Master Thesis Timeplan

Jonatan Olofsson [jonol402]

January 15, 2012

Introduction

This paper contains the description of the tasks outlined for the master thesis work performed by the author.
Due to physical size constraints, a Gantt chart and relevant statistics are separated from this document, but are relevant for the reader.
The numbering used in this document corresponds directly to the WBS[1] numbering in the activity list appended separately.

Comment regarding the timespan

The project timespan is limited by the LiU semester VT2012. The semester spans from 2012-01-16 to 2012-06-02[2] which corresponds to 20 weeks. The suggested plan also takes into account a reasonable amount of time for the opposition for preparation, and suggests that a more or less final report should be available as of one month before the final presentation.
This limitation in turn imposes a deadline to the project goal - the report is set due two weeks after the milestone where autonomous landing will be presented.

Workload

As shown in in the appended plot, the workload has been roughly evenly distributed over the semester. It should be noted in the plot that 400 hours - 4 hours a day - have been assigned to ”unspecified writing of report”. Since this time is distributed evenly over the entire semester in the plot, the exact amount of working time is somewhat incorrect. The (unrealistic) even distribution should however assure that time is available to the writing of the report throughout the entire project.

Availability of resources

The proposed schedule is quite narrow, and even if I have both tried not to underestimate the work required, and left about 10 % of the time unplanned, there is still little room for delays because of inavailability of hardware. Inavailability of tutoring time is less critical, since the shedule is laid in such a way that work always can be made simultaneously on several activities. It should thus be possible to continue work on other activities while waiting for the availability of a tutor.

[1] Work Breakdown Structure
[2] http://www.lith.liu.se/kalender/terminstider/

1 Primary Linux Computer

The primary Gumstix module will be central to the state estimation and control of the LinkQuad. A lot of the work has been done previously that can be used on this computer, and a lot of the work is therefore focused on assembling existing parts and get it to work together. Also, the time for tuning the parameters should not be underestimated.

1.1 Kalman Filter

1.1.1 Implementation

There exists today an implementation in C++ of a non-linear Unscented Kalman Filter that will be used. There is some implementation details to get it running on the Gumstix module, as well as a few minor theoretical questions in the implementation that I would like to adress.

1.1.2 Tuning

The tuning of filter covariances can be both cumbersome and time-consuming. The work is eased with the availability of the VICON-lab, where a ground truth can be obtained to enable an off-line tuning loop.

1.2 Controller

A study have to be performed here to decide upon control type, control variables and control model.

1.2.1 Implementation

After the type of controller has been decided upon, the work with implementing it is quite easy work.

1.2.2 Modelling

The controlled system should be modelled in some way. In case LQ control is used, a simple state-space model should be devised. If a PID controller is used, using an IMC tuning model is beneficial and a model should thus be constructed in either case.

1.2.3 Parameter Tuning

Here the use of a simulation environment is much beneficial. It might be of interest to introduce a disturbance model in the simulator to be able to test more realistic flight cases.

1.3 Communication

1.3.1 Internal Communication

The internal communication on the Gumstix module will be made using a simple signaling framework. There should not be much work here except compiling it on the Gumstix.

1.3.2 Serial MCU Communication

The serial communication on the Gumstix should be implemented already through the existing LinkBoard communication library.

1.3.3 Ground Ctation Communication

Communication with the ground station should be implemented already through the existing LinkBoard communication library.

1.3.4 Gumstix Communication

Communication with the secondary Gumstix should be implemented already through the existing LinkBoard communication library.

1.4 Logic

1.4.1 State Machine Implementation

The behaviour of an UAV can generally be described as actions performed in one of several states. A general state machine should therefore be constructed with easily defined actions and transition rules. This will be beneficial as one can focus on implenting each given action independently and let the rules define when each action should be selected.

1.4.2 Hover Mode

When the LinkQuad is in hover mode, is only control goal is to stand still in the air. This is the first mode to be implemented, and acts as a test case for further development of more advanced flight modes.

1.4.3 Milestone: Stable Hovering

Here, stable hovering with the LinkQuad should be demonstrated. Optionally, reference hovering positions may be changed to demonstrate movement.

1.4.4 Landing mode

1.4.4.1 State Identification

When performing a full landing, there are several substeps that need to be identified.

1.4.4.2 State Transition Rules

Each substep of the landing have to be defined with its own set of transition rules that define which control state should be activated. This work also includes the detection of a completed landing.

1.4.4.3 Reference Generation

A lot of the work after implementing reference following for the LinkQuad will be spent on creating viable route references that should be followed.

1.4.4.4 Milestone: Autonomous landing

Here, an autonomous landing should be demonstrated.

1.4.4.5 Milestone: Repeated Autonomous Landing

A short while after the previous milestone, repeated landing should be performed where time has been available to fix issues that arised during the first demonstration. This should allow for repeated successful landings of the LinkQuad.

1.5 Gumstix Integration

Since the Gumstix is a full-fledged Linux computer, compiling existing code is not expected to be a big issue.

2 Secondary Linux Computer

The second Gumstix will be running the video processing algorithms and communicate results to the primary Gumstix.

2.1 PTAM Integration

PTAM is a library for monocular SLAM developed at the University of Oxford. Is is in a very experimental state, but is what was used in e.g. [Klein and Murray, 2007, Weiss et al., 2011]. Since getting it to run seems non-trivial, a lot of time has been assigned to getting it integrated.

2.2 Gumstix Integration

Since PTAM may be severely platform-dependent, the implementation will be done directly on the Gumstix.

3 Sensory MCU

3.1 Sensor Interfacing

This should be mostly implemented already with code from IDA

3.2 Milestone: Plotted Sensordata

This milestone is a confirmation that the sensor interfacing is working. Live sensordata should be plotted - filterer or unfiltered.

4 Control MCU

4.1 Control Servos

This should be mostly implemented already with code from IDA

4.2 Milestone: Moving Servos

This milestone should display the availability of control over servos from the Gumstix computer.

5 Ground Station

5.1 Interfacing the LinkQuad

Some kind of interface is needed with the LinkQuad to be able to communicate commands to the logic on the primary Gumstix module. This may be fairly easy to implement, thanks to IDA code, but should nonetheless be well thought through.

6 Peripheral

6.1 Simulation Environment Integration

The LinkQuad has recently been extended with a ROS based simulator. It would be of much value to interface the new developments with this simulator for control parameter tuning etc. This may be non-trivial however and has, as it is deemed an important task, been assigned quite some time.

7 Report

7.1 Unspecified Work

Half the work on the thesis have been assigned to this ”unspecified work” on the report. Hopefully, this will give enough time for a good report and not least, time is assigned to this task from the very beginning of the project.

7.2 Environment Setup

Starting a report of this size requires some setup-time where a formal LaTeX directory structure, figure generation tools and document generation scripts are setup to work together.

7.3 Milestone: Chapter Outline

On this milestone, a preliminary outline of the report’s table of contents should be presented.

7.4 Milestone: Filtering and Control Chapters

Ready for critique.

7.5 Milestone: Video Processing Chapters

Ready for critique.

7.6 Milestone: Hover State Evaluation Chapter

Ready for critique.

7.7 Milestone: All Chapters Ready for Critique

The aim is that the report will be largely finished by the time of this milestone. Some fine-tuning will remain, but the report should be finished to be critiqued.

8 Administration

Some time in the thesis project should be devoted to administration

9 Opposition

This activity contains the preparation time for opposing a fellow student's thesis.

10 Presentation

This activity is dedicated to the preparation of the final presentation of the thesis work.

References

Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

S Weiss, D Scaramuzza, and R Siegwart. Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.