Lucerne University of Applied Science & Architecture

ARIS - Data Fusion for a Sounding Rocket

Bachelor Thesis

Author:	Michael Kurmann
Supervisor:	Prof. Marcel Joss
Expert:	Werner Scheidegger
Industrial Partner:	ARIS (Akademische Raumfahrt Initiative Schweiz) Oliver Kirchhoff
Submission date:	22.12.2017
Classification:	Access

Declaration

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from sources are properly denoted as such. This paper has neither been previously submitted to another authority nor has it been published yet.

Horw, April 26, 2018

Abstract

This is the Abstract

Contents

Chapter 1

Introduction

1.1 Tipps/Notes

Pictures for what: Aris logo ? Atmospheric model ? Different sensors ? Sensor Network

Problems found so far:

- How to calculate Height out of Pressure/Temp/Humidity Fabian version: $44330 * (1 - (\frac{pressure}{101325})^{\frac{1}{5.255}})$
- How to parameterize the different sensors (Measuring, Test Flight, Data Sheet) ?
- How to fuse together Data from Sensors that have different Taus, especially those who are slower than the Loop-Time ?
- How to integrate AirBreaks/Drag Force of Air/ Trust of Motor a input value?
- What are the different noise factors and when do they occur ?
- The up-flight is rather short: about 25 seconds, so the Fusion should have a small settling time
- The Micro-Chip on which it is used is no the fastest : 168 MHz clock
- The Ram on the Chip is not endless: Maximal space for the Sensor fusion is about 10kB
- The Sensor Fusion should be as modular as possible so that it also can be used in the next competition
- The Sensor Fusion has to be as sturdy as possible so that it will not fail if a problem occurs
- The Fusion should make a state Estimation as precise as possible.
- There are a lot of different variables: 3 Positions, 1 Speed, 3 Accelerations, 3 Lagen, Time, Pressure, Tempterature, Humidity, Up-/Downforce.
- Especially the Input Value u which is the force onto the rocket is difficult to define (Drag, Trust = acceloration depends on wheigt which changes over time).
- The different Sensor have different weaknesses:
  - Accelerometer: Offset, drift, weak to vibrations
  - Gyro: Weak to Vibrations
  - Barometer: Many uncertenties, unpercise
  - GPS: Slow (max 5Hz)

The Academic Space Initiative Switzerland (ARIS) is student group which tries too compete in the yearly Intercollegiate Rocket Engineering Competition (IREC). To aim for the right apogee (3000 m) a Control algorithm is implemented. This algorithm relays on the information of different sensors to determine the rockets actual state. Cause there are different sensors to measure the same value a algorithm which fuses those data would come in handy. With this fusion algorithm it should also be possible to be more accurate as with each sensor on its own. For this the problem as well as the desired solution will be defined in this chapter. After that the dynamics of the rocket as well as the parameter of the different Sensors are defined at beginning of chapter 2, the most suited algorithm will be chosen. Chapter 3 will then describe how this fusion algorithm implemented in a simulation in detail. To verify that the implementation is working as intended, the fused data will be held against the ground truth which are provided from this years test flights in chapter 4. In the last chapter 5 a summary of the achieved knowledge will be stated. The purpose of this thesis is to find and implement the algorithm which is most suitable for this task.

1.2 Purpose

The hardware as well as the most of the software parts that will be used for this competition is already defined. Also it is a suitable assumption that the sensors and the dynamics of the rocket will be stay more or less the same for the competitions coming. Therefore this thesis will mainly focus on finding a algorithm for this given surroundings, but it is also will try to find as modular solution as possible, so that achieved knowledge can be used in further competition.

1.3 Research

Sensor/data fusion and state estimation is an well established engineering field. Therefore there is already a lot of previous work which can be used in this thesis. For this thesis two books are used which provide the needed theory, this are by name David (2004) which contains basic theory about kalman filters. The second book Simon (2006) contains more about different approaches of state estimation and provides also different solution to common problems that occur while implementing a state estimation. In addition the Master Thesis Bryan (2012) accesses more ore less the same issue as this thesis. Therefore it will be used mainly in the conceptional part of this paper.

1.4 Sensors

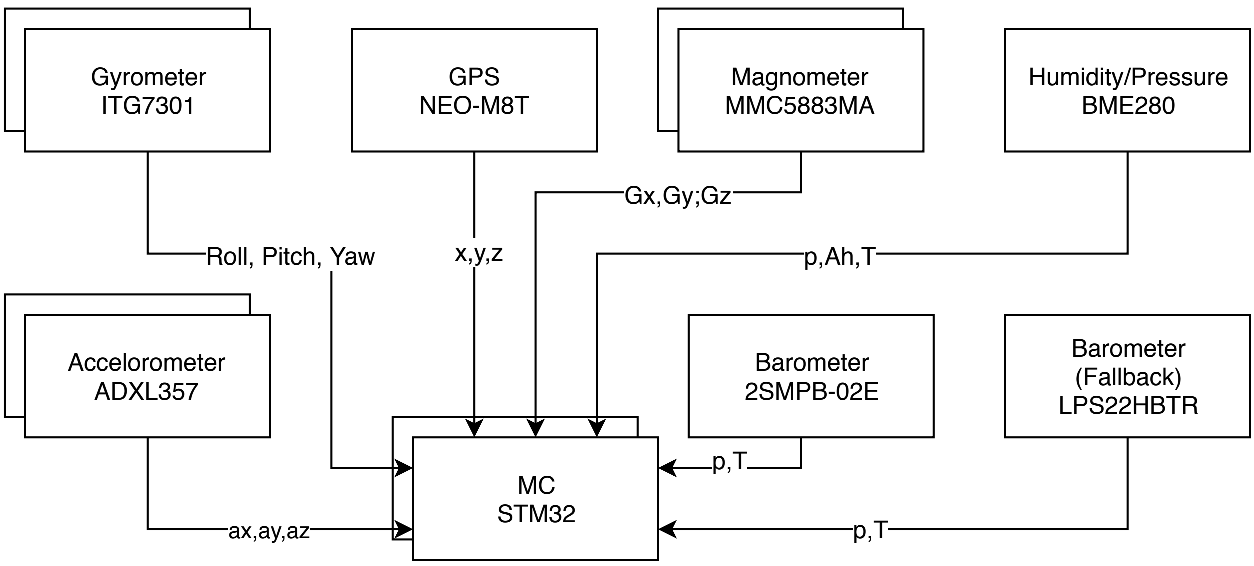

Figure 1.1: Sensor Network

As mentioned above different sensors are used in this years competition. This used sensors and their settings will be described in this chapter.

1.4.1 Accelerometer

First of all, comes the accelerometer. This is a well established and widely used sensor. It measures the acceleration that do force on the sensor in the three space dimensions. This years accelerometer is adxl357 which will be sampled at 1000 Hz. It has an accuracy of bla meters per seconds squared.

1.4.2 Gyrometer

The Gyrometer is needed to measure the posture of the Rocket. This is espacially needed to determine if the rocket has a pitch angle. If so the pure acceleration on the z-axis can be calculated. The used gyrometer is ITG-3701.

1.4.3 Barometers

Barometers are widely used in aviation, cause with a common pressure model the height can be calculated out of the measurements that the barometer takes. In this years competition three barometers are used.

This are by name 2SMPB-02E and LPS22HBTR. They will be used with a sampling rate 100Hz and 50Hz.

1.4.4 Temperature

The temperature is maybe needed to make the height out of the barometer better because most of the atmospheric model depend on the pressure as well as the temperature. This temperature will be provided by the different barometers which each posses a separate temperature sensor.

1.4.5 GPS

For next years competition differential GPS will be implemented with the help of two μblocks modules. This taken measurements are more precise as the rest of sensors but are taken much slower on a rate like 1 Hz. Therefore the algorithm should takes those provided measurements and interpolate between them with the data from the other sensors.

1.5 Problems

Out of the research and the previous competition, different problems appeared that need to be addressed in this thesis to ensure an as good solution as possible.

1.5.1 Different Sensors

First of all there are different sensors which all do measure different values and have different parameters (precision, sampling time). So the algorithm has to use out the strengths of the different sensors to cancel out their individual weaknesses. Additionally, because this algorithm is system critical, it has to be reliable enough that it still is working properly if sensors failing.

1.5.2 System Load

The cycling time will be around 1 ms on a embedded system. This time was chosen on the behalf that it would be difficult to get the exact needed cycling time on ensure the needed controlability of the rockets apogee. Therefore the system load that the algorithm can cause, has to be strongly limited, so that it can be run on this given system. The system this year is an 32 bit Arm Processor which runs on 168 MHz, assumed that the algorithm has at maximum the half of a software cycle, the maximum given clock cycles are around 84 000. This number should not change in great manner over the next competitions.

1.5.3 Precision

The Precision is after the system load the most critical attribute, if the algorithm does not get into the required accuracy the whole thing is more or less for nothing. The Control stated that the maximal error between the estimated and ground truth height should not exceed two meter. This accuracy is needed to proper control the aim of the apogee.

1.5.4 Settling Time

The settling time defines the time span when the first reliable measurements arrive after burnout until the estimation is into the required precision. This time span has to be small enough to ensure that the controlling has enough time to aim for the desired apogee. In the current system the burnout occurs occurs around 3-3.5 seconds after ignition, whereas the whole flight upwards only takes around 23 seconds. Therefore the settling time needs to be around just one second so that the control has as much time as possible for the controlling.

1.5.5 Reliability

Due to given surroundings that come if a sensor package is placed into a rocket, the assumption has to be made that it will be possible that sensors fail in execution. Therefore the algorithm should provide the reliability of still working in a proper manner with some sensors failed. So that the execution of the controlling software in terms of functionality, but locally it has not to be as accurate as it would be with all sensors working.

1.5.6 Modularity

Although it can be assumed that the sensors will stay more or less they same over the next competitions, it is not ensured that exactly this sensors will be used. Therefore the presented algorithm should provide the possibility to exchange the sensors, as long as they resemble the old sensor in a feasible way. This will ensure a long term use of the provided algorithm.

1.6 Requirements

Requirement	**Rating**	**Aim**	**Importance**
System Load	# Calculation steps per loop	< 5000	Critical
Precision	Error between estimation and ground truth	< 2m in Z	High
Settling time	Time from first reliable to optimal estimation	< 1 s	High
Reliability	Functioning Estimation with #failed sensors	2-3 sensors	Medium
Modularity	Effort needed to change a sensors	< 10 h work	Desirable

Table 1.1: Requirements table

As seen in the table 1.1 five requirements were drown out of the problem analysis. First of all, there is critical requirement the system load. This is given as critical cause it is needed that the algorithm is small enough to be run on a embedded system. Any solution that would not fit this requirement would be pointless in the frame of this thesis. Secondly there are two requirements which are tied together, the precision and settling time. Where the precision describes what a optimal estimation is under the context of this thesis, the settling time relies on this to be defined. The desired precision will be needed to ensure a possible good control.

1.7 Desired Solution

The desired solution should met the given requirements as optimal as possible. While doing this it should also not be more complicated than needed to make a future use as easy as possible. Cause of this the modularity is an important requirement to ensure this.

Chapter 2

Approach

How I want to get to the Solution and why I choosed it also how it works. Also there should be some theory about state estimation and the kalmanfilter used in this

2.1 Tipps/Notes

- Don't forget point of origin / reference system - Don't forget the pitch angle Maybe do rocket model here

2.1.1 Barometer

-Use density as a statevector
-Use exponential atmosphere method.
-Increasing the R measurement noise matrix when rocket is ascending access the rising uncertainties

2.2 Verification

First of all the test concept has to be defined on which the developed algorithms will be tested. For this the following concept 2.1 was developed.

The theory behind this is that a trajectory is generated by the simulation, which should resemble a real trajectory as good as possible. This function was provided by the simulation team of ARIS from the last years competition. Then this trajectory is applied on the sensor models which are developed in chapter 3. This generates so called perfect sensor data. After this the noise is applied which will also be developed in chapter3, which will result in real sensor data. This noise is drawn out of the sensor log data from the previous test flights. After the realistic sensor data is generated, it serves as the input to the different estimation algorithms.

Then to verify the functionality of those algorithms, the estimated trajectory is then compared against the generated trajectory.

2.3 Sensor models

2.4 Noise generation

2.5 State estimator

Which ones are there and what are their positives and negatives, in short how do they work I will use kalman filter with time depending system and sensor noise, but i have yet to define how i do this in peticullar. I should also make some pictures and such things.

There are many possibilities to do sensor fusion. first of all an all new algorithm could be developed which accesses the stated problems directly. While this solution would be preferebal regarding the efficency, the time ad knowledge needed for this task would exceed the resoursces given in this thesis by far. Also as stated in chapter 1 a lot of theoretical as well as practical pre work is already done and therefore should be used.

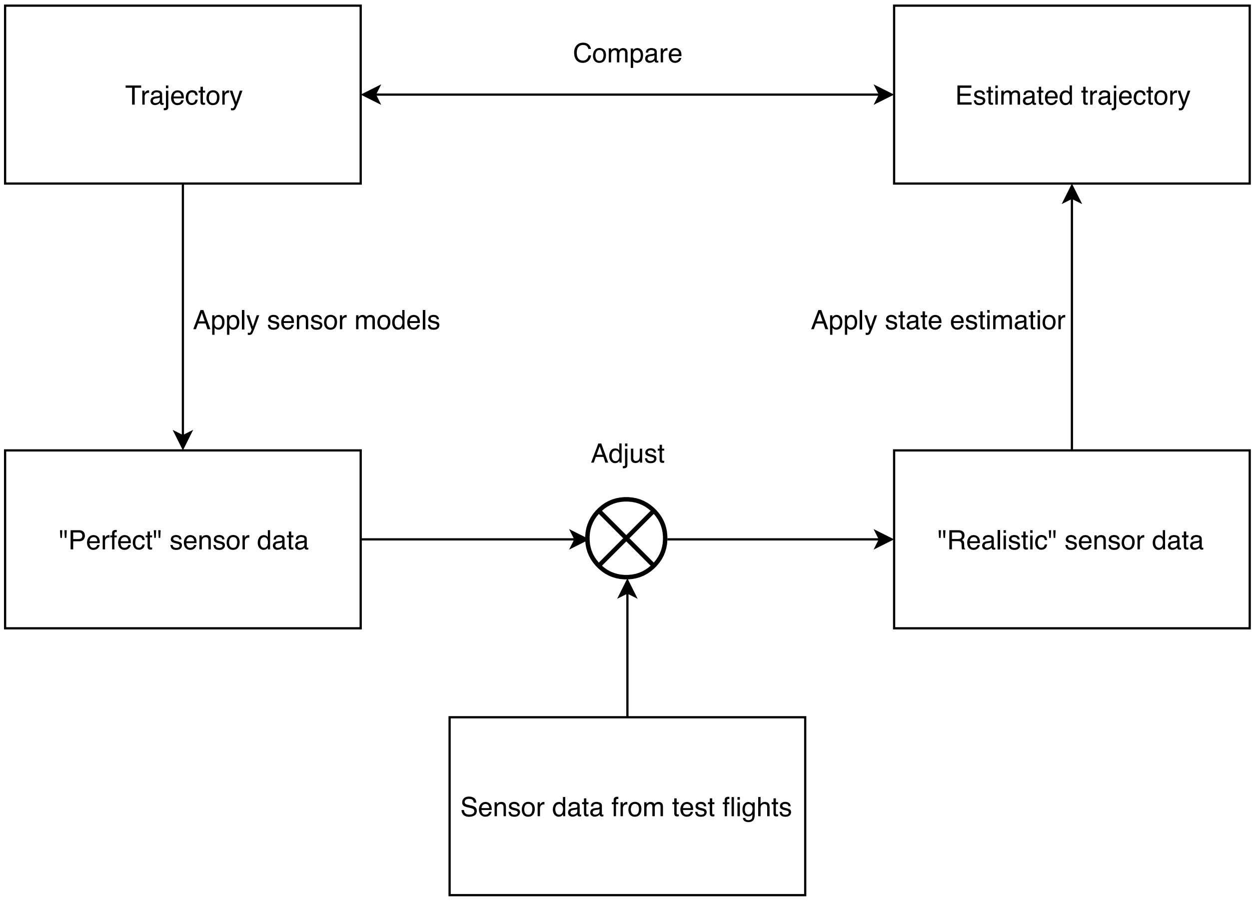

Figure 2.1: Verification Concept

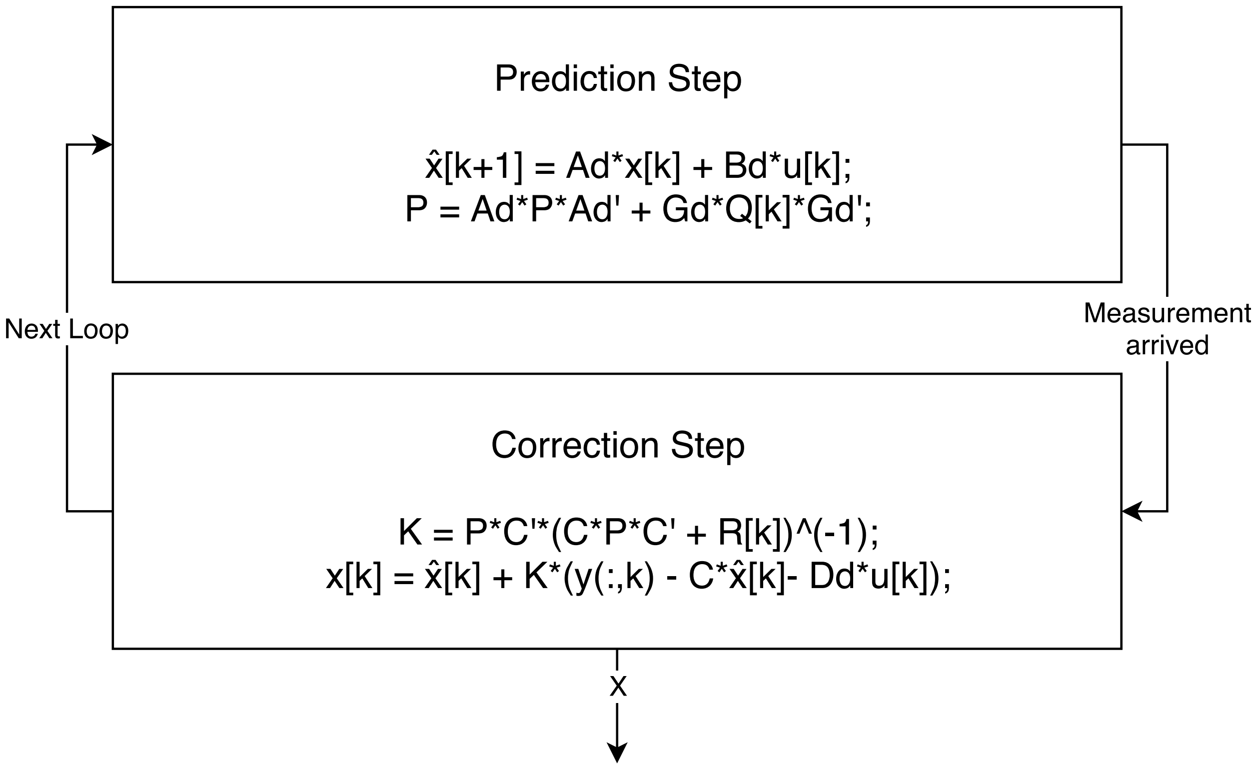

Figure 2.2: Kalmanfilter

2.5.1 Kalmanfilter

The discrete Kalmanfilter was first introduced by Kalman in the year 1960. Its structure provides the optimal estimation of the standard deviation estimation error as long as the noise is Gaussian. But there lies the problem, a physical system is most often not linear. Also the estimated system as well as its variances over the time have to be known to provide this optimal estimation. If the noise matrices of the system are static, the filters gain matrices aim for a fix value and can therefore be calculated in beforehand. This reduces the computational effort by a significant amount. David (2004). It should be mentioned that even if the noise is not Gaussian, the kalmanfilter is still the best linear estimator as long as the system and its properties are well known Simon (2006).

2.5.2 ROSE

The ROSE(rapid ongoing stochastic estimator) is in simple terms three kalman filters in one. Where the main filter is used as stated above, the additional two are used to estimated the the system noise as well as the measuring noise. Therefore this sensor preforms better than the traditional kalman filter if those noises change over time in a not known fashion and has therefore be estimated. Due to this, this sensor needs more computational effort David (2004).

2.5.3 Extended Kalmanfilter

The extended kalmanfilter provides additional parts to better access nonlinearity in the observed system. This by not estimating the state of the system but by estimating the linearized change of the state to the past sate. For this the systems equations have to be derived around the current working point in every estimation state. This is some sort of Bootstrap solution because the nominal point on which the derivation happens are estimated in the process and this estimates are then used to estimate the change between this estimation and the next. Therefore the computational effort exceeds even further Simon (2006).

2.5.4 Unscented Kalamnfilter

The unscented kalmanfilter takes the unscented transformation in use to calculate the different interpolating steps. The unscented transformation uses different points on which the mean and the covariance of a state is known to estimate the change in the next iteration much better than the normal linearized approach. But for this the unscented Kalmanfilter needs also to apply the unscented transformation onto the state vectors in each iteration and does therefore need even more computational effort.

2.5.5 H∞ filter

The H∞ filter is a more diverse approach then the ones described above. It was developed to access the problem when the to be observed system espacially its noise is not well known. Also in contrast to the kalman filters the H∞ filter minimizes the worst-case estimation error in stand of the standard deviation of the estimation error.

2.6 Choosing

If the requirements table 1.1 is taken into the consideration of finding the optimal solution, two main requirements occur that define this decision. First the system load is a critical requirements and has therefore to be addressed in this process. Also for the requirement of modularity the algorithm should be as simple as possible. If taken in regard that the system is more or less well known and that the noise can be determened with the simulation and the log data from previous test flights, a normal kalmanfilter seems to be the most fitting solution. This because the performance of the rocket and the sensor should stay the same during each flight.

2.7 System Model

The developed system or how you want to call it it doesn't really matter. will be hold simple to reduce the system load as well as prevent nonlinearities. Also the important values to estimate are at first hand the vertical height and speed, so for a first implementation just variables that can bed used do determine those both will be used. This are mainly the height and speed from the GPS, the vertical acceleration from the accelerometer as well as the pressure and temperature from the pressure sensors. But even with such simplification there are different possible system description which have to be taken into account to find the best suitable.

2.7.1 Point Mass

The most simple possible model would be, that the rocket would be resembled as a simple point mass which flies perfectly vertical upwards. For this only three state variables would be necessary, the vertical acceleration, the vertical speed and the height.

$$x = \begin{bmatrix} h_z \\ v_z \\ a_z \end{bmatrix}$$

This would reduce the dynamics matrix of the system to a 3x3 matrix with with only two 1 in it:

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

As normal in engineering such a simplification comes with a cost. With this system description the output from the barometer would have to be transformed into the height before they could be taken into the system. Due to this the properties of these sensor could not be estimated correct because the value was transformed in a non linear way before it entered the system. In addition the same problem occurs with the accelerometer. If the rocket develops a pitch angle not equal to 0 during the asccending, it would be measured wrong. To counter this error, the measurements of the accelerometer would have to be weighted with the angle of the gyrometer before entering the system. This weighting is also non linear and the values of the gyrometer are not filtered, which would make the estimation even more uncertain.

2.7.2 Point Mass with Pressure and Temperature

To taken into account to problem stated above, pressure and temperature can be taken into the state vector and therefore be estimated.

$$x = \begin{bmatrix} h_z \\ v_z \\ a_z \\ p \\ T \end{bmatrix}$$

While this solves the problem stated above, it also produces a new. The system model can only describe linear dependencies between the state variables, but the relation between the pressure and the height is clearly non linear in each atmospheric model. This dependency can be linearized, but if done so, it does resemble the athmosperic model with less accuracy. This as to be taken into account an has to be 'told' to the state estimator. The change of the temperature depending on the height is a difficult manor. But to make it simple I use KT_h = -0.0065 °/m in this.

This results in a 5x5 dynamic system matrix which looks like this:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ KP_h & 0 & 0 & KP_T & 0 \\ KT_h & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.7.3 Point Mass with Angle, Pressure and Temperature

The same solution as above can also be applied for the pitch angle. The linearization of small angle changes in the sinus is the angle itself. The assumption that the pitch angle of the rocket wont be to great during the ascention is suitable due to its flight caracterices. This would change the state vector from above into the following.

$$x = \begin{bmatrix} h_z \\ v_z \\ a_z \\ \varphi_{pitch} \\ p \\ T \end{bmatrix}$$

But now how to do this ? describe here how the system description looks in particular
Those described different system will no be implemented in matlab to find the best possible.

2.8 Explenation

How it 'should' work in detail

Chapter 3

Implementation

How it will be Implemented MQTT !!!

3.1 Tipps/Notes

make plots from the different sensors and the log data to show how it was produced an how it resembles the real data.

3.2 Sensor Model

How do i model the different sensors to mimic the real sensors as good as possible.

3.2.1 GPS

Perfect Signal

The perfect GPS signal is rather simple to generate out of the trajectory. For this, just the sampling time has to be adjust. A suitable sampling rate would be between 0.5 and 2 Hz. To maintain the vectors length which simplifies the later use in the estimation algortihm, the signal is aquired with a zero order hold convertion like this:

Noise

The noise capacites of GPS is not real white. It more resembles a brown noise because it has a slow ocsillation over it. I have to determine how i will remodel this in perticular.

Real Signal

3.2.2 Accelorometer

3.2.3 Gyrometer

3.2.4 Barometer

3.3 System Model

6*6 with moving point mass and temperature. 4*4 with just temperature. 3*3 just moving point mass.

3.4 Simulation

The simulation will be implemented in matlab or is already with the different system models.

Chapter 4

Tests

Here come the tests

Chapter 5

Conclusion

5.1 Thanks

Thanks to the Aris Team espacially Thomas and Fabian Also Thanks to Lukas and Jossely

Bibliography

Bryan, T. M. (2012).

David, W. (2004). Application of the kalman filter to rocket apogee detection.

Simon, D. (2006). *Optimal state estimation : Kalman, H∞, and nonlinear approaches.*

Appendices

This is the Appendix