**Designing and making judging aid for canoe slalom**

**CM3203 – One Semester Individual Project – 40 Credits**

**Final report**

A red sign with white text

Description automatically generated with medium confidence

**Cardiff University School of Computer Science and Informatics**

**Author: Harry Batchelor (C18163377)**

**Supervisor: Dr Ian Cooper**

**Moderator: Dr** [**Yipeng Qin**](https://pats.cs.cf.ac.uk/!user_info?u=scmyq)

# Abstract

In most professional sports some form of video judge is being implemented. Cricket was one of the earliest sports to implement this technology with Hawk-eye back in 2001. With bowlers, bowling at around 90mph, it can be hard for TV spectators to follow the, but with Hawk-eye it allows the spectators and the umpires to replay these fast pasted moments to allow them to make better decisions on close calls.

Canoe slalom is no different. With athletes trying to reduce their times by seconds and cutting it as close as possible to the gates, a judge can only see so much from the bank. This project aims to design and make an accessible system for these canoe slalom judges to allow them to make better split decision calls. The project also aims to evaluate the success of the solution and verify if it could be scaled up and used in competitions at all levels.

# Acknowledgments

I would like to thank my supervisor, Dr Ian Cooper whose friendly advice and guidance throughout the whole project allowed me to work through the hurdles I encountered in the project.

I would also like to thank members of Seren Dwr Canoe club for allowing me to come down to their training sessions and collect data.

Table of Contents

[Abstract 2](#_Toc97303574)

[Acknowledgments 2](#_Toc97303575)

[Introduction 2](#_Toc97303576)

[Background 3](#_Toc97303577)

[A beginners guide to canoe slalom 3](#_Toc97303578)

[Approach 3](#_Toc97303579)

[Implementation 3](#_Toc97303580)

[Results and evaluation 3](#_Toc97303581)

[Future work 3](#_Toc97303582)

[Conclusions 3](#_Toc97303583)

[Reflection on learning 3](#_Toc97303584)

[Appendix 3](#_Toc97303585)

[References 3](#_Toc97303586)

# Introduction

As with any top level sport, canoe slalom races often come down to the finest margins, as such paddlers have been trying to find the smallest margins possible to help them win races. This has evolved over time with boat manufacturers using the lightest materials to reduce weight and allow the paddler to move faster. In more recent years, manufacturers such as PeakUK have been developing garments such as the racer ST2020[[1]](#footnote-1)which take the vital buoyancy aid on the front of the paddler and integrate it into the spray deck, this allows for the paddlers chest to be unimpeded with the traditional buoyancy aid. Although these new designs only eliminate a few millimetres from the chest, it allows the paddlers to get even closer to the poles. This new innovation meant that PeakUK athletes at the 2020 Tokyo Olympics were able to win a total of 9 out of the 12 possible medals[[2]](#footnote-2).

Figure 1 - PeakUK Racer ST2020, showing now foam on the front, to allow for tighter turns around the poles. Image Source: (PeakUK, 2020)



With the margins being as thin as mentioned above, the pressure that the judges are under to make accurate calls are huge, especially when a penalty means the difference between 3rd and 4th place, as with the Olympic games this summer[[3]](#footnote-3). That is why, my project aims to create a product that will aid judges in making these split decision calls while the race is still going, without necessarily using a video judge, which can be a lengthy process, and is also only available at the top level of racing.

Although the project is aimed for the judges use at the most competitive levels of sport, it could also be used as a training tool for all the athletes in the sport. Canoe slalom coaches have to perform two job at the same time, one as a judge making sure their paddler makes it through the gates and secondly as a coach to see what the paddler can improve on. If my project was used, the role of the judge can be taken away and the coach can focus on coaching. The coach will also be able to gain more data for quantitative feedback from the various sensors and outputs on the poles allowing them to gain tip for which poles their paddler can get even closer on without.

As of the 2018 season, the ICF (international canoe federation) has brought in a new system for video judging for only the world cups and the world championships[[4]](#footnote-4). Compared to tennis or cricket who have been using video judging since 2001. Even then in canoe slalom there is a very large team of judges, some watching the live video feed of the run and some reviewing the replay of any contested decision. Whereas in a sport such as rugby, there is only one video judge who runs it all. My proposed project would utilise the judges already on the river bank. These judges would first watch the paddler come through the gates and if they then decided there is a close call they can refer to the mobile device which will have data for them to review instantly. All of this will reduce the pressure on the judges, reduce the number of judges needed, and cut down on the time taken to the review close calls.

# Background

### A beginners guide to canoe slalom

The main focus of this project isn’t to teach the reader about the intricacies of canoe slalom, an understanding of this sport is useful to fully understand the project. A full list of canoe slalom terminology can be found in the appendix.

In canoe slalom athletes race down a roughly 200 meter long course consisting of a minimum of 18 and maximum of 25 different gates, of which 6 must be upstream gates. Paddlers can occur time penalties of 2 seconds for having a touch on the poles, this can be from any part of the paddlers body, or any of their equipment. The athlete can also gain a 50 second penalty for not having their complete head go in between the two poles, or for navigating the course in the wrong direction or order.

The way this is judged is with two judges having a set of 4-5 gates between the two of them. They will watch the paddler head down the course and report back to the race control what the results are. This will be in the form of numbers, for example a “zero, zero, 50, two” means that the paddler has successfully navigated the first two gates, missed the third gate, and touched the fourth gate. At the top level of the sport, if there is a disagreement between the two judges or they aren’t sure on what to give they might be able ask the video judge to review the footage they have.

* 1. Current implementations

A picture containing text, outdoor

Description automatically generatedOn a typical canoe slalom broadcast such as the Olympics or the world cups we never get to see any of the judging or video judging taking place. The viewer might get some slow motion replays of the gates if there is time between each paddler, but there is never any formal type of judging shown. The only aid that the viewer has is a small icon showing if the paddler has hit or missed the gate.

Figure 2 – Typical canoe slalom world cup broadcast, with icon showing penalties. Image Source: (Planet Canoe, 2021)

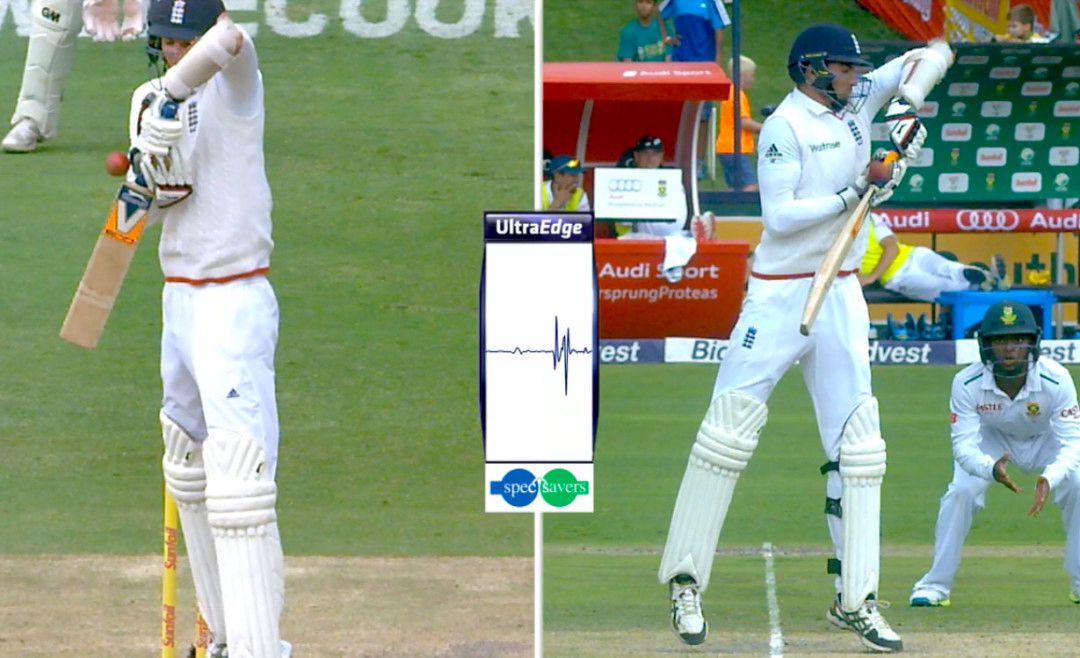
Although my project isn’t aimed towards viewers of canoe slalom, there is scope for the acceleration graphs produced by the accelerometers to be displayed alongside the replay of the gate that is under question. This would be something similar to ultra-edge in cricket where they have the sound wave produced, helping the viewers and umpires make their decision.

Figure – 2 Ultra-edge used in cricket. Image Source: (inshorts, 2016)

* 1. Tools used

#### Flask

The first and arguably the most important tool I used was Flask. This is a popular web app framework for python. I decide to use Flask over other frameworks such as Django as I already had some experience with using Flask in other project. Flask is also lighter weight which was important for me as everything would be running of a raspberry pi, which are notorious for struggling to run programmes.

Flask was able to provide the basic functionality to get everything the canoe slalom judge would need onto a web app. This was a vital criteria for the project, without the use of a web app the judges would need a lot more training on the programme to be able to use it effectively. Flask also has different protocols which have been very useful when building the programme. The most common request I have been using has been the GET request allowing me to get data from the flask server, this can be anything as simple as a flask route or html template.

#### SQLite

SQLite has been another backbone to the project. SQLite is a small and fast database engine. SQLite allowed me to easily link the data from the accelerometers to the flask server talked about above. Storing the data within the database also allowed for easy retrieval for plotting graphs to show changes in the accelerometer date.

The reason I picked SQLite compared to other database engines such as NodeJS or MongoDB was that I already had a good understanding of SQL. SQLite is also a fully self-contained system meaning it is very simple to set up and get running quickly, and it also doesn’t take up much storage or processing power which are both key factors when dealing with a raspberry pi.

#### Matplotlib

Once I had the basic functionality of the programme down I started to work more with Matplotlib. This is a python library built upon NumPy for python, it allows the user to create all sorts of static, animated, and interactive graphs. There are a few other alternatives to matplotlib, but none based around python. I have used a charting programme based on JavaScript which I will move onto now.

#### HighCharts

HighCharts is very similar to the aforementioned Matplotlib, this time based around JavaScript. HighCharts has been used to take the accelerometer data from the Live system to the flask site. I’ve used the animation functionality of HighCharts to get a set of graphs which are atomically refreshing at a high rate to allow the judge to effectively see a live update of the accelerometer reading in a user-friendly form.

#### VNC viewer

VNC viewer has been a key productivity booster for this project. VNC viewer enables me to remotely access the pi meaning I don’t need to have a keyboard or mouse plugged in. This is a key problem as the slalom pole can often be a couple meters above the water. VNC is automatically installed on all raspberry pi meaning it is a no brainer to use. The other alternative that I could have used would have been SSH, this allows for command line execution on the pi. The only problem with this is there no graphical user interface as with VNC. This was a key downfall of SSH as I knew I would be needing to look at different graphs and different code, therefore SSH would lower my productivity as I would be needing to upload everything to git every time I made a change.

#### Hardware

The main hardware I have used during the project is a raspberry pi model B, two Adafruit ADXL345 accelerometers and one Raspberry pi camera. I decided this was this going to be the best solution after researching different microcontrollers. The main competition would have been using an Arduino. I decided to go with the raspberry pi as some of the other tools I have used weren’t available on the Arduino. Another benefit of the pi was that my supervisor Dr Ian Copper already had one spare which I could borrow for the project, allowing me to swiftly move onto the coding stage.

The Adafruit ADXL345 were used as they are readily available for a low cost, a pack of 4 only costing around £8. Although a pack of 4 was purchased, only two were needed. This was first because on one slalom gate there are two poles. Secondarily the amount of data coming from four accelerometers would have been too much to handle for one raspberry pi.

#### Ajax

Ajax has been one of the backbones for the Live system. Ajax has allowed me to retrieve data from different parts of the pi, asynchronously without having to update or refresh the page. This has been implemented in the retrieval of the sensor data, and also the retrieval of the camera live stream.

# Approach

## Fundamental system behaviour

I started off the project by defining a few basic requirements for the system, they are:

* The final product must use some form of sensor to track the movement of the pole
* Solution could use a camera to help the judges
* The solution must be presentable and easy to use for the judges
* There should be a web app as the final solution
* On the web app there should be a graph to show the changes of acceleration
* The graph could show rate of change
* The system could produce a judgement on if the pole has been hit or not

The first design decision I came to was using accelerometers as the main sensor in the solution. I came to this conclusion because after doing some research I found that the Adafruit ADXL\_345 sensor was cheap, small, and compatible with the raspberry pi, even coming with its only library. The documentation on the device was also excellent and very detailed, along with other people’s implementation of the accelerometer. All of these things combined made it a no-brainer to choose accelerometers and especially the ADXL\_345

Ease of use was a major underlying system behaviour that I defined. This is because I started off defining my target audience and users, this made it easier to clarify exactly what my final project will be. I found that my main users would be the judges at canoe slalom races, who are often just volunteers, who possible haven’t raced canoe slalom before, nor had any deep understanding of computers, raspberry pi’s, or the accelerometers I used. This meant that I had to focus on making the user experience as simple as possible.

A web app is essential for the final product to allow the product to be easy to set up for any user as well as making it cheaper. This is because 87% of people in UK own a smartphone[[5]](#footnote-5) meaning that there is a high chance all of the judges there have their own smartphone, meaning all the judge will need to do to access the system is just enter a web address. Because the system is then designed to run on a web app, I needed to make sure when developing it I make sure to take this into account.

## Live System

## Test System

## System Flow

# Implementation

## Live System

## Overview

## Live Data and charts

## Camera

## Test System

## Overview

## Database management

## Network Overview

# Results and evaluation

## Evaluation of Live system

## Analysis of results

## Conclusion of evaluation

## Discussion of Results

# Future work

# Conclusions

# Reflection on learning

# Appendix

## Code

## Glossary of Canoe Slalom terms

**Downstream –** The direction the water is flowing

**Upstream** – the opposite direction that the water is flowing

**River left –** The left hand side of the river, if you are looking at it downstream

**River right –** The right hand side of the river, if you are looking at it downstream

**Spray Deck –** the piece of neoprene that covers the cockpit of the kayak to stop the paddler falling out

# References

1. https://peakuk.com/index.php?route=blog/blog&blog\_id=283 [↑](#footnote-ref-1)
2. https://peakuk.com/index.php?route=blog/blog&blog\_id=340 [↑](#footnote-ref-2)
3. https://www.canoeicf.com/sites/default/files/tokyo\_2020\_olympic\_games\_canoe\_slalom\_results\_book\_v1-3\_og2020-\_csl\_b99\_csl.pdf [↑](#footnote-ref-3)
4. https://www.youtube.com/watch?v=symeJ4kKsCY [↑](#footnote-ref-4)
5. https://cybercrew.uk/blog/how-many-people-own-a-smartphone-in-the-uk/#:~:text=As%20of%20March%202020%2C%2087,the%20UK%20were%20smartphone%20owners. [↑](#footnote-ref-5)