Final Year Project Report

Full Unit - Project Plan

Autonomous Micro Air Vehicles: Enhanced Navigation in GPS denied environments.

Roger Milroy

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science (Artificial Intelligence)

Supervisor: Professor Sara Bernadini

Department of Computer Science
Royal Holloway, University of London

November 7, 2019

Table of Contents

Abstract

I am proposing to use the newly developed technique of Graphical Recurrent Inference Networks in order to improve state estimation in GPS denied environments. I will compare performance to the technique developed by Engel, Sturm and Cremers [5].

One of the key challenges of operating with Micro Air Vehicles (MAVs) is payload and power constraints, both of which also limit onboard computation. In the approach by Engel, Sturm and Cremers they communicate with an offboard laptop in order to overcome this issue. This however obviously introduces other issues such as range and latency. I would like to see whether using GRIN and a different platform will enable me to implement a similar system onboard the drone.

This project will allow me to work with diverse technologies, including state of the art Deep Learning techniques as well as different hardware. This will present challenges in integration that will enable me to develop skills that will trasfer into robotics in general which is an area that I would like to become more involved with.

Specific goals are: to learn how to program in the ROS framework, to integrate two different pieces of hardware into a functioning package, to combine seperate research output into a single application and to increase the accuracy of position of a drone in space without GPS with only onboard computation.

Programming in ROS is a core skill needed by anyone hoping to develop code in robotics, and will be key to the success of this project. Hardware integration is often challenging but in terms of creating a funtioning product it is extremely important. Combining ideas and techniques that have been expressed in research papers into a cohesive whole is a skill in and of itself. Translating mathematical formulations into code and getting it to function correctly is the core challenge of this project. Finally the aim of the project is to demonstrate a technique that can increase the accuracy of position flying without GPS with onboard computation.

Chapter 1: **Background Reading**

1.1 Sources

I read the below listed books, papers and websites while considering my project. I have included a brief explanation of why I read them and how they relate to the project proposed.

- **A Concise Introduction to Models and Methods for Automated Planning, Geffner and Bonet** [6] - I read this book while considering my project. In the end I decided not to use planning in my project and to focus instead on technologies that would enable better performance of drones that do use planning.
- **Artificial Intelligence: A Modern Approach, Russell and Norvig** [10] - As one of the key books on the subject of AI, I will be using this book as a reference throughout the project.
- **Camera-Based Navigation of a Low-Cost Quadrocopter, Engel et al.** [5] - This paper outlines the technique that I want to try and improve upon. One of the key tasks will be to port their implmentation onto the platform that we are using.
- **Accurate Figure Flying with a Quadrocopter Using Onboard Visual and Inertial Sensing, Engel et al.** [4] - This paper extends the ideas in the previous paper and uses them to complete accurate figure flying. I will attempt the same in order to properly evaluate the two approaches.
- **Recurrent Inference Machines for Solving Inverse Problems, Putzky and Welling.** [8] - This paper outlines the techniques used in GRIN and as such is part of the motivation for implementing this technique in this domain. GRIN at it's core is a technique for improving upon model based mathematical models by adding learned error correction.
- **DJI OSDK website** [3]- This is the reference page for the OSDK with which I will be controlling the drone programmatically. This is obviously extremely important to the success of the project.
- **hku_100_gazebo, GitHub caochao39** [2] - This repository contains code and explanation for creating a working simulation of a Matrice 100 in Gazebo. I will attempt to use this package to simulate effectively the Matrice in Gazebo.
- **Medium webpage for setup of Gazebo, Tahsincan Kose.** [7] - This article extends the above repository enabling Hardware In The Loop testing of the Matrice 100 in Gazebo.

Chapter 2: **Project Specification**

2.1 Term 1

The first term will consist mainly of proof of concept programs and reports that will develop my competence with, and understanding of, the necessary technologies that my project involves. This includes some background theory as well as coding. This will serve to both verify that the project is achievable in its current form, the second is to ease the programming load in the second term.

2.1.1 Proof of Concept Programs

1. **Simulate Matrice 100 in Gazebo** - It is important to have a standalone simulation of the model in Gazebo as this will allow testing without having the physical drone.
2. **Get sensor data from the model in Gazebo** - I will need access to the sensor data for higher level tasks so this will be very important.
3. **Port TUM code to ROS Melodic** - This is of crucial importance as this forms the backbone of the project. It currently targets ROS Indigo which is two releases behind. I will assess the workload for this ASAP.
4. **Run TUM code in Gazebo with Hector quadcopter model** - In order to validate the functioning of the TUM code I will run it in Gazebo. This will also involve creating an adapter to communicate with the Hector quadcopter implementation and tuning some variables in the TUM code.
5. **Get ROS and PyTorch working on the Jetson Nano** - In the second term I will be deploying the technique onto the Matrice and the Jetson will form the onboard compute capability. This means it will have to run both ROS and PyTorch in order to compute position as well as relay control commands to the drone.
6. **Write a GRIN implementation for Kalman Filter** - In order to validate the technique, as well as develop my skills in implementing it I will create a relatively simple GRIN. This will be relatively similar to the one suggested in the paper in order to allow comparison as well as limit complexity. This will validate my ability to implement it on the more complex case that is the TUM code.
7. **Train and run the GRIN on dummy data** - This is primarily to validate the correct functioning as well as identifying best practices for training the networks.

2.1.2 Reports

- **Kalman Filters for Sensor Fusion** - Both of the methods I will be contrasting use Kalman Filters and Extended Kalman Filters to integrate different sensor data. I will explore the theory behind this approach to sensor fusion.
- **Graphical Recurrent Inference Networks** - I will write a report covering the technique that I plan to use and implement. This will cover what it is, why it is useful and a brief analysis of its advantages and disadvantages particularly in the context that I wish to use it.

- **Applications of Navigation in GPS Denied Environments** - I will explore the current and potential applications of these techniques. What are the challenges in expanding operations in these environments. What might the impact of expanded applications be. This report will have some relevance towards the Professional Issues section of the Final Report.
- **Graph Convolutions** - GRIN uses convolutions over graphs in order to integrate neural networks with existing model based approaches. I will explore the theory underlying it and how it will apply to implementing GRIN.

2.2 Term 2

The focus of the second term will be implementing the project. This will involve translating the experience, skills and knowledge gained in the first term into a functioning product.

2.2.1 Milestones

1. **Write the GRIN implementation to go on top of the TUM code.** - This is the core of the project. I will create the GRIN and integrate it with the TUM position estimation algorithm.
2. **Gather data and train the GRIN** - I will be using Gazebo to gather most training data due to the ability to recover ground truth data. One challenge of this is that the dynamics of the simulated Hector Quadcopter are likely to be different to the Matrice. If I have time and access to the physical drone I will train with a model of the Matrice instead.
3. **Test and evaluate performance in Gazebo** - This will form the core of the evaluation of my project. I will demonstrate the final achieved accuracy compared with the raw TUM code over a variety of figure flying demonstrations.
4. **Connect the Jetson Nano to the Matrice 100** - In order to demostrate it's real world effectiveness I will need to connect the Jetson to the Matrice. This involves data connection via a UART as well as a power connection for which I will need a converter. There will also be software interfacing that I will need to overcome in order to successfully communicate between the two.
5. **Write an adapter to interface with the OSDK** - The code that I write up to this point will be targeting the Hector Quadcopter implementation (assuming that I have not had physical access to the drone up to this point). In order to operate the Matrice however I will need to communicate with the OSDK. For this I will write an adapter to mediate between different representations.
6. **Deploy on Matrice and Jetson and verify real world performance** - If all has gone well then I will be in a position to demonstrate and evaluate the performance of the project in the real world. This will be the culmination of the project if it goes well.

2.2.2 Reports

- **The structure of the TUM code** - The TUM code is a good example of programming capabilities in ROS. I will examine their structure and the design choices they made in order to better decide how I structure my extension.

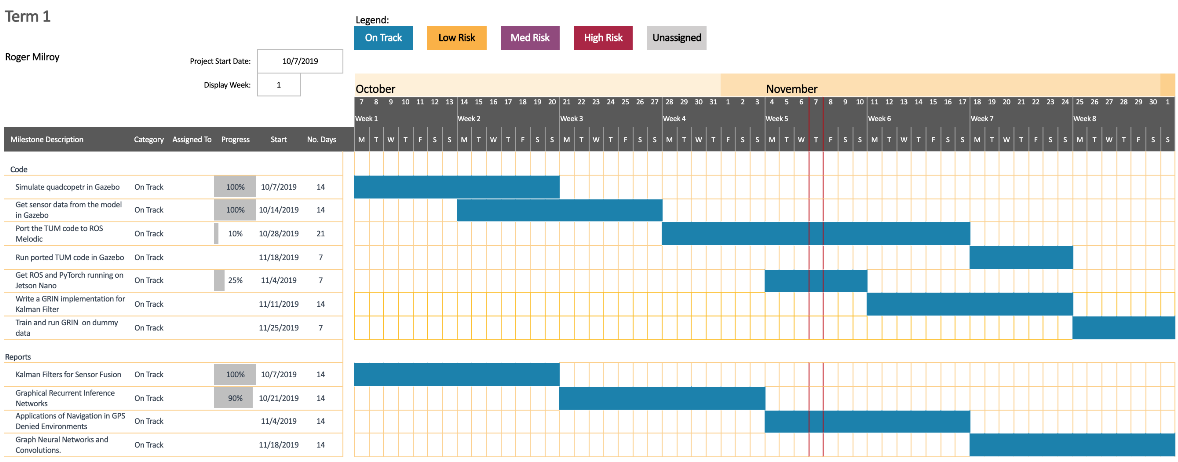
- **Hardware integration.** - I will write a report covering common hardware integration challenges and potential solutions. I will naturally focus on issues relevant to the project though I will organise them under general themes so that general applicability is still maximised.
- **Alternative Navigation Techniques** - It is important when implementing a system like this that you examine and evaluate alternatives. This report will itemize and evaluate a variety of alternative approaches and their relative advantages.
- **Vision Based Navigation Techniques** - I will also explore alternative vision based approaches to this task. I will summarise their advantages and disadvantages, where they are most appropriate and how they relate to the project I am implementing.

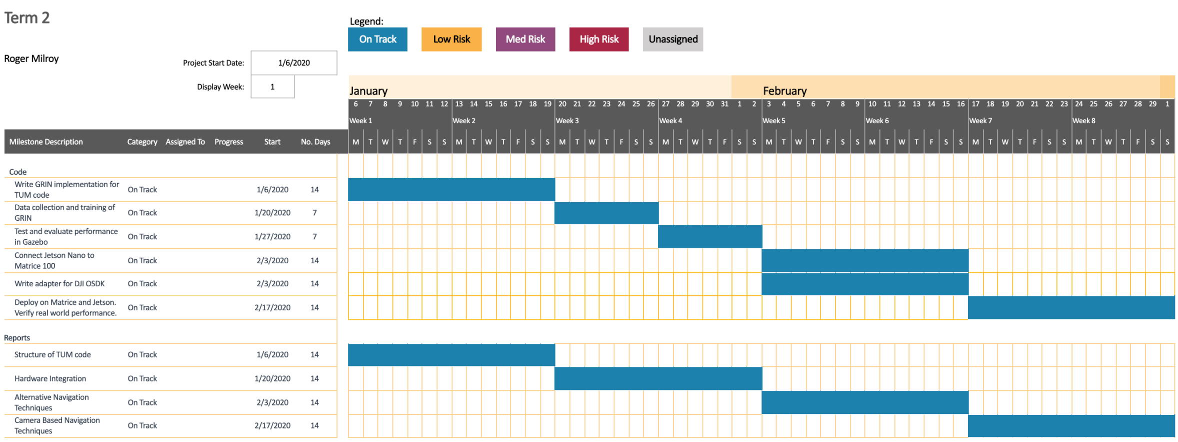
2.3 Timing and Gantt Chart

2.3.1 Motivation

The strategy I have chosen in planning my milestones is intentionally conservative. I am planning for a slow start due to the number and complexity of new technologies and concepts. Once those have been successfully learnt the pace will increase substantially. There is also considerable risk in hardware interfacing so I am completing that phase as early as possible.

I have also allowed some time at the end of each term as a buffer. This is primarily because I understand the limitations of my work and timing estimates. This buffer zone should increase the likelihood of overall project success.





Chapter 3: **Risks and Mitigation**

3.1 Risks

3.1.1 Major Risks

There are a few major risks to the success of the project.

- **Access to information about GRIN.** - While information about RIMs is published and therefore publicly available, material about GRIN is currently unpublished. I have contacted the lead researcher to try to get access but this is by no means guaranteed. This presents the risk that I may have insufficient material to successfully implement the desired project.
- **Porting the existing code of Engel et al.** - As the project was targeted at a completely different platform there is no easy way to assess the difficulty of porting to our platform. This could mean that no comparison is possible.
- **Computational Load** - As the technique of GRIN is very new, there is no current analysis of the computational load. This may make it infeasible to deploy on a Raspberry Pi onboard the drone.
- **Resources for Training** - I am anticipating training the GRIN to take considerable computational resources. It is possible that I will have miscalculated the amount of resources I will need for this stage.
- **Overreach** - This is probably the key risk to this project. It is quite ambitious and presents the risk that the key element of the project will not be completed satisfactorily in time.

3.1.2 Minor Risks

- **Access to the Drone** - There is only one drone between 4 students. This will mean time for development with the drone will be limited.
- **Hardware Issues** - As with any project involving hardware there exist a multitude of potential issues in the interface of hardware and software.

3.2 Mitigations

As with all risks there are actions we can take to minimise the risks. At the same time it is a good idea to plan for the worst case scenario and have a good idea of how to recover and move forward.

- **Access to information about GRIN.** - I will make the best use of my available resources to get in touch with the lead researcher in order to access the necessary information. If I am unable to get access I will pivot the project towards a direct comparison of a reinforcement learning based approach to compare with Engel et al.
- **Porting the existing code of Engel et al.** - One of my first tasks will be to assess the scale of the difference in code base, technologies and architecture. This should allow me to make an early determination if it is feasible or not. If it is not then I will shift from porting to reimplementing if that appears to be feasible. Again if not I will simply compare the results I am able to achieve directly to the results they present.
- **Computational Load** - It is entirely possible that GRIN will not be computationally feasible within the computational and time constraints. If this is the case, I will assess the possibility of transfer training a smaller network. I saw the possibility of this first in An On-device Deep Neural Network for Face Detection [1] and I would use the tips in Fitnets: Hints for thin deep nets [9] if I get to this point. If that is still not possible then I will assess the possibility of using a remote computer, this is not however desireable.
- **Resources for Training** - I am planning to use credits that I have available for AWS towards this. If this proves insufficient I may ask the department for assistance with access to GPU instances.
- **Overreach** - In order to minimise this risk I have already made some decisions relating to timing that I outlined above. In addition to this I have organised the project so that I am being as incremental as possible with lower risk elements coming first so that I should have deliverables to demonstrate even in the worst case.
- **Access to the Drone** - This risk is entirely manageable. The first term will largely not require physical access to the drone and where possible I will be using simulations to reduce dependency further. In any case it shouldn't be too hard to organise access when needed.
- **Hardware Issues** - These are quite hard risks to remove in that they are mostly unknown and dealing with them will largely have to be reactive. That being said I will be verifying various levels of the different hardware in the first term in order to get ahead of any potential issues. I should have a good handle on the hardware by the end of the first term and therefore be better able to plan a hardware strategy for the second term.

Bibliography

[1] An on-device deep neural network for face detection - apple. https://machinelearning.apple.com/2017/11/16/face-detection.html, Nov 2017.

[2] caochao39. caochao39 on github hku_m100_gazebo. https://github.com/caochao39/hku_m100_gazebo.

[3] DJI. Onboard sdk overview. https://developer.dji.com/onboard-sdk/.

[4] Engel, J., Sturm, J., and Cremers, D. Accurate figure flying with a quadrocopter using onboard visual and inertial sensing. *IMU 320* (01 2012).

[5] Engel, J., Sturm, J., and Cremers, D. Camera-based navigation of a low-cost quadrocopter. pp. 2815–2821.

[6] Geffner, H., and Bonet, B. *A Concise Introduction to Models and Methods for Automated Planning: Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1st ed. Morgan & Claypool Publishers, 2013.

[7] Kose, T. Implementing dji m100 emulator in ros-gazebo for hardware in the loop. https://medium.com/@tahsincankose/implementing-dji-m100-emulator-in-ros-gazebo-for-hitl-e79f4bb077f6, May 2019.

[8] Putzky, P., and Welling, M. Recurrent inference machines for solving inverse problems. *CoRR abs/1706.04008* (2017).

[9] Romero, A., Kahou, S. E., Montréal, P., Bengio, Y., Montréal, U. D., Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *in International Conference on Learning Representations (ICLR* (2015).

[10] Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.