A Low-Cost Approach to Autonomous Litter Collection

By James Heselden

Completed as part of a third year BSc in Computer Science

School of Computer Science

University of Lincoln

2019

Acknowledgements:

I wish to thank all of those around me who helped to keep me on track, focused, and offered help when I was stuck. I hope I have helped you all as much as you have helped me.

I would like to give special thanks to Jack Stevenson, your assistance went above and beyond with helping debug my initial mess of a system, your knowledge on areas I was unsure of helped a lot when it came to work in Ubuntu and Raspbian; and I am sure there is much more I am missing out.

Contents

[Part A: Introduction and Background 4](#_Toc5203904)

[Chapter 1: Introduction to Problem 4](#_Toc5203905)

[Chapter 2: Background Domain Research 4](#_Toc5203906)

[Part B: Methodology: 6](#_Toc5203907)

[Chapter 3: Project Management 6](#_Toc5203908)

[Chapter 4: Methodology Planning & Software Development Lifecycle 7](#_Toc5203909)

[Sub-System 1: Background and Foreground Extraction 8](#_Toc5203910)

[Sub-System 2: Object Identification and Litter Filter 8](#_Toc5203911)

[Sub-System 3: Localisation & Movement System 8](#_Toc5203912)

[Sub-System 4: Robot Development/Build 8](#_Toc5203913)

[Sub-System 5: Camera Setup 8](#_Toc5203914)

[Sub-System 6: System-wide Communication 10](#_Toc5203915)

[Part C: Conclusion 11](#_Toc5203916)

[Chapter 5: Evaluation through Metrics 11](#_Toc5203917)

[Chapter 6: Achieving the Aim 11](#_Toc5203918)

[Chapter 7: Changes to Development 11](#_Toc5203919)

[Part D: Reflective Analysis 11](#_Toc5203920)

[Chapter 8: WW and EBI 11](#_Toc5203921)

[Chapter 9: Further Research / Research Limitations 11](#_Toc5203922)

[Part E: References 11](#_Toc5203923)

# Part A: Introduction and Background

## Chapter 1: Introduction to Problem

The aim of the project is to design and develop an autonomous system to deploy a robot to pick up litter. The project is focused around the requirements of the system being low-cost, low-maintenance, and high-efficacy. To ensure these requirements are met, special emphasis is placed on them during the decision-making processes throughout the development.

|  |
| --- |
| Abstract—Path motion object detection based on video is a fundamental part of intelligent transportation systems, In the aspect of background extraction, this paper compared all existing theories and algorithms, aimed at specific objects (city expressways or high-speed Road), and combined with the virtual loop set method. This paper proposed an extraction and updating algorithm based on the sub-segmentations of invariant background, which greatly increased the time efficiency of the background extraction. It achieved great results of accuracy and real-time of this algorithm under background extraction. |
| With the application of autonomy invading all areas, I say why not anti-littering  What are the biggest hurdles preventing autonomy in different areas?   * Cost? Effectiveness? Maintenance/Complexity? Social/Legal/Ethical?   This project plans to investigate, design and develop an autonomous system to deploy a robot to identify litter within an independent live video feed. |

## Chapter 2: Background Domain Research

**ESSENTIALLY A LIT REVIEW:**

A comprehensive review of literature will provide background to the project, and give weight to the decisions you have made.

This section establishes what you intended to do and shows the reader that what you have done is the result of academic study.

System Structure:

The first thing I need to contextualise, is why I set up the system in an isolated way. So why did I?

*What system structure could I have? Middleware approach? Review literature, and find some which talk about the benefits of a middleware approach to robotics and larger scale systems of interconnected systems…*

*Do a small critical evaluation of different middleware systems… Talk about how ROS is a good tool for connecting complex parts of a system together. <https://www.hindawi.com/journals/jr/2012/959013/>*

*Talk about and critically evaluate some basic cost reduction techniques like using cheaper materials and parts like USB webcam, over integrated circuitry on the robot.*

System Components:

Then I need to talk about why I split the project into the sub systems I did.

The specific implementations of each of the sub systems are described below, so the details on them I don’t need to worry about here. But I do need to describe why I chose to make each of these systems isolated.

*List off the individual sub-systems along with the system diagram.*

*Describe each one in turn, and describe what it contains, and why this is considered independent.*

*Try to find some benefits online and in research as to why this system should be independent.*

# Part B: Methodology:

## Chapter 3: Project Management

During the initial conception of the project, a plan was put forward to lay out the time scales of each of the tasks, so as to get a better perspective of the project. The Gantt chart laid out 4-5 distinct sections of the project, basic image processing, basic robot, advanced image processing, and advanced robot. Each of these sections was given a defined milestone of which the section must be completed by, and smaller milestones which individual components must be completed by.

In actuality, the project deviated from this quite dramatically for a few reasons, the first was the time estimation for building the robot, where the building of the robot took significantly longer than expected due to lack of experience and overestimation of ability. For the implementation of advanced image processing, the aim was to develop a ML approach to identification, however when research was conducted, it was found that existing cloud-based systems could offer much more advanced functionality than could be made with the time and resources available, so this was implemented within a couple days, rather than the 6 weeks planned. The advanced robotics mapping was also removed from the project as for a proof of concept, this feature was far too complex to implement.

Throughout development, goals were set weekly to ensure the development continued smoothly, without much delay. The weekly goals were defined at the start of each week, as small achievable aims such as “Implement a mean and median background construction script & test automatic connection between camera and server”. Weekly goals were used to ensure priority lists were kept up to date for changes which occurred throughout the project, and they proved to be a helpful tool to the project, for instance, after the development of the robot stagnated and delayed the Gantt chart time estimations, the project worked solely off of the weekly aims. These being developed at the start of each week, ensured focus was being placed on the high-priority tasks which working solely off the Gantt chart did not allow.

The project initially was aimed as following a waterfall approach due to the structure of the system, and the impact of testing the system in an outdoors environment, however as the development continued and new understanding was found on the style and structure of the control system, the project became more of an adaptive waterfall approach, where each sub system in the project was developed under an independent adaptive waterfall methodology to ensure the systems were able to adapt to the growing demands. This was a very adaptive approach to the development of a system with this type of structure, as each individual sub-system was developed to a high quality without too much back-tracking on issues. The systems themselves were all quite small meaning that going back a level of the waterfall did not cause much issue, but together they combined to a strong project, which was well developed to meet the aims set out.

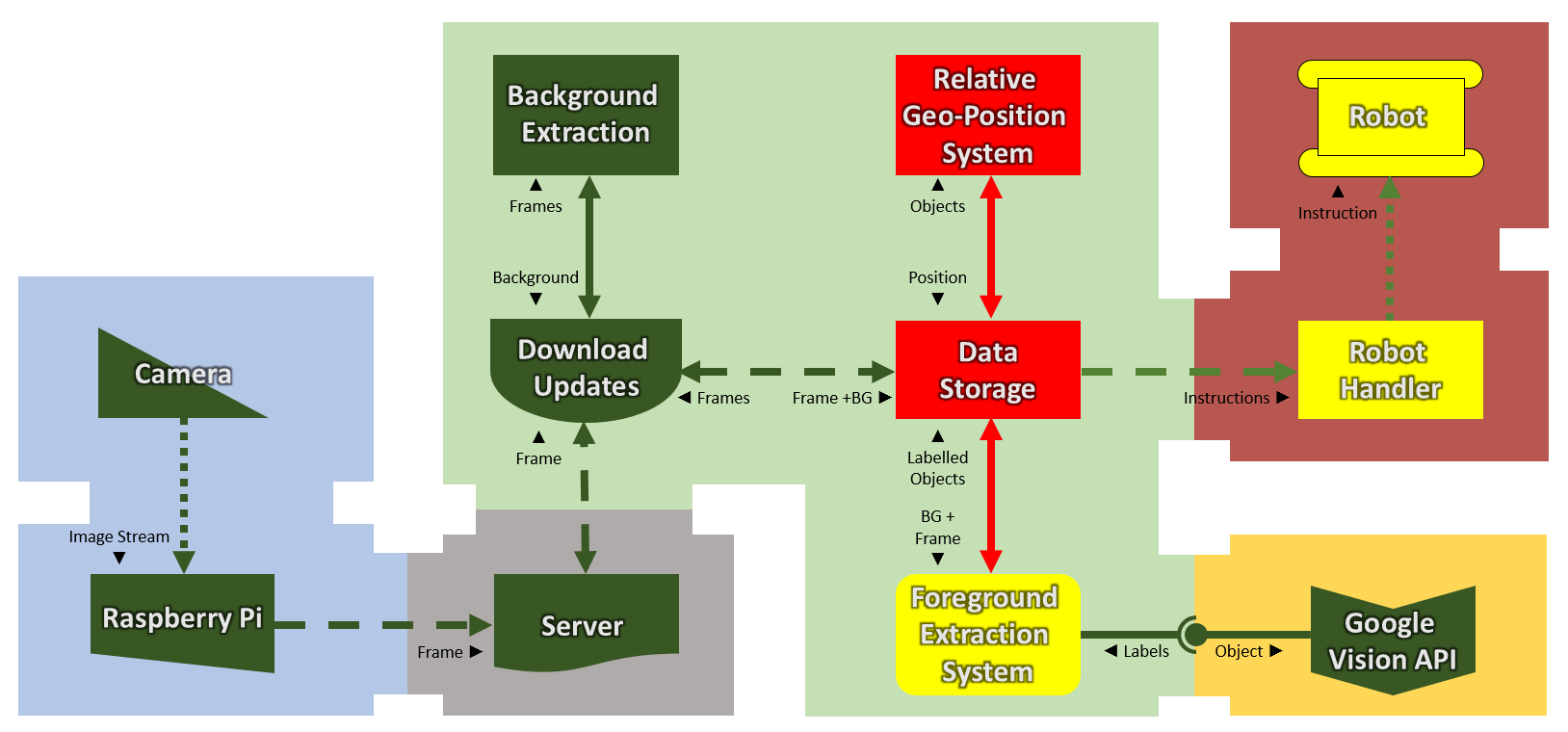
## Chapter 4: Methodology Planning & Software Development Lifecycle

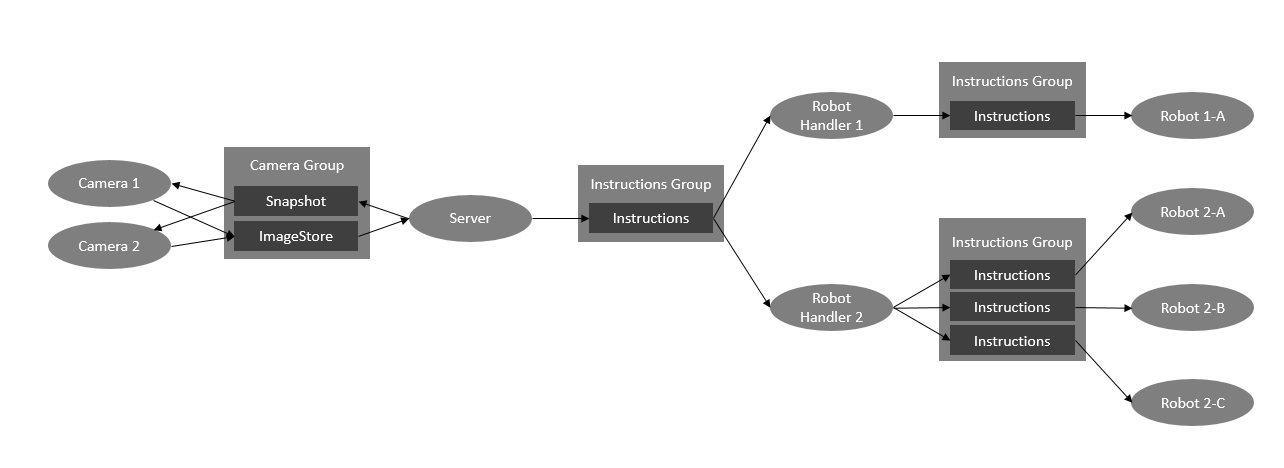
Following the adaptive waterfall approach to developing individual sub-systems, meant each sub-system went through requirements gathering, design, testing and evaluation. Which meant nearly the entire software development lifecycle was met during each stage of the development.

As described previously, the project is structured as a series of independent systems with an intercommunication structure set up to allow data passing, analysis and control. Each system was built independently with specific input and output structures.

The process in which each system was built is described below. For each sub-system, a review of best practices and methods was carried out, followed by an analysis of the best evaluation metric. This is followed by a basic implementation of the most appropriate systems, an evaluation of them and finally the implementation into the full project.

The connection of the sub-systems resembles the following diagram:





### Sub-System 1: Background and Foreground Extraction

**So, what do I want to talk about here… obviously I want to talk about the research I did on object extraction. What research did I do?**

I began the research with basic stacking in order to get the background from which the foreground would be extracted from with each subsequent image, I tried mean, median and mode stacking, but all gave different errors, in the end the best was mode.

Mode stacking on a relatively small timed cycle was found to be the most effective and allow the smallest impact of changes. Mean is impacted by random sparks, and when there is a lot of change, it cannot identify the background. While mode is not impacted by random changes. It takes what is most static in the frame-set, which should always represent the background.

The biggest issue with applying mode stacking is impulse valued noise occurring. Without by the least, an identification system for where these problems are occurring, this method is unusable in practice. Attempts to remove IVN mostly revolves around adaptive median filters [https://www.sciencedirect.com/science/article/pii/S1047320314001631], however while smoothing would work to reduce these regions, this came with its own problems, that being the distortion from the new images, which would likely cause errors with the extraction. The maths behind why mode filtering is so effective at detecting change stems from the lack of maths applied to the values directly, this ensures the comparison with a live image once the background is generated would be accurate, applying median filters would distort this clarity.

Despite its disadvantages, the method worked well, with extraction of foreground objects as shown in figure 1 after a stacking of 20 images, with a temporal spacing of 1 minute; note the noise around the middle of the generated background.



Figure . From top to bottom: (a) Mode stacked image; (b) Input image; (c) Foreground extracted

An adaptive algorithm was also developed to use entropy-based stacking similar to blur detection. In this, regions would be weighted based on their activity, with low active areas being discarded. This process was very resource intensive for the initial generation of background, however it meant the background would work regardless of the state of the environment, weather, and lighting conditions. The biggest disadvantage to this was the reduced complexity of the resulting image, where due to the bluring performed on the image, the sampling dropped significantly, and there was a strong difference between the background and the new input. **TALK ABOUT HOW TREES AFFECTED THE GENERATED BACKGROUND** [https://res.mdpi.com/applsci/applsci-08-00807/article\_deploy/applsci-08-00807.pdf?filename=&attachment=1]

**Tools:**

Considerations for toolsets for this is… MATLAB and PYTHON w/ Numpy, benefits of python is latency reduction, however MATLAB offers more off the shelf functionality, which for a prototype, meant testing could be completed in a significantly shorter time.

Entropy blurring was attempted but the implementation for such a system meant the quality of images were reduced making small objects harder to identify.

Stacking with edge detection was tested, however for it to work the quality of the camera needed to be improved, and the material patterns on the floor, greatly impacted the quality of the output.

Foreground Extraction is background subtraction. It is simply the new frame minus the background, where the value is 0 or near-0 then there is no change and should be disregarded.

|  |
| --- |
| RESEARCH POINTS: |

#### Step 1: Research into tools and methods to set up the sub-system

#### Step 2: Define appropriate evaluation systems

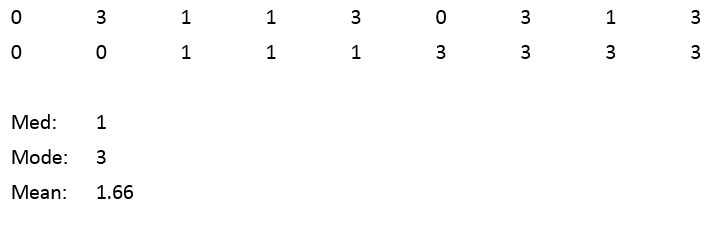
#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

250cm finds .5cm object reliably with mode stacking.

|  |
| --- |
|  |



#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

### Sub-System 2: Object Identification and Litter Filter

The first thing to do was to find out how we could identify litter in the first place once the object is extracted, so what characteristics could be used to differentiate objects, in order to be able to recognise them?

Some listed are, hue, intensity, saturation, glare, shape, sharpness. After some early consideration around the type of things being identified, as it is litter, the shape could change, consider a new packet of crisps and how that same packet would look screwed up into a ball; or the distortion of a crushed can of cola. These understandings meant that any sort of basic approach with regards to object recognition could not be done on shape, thus colour based visual recognition was the most appropriate to go for.

Additional methods were also tested using more complex systems like entropy analysis. The aim for this section of development was to gather data using values identified and test the effectiveness of a handful of clustering algorithm on the data such as Simple K-Means, KNN and Random Forests.

It was found before the clustering was applied to the data, that there was a more effective and time efficient approach to implement for the identification; the application of cloud-based APIs. The Google Vision API was found, through [<https://activewizards.com/blog/comparison-of-the-top-cloud-apis-for-computer-vision/>] to be the most effective, and so was deployed to the system with a simple JavaScript file which formatted and passed the message to the server.

There is a big disadvantage to using this setup and that is the requirement for the server to be connected to the internet, which adds potential leaks for security in an otherwise enclosed system, there is also an added cost element, where for testing purposes, the account being used is limited to so many requests per day for free, however as time goes on and further development to the system is completed, a more adaptive approach using a custom built ML system as planned may be more appropriate, perhaps using some sort of deep convolutional artificial neural network.

#### Step 1: Research into tools and methods to set up the sub-system

#### Step 2: Define appropriate evaluation systems

#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

### Sub-System 3: Localisation & Movement System

#### Step 1: Research into tools and methods to set up the sub-system

##### Localisation:

Localisation is on the harder spectrum of tasks when it comes to autonomous robotics, with it often requiring an expensive and highly calibrated tools such as laser scanners, Lidar or depth sensors. There has been a recent increase in localisation using cheaper alternatives such as **MONO CAMERA LOCALISATION BY <INSERT NAME HERE>,** using these types of setups allow the cost for building robots to go down, allowing for greater accessibility to the field. These robots often still require cameras which can cost over £100 like the Kinect.

Due to the nature of a small robot designed to pick up litter in an outdoors environment, the robot could be subjected to harsh and unclean conditions which could make a mounted camera unusable for effective localisation and planning. As such, the project has been designed away from including a mounted camera on the robot, and has instead chosen to adopt a style of localisation using external cameras. This method as described by **<** **J-AND-Y >**, **COULD BLAH BLAH BLAH**.

Removing the mounted camera, also leads to other benefits such as an easier way to detect humans approaching the robot, and easier maintenance on the robot itself, as the number of parts is decreased. By removing the camera, the communications with the robot become a one way interface, leading to less demand for the robot handler, the weight of the robot decreases, requiring a less intensive battery, and the impact of a robot being broken or stolen is lass impactful for the client.

The system is not without fault where the communications is concerned, as the robot requires a direct connection to its control hub receive any commands, the system has a larger latency, leading to the robot becoming less responsive to immediate change in the environment around it, however this is also countered by what is arguably the most important benefit which is the lack of a strong computation device mounted on the robot itself, as the robot only requires the ability to receive and process communications to send to the motors, an expensive, lightweight computer is not required.

The robot also becomes completely useless with respect to the environment outside the fixed camera’s visibility. The system also has fault with costs relative to the ratio of robots to coverable land, where having a fixed camera on a robot may be cheaper if there is only 1 robot patrolling a large facility compared with many fixed cameras to cover the entire traversable area.

As the project is focused around a proof of concept for the implementation, a more adaptive approach has been used for the robot control, with a simply system of 3 point alignment being implemented. In this localisation system, 3 points are identified, being the litter, the front and the end of the robot. The robot is spun till it is facing the litter at which point the three points are aligned, then the robot is moved forward until it reaches the litter.

#### Step 2: Define appropriate evaluation systems

#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

### Sub-System 4: Robot Development/Build

Decided to begin with the low-cost approach of using a basic toy off-the-shelf £12 RC car, and rewiring it. Breadboarding a motor driver chip to control it was not successful and broke the Pi, diodes, LEDs, and breadboard. So, decided to look for pre-built alternatives, many worked off of Arduino Uno, and this gave me an idea of prices. I bought a kit for £30, however it did not include the electronics or any assembly instructions, so it was returned. A motor control board was bought prebuilt for £10.

… PENDING RESULTS…

|  |
| --- |
| RESEARCH POINTS:  <https://www.amazon.com/gp/product/1457186039/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1457186039&linkCode=as2&tag=therobpod-20&linkId=QHNJA3OMPG5P7T4Z> |

#### Step 1: Research into tools and methods to set up the sub-system

#### Step 2: Define appropriate evaluation systems

#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

### Sub-System 5: Camera Setup

**Was it better to use a request acknowledgement or just a timed system? What transfer protocol would be best? FTP was tried but ran very slowly? Would the system be able to work on low-power hardware? How low-powered?**

##### Hardware:

The initial plan for the camera system was to make use of a wall mounted camera, as opposed to a robot mounted one. This decision stemmed around system control, where the system itself would use the robot to as a tool to complete its overall goal, rather than the robot using the camera to complete its own independent task. The server in this instance works as a hub to connect all other devices, be them cameras or robots.

Before considering the specific camera to use, some decisions had to be made, in terms of the type of connection to the server, the cost limitations, and the quality requirements.

For the quality requirements, two cameras were used and tested, with the aim to find if the lower cost camera (a £10 USB camera) was able to perform nearly equally to the more expensive (£30 4k action camera). To ensure the cameras worked for the environment, a few small tests were carried out with the quality of the output image, where a scenario was developed in which litter would be placed haphazardly in a region, and the cameras outputs would be tested in their quality when performing the stacking. It was found through these tests that a basic £12 USB webcam, had high enough quality to detect a piece of paper 50mm wide, from 2.5m away using mode stacking, so the 4k action camera was abandoned. The Pi cam was also reviewed however the module only offered 8MP, which was twice that of the USB webcam, however it was significantly more expensive, costing £24. The benefit of its resolution and WORD THAT MEANS SNUG are clear, however for the task required, much better then what the USB webcam offers is not quite needed.

The next stage was choosing a suitable connection from the camera to the server. For simplicity of a project on this scale, cost and time limitations, the most appropriate method would have been to set up the camera on the server computer itself, however a more scalable and technically challenging approach was taken, which involved connecting the camera to a piece of low-power hardware, in this case a raspberry pi, such as with the robot, and feeding the images through the local network to the server. The Raspberry Pi 3B+ costs £34; due to this expense, and the relatively low amount of processing on the board, testing was also carried out with running the setup on the lower power and cheaper board, the Raspberry Pi ZeroW+, which costs only £10. Testing was done on the effectiveness of the Pi ZeroW for the type and scale of file communications such as would be used in this system. This was **NOT SURE, NO DONE TESTING YET**

**::: INSERT IMAGE OF TESTING RESULTS :::**.

##### Software:

Was going to use Wired USB Camera but Changed to use (wireless)/Wi-Fi Camera as wired didn’t allow for having multiple cameras over a large area. This was simpler and easier to implement for cost reduction.

Changed to use Wired USB Camera connected to RPI as it was easier than connecting cam to internet or setting up own campus-wide subnet or adding own routers for each Wi-Fi cam.

#### Step 1: Research into tools and methods to set up the sub-system

#### Step 2: Define appropriate evaluation systems

#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

### Sub-System 6: System-wide Communication

ROS ROS ROS ROS ROS!!!

<http://wiki.ros.org/ROS/TCPROS>

#### Step 1: Research into tools and methods to set up the sub-system

#### Step 2: Define appropriate evaluation systems

#### Step 3: Define requirements for the sub-system

#### Step 4: Design the sub-system using the best method

#### Step 5: Test effectiveness of systems researched in context

#### Step 6: Build the most appropriate

#### Step 7: Evaluate the efficacy of the sub-system

# Part C: Conclusion

## Chapter 5: Evaluation through Metrics

## Chapter 6: Achieving the Aim

## Chapter 7: Changes to Development

# Part D: Reflective Analysis

## Chapter 8: WW and EBI

As described before, there were limits to what could be achieved considering the scale and complexity of the project along with the time available to achieve the aim. The underestimation on time for building the robot, along with the lack of knowledge in the area delayed the project significantly, this impacted the development…

## Chapter 9: Further Research / Research Limitations

# Part E: References