Enabling Multirotors to Perform Construction Tasks Using Swarm Algorithms

Jens-Christian Finnerup

Kongens Lyngby 2016

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

The goal of the thesis is to ...

Summary (Danish)

Målet for denne afhandling er at ...

Preface

This thesis was prepared at DTU Compute in fulfilment of the requirements for acquiring an M.Sc. in Engineering.

The thesis deals with ...

The thesis consists of ...

Lyngby, 01-August-2016

Not Real

Jens-Christian Finnerup

Acknowledgements

I would like to thank my....

Contents

Summary (English) **i**

Summary (Danish) **iii**

Preface **v**

Acknowledgements **vii**

1 Introduction **1**
1.1 Problem Description 2
1.1.1 Main Problem 2
1.1.2 Sub Problems 3
1.2 Project plan 3

2 Methodology **5**
2.1 Control Problem 5
2.2 Simulation Implementation 6
2.3 Validation 6
2.4 Test Metrics 7
2.5 Control Implementation 7
2.6 Optimization 8

3 Simulation **9**
3.1 Software 9
3.2 Specifications & Assumptions on Physics 10
3.3 Simulation vs. Animation 11
3.4 Reality Criteria 11
3.5 Control Problem 11
3.6 Implementation 11

3.6.1 Sources and Sinks 12
3.6.2 PID Controller 12
3.6.3 Gravity & Impulse 12
3.6.4 Control Loop Controller 12
3.7 Validation 12
3.8 Test Metrics 12

4 Swarm Logic & Swarm Algorithms 13
4.1 Multi-body vs. Multi-agent path planning 13

5 Design 15

Bibliography 16

Chapter 1

Introduction

In today's world, we increasingly rely on robotics to perform a wide range of tasks, with use cases ranging from complimenting, or even substituting, human labour in factories to more complex tasks such as automated aerial photography. Though technology continues to improve, the use of robotics is often confined to within predictable environments, where machines are told what to do and when to do it. More recently though, as control algorithms improve, autonomous robots are starting to become a practical reality, with increasing ability self-plan and work unsupervised. As such we experience a paradigm shift, where robots are no longer limited to predictable and confined environments, but can act freely and adapt to changing circumstances.

As a consequence of this, new use cases and areas within robotics are starting to gain interest. Specifically the area of autonomous flying vehicles is gaining interest with companies such as amazon, preparing to use flying drones to deliver packages. This requires the drone to be able to plan according to changes in the environment as well as being able to coordinate its movements in relation to other drones around it. At a currently more theoretical scale, using flying drones to perform construction tasks has been subject for testing by universities such as *ETH Zurich*, where micro drones were used to construct a 6 meter tall tower [ALH$^+$14].

The task of using flying vehicles to perform construction is especially interesting because it combines many of the challenges that face robotics today, as agents become more autonomous. A construction task work as a great test bed, for testing flying drones as it involves multiple agents working on the same problem as once and as such they must have the ability to plan according to changes in their individual environment.

One solution to the challenges of enabling drones to collaborate, is to create individual sectors of space in which only a single agent is allowed to move at a certain time. This is a simple way to prevent collisions between the drones, but it also poses some optimization challenges, as the path of the drones cannot intersect. This means that even if a path for a drone towards a goal is optimal (shortest), it cannot be used if it intersects with the path of another agent. Another drawback of this simple solution is that it limits the amount of agents able to act with in a certain space, to a small number, as reach has to have its own designated area. As such there is great motivation to design control algorithms, which can enable drones to collaborate on tasks, while allowing them to avoid collisions without limiting their flight space.

This thesis focuses on how create and use algorithms from the field of swarm robotics as control for flying drones. The different algorithms will be implemented on a simulation test bed, to compare their complexity and their ability to give optimal movements to the drones within the simulation. For the comparison, different metrics will be used measure effectiveness in allowing drones to solve a collaborative problem. As a basis for the comparison, this thesis will document the implementation of the simulation platform and the control algorithms.

1.1 Problem Description

The following will outline the detailed goal of this thesis, as well as the subgoals necessary to achieve it.

1.1.1 Main Problem

To create and use swarm algorithms to enable quadcopters to coordinate their movements in a multi-agent environment.

1.1.2 Sub Problems

- To design and implement a simulation platform, based on a physical model, within which the algorithms can be used to control simulated agents
- To design a problem for the agents to collaborate on, to use as a standardized test for performance of the algorithms
- To determine the correct metrics for optimizing and comparing the swarm algorithms
- To implement swarm algorithms based on existing concepts, and test them using the metrics and the simulation platform
- To optimize the control algorithms based on pre mentioned tests, and analyze potential areas of improvement

1.2 Project plan

We note that the contents of the project plan is also something we would like to see in the introductory chapter of your thesis. In fact, you can reuse your final project plan (possibly extended) as the introduction. If you prefer to write an introduction from scratch, it is, of course, important that it is consistent with the final project plan.

Chapter 2

Methodology

In order to evaluate the different control algorithms and their performance, the approach towards designing and testing the algorithms must be systematic and concise. The different subproblems specified in 1.1.2, can be said to either relate to either the simulation design or the algorithm design. As such, for the purpose of this thesis, a different approach will be used when dealing with constructing the simulation and when designing and testing the control algorithms. In this chapter, I will outline these approaches and the methodology used. The first three sections of this chapter will deal with the simulation methodology, whereas the latter three will deal with the implementation and testing of the control algorithms.

2.1 Control Problem

The main research problem os this thesis involves building and evaluating different control algorithms, and as such each algorithm must be subject to the same evaluation technique. This evaluation technique will throughout the thesis be referenced to as the *Control Problem*, and the purpose of the agents will be so solve this problem collectively. As such the problem chosen must be the same problem throughout every simulation and independent of the control algorithms themselves. The problem must also be able to be solved collectively, by multiple agents at the same time, while still posing enough challenges to properly test the control algorithms. Posing dangers to the agents, such as the risk of collision.

This control problem will exist within the simulation as a physical task that the agents have to carry out. Chapter 3.5 will outline the concrete control problem used, as well a discussion on the specific control problem's advantages and limitations.

2.2 Simulation Implementation

The control problem and the simulation are mutually dependent, as the problem has to able to exist and be solvable within the simulation. As such the simulation has to be designed with the control problem in mind, but the control problem also has to be made with the limitations of the simulation taken into account. The simulation aspect is crucial to the validity of the results expressed in this thesis. For the purpose of solving the problems outlined in this thesis, the simulation will be a virtual environment, in which drones can be deployed. The drones will have to be subject to the same physical laws, as they would in real life such that the findings and test results can by applied to issues and control techniques in the real world.

In order to reduce complexity of the simulation, some physical laws are not taken into account when designing the simulation. This is done based on assumptions on which laws influence the results. An outline of the simulation design, can be found in Chapter 3.

2.3 Validation

Validation is a method to ensure that the simulation is mimicking real life physics and not simply animating the desired behavior of the drones. The validation will be carried out iteratively along with the construction of the simulation, to monitor the progress of the simulation construction, and act as a sort of checklist to the accuracy and capabilities of the simulation. The validation phase consists of multiple validation tests, meant to test if the simulation actually simulates the physical mechanics required. These mechanics can range from gravitational pull to accurate impulse and mass on the bodies in the simulation. The validation tests will when range from checking the accurate acceleration of drones in free fall etc. These validation tests are also important because the simulator must have the predictable behavior across many scenarios, such that they can be compatible. The validation tests will be discussed and outlined in Section 3.7

2.4 Test Metrics

Up until now this chapter on methodology has been focused on systematically creating and validating the simulation platform. The following part will be covering my approach to creating and testing the control algorithms.

The comparison of the algorithms tested within this thesis, will be done based on their ability to solve the Control Problem. However the control problem must be designed in such a way, that it can be solved to various degrees. To measure how well the control problem is solved, some metrics must be chosen so that they represent the challenges of the control problem. I.e. if the control problem challenges the agents in collision avoidance, a test metric could be each agent's distance the all other agents over time. Time to solve the problem will also act as a simple test metric. The metrics chosen will be covered in Chapter 3 and will in part be based on the challenges of swarm robotics covered in `Multi-robot navigation in formation via sequential convex programming` [AMBR15].

2.5 Control Implementation

The algorithms tested within this thesis and presented as *swarm algorithms* will be covered in depth in Chapter 4. When implementing and testing the algorithms in this paper, the algorithms will be partly implemented through experimentation. This is due to the fact that swarm algorithms such as the *Artificial Bee Colony* (or ABC) [BRG^{+}11], cannot directly be mapped to the issue of controlling quadcopters to perform construction tasks. The ABC has a different number of agents and multiple categories. Because of this, mapping some of the algorithms will involve some experimentation and for the sake of methodology it must be stated that algorithms will undergo a level of personalization when implemented. This will be documented along with how much the algorithms ends up differing from its theoretical counterpart.

2.6 Optimization

In contrast to the Simulation Implementation, the implementation of the control algorithm will not be followed by a validation step in the same way. This is due to the fact the control algorithms cannot be validated with the same fixed criteria, as their desired behavior deviates for each algorithm. Instead, they will undergo optimization. This refers to the activity of not only mapping the control algorithms to control quadcopters, but also tuning them in an attempt to maximize their performance.This will be in benchmarked with the test metrics and the control problem within the simulation. The optimization will only be done on the control algorithms themselves, and not by tuning simulation variables, as all algorithms bust be tested on the same simulation. The simulation variables will be predetermined in the *Validation* phase (see Section 3.7) and hereafter remain static.

Chapter 3
Simulation

As mentioned previously in this paper, there are many advantages to performing simulation vs. carryings

The simulation environment, hereafter referred to as simple *the simulation*, will be the abstract platform on which the quadcopters and their control algorithm can be tested. In this chapter i will outline the construction of the simulation platform, as well of the validation used to ensure that the simulation follows the required physical mechanics.

3.1 Software

The software used in this paper to create the simulation platform is Matlab Version 8.7 (R2016a) with Simulink [?]. Simulink was chosen based on its ease of use and ability to perform simulations continuous time. Simulink also has a 3D graphics engine called `VR3D`[?], which can be used to model 3D animations from any signal output from the simulation model. I.e. if the simulation models a flying quadcopter, the position values of the quadcopter can be sent as signals to the 3D model. The ability to 3D animate the movements of the quadcopters is important, as *seeing* their actual flight paths can be a simple way to discover errors in either the simulation model or the control algorithm.

Parameter	Description	Value
Agent-type	What type of agent to be simulated	Quadcopter
Y-acceleration	Mechanic to simulate Gravity	$9.8m/s^2$
Number of agents	The number of quadcopters acting to solve the control problem	5
Mass of agent	The assumed mass of one agent throughout the simulation	$150g$
Number of boxes	The number of boxes to be moved to their respective goal position	9
Mass of box	The assumed mass of one box throughout the simulation	*massless*
Maximum velocity	The limit of the velocity of any object on every axis	$15m/s$
Maximum acceleration	The limit of the acceleration of any object on every axis	$9.8m/s^2$
Attach threshold	The maximum distance from an agent to a box, before the box can be attached	$2cm$

Table 3.1: Environmental Specifications

3.2 Specifications & Assumptions on Physics

The goal of the simulation is to simulate multiple flying quadcopters, and as such physical mechanics play an important role of the simulation model. This section will list different physical parameters, a description of them as well as their importance in the simulation model. Due to the desire to limit complexity some physical parameters, such as wind, will be assumed to be non-existent in the simulation model. Others such as gravity and impulse play a more vital role in producing accurate test results.

3.3 Simulation vs. Animation

3.4 Reality Criteria

3.5 Control Problem

The Control Problem of the simulation, will be a construction task.

3.6 Implementation

After assessing the needs of the simulation model, the actual construction in simulink is done by connecting different modules in a circuit like manner. Explanations of the simulation construction will be done with references to the finished model, which can be seen in Figure 3.1. The modules and submodules of the simulation model have been labeled with a title, describing their function.

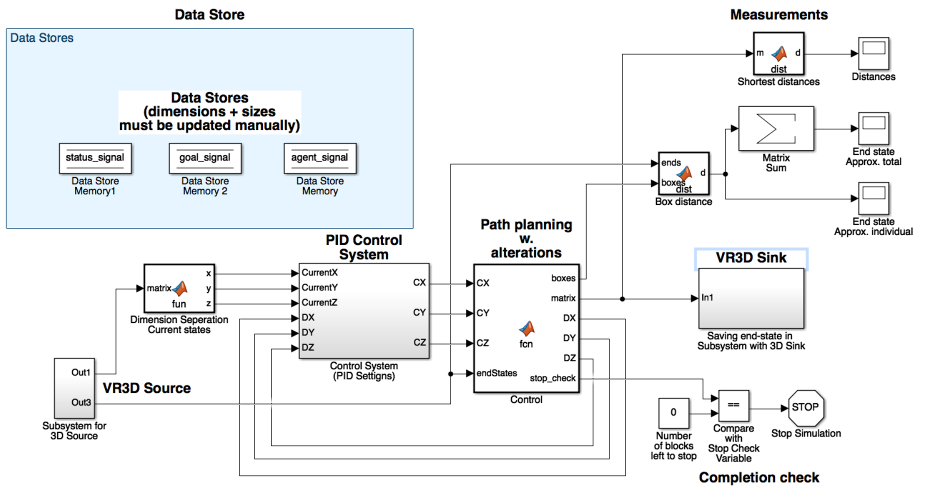

Figure 3.1: Overview of finished model in Simulink

The following subsections will outline how the different model requirements outlined in Section 3.2, were achieved in a step wise manner.

3.6.1 Sources and Sinks

The simulation is carried out with a start and an endpoint

3.6.2 PID Controller

As mentioned in Section 3.3, it is important the control systems cannot alter the position of the quadcopters directly, but instead gives instructions to the quadcopters to change their thrust etc.

3.6.3 Gravity & Impulse

3.6.4 Control Loop Controller

3.7 Validation

3.8 Test Metrics

CHAPTER 4

Swarm Logic & Swarm Algorithms

4.1 Multi-body vs. Multi-agent path planning

Chapter 5

Design

Bibliography

[ALH$^+$14] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea. The Flight Assembled Architecture installation: Cooperative construction with flying machines. *IEEE Control Systems*, 34(4):46–64, August 2014.

[AMBR15] J. Alonso-Mora, S. Baker, and D. Rus. Multi-robot navigation in formation via sequential convex programming. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4634–4641, September 2015.

[BRG$^+$11] Preetha Bhattacharjee, Pratyusha Rakshit, Indrani Goswami, Amit Konar, and Atulya K. Nagar. Multi-robot path-planning using artificial bee colony optimization algorithm. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, pages 219–224. IEEE, 2011.

[CPD$^+$14] Ruaridh Clark, Giuliano Punzo, Gordon Dobie, Rahul Summan, Charles Norman MacLeod, Gareth Pierce, and Malcolm Macdonald. Autonomous swarm testbed with multiple quadcopters. In *1st World Congress on Unmanned Systems Enginenering, 2014-WCUSEng*, 2014.

[DHB15] Shreyansh Daftry, Christof Hoppe, and Horst Bischof. Building with drones: Accurate 3d facade reconstruction using mavs. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3487–3494. IEEE, 2015.

[HM99] Owen Holland and Chris Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial life*, 5(2):173–202, 1999.

[LGB$^+$16] Pierre LATTEUR, Sébastien GOESSENS, Jean-Sébastien BRETON, Justin LEPLAT, Zhao MAb, and Caitlin MUELLER. Drone-Based Additive Manufacturing of Architectural Structures. 2016.

[mat16] matlab matlab. MATLAB and Simulink for Student Use - Math software for engineering and science students - MathWorks Nordic, 2016.

[PKG$^+$02] Giovanni Cosimo Pettinaro, I. Kwee, Luca Maria Gambardella, Francesco Mondada, Dario Floreano, Stefano Nolfi, J.-L. Deneubourg, and Marco Dorigo. Swarm robotics: A different approach to service robotics. In *33rd International Symposium on Robotics*, pages 71–76, 2002.

[SMVDPB05] John A. Sauter, Robert Matthews, H. Van Dyke Parunak, and Sven A. Brueckner. Performance of digital pheromones for swarming vehicle control. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 903–910. ACM, 2005.