

# Continuous Pursuit-Evasion with Aerodynamic Constraints

Part II Computer Science Project Proposal

Brace Godfrey, dg681

October 3, 2025

<b>Project Originator:</b>	Brace Godfrey
<b>Day-To-Day Supervisor:</b>	Ramsey Faragher
<b>Marking Supervisor:</b>	Alastair Beresford
<b>Director of Studies:</b>	Ramsey Faragher
<b>Project Checkers:</b>	??? and ???

# 1 Introduction

Pursuit-Evasion is a problem domain in which one or more pursuers attempt to capture one or more evaders in some environment. This field has applications and links to game theory, spatial algorithms and airflight kinematics. In this project I will consider a continuous space populated by aerodynamically-accurate entities and investigate how a Markov Decision Process (MDP) can be used to model and solve pursuit-evasion problems in this space.

The core of my project will incorporate work from [1] where I will implement their **FastMDP** algorithm for efficiently solving pursuit-evasion problems in continuous space using a Markov Decision Process. I will extend this work

I will use python

## 2 Structure of the Project

The project will be structured as XYZ

Visualisation

Modelling

Markov Decision Processes

The project will have the following extensions

### Extension X

X

### Extension Y

Y

### Extension Z

Z

### **3 Starting Point**

My starting point is XYZ

## **4 Evaluation**

### **4.1 Success Criteria**

For the project to be deemed a success, the following must be successfully completed:

1. X
2. Y
3. Z

### **4.2 Evaluation Plan**

It will be evaluted as XYZ

## 5 Timetable and Milestones

### Weeks 1 to 2 (17 Oct 22 - 30 Oct 22)

Proposal Submitted

Learn best software development practices in OCaml and research tools available for creating the frontend of a compiler (e.g. lexer and parser tools).

Familiarise myself with Go syntax. Begin designing the syntax for Kautuka.

Begin writing the Introduction draft in dissertation.

### Weeks 3 to 4 (31 Oct 22 - 13 Nov 22)

Finish designing the syntax for Kautuka.

Implement the lexer and parser for the language and write tests to verify that the output abstract syntax trees are correct. Write a compiler to convert the output abstract syntax tree into Go code, which at this point will closely resemble the source code we started with.

Finish writing the Introduction draft in dissertation.

Milestone: Given a Kautuka program, generate the corresponding abstract syntax tree

### Weeks 5 to 6 (14 Nov 22 - 27 Nov 22)

Design and formalise effect tracking for my language. Then implement this into the compiler to generate lists of effects for all functions in the program. Add more tests to verify that this has been done correctly.

Begin writing the Implementation draft in dissertation (specifically adding my formalised Kautuka syntax and effect tracking inference rules).

Milestone: Given a Kautuka program, determine the list of effects for all of the functions

### Weeks 7 to 8 (28 Nov 22 - 11 Dec 22)

End of Michaelmas Term - start of Christmas holidays.

Design and formalise my proposed bound-based type system. Then implement this into the compiler. Collect data surrounding the execution time for the standard functions and the overheads involved with creating threads.

Continue writing the Implementation draft in dissertation.

Milestone: Extend Kautuka with my bound-based type system, represent these bounds in the generated abstract syntax tree

## **Weeks 9 to 11 (12 Dec 22 - 1 Jan 22)**

Finish collecting data surrounding execution times and create an algorithm to predict how long functions will take on inputs of different sizes. Using my type system and the collected data, now extend the compiler to estimate the runtime of the user's functions. Finally we can compile functions into different threads if there is sufficient cost benefit.

Start my evaluation.

Finish writing the Implementation draft in dissertation.

Take time off for Christmas

Milestone: Estimate the runtime of all user defined Kautuka functions. Compile Kautuka to parallelised Go based upon the analysis

## **Weeks 12 to 13 (2 Jan 22 - 15 Jan 22)**

Add better type system inference and create further tests to ensure that the program is compiled correctly.

Evaluate the runtime of Kautuka compared to the semantically equivalent, sequential Go code.

Begin writing the Evaluation draft in dissertation.

Milestone: Finish core project implementation and pass all success criteria

## **Weeks 14 to 15 (16 Jan 22 - 29 Jan 22)**

End of Christmas holidays - start of Lent Term.

Slack time to finish core project implementation.

Make a start on progress report.

## **Weeks 16 to 17 (30 Jan 22 - 12 Feb 22)**

Finish progress report for the deadline Fri 3 Feb 2023. And create progress report presentation for the deadline Wed 9 Feb 2023.

Finish writing the Evaluation draft in dissertation.

Milestone: Submit progress report

## **Weeks 18 to 19 (13 Feb 22 - 26 Feb 22)**

Make a start on extensions if time permits. Mainly focus on extensions 1 and 4.

Start Conclusion draft in dissertation.

## **Weeks 20 to 21 (27 Feb 22 - 12 Mar 22)**

Continue with extensions if time permits. Now focus on the remaining extensions.

Finish Conclusion draft in dissertation.

Milestone: Finish dissertation draft

## **Weeks 22 to 23 (13 Mar 22 - 26 Mar 22)**

Slack time to finish the dissertation draft.

## **Weeks 24 to 25 (27 Mar 22 - 9 Apr 22)**

Submit draft of the dissertation to my supervisor and DoS for review by the deadline Fri 7 Apr 2023. Begin exam revision and review code repository in the meantime (ensure that my code is clear, readable and well structured).

Make improvements to the dissertation based upon this feedback.

## **Weeks 26 to 27 (10 Apr 22 - 23 Apr 22)**

Finish improvements to the dissertation and submit to my supervisor a second time for final review.

Make last adjustments and focus on making the dissertation presentable and easy to read.

## **Weeks 28 - 29 (24 Apr 22 - 7 May 22)**

Slack time for any extra additional evaluation and tests that may be discovered during final reviews of my dissertation, and start revision for exams.

Milestone (28 April 22): Submit Dissertation (2 weeks in advance of final deadline)

## **6 Resource Declaration**

I will be using my personal laptop (Macbook Pro 2021 — 16GB RAM, Apple M1 Pro) as my primary machine for software development. As a backup, I will use my personal desktop computer (32GB RAM, Intel 11th Gen Core i9) or a remote server provided by SRCF (student run computing facility). I will continuously backup my code and dissertation with Git version control and push to a remote GitHub repository.

## References

- [1] J. Bertram and P. Wei. “An Efficient Algorithm for Multiple-Pursuer-Multiple-Evader Pursuit/Evasion Game”. In: *AIAA Scitech 2021 Forum*. 2021.