupted data is recognized by checks and the data is lost</td>
<td>disturbances, out of range</td>
<td>checksum</td>
<td></td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>D.201</td>
<td>RC (manual control)</td>
<td>used for manual control of the mav by pilot and to switch a few predefined parameters</td>
<td>no connection</td>
<td>mav is uncontrollable, attitude is unstable, translation's system will diverge</td>
<td>out of range, damage of the receiver or sender</td>
<td>none</td>
<td></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>D.202</td>
<td>RC (manual control)</td>
<td>used for manual control of the mav by pilot and to switch a few predefined parameters</td>
<td>corrupted data</td>
<td>corrupted data is detected and is not used</td>
<td>disturbances, out of range</td>
<td>checksum</td>
<td></td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>Nr.</td>
<td>Description</td>
<td>Interface</td>
<td>Function</td>
<td>Failure Mode</td>
<td>Effects</td>
<td>Cause(s)</td>
<td>Current controls</td>
<td>Occurrence</td>
<td>Severity</td>
<td>Detection</td>
<td>RPN</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-----------</td>
<td>----------</td>
<td>--------------</td>
<td>---------</td>
<td>---------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>F.501</td>
<td>Mission Planning and Control</td>
<td>interfaces the mission to the flight control computer and is responsible for the trajectory based on the data received over the communication channels.</td>
<td>Navigation</td>
<td>failure of the module</td>
<td>no mission operations remain the way they were</td>
<td>software bug, communication failure</td>
<td>none</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>F.502</td>
<td>Mission Planning and Control</td>
<td>manages the mission, verifies the mission, and is responsible for the trajectory based on the data received over the communication channels.</td>
<td>Reception</td>
<td>no commands from the communications module</td>
<td>aircraft continues the current mission or if in manual mode, the aircraft gets unstable and remains on the current set points</td>
<td>communication failure</td>
<td>switch to a safe hover state with altitude</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.503</td>
<td>Mission Planning and Control</td>
<td>manages the mission, verifies the mission, and is responsible for the trajectory based on the data received over the communication channels.</td>
<td>Navigation</td>
<td>navigation controller remains inactive</td>
<td>no data from the data channel</td>
<td>no data from the rc channel</td>
<td>none</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.504</td>
<td>Communication</td>
<td>interfaces the command controller with the low-level communication task, provides the communication data to the mission planning and control module.</td>
<td>Communication</td>
<td>no data from the data channel</td>
<td>no trajectory information</td>
<td>no attitude information</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.505</td>
<td>Communication</td>
<td>interfaces the command controller with the low-level communication task, provides the communication data to the mission planning and control module.</td>
<td>Communication</td>
<td>no data from the data channel</td>
<td>no data from the rc channel</td>
<td>no data from the rc channel</td>
<td>no data from the rc channel</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.506</td>
<td>Communication</td>
<td>interfaces the command controller with the low-level communication task, provides the communication data to the mission planning and control module.</td>
<td>Communication</td>
<td>data from a foreign transmitter on the rc channel</td>
<td>wrong data</td>
<td>another known configuration is impossible</td>
<td>is almost impossible with a correct configuration</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.507</td>
<td>Communication</td>
<td>interfaces the command controller with the low-level communication task, provides the communication data to the mission planning and control module.</td>
<td>Communication</td>
<td>data from a foreign transmitter on the communication channel</td>
<td>wrong data</td>
<td>another known configuration is impossible</td>
<td>is almost impossible with a correct configuration</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.508</td>
<td>Communication</td>
<td>interfaces the command controller with the low-level communication task, provides the communication data to the mission planning and control module.</td>
<td>Communication</td>
<td>data from a foreign transmitter on the communication channel</td>
<td>wrong data</td>
<td>another known configuration is impossible</td>
<td>is almost impossible with a correct configuration</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.509</td>
<td>Communication</td>
<td>manages the mission, verifies the mission, and is responsible for the trajectory based on the data received over the communication channels.</td>
<td>Communication</td>
<td>no data from the communication channel</td>
<td>navigation controller remains inactive</td>
<td>no data from the communication channel</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.510</td>
<td>Position Estimation</td>
<td>estimates the position of the system using sensor data and/or data from external devices.</td>
<td>Position Estimation</td>
<td>no mission data</td>
<td>no position information</td>
<td>no position information</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.511</td>
<td>Attitude Estimation</td>
<td>estimates the attitude of the system using sensor data</td>
<td>Attitude Estimation</td>
<td>no mission data</td>
<td>no attitude information</td>
<td>no attitude information</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.512</td>
<td>Attitude Estimation</td>
<td>estimates the attitude of the system using sensor data</td>
<td>Attitude Estimation</td>
<td>no mission data</td>
<td>no attitude information</td>
<td>no attitude information</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>F.513</td>
<td>Attitude Estimation</td>
<td>estimates the attitude of the system using sensor data</td>
<td>Attitude Estimation</td>
<td>no mission data</td>
<td>no attitude information</td>
<td>no attitude information</td>
<td>navigation controller remains inactive</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
</tbody>
</table>
### Table A.6: System FMEA sheet for flight electronics (Continuation)

<table>
<thead>
<tr>
<th>Nr</th>
<th>Part Interface Process</th>
<th>Function</th>
<th>Failure Mode</th>
<th>Effects</th>
<th>Cause(s)</th>
<th>Current controls</th>
<th>Occurrence</th>
<th>Severity</th>
<th>Detection</th>
<th>RPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>F601</td>
<td>Stabilization Control</td>
<td>stabilize and control the flying platform using the attitude information</td>
<td>Control aberrations does not correspond to reality</td>
<td>In case of propulsion unit failure, the nominal control system would fail</td>
<td>none (robust controller)</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>F602</td>
<td>Stabilization Control</td>
<td>stabilize and control the flying platform using the attitude information</td>
<td>nonlinear effects in case of high angles or dynamical flight</td>
<td>weak performance of the control system or even instability</td>
<td>linear control law combined with fast dynamical flight</td>
<td>none (robust controller)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>F603</td>
<td>Stabilization Control</td>
<td>stabilize and control the flying platform using the attitude information</td>
<td>no attitude information available</td>
<td>mav gets unstable</td>
<td>imu failure (sensors, estimation,…)</td>
<td>none</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
</tr>
<tr>
<td>F604</td>
<td>Stabilization Control</td>
<td>stabilize and control the flying platform using the attitude information</td>
<td>bad control gains</td>
<td>change in the system properties</td>
<td>payload, geometrical configuration,…</td>
<td>none (robust controller)</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>48</td>
</tr>
<tr>
<td>No.</td>
<td>Part Interface</td>
<td>Process</td>
<td>Function</td>
<td>Failure Mode</td>
<td>Effects</td>
<td>Cause(s)</td>
<td>Current controls</td>
<td>Occurrence</td>
<td>Severity</td>
<td>Detection</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>----------</td>
<td>----------</td>
<td>--------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>G.101</td>
<td>Propulsion Unit</td>
<td>Generating Thrust (Input: Command and Electrical Power, Output: Thrust)</td>
<td>loss of power connection</td>
<td>deactivation of the unit</td>
<td>can be caused by overheating and auto-desoldering of the cables</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>G.102</td>
<td>Propulsion Unit</td>
<td>Generating Thrust (Input: Command and Electrical Power, Output: Thrust)</td>
<td>continuous (weak) overloading the propulsion unit</td>
<td>heat up of the motor and/or BLDC, which leads to self-destruction</td>
<td>BLDC/motor mismatch, no time depended saturation for the control output</td>
<td>overloading is used to extend the dynamic range of the actuators, but no stop mechanism to keep the overloading short</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>G.103</td>
<td>Propulsion Unit</td>
<td>Generating Thrust (Input: Command and Electrical Power, Output: Thrust)</td>
<td>fast strong overloading the propulsion unit</td>
<td>heat up of the motor and/or BLDC, which leads to self-destruction</td>
<td>BLDC/motor mismatch, no saturation for the control output</td>
<td>saturation of the control output</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>G.104</td>
<td>Propulsion Unit</td>
<td>Generating Thrust (Input: Command and Electrical Power, Output: Thrust)</td>
<td>communication bus failure</td>
<td>BLDC waits for new commands and shuts down the propulsion unit</td>
<td>short circuit on the bus, loss of cable connection</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>G.201</td>
<td>Brushless Motor Controller (BLDC)</td>
<td>Control and drive the Brushless Motor</td>
<td>MCU crash</td>
<td>failure of the propulsion unit, possible short circuit of the battery connection, that leads to the destruction of the BLDC</td>
<td>software error</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>G.202</td>
<td>Brushless Motor Controller (BLDC)</td>
<td>Control and drive the Brushless Motor</td>
<td>overheating</td>
<td>destruction</td>
<td>overloading</td>
<td>saturation</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>G.203</td>
<td>Brushless Motor Controller (BLDC)</td>
<td>Control and drive the Brushless Motor</td>
<td>communication bus errors</td>
<td>wrong set points</td>
<td>wrong thrust</td>
<td>check sums</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>G.301</td>
<td>Brushless Motor</td>
<td>Converting electrical power to mechanical rotational power</td>
<td>overheating</td>
<td>damaging the unit</td>
<td>overloading</td>
<td>setpoint saturation</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>G.302</td>
<td>Brushless Motor</td>
<td>Converting electrical power to mechanical rotational power</td>
<td>mechanical failure</td>
<td>destruction of the motor, loss of rotor, vibrations, ...</td>
<td>fatigue, overheating, ...</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>G.401</td>
<td>Rotor</td>
<td>Converting Rotational Power to Thrust</td>
<td>partial rotor damage</td>
<td>change in the aerodynamic properties, vibrations</td>
<td>collision, wear in material, fatigue</td>
<td>none</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>G.402</td>
<td>Rotor</td>
<td>Converting Rotational Power to Thrust</td>
<td>eccentricity of the rotor</td>
<td>strong vibrations, dirt on the stator, rotor damage</td>
<td>none</td>
<td>none</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>46</td>
</tr>
<tr>
<td>G.403</td>
<td>Rotor</td>
<td>Converting Rotational Power to Thrust</td>
<td>loss of rotor</td>
<td>loss of thrust</td>
<td>failure of the rotor attachment</td>
<td>none</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>Nr.</td>
<td>Part</td>
<td>Interface</td>
<td>Process</td>
<td>Function</td>
<td>Failure Mode</td>
<td>Effects</td>
<td>Cause(s)</td>
<td>Current controls</td>
<td>Occurrence</td>
<td>Severity</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>-----------</td>
<td>---------</td>
<td>----------</td>
<td>--------------</td>
<td>---------</td>
<td>----------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>E.101</td>
<td>Accelerometers</td>
<td>Sense the acceleration (digital interface)</td>
<td>no signal</td>
<td>no long term attitude reference</td>
<td>communication failure, no responsive to a damaged sensor</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>E.102</td>
<td>Accelerometers</td>
<td>Sense the acceleration (digital interface)</td>
<td>faulty signal</td>
<td>leads to wrong attitude information or faulty drifting attitude information</td>
<td>high-dynamical flight, no detection to strong vibrations, sensor misalignment</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>E.103</td>
<td>Accelerometers</td>
<td>Sense the acceleration (digital interface)</td>
<td>noise</td>
<td>weak sensor performance</td>
<td>vibrations, high-dynamical flight</td>
<td>data filtering</td>
<td>1</td>
<td>5</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>E.104</td>
<td>Accelerometers</td>
<td>Sense the acceleration (digital interface)</td>
<td>bias signal</td>
<td>leads to wrong long-term attitude information</td>
<td>bad calibration, temperature changes, disturbances in the power supply</td>
<td>calibration and referencing at startup</td>
<td>4</td>
<td>2</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>E.105</td>
<td>Accelerometers</td>
<td>Sense the acceleration (digital interface)</td>
<td>misalignment</td>
<td>weakens the performance of the control system</td>
<td>bad calibration or misaligned attachment</td>
<td>calibration and good attachment</td>
<td>2</td>
<td>3</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>E.201</td>
<td>Gyroscopes</td>
<td>Sense roll rates to sense high frequency attitude changes (analog signal)</td>
<td>no signal</td>
<td>fast attitude changes can't be sensed, depending on the sensed signal, the error signal variable</td>
<td>sampling failure, no responsive to a damaged sensor</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>E.202</td>
<td>Gyroscopes</td>
<td>Sense roll rates to sense high frequency attitude changes (analog signal)</td>
<td>noise</td>
<td>can bias the data</td>
<td>calibration, temperature</td>
<td>data filtering and change the roll rate to a short term attitude information</td>
<td>4</td>
<td>4</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>E.203</td>
<td>Gyroscopes</td>
<td>Sense roll rates to sense high frequency attitude changes (analog signal)</td>
<td>faulty signal</td>
<td>leads to wrong attitude information or faulty drifting attitude information</td>
<td>high-dynamical flight, no detection to strong vibrations, sensor misalignment</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>E.204</td>
<td>Gyroscopes</td>
<td>Sense roll rates to sense high frequency attitude changes (analog signal)</td>
<td>biased signal</td>
<td>leads to wrong long-term and short-term attitude information</td>
<td>temperature changes, bad calibration, noise, strong vibrations</td>
<td>calibration and averaging at startup</td>
<td>4</td>
<td>2</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>E.205</td>
<td>Gyroscopes</td>
<td>Sense roll rates to sense high frequency attitude changes (analog signal)</td>
<td>misalignment</td>
<td>weakens the performance of the control system</td>
<td>bad calibration or misaligned attachment</td>
<td>calibration and good attachment</td>
<td>4</td>
<td>4</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>E.301</td>
<td>Magnetometers</td>
<td>Sense the 3D magnetic field</td>
<td>no signal</td>
<td>no yaw information available</td>
<td>communication failure, no responsive to a damaged sensor</td>
<td>none</td>
<td>2</td>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>E.302</td>
<td>Magnetometers</td>
<td>Sense the 3D magnetic field</td>
<td>faulty signal</td>
<td>sensor data invalid and can not be used</td>
<td>air on the sensor, cap on the sensor</td>
<td>none</td>
<td>2</td>
<td>5</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>E.303</td>
<td>Magnetometers</td>
<td>Sense the 3D magnetic field</td>
<td>external disturbances</td>
<td>sensor bias</td>
<td>magnetic materials</td>
<td>calibration</td>
<td>4</td>
<td>4</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>E.401</td>
<td>Pressure Sensor</td>
<td>Measure the pressure to sense the altitude</td>
<td>no signal</td>
<td>no altitude sensor</td>
<td>communication failure, no responsive to a damaged sensor</td>
<td>none</td>
<td>4</td>
<td>2</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>E.402</td>
<td>Pressure Sensor</td>
<td>Measure the pressure to sense the altitude</td>
<td>faulty signal</td>
<td>sensor data invalid and can not be used</td>
<td>light on the sensor</td>
<td>cap on the sensor</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>E.403</td>
<td>Pressure Sensor</td>
<td>Measure the pressure to sense the altitude</td>
<td>external disturbances</td>
<td>sensor bias</td>
<td>wind disturbances, pressure changes due to the weather &amp; temperature</td>
<td>protecting the sensor from wind and ensuring a good atmosphere</td>
<td>3</td>
<td>5</td>
<td>27</td>
<td>27</td>
</tr>
</tbody>
</table>
A.2 High Priority Risks

The following table lists all identified risks with a RPN above 30.
<table>
<thead>
<tr>
<th>RISK #</th>
<th>FMEA #</th>
<th>System Level</th>
<th>Component</th>
<th>Failure Mode</th>
<th>O</th>
<th>S</th>
<th>D</th>
<th>RPN</th>
<th>PROPOSED ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A.203</td>
<td>Top Level</td>
<td>Flight Control Electronics</td>
<td>MCU crash/damage</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td>REDUNDANCY &amp; MONITORING</td>
</tr>
<tr>
<td>2</td>
<td>A.401</td>
<td>Top Level</td>
<td>Propulsion System</td>
<td>generating strong vibrations</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td>ALGORITHM THAT IDENTIFIES THE FAILED PROPULSION UNIT AND IF POSSIBLE SHUT IT DOWN</td>
</tr>
<tr>
<td>3</td>
<td>A.405</td>
<td>Top Level</td>
<td>Propulsion System</td>
<td>failure of a propulsion unit</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td>REDUNDANT PROPULSION SYSTEM (4+ PROPULSION UNITS)</td>
</tr>
<tr>
<td>4</td>
<td>F.601</td>
<td>Stabilization Control</td>
<td>control allocation does not correspond to reality</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td>ADAPTIVE CONTROL ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>F.603</td>
<td>MCU</td>
<td>Stabilization Control</td>
<td>no attitude information available</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>75</td>
<td>MINIMAL REDUNDANT IMU SENSORS TO PERFORM EMERGENCY PROCEDURES</td>
</tr>
<tr>
<td>6</td>
<td>A.205</td>
<td>Top Level</td>
<td>Flight Control Electronics</td>
<td>Actuator setpoint output failure</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>60</td>
<td>REDUNDANT CONTROL SYSTEM</td>
</tr>
<tr>
<td>7</td>
<td>A.301</td>
<td>Top Level</td>
<td>Battery</td>
<td>discharging under the required operating voltage</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>60</td>
<td>GOOD BATTERY MONITORING SYSTEM AND ALERTING OF THE USER AND EMERGENCY PROCEDURE</td>
</tr>
<tr>
<td>8</td>
<td>A.201</td>
<td>Top Level</td>
<td>Flight Control Electronics</td>
<td>Power Supply Failure</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT POWER SUPPLY</td>
</tr>
<tr>
<td>9</td>
<td>A.302</td>
<td>Top Level</td>
<td>Battery</td>
<td>sudden failure of the battery</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT BATTERY, SPLIT THE EXISTING CAPACITY IN TWO PARTS AND BALANCE THE LOAD</td>
</tr>
<tr>
<td>10</td>
<td>A.601</td>
<td>Mechanical Integrity</td>
<td>Power Cabling</td>
<td>shorts circuit</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT COMMUNICATION BUS</td>
</tr>
<tr>
<td>11</td>
<td>A.501</td>
<td>Top Level</td>
<td>Actuator Signal Cabling</td>
<td>shorted on the signal bus</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT POWER SUPPLY</td>
</tr>
<tr>
<td>12</td>
<td>C.101</td>
<td>Flight Control Electronics</td>
<td>Power Supply</td>
<td>low/high/no output</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT POWER SUPPLY</td>
</tr>
<tr>
<td>13</td>
<td>C.301</td>
<td>Flight Control Electronics</td>
<td>MCU</td>
<td>mcu reset</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT MCU</td>
</tr>
<tr>
<td>14</td>
<td>C.302</td>
<td>Flight Control Electronics</td>
<td>MCU</td>
<td>mcu freeze</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT MCU</td>
</tr>
<tr>
<td>15</td>
<td>E.101</td>
<td>IMU Sensors</td>
<td>Accelerometers</td>
<td>no signal</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>16</td>
<td>E.102</td>
<td>IMU Sensors</td>
<td>Accelerometers</td>
<td>faulty signal</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>17</td>
<td>E.201</td>
<td>IMU Sensors</td>
<td>Gyroscopes</td>
<td>no signal</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>18</td>
<td>E.201</td>
<td>IMU Sensors</td>
<td>Gyroscopes</td>
<td>faulty signal</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>19</td>
<td>F.304</td>
<td>MCU</td>
<td>Communication data from a foreign transmitter on the rc channel</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>RC IDENTIFICATION</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>F.501</td>
<td>MCU</td>
<td>Attitude Estimation</td>
<td>no imusensor data</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT SET OF MINIMAL SENSOR</td>
</tr>
<tr>
<td>21</td>
<td>G.301</td>
<td>Propulsion Unit</td>
<td>Propulsion Unit</td>
<td>loss of power connection</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>NONE</td>
</tr>
<tr>
<td>22</td>
<td>G.303</td>
<td>Propulsion Unit</td>
<td>Brushless Motor Controller</td>
<td>MCU crash</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT PROPULSION UNITS</td>
</tr>
<tr>
<td>23</td>
<td>G.405</td>
<td>Propulsion Unit</td>
<td>Rotor</td>
<td>loss of rotor</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>50</td>
<td>REDUNDANT PROPULSION UNITS AND DETECTION ALGORITHM</td>
</tr>
<tr>
<td>24</td>
<td>A.402</td>
<td>Top Level</td>
<td>Propulsion System</td>
<td>deviation from desired torque</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>48</td>
<td>ADAPTIVE CONTROL ALLOCATION</td>
</tr>
<tr>
<td>25</td>
<td>E.303</td>
<td>IMU Sensors</td>
<td>Magnetometers</td>
<td>external disturbances</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>48</td>
<td>DETECTION ALGORITHM</td>
</tr>
<tr>
<td>26</td>
<td>F.604</td>
<td>MCU</td>
<td>Stabilization Control</td>
<td>bad control gains</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>48</td>
<td>NONE</td>
</tr>
<tr>
<td>27</td>
<td>G.401</td>
<td>Propulsion Unit</td>
<td>Rotor</td>
<td>partial rotor damage</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>48</td>
<td>ADAPTIVE CONTROL ALLOCATION</td>
</tr>
<tr>
<td>28</td>
<td>A.204</td>
<td>Top Level</td>
<td>Flight Control Electronics</td>
<td>IMU failure (part / total)</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>45</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>29</td>
<td>E.302</td>
<td>IMU Sensors</td>
<td>Magnetometers</td>
<td>faulty signal</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>45</td>
<td>NONE</td>
</tr>
<tr>
<td>30</td>
<td>F.201</td>
<td>MCU</td>
<td>Navigation Controller</td>
<td>no position information</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>45</td>
<td>NONE</td>
</tr>
<tr>
<td>31</td>
<td>G.304</td>
<td>Propulsion Unit</td>
<td>Propulsion Unit</td>
<td>Communication bus failure</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>46</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>32</td>
<td>G.302</td>
<td>Propulsion Unit</td>
<td>Brushless Motor</td>
<td>mechanical failure</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>40</td>
<td>DETECTION ALGORITHM</td>
</tr>
<tr>
<td>33</td>
<td>C.201</td>
<td>Flight Control Electronics</td>
<td>Communication to Ground</td>
<td>no connection</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>36</td>
<td>EMERGENCY SCENARIO</td>
</tr>
<tr>
<td>34</td>
<td>E.201</td>
<td>IMU Sensors</td>
<td>Gyroscopes</td>
<td>noise</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>36</td>
<td>NONE</td>
</tr>
<tr>
<td>35</td>
<td>G.302</td>
<td>Propulsion Unit</td>
<td>Propulsion Unit</td>
<td>Continuous /weak overloading the propulsion unit</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>36</td>
<td>NONE</td>
</tr>
<tr>
<td>36</td>
<td>D.301</td>
<td>Communication Modules</td>
<td>RC (manual control)</td>
<td>no connection</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>32</td>
<td>EMERGENCY SCENARIO</td>
</tr>
<tr>
<td>37</td>
<td>A.601</td>
<td>Mechanical Integrity</td>
<td>Power Cabling</td>
<td>propulsion units unpowered</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>30</td>
<td>NONE</td>
</tr>
<tr>
<td>38</td>
<td>A.603</td>
<td>Mechanical Integrity</td>
<td>Power Cabling</td>
<td>flight electronics not powered</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>30</td>
<td>REDUNDANT POWER CABLE TO THE REDUNDANT FLIGHT ELECT.</td>
</tr>
<tr>
<td>39</td>
<td>A.502</td>
<td>Mechanical Integrity</td>
<td>Actuator Signal Cabling</td>
<td>signal line physically broken</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>40</td>
<td>REDUNDANT COMMUNICATION BUS</td>
</tr>
<tr>
<td>40</td>
<td>D.201</td>
<td>Communication Modules</td>
<td>RC (manual control)</td>
<td>corrupted data</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>30</td>
<td>EMERGENCY SCENARIO</td>
</tr>
<tr>
<td>41</td>
<td>F.201</td>
<td>MCU</td>
<td>Navigation Controller</td>
<td>no attitude information</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>30</td>
<td>REDUNDANT SENSOR AND FAILURE DETECTION ALGORITHM</td>
</tr>
<tr>
<td>42</td>
<td>G.302</td>
<td>Propulsion Unit</td>
<td>Brushless Motor Controller</td>
<td>overheating</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>NONE</td>
</tr>
<tr>
<td>43</td>
<td>G.301</td>
<td>Propulsion Unit</td>
<td>Brushless Motor</td>
<td>overheating</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>30</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Table A.6: FMEA entries with an RPN > 30 [High Priority Risks]
Bibliography


