UNIVERSITY OF BIRMINGHAM

Interim Group Report

“Dancing Sweeping Brush”

Team A - MEng Group Project

2013-2014

1. Executive Summary

In this interim project report, the core components of a system to balance an inverted sweeping brush using feedback from an active vision system are researched, designed and developed. A vision system capable of determining the position and angular deviation of a brush from vertical has been implemented in C++ with functions from the OpenCV library. PID and LQR methods of control are explained and simulations evidenced. Communication protocols are discussed and full system integration plans proposed. A platform capable of moving smoothly in two orthogonal planes is planned. All subsystems have been implemented to some degree in one plane in order to learn from more simplistic objectives. A thorough plan for the expansion of these systems in the future is outlined.

2. Introduction

Have you ever attempted to balance a sweeping brush upside down in the palm of your hand? If your hand remains stationary, the brush is unstable – and will begin to fall. To keep the brush balanced, and vertically upright, you must move your hand in response to the fall. What you are doing here is using your eyes to detect the deviation of the brush from the vertical; estimating the adjustment required to correct this deviation in your head and applying feedback through the movement of your hand. As part of the third-year MEng group project, we are required to design, prototype and demonstrate an engineering system which tackles this same issue. Two cameras will be the systems eyes; a control algorithm will be the systems brains and a moving platform will do the job of a human hand. An extension to this task is to make the brush do small oscillations (dance), whilst still remaining balanced.

This interim report describes the key steps we have taken on the path to this end goal. We will outline key research sources in identifying potential methods and related challenges. This will lead on to our own experimental methods and calculations, which have influenced our design decisions and subsystem implementation so far. Since the task and process for this project are only loosely defined – we have been entirely responsible for organising our time, delegating tasks, finding and selecting suitable research sources and obtaining necessary laboratory equipment, software & analytical models. We have also had to determine exactly what the project is about and what would constitute a working system. Hopefully, this report will demonstrate a logical and professional approach to the problem so far.

There are number of challenges which have been clear from the outset. One of these is related to system integration. The total system constitutes a number of distinct parts, which are required to communicate with one another. It is critically important, therefore, that we define subsystem requirements properly and practise clear communication between team members.

3. Sub-System Breakdown

In this section we will be discussing the process we took in deciding how to split the project up into sub-systems. To begin, we thought of the entire system as a black box diagram (see Figure 1). By having no regard to the internals of the system or how it worked, we could focus solely on the inputs and required outputs to it. By using this trail of thought we decided that the only inputs to our system would be that of the two cameras.

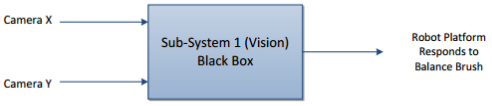


Figure 1: Black box of vision subsystem

As a result the system will read the data inputted by these cameras; after this has traversed through the system it will then respond to reset the inverted brush to a perpendicular position. Its output will therefore be the response of the platform. We called this the "black box idea", the ***vision sub-system*** as it concentrated solely on the input from the cameras and use this to acquire data about the current angle of the inverted brush and potentially the position of the cart.

The next stage in this process was to break down Figure 1 to include a second black box. To do this we wanted to think about how the robot would need to respond in order to correct the household brush back to its perpendicular position. For this we would needed to presume that the data from the cameras would be inputted to this sub-system and the output would be a set of clear directions that would tell a platform in what direction it should move, for how long and at what velocity to ensure that the brush remained back to its upright position. This led us to the idea for our second sub-system. We called this the ***control algorithm sub-system***. Its aim would be to take the data about the angle of the inverted brush from the cameras and to then issue these directions.

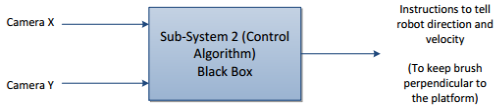


Figure 2: Black box of control algorithm subsystem

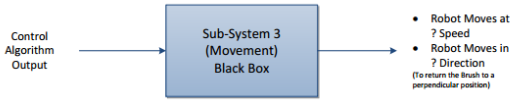


Figure 3: Black box of movement subsystem

The third stage in our system break down was to consider what would happen once we have processed the information about the position of the brush by the cameras in the vision sub-system and after the data had been converted into a set of useful instructions by the control algorithm sub-system. The next stage in our thought process was to consider how the robot could respond. This sub-system we labelled as the ***movement sub-system***. Its input would be the output from the control sub-system and would be a set of instructions that would be designed to tell the robot what it would need to do in order to balance the brush upright. As a result, its output would be to carry out these instructions.

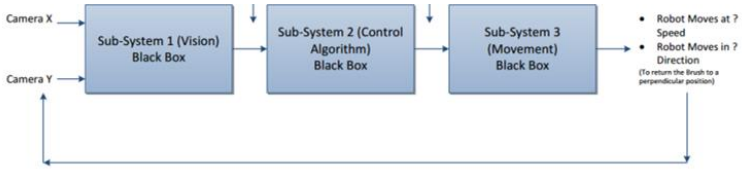


Figure 4: First view of system breakdown

This led us to the following set of sub-systems. Figure 4 shows the process that the system would follow in order to view that the brush is not at a perpendicular position to the platform, and hence to rebalance it. This process would then repeat.

However, we felt that this view of the system was not complete. At this point we considered the arrows in the drawing above. They would indicate the links between each sub-system. In the idea of the one between the vision sub-system and the control algorithm sub-system, we knew that this was in the form of a USB cable running from the camera. This left us with the link between the control algorithm sub-system and the movement sub-system.

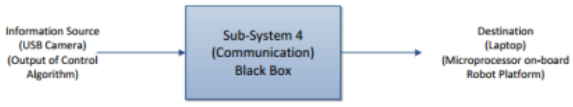


Figure 3: Black box of communication subsystem

and deliver them to the movement sub-system. This could be taking the directions about which motor to rotate clockwise at $2ms^{-1}$ and delivering this to the motor in question. The final result of our system breakdown is as follows:

As a result this led to a fourth sub-system being created: ***Communication Sub-system***. Its role is to take the values produced at the output of the control algorithm sub-system

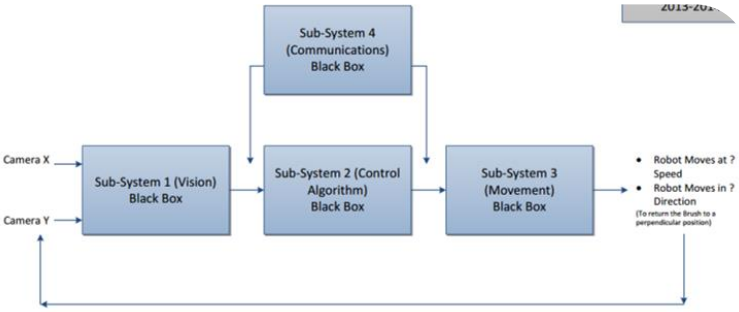


Figure 4: Complete breakdown of subsystems

4. Personal objectives of each team member for the project

The Personal Objectives section illustrates the goals that each team member would want to achieve at the end of this project. The table below shows the members and their respective expected outcomes from the project.

Team Member	Objectives
Matthew Cottrell (Team Leader)	Wants to develop skills in project management to further knowledge of Project Engineering gained from employment experience and from EE2G1
Raymund Lagua (Treasurer)	Wants to focus on the vision sub-system and aims to implement this
Faisal Ahmed	Wants to apply his study of control systems and aims to learn about how this can be integrated into the system
Jonas Chan Leong Sean	Wants to apply himself to research control systems and control algorithms to drive a machine
Adesegun Adepegba (Scribe)	Wants to be able to successfully design and build a wireless communication channel
Yew Jun Hoong	Wants to improve skills in programming. In particular wants to apply himself to the construction of the platform and the control of motors
Lee Kok Li	Wants to be able to successfully design and build a platform
Thomas McKay-Smith	Wants to investigate a topic of engineering that he is not currently exposed to

5. Team Structure

The aim of section 5, was to find where each team member's strengths lie. This meant that we were able to arrange ourselves into sub-systems where we would have a chance to achieve our personal objectives. As a result, Figure 7 illustrates the sub-system that each team member will be working in during this project.

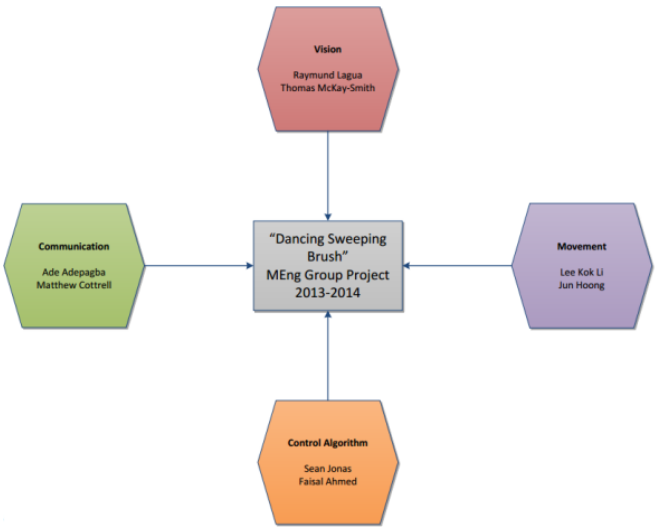


Figure 5: Team A Structure

6. Analytical Approach to Problem

Before we considered what we wanted to build and how to design this, we first had to consider the approach we were going to take for this project. We had to consider whether we were going to take a practical approach or whether to follow an analytical approach. Both approaches had advantages and disadvantages that would affect the way that we structured the work that we carried out as a team and our ability to reach decisions.

We felt that an analytical approach consisted of considering the theory of the problem and then to make decisions based upon this. As a result, the positive aspect of this is that we would understand the working behind each item we use and should understand how each piece of the puzzle will ultimately fit together. Thus less time would be spent integrating sub-systems. However, producing the theory would take up valuable time, which could delay the build of the project.

We felt that a practical approach would be the opposite of this and would consist of purchasing hardware and tuning this until a working project was established. The advantage of this; is that with a considerable budget, the project would start to be built quite early on. However we felt that due to a lack of analysis of how each piece of hardware will fit together, this could result in a more significant amount of time spent trying to integrate all sub-systems together. An example of a practical approach would be rapid prototyping. This is where items are purchased and then replaced if they are not fit for purpose. We felt that this would indicate poor budget management as additional items would need to be purchased until a working system was achieved.

To conclude, we opted to carry out an analytical approach to this project. This meant that during semester 1, we have created technical documents based upon the theory, physics and mathematics that we felt was required to make crucial decisions in our project. In this report we will look into the design choices that we have already made and justify our reasons behind them.

6.1 Vision

Introduction

In this part of the report, we aim to outline a number of secondary research sources which have influenced

our approach into a *"real time active vision system"*. These sources have helped us identify important issues we will need to address and given us insight into potential hardware, software and computation choices.

The real time active vision system must:

- Be based on simple and cheap USB HD web cameras
- Use 2 orthogonal planar views of the brush to detect angular deviations from the vertical in the 2 view planes
- Not need extensive 'tuning' to specific foreground or background objects

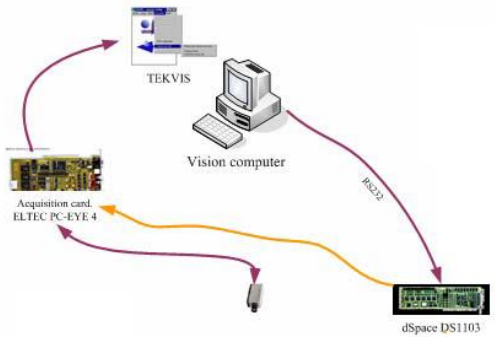


Figure 6: TEKVIS Vision Integration system

The source (Haoping Wang, 2008) is similar to ours, in the sense that their objective is to stabilise an inverted pendulum in both X and Y directions. Vision-wise, they have solved the problem by using the following hardware: an *ELTEC PC-EYE 4 image acquisition card, an IR CCD (Jai M50 IR) camera and a dSpace DS1103 card*. The ELTEC PC-EYE was used to obtain a signal from the camera and transfer it to the computer to be used by the software. They used a low cost CCD camera as opposed to hi-tech digital cameras which could provide higher sampling rates, spatial resolution and improved signal/noise ratio. By using a dSpace card they were able to generate a periodical pulse signal that triggered the camera within a period similar to that of the sampling rate. The software that they have used to monitor and detect two points is *TEKVIS*. In terms of video sampling they have used a rate of 25 frames/sec with a resolution of 640 x 480 pixels in non-interlaced mode. They used a sampling rate of T=40ns. This includes the acquisition, processing and transferring times.

The method that they used to calculate the angular deviations was to take two reference points, represented by a lit LED. One LED was placed on the upper tip of the pendulum (x_a, y_a) and the other on the pendulum's pivot (x_b, y_b). These were then used to represent the coordinates of the two points, relative to the image stream from the camera. From the given coordinates, they computed the angular deviation of the inverted pendulum by using the following formula:

$$\theta = tan^{-1}\left[\frac{(y_a - y_b)}{(x_a - x_b)}\right]$$

In our opinion, the method that they have used is unsophisticated, however it does what is required. They have used kept their budget spend low by using a cheap CCD camera. However, we are not persuaded by the need of an image acquisition card. Apart from the fact that it will cost more, I believe that this can be omitted by choosing a USB camera and compatible software and computer.

The objective of this particular project (Kentaro Hirata, 2007) is to control and stabilise an inverted pendulum through visual servoing. Unlike our task, the pendulum in this project is restricted to move only in one direction. They have used a simple Logicool Qcam Pro 4000 USB camera and an image processing PC to generate their visual feedback system. The web camera that they have used provides a maximum video

Figure 7: Image Processing using AD/DA

resolution of 640 x 480 pixels and comes with USB 1.1 connectivity. The image processing PC obtained a Pentium 4 2.8 GHz CPU and a memory of 1GB.

In terms of software used in the project, they utilised Microsoft's DirectShow which is a software library used to handle multimedia data. They targeted a sampling rate of 30Hz which required the camera to capture 30 frames/sec. The chosen video resolution was QVGA (320 x 240 pixels). They encountered a problem with sampling at this rate as the data rate was too much for the USB to handle. They required 320 x 240 x 8 (coloured) x 30 (fps) $\simeq$ 55.29 Mbps whilst USB 1.1 could only handle a theoretical maximum data rate of 12 Mbps. To solve this they looked at cameras with USB 2.0 connectivity which allowed a theoretical maximum speed of 480 Mbps.

As mentioned, they used DirectShow to sense the two points, represented by lit up LED's. And as similar to the aforementioned project, they had also calculated the angle of deviation by calculating the arctangent of the difference between the two point's X and Y coordinates which are relative to the screen.

In our opinion, we believed that this was a very good example of a visual system that didn't require much hardware. The camera was connected directly to the image processing PC which was able to process the data at the frequency that was required.

We chose this particular project (Kazi Mahmud Hasan, 2012) as it focused on a different method of image processing to that of the ones mentioned above. It is similar in terms of hardware used as the image acquisition device is connected directly to the image processing device without any data acquisition boards in between. Although we do not know which specific camera or PC was used, we know that it is a USB web camera, similar to what we need to use.

The image processing software used in this project was MATLAB, which contains its own Image Acquisition toolbox and provides support for the problem of object detection and tracking. The first step taken to perform image processing was to initialise the image acquisition. The container for the image stream was created using 3D matrixes. The next step was to detect the specific colours. To achieve this, a method was used to detect and isolate the colour of an object. Image for the webcam was of the Red Green Blue (RGB) colour space. They separated these three colours from the original 3D matrix which resulted in the formation of three different matrices. This is followed by the calculation of the different intensities of the three colours in the RGB colour space. They have used the variables to determine the minimum and the maximum values of red, green and blue in the object that was tracked. The following equation summarises how they isolated a specific colour from an image: $X = (R \geq red.\min \& R \leq red.max) \& (R \geq green.\min \& R \leq green.max)$

$$\& (R \geq blue.\min \& R \leq blue.max)$$

In this case, X will represent a binary image where the presence of colour will be represented by a '1' and the absence of colour by '0'. When red was the colour set to be detected, any object that is red will appear white and any other colour will appear as black. This allows the detection of the object.

Filtering the noise is also important as this will remove small areas of the same colour that are irrelevant to tracking. MATLAB contains several functions that will perform the filtering. Examples of the most commonly used functions are *imerode()*, *imdialate()*, *imclose()*, *imopen()* and *imfill()*. The first filter that was used was *imclose()*. This function closes, fills and smoothens the gaps between two objects. This was used to ensure that the shape (circle) that was tracked was as clear as can be.

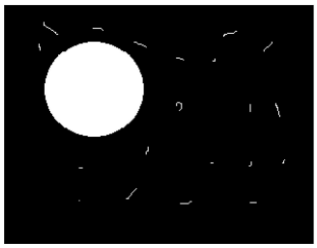


Figure 8: Visible noise on an isolated colour of an object

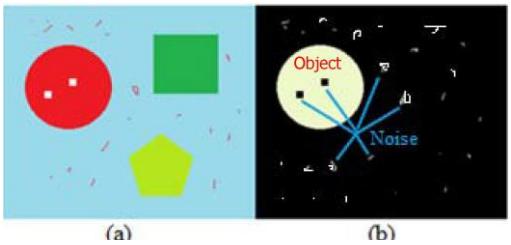


Figure 8: Detect and isolate colour of an object

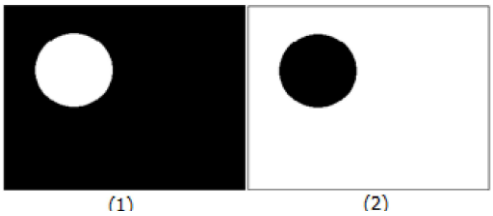


Figure 10: Result of filtering

The next function, *imfill()*, fills any black areas within the object and to complete the filtering of noise, the *imopen()* function was used to get rid of any other scattered noises in the image. Figure 10 presents the result of filtering.

To calculate the centre of the object, they first had to find out the first and the last column in the matrix that contained a '1' and also similar for the row. From this they were able to calculate the centre of the circle in regards to the X axis, and also the centre of the circle in regards to the Y axis. By using this they were able to calculate the mid-point of the object.

In my opinion I think that this method is feasible and we will definitely be able to implement this into our system. From this method we could create two unique coloured points that will be tracked, calculate the centre point of those points and from that we will be able to calculate the angle using the methods used in the aforementioned projects. This was demonstrated during the 1st demonstration and presentation. Upon conducting further research on different methods of visual tracking, we discovered that there are various approaches to creating a consistent and efficient vision system.

(See appendix 7 for further justification of background subtraction)

6.2 Control

In this part of the report, we aim to outline a number of secondary research sources which have influenced our approach for our *"control system"*. As for vision, we first started by considering the following specifications that aided us when searching for similar projects. The control system must:

- Be wireless or wired
- Be run on general purpose hardware
- Use traditional or 'fuzzy' control techniques

Analysing Available Choices and Alternatives

Controllers

Proportional Integral Derivative (PID) Controllers

From this source (Sharifi, 2010), we understand that the basic idea of the PID Controller is to examine the signals from the sensors placed in the system and then to feed errors back to the input in order to readjust the behaviour of the system. A typical unity feedback system was considered in this source. This can be seen in Figure 11 below:

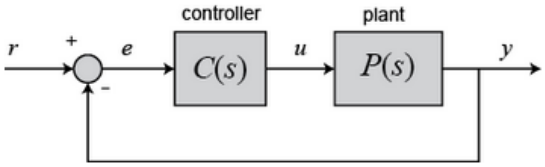


Figure 9: Schematic diagram of a typical unity feedback system

To do this we would need to determine the PID values of the controller. PID stands for Proportional gain (K_P), Integral gain (K_I), and Differential gain (K_D). The gains of a controller possess several characteristics on the rise time, overshoot, settling time and the steady state error of the system. This source (Sharifi, 2010) shows how the obtained PID values can then be fed into the PID controllers to stabilize the system.

Linear Quadratic Regulator (LQR) (Lenka, 2011)

The process of the linear quadratic regulator is somewhat similar to the PID Controller except that instead of finding values for PID gains, the LQR method requires determining the feedback gains of the controller using state space modelling. The gains collected are then fed back into the state space of the system as shown in Figure 12:

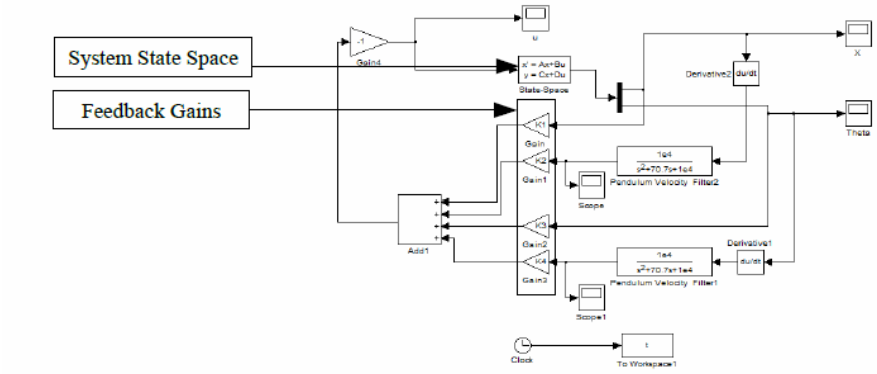


Figure 10: Linear Quadratic Regulator using SIMULINK

System Modelling and Simulation

System modelling is important, as the values obtained from the controllers are first tested out in the modelling and simulation software using real parameters of the system such as the weight of the cart, weight of the broom, moment of inertia of the broom and so on. Based on our research, most of the modelling process is done using MATLAB. We found that this is due to the fact that MATLAB allows the user to build and test controllers to model the inverted pendulum. Further to this optimization can be reached before implementation on the actual pendulum equipment. This would allow for faster development and the potential opportunity to investigate different methods of control, so as create the best possible control algorithm for our system. We found that modelling is mostly done using Simulink, an extension of the MATLAB software. (Lenka, 2011)

Programming Languages

Programming purely in C language allows the use of microcontroller devices such as the Peripheral Interface Controllers (PIC) and Arduino boards. Unlike the PIC and the Arduino, LabVIEW is a piece of system design

software which allows the user to incorporate the C language, comprehensive toolkits and control applications and embed this into the system itself.

Choosing the Most Appropriate Choice and Alternatives

The control system team made a decision on the most suitable and appropriate choice for each operation. Regarding the controller, it was decided that working on both the PID Controller and the LQR Controller would be beneficial. This is due to the fact that although it would be easier to find the necessary values for the PID controller, the sources that we have considered have shown that the LQR method of control could give better performance. To conclude, the team have already had some experience at using MATLAB to control an inverted pendulum in EE2C2 (Bryds, 2013), the control team aim to look at modelling the virtual inverted pendulum using real parameters. The aim of this will be to model the system behaviour. The team has also decided to carry out further research into using LabVIEW. The reason for this is that it presented us with the option of reducing the amount of programs that will need to be built and hence apply more time towards the understanding of the concept of the inverted pendulum.

6.3 Movement

In this part of the report, we aim to outline a number of secondary research sources which have influenced our approach for our "The balancing platform". As for vision and control, we first started by considering the following specifications that aided us when searching for similar projects. The balancing platform must:

- Be robust enough to hold a standard household sweeping brush
- Be controllable and steerable in the X and Y directions on a flat planar surface

The aim of this section is to focus on the applications of similar projects that have been carried out and to look at the movement methods that they have opted for. From this we aimed to obtain possible approaches that have been successful. To successfully move a robot, we needed to consider methods of movement as well as how we could possibly drive this method of movement.

Source 1 (Conradt et al., 2009)

This project is similar to ours as it required the inverted pendulum to be balanced in both X and Y directions. The balancing hardware consisted of two parallel linear rollers that act as the platform for the cart to balance the pendulum. Linear rollers were used to provide minimum friction for both directions whereas two independent servos were then used to control the movement of the cart. They have considered a Futaba BLS451 which is a brushless DC motor servo. This appeared to be because of it having sufficient torque and a fast response time to move the cart according to their control system in order to balance the pendulum. Each servo was then programmed so that it would start in its centre position. The reason for this is so as to provide the maximum possible movement when the balancing software starts to send PWM signals. In our opinion, the movement methods used in this source would be suitable for a pendulum that is relatively small, thus only needing the platform to move by a small amount to keep it balanced. This is because of servos having a limited rotation. This would mean that a servo with a large wheel/gear would be needed to balance a pendulum with a greater length, due to a larger movement being required. We found that a larger servo would be more expensive to obtain and therefore it is not a wise choice for our project.

Source 2 (Bjarenstam & Lennartsson, 2012)

This project aimed to control a robot on a ball, using omnidirectional wheels (see appendix 4 [Omniwheel explanation]). The difference between this project and ours is that instead of balancing a pendulum on a robot, this project looked at balancing the robot itself. The concept itself is almost the same as movement is required to maintain balance. Based on that, the omnidirectional wheel can change the direction of movement by sliding rather than steering. Therefore, a robot with Omni direction can reach a specific location without rotating. In this source, three Omni wheels were placed at 120° to each other; similar to a triangle, so that the robot could be balanced. Three 125mm double plated Omni wheels; manufactured by Rotacaster, were used. The reason being is that the double plated construction offered a smoother run compared to single as can be found in the report. Each wheel was then attached to a motor. In our opinion, omnidirectional wheels appeared to be a suitable option for method of movement needed for the movement sub-system. A three wheel configuration was used in this source because it is placed on a ball thus three wheels will provide suitable support. Whereas a four wheel configuration would appear to be more efficient on a flat surface as it would allow our system to drive with two wheels in any direction compared to three wheel configuration that would only move with one.

Verdict:

In conclusion, we felt at this stage we could rule out the use of servo motors due to them having limited rotation. In section (8.3.1) we will discuss the testing that was carried out on a unipolar stepper motor to consider its suitability for our system.

6.4 Communication

Unlike the other three sub-systems, *"Communication"* only had one specification:

- Be wireless or wired

The aim of the section will be to review appropriate literature available to help us compile a list of the possible communication methods that we have available to us for this project. By the end of this section we will have reached a decision on the communication method(s) that we wish to conduct further research into (these options will then be discussed in section 8.4.

6.4.1 Wired Communication Methods

RS232

This source (Holzapfel, May 1998) discusses the results of a similar project to ours using RS232 as one of the communication methods. This project appeared to be similar in sub-system structure to our project as well by separating out into vision, mechanical, control and actuator. This project uses a video camera to monitor the angle of an inverted pendulum that is positioned on a cart. If the angle of the pendulum is considered unstable then a DC motor rotates a ball screw. This movement displaces the cart, thus correcting the inverted pendulum. The camera is connected to a vision system, which in turn communicates to a personal computer via an RS232 serial communication link. This study merits the use of RS232 by stating that no image information regarding the position of the sled and the angle of the inverted pendulum would be lost when transmitting data over an RS232 serial communication channel. The successful application of RS232 in this project amplifies its qualities to us and means that we are more likely to consider it for our Dancing Sweeping Brush.

RS422 and RS485

The source (Lei Shi, RS485/422 solution in Embedded Access, 2009) discusses using RS422/RS485 as a serial bus in embedded systems. From this source it is clear that the main advantage of RS422/485 is that it is used for medium and long-distance data communication. This is supported within this source by the statement "RS232 is unsuitable for medium-long-distance data transmission, whereas RS485/422 can make up for those shortcomings". The source (Jia Hui-juan, 2010) considers the technology of RS485 over Ethernet. It states that "RS485 is a standard based on RS422, RS232 set by EIA." From this source and (Lei Shi, RS485/422 solution in Embedded Access, 2009) it seems that RS485 is an upgrade of both RS232 and RS485. From these two sources it seems clear that RS422 and RS485 are improvements over RS232 serial communication. As neither of these sources are from projects that are related to ours; Dancing Sweeping Brush, it would seem that they do not carry much weight. As a result we feel that they would not be appropriate methods for communication for our project as we feel that medium/long distance communication will not be necessary.

USB

This source (Bruce, Communication Part 1: Communication between Computers, 2009) discusses the use of a serial to USB communication channel. In this source Bruce focuses on the application of a USB communication channel within an Arduino Uno to Computer connection. She states "The USB chip on your Arduino, presents itself to the operating system of your desktop computer as a serial port and sends data through the USB connection at the rate you choose (e.g., 9600 bits per second). The desktop's USB controller only passes you the bytes you need" This is a significant point as USB ports are readily available on almost all computers. This means that we would not have to purchase any specific adaptors or cables in order to communicate between two devices. The result of this is that we would not have to use a specific computer or purchase a specific adaptor, thus making our communication method transferrable if we had to change the computer that we wished to use.

6.4.2 Wireless Communication

Bluetooth (802.15.1WPAN/Bluetooth)

The source (Hans-Dieter Schutte, 2004) concerns the RS422 navigation sensor networks for integrated bridges on vessels and replacing them with Bluetooth communication. This project aimed to look into the feasibility of this replacement and considered that "A major application for Bluetooth communication is cable replacement" and by concluding with "Bluetooth offers sufficient transmission capacity, robust communication, and adequate coverage, even for a large bridge of up to 50 m". This project gave us confidence in Bluetooth communication and that it could be used comfortably as a suitable wireless replacement for a RS422 Wired Communication channel and hence for the RS232 option discussed in section (6.4.1) up to a range of 50m.

802.15.4/ZigBee

The IEEE 802.15.4 and ZigBee are two methods for wireless communication, which have many similarities, but due to key differences between the two, we are going to consider the two as separate options in regards to wireless communication.

The IEEE 802.15.4 is a communication protocol developed by the IEEE for "low data rate, simple connectivity and battery applications in mind", whilst the ZigBee standard is "a protocol that uses the 802.15.4 standard as

a baseline and adds additional routing and networking functionality". (Digi, Demystifying 802.15.4 and ZigBee, 2008)

This means the ZigBee standard is built on 802.15.4, hence the similarities; however, we are considering the 802.15.4 as it is more applicable for our project compared to ZigBee. This is because we require the use of wireless communication for simple data communication between PC and a microcontroller, meaning the ZigBee standard's additional mesh networking features are not required for this project.

Digi produces XBee wireless 802.15.4(series 1) and ZigBee (series 2) modules, which both have a signal frequency of 2.4GHz and RF data rates of 250Kbps. Maximum range for the 802.15.4 module is of 90m, which is fairly smaller than the ZigBee module's 120m maximum range, and as we are most likely going to build a robot that is going to move around in a room, the extra-large range of the ZigBee module will be wasted. Compared to Bluetooth, the 802.15.4 and ZigBee module have a great range for communication, but this comes at a cost in regards to data rate. (Digi, XBee wireless RF modules, 2013)

Verdict: Wired Communication vs Wireless Communication

To conclude, we have considered the applications of a range of wired and wireless communication options. At this stage; from the sources of literature reviewed, we were able to rule out several possible options. The remaining options we considered in greater detail, these are RS232, due to its familiarity and that the sources researched, were concerning a project most similar to that of which we are undertaking. As a result this appeared to be our favourite choice for wired communication. Aside from this, 802.15.4 also had some attractive selling points that prevented us from ruling out wireless communication.

7. Experimental Verification of analytical material

Before we considered designing or building any aspect of our system, we first decided to focus on the brush we wanted to use for our project and review its behaviour. The aim of this was to help us decide upon key parameters that would help us model the physics of the brush and could be used in our control algorithm. The items that we considered were:

- To measure the total mass of the brush that we plan to use in our project
- To measure the time it takes for the brush to fall from an upright position through 90°
- To consider how the system would need to move in order to balance an inverted brush.
- To set a value for the mass of the cart
- To determine whether the distance of the brush tilting away from vertical 90 degrees could be balanced back with the same distance by applying a parallel force.

7.1 Experiments & Results

7.1.1 Mass of the brush

We started by measuring the length (from end of handle to end of head) of our wooden brush and finding it to be 1306mm, with a mass of 0.7kg. The mass was measured using a set of luggage scales

Figure 11: Measuring of wooden brush

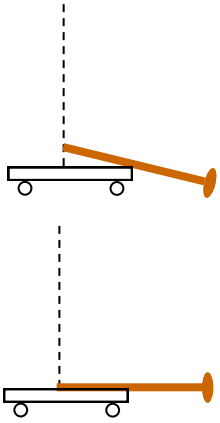
shown in Figure 13. Whilst the dimensions were measured using a tape measure.

The measurement of the mass of the brush is only up to one decimal point due to the precision of the luggage scale, thus the error being ± 0.05kg. Likewise, the measurement of the dimensions of the brush has an error of ± 0.5mm due to the smallest increment being 1mm.

7.1.2 Stability of the brush

The second experiment we undertook was to consider the range of response times that we would need to make our system react within. To do this, we considered the minimum possible time it takes for the broom to fall through its maximum possible angle T_{max} (from upright through 90°). This took 1.4 seconds. We measured this process using a pair of stopwatches; for redundancy. Upon this consideration, we had a benchmark that we could use to help us decide whether an item of hardware would take too long to respond, i.e. the maximum possible response time.

This is significant as it means that our system will need to detect the angle between the brush and the perpendicular axis, process this data through the control algorithm and communicate this to the movement subsystem and then actually move within 1.4 Seconds in order to correct the brush whilst it is falling. A response time greater than 1.4 Seconds could result in one of two situations:



- The brush falling off the platform: This would mean that we would be unable to correct the angle of the brush.
- The brush will lie flat on the robot platform. This will occur if the distance between the pivot point and the brush's centre of gravity is less than or equal to the radius of the platform.

By looking at the sub-systems involved in making the entire system we can see where possible delays in response time can occur:

Vision: The quality of image that both cameras will be taking and how quickly this data can be sent to the control algorithm. (Denoted by T_V – Time taken to transfer data from the cameras)

Control Algorithm: The Processing capabilities of the computer/microprocessor that will contain the control algorithm to extract useful data from the data sent by each camera. (Denoted by T_P – Processing time of computer/microprocessor)

Communication: The time it takes to send the data outputted from the control algorithm to the movement sub-system (motors). (Denoted by T_C – Time needed to send data from destinations to sources)

Movement: The Response time of the circuit/ how acceleration generated by the motors that we wish to use. (Denoted by T_M – Response time of any motors that are used to move the cart in a way that would correct the brush)

Based on this we can say that:

Total Response time of system $(T_{Total}) = T_V + T_P + T_C + T_M$

As well as this

The Total Response time of our system must be less than the Maximum Response Time

$$T_{Total} < T_{max} \text{ OR } T_V + T_P + T_C + T_M < 1.4 \text{ Seconds}$$

7.1.3 Mass of the cart

At this point in the project we had not considered the potential mass of the cart. This was decided to be an important consideration and would be needed to simulate the response of the control system. We decided to set this at 1 kg. As mass of the cart is an important value for the control system based on F(force) = Ma(mass x acceleration). An estimated value will never be a substitute for a measured value, but this value could only be found once the moving platform had been constructed. If we had not decided to estimate this value, then perhaps we would not have made as much progress in semester 1, or we would have constructed a platform that would have not been thought out.

7.1.4 Experiment with a movable chair

For this experiment, we rethought the idea of balancing an inverted brush on your hand and took this one step further by attempting to stabilise a plastic brush on a movable chair. This was used to help us determine whether the brush could be re-stabilised by applying a force towards the direction of the tilted brush with the same distance. Although we were not successful in balancing a brush for very long using this method, we could relate the theory behind it and were able to consider the range of movement of the moving platform and forces that would need to be applied to our system to balance the brush.

7.1.5 The Physics of our System

Our next step was for us to consider the physics of the system. This document aimed to take the experimental results and to physically model the movement of the brush. The aim of this was to help us understand the forces that would be acting on the brush in two dimensions. By looking at only two dimensions, we simplified the problem for one plane. Although on first inspection, we planned this document to coincide with a two dimensional prototype that would only allow the brush to fall in one plane and to be viewed with only one camera, we realised that this view would not need to be changed when a second camera is added. The reason for this is that both cameras will be positioned at a right angle to each other, thus will measure the angle of the brush as a mixture of x and y. but essentially both cameras will see the same two dimensional view of the brush. The application of this is that we were able to consider the effect that the control algorithm will need to have to balance the inverted brush, as well as to consider the vision system it.

8. System Design and Early Stages of Implementation

We have outlined the sources of literature and primary research which has influenced our early design decisions and methodology. In this section, we will discuss the four subsystem designs we have arrived at as a result of the previous sections. We will also explain the first implementation stages of these – including problems encountered and methods to overcome these.

8.1 Computer Vision

This subsystem requires both hardware and software elements. Hardware is required to capture live video of the inverted brush and for running the computer vision software.

8.1.1 Hardware

The vision system requires two types of hardware:
- Computer for processing (PC/Microcomputer/PIC)
- Cameras (2 HD USB webcams)

Firstly, we have chosen to use a standard computer with Windows OS to run our program code. One reason for this is that we already have access to university computers and so this option requires no additional spending from our budget. The main reason, though, is that this is the simplest way to get this subsystem up and running. We have experience with Microsoft Visual Studio, and with installing 3rd party libraries on a Windows computer. Additionally, we have avoided any issues that could occur from using lower speed processors. Having said this, we have not ruled out a switch to a microcomputer or microprocessor further on in the project. We have retained this flexibility, by choosing to program with the C++ language. It would be desirable for us to switch the processing to a lower cost and smaller device – but this is not a mandatory requirement and so, initially, we have chosen to prioritise a high-speed working system that can be developed fast.

Our second hardware decision was the choice of cameras. The cameras must be cheap HD USB webcams, so our options were fairly constrained to begin with. The two cameras are inputs to the vision subsystem and also the very first inputs to the entire system. A poor camera choice could cause knock on effects. In choosing cameras, we considered:

Price – We were given the subjective requirement of needing to use 'cheap' cameras. With this in mind we want to keep the price as low as possible while not sacrificing any necessary performance.

Image Quality – A high-resolution video means that there are more pixels that allow for more accurate detection, classification and identification of objects.

Data Transferred per Frame - This determines the amount of memory each frame uses. We need to be careful with this as we need to make sure that we are able to transfer this data without any delays.

Frames per second (FPS) - The higher the FPS, the smoother the video stream. By having a high FPS, we can analyse new data more frequently.

Field of View – The cameras will be static. The field of view must be large enough to see the cart and broom without distortion when its horizontal displacement, from the starting position, is maximum.

Cable - We need to consider the placement of cameras in the room, the cables must reach the PC. We can also use USB extension cables.

Connection - We must take into consideration the method of transfer and the speed of data transfers between the webcam and the PC.

System Requirement - The PC must be able to run the webcams and their software smoothly.

See Table 1 in Appendix 1

8.1.2 Calculating the size of each video frame

This will depend on the resolution of the image and the number of colour channels it uses (typically 3 for colour, 1 for grayscale).

One video frame of an **uncompressed** coloured video with a resolution of *640*480* contains *307,200* pixels. Each pixel requires 3 bytes of memory therefore, ~ *0.92* Megabytes of memory is transferred to the PC for each video frame.

If we are to going to configure the camera to take 30 FPS, one second of the video would use ~ *27.6* Megabytes.

The following are the other resolutions that we will consider.

720p HD uncompressed (compressed): 2.76 MB * 30 = 82.94 MB/s (7.39 MB/s)

1080p HD uncompressed (compressed): 6.22 * 30 = 186.62 MB/s (12.94 MB/s)

8.1.3 USB Transfer Speeds

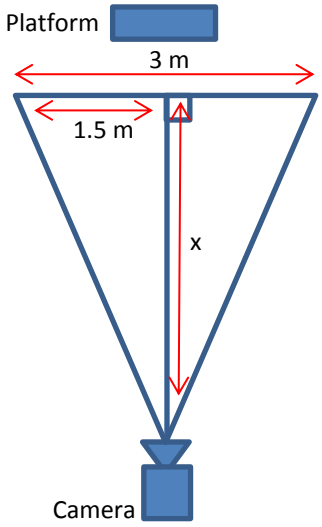
USB 2.0 Hi-Speed - Theoretically, USB 2.0 Hi-Speed mode can handle transfer speeds of up to 480 Mbps or 60 MB/s. However, no USB 2.0 connection could ever come close to the theoretical maximum throughput, making the data transfers at around 40MB/s. If we are going to use a video resolution of 640*480, we are going to need 27.6 MB/s therefore this will be sufficient. However when we use 720p HD resolution we need 82.94 MB/s which means that we will not be able to stream at 30 FPS. We will have to reduce the FPS. The same applies for 1080p HD.

USB 3.0 - The theoretical speed of USB 3.0 is 4.8 GB/s although similar to the USB we are not able achieve this speed. Typically, the real world transfer speed of USB 3.0 is ~ 100 MB/s. This will now allow us to use 720p HD at 30 FPS. However we are still not able to use 1080p at 30FPS.

Logitech C920 - A useful feature of this camera is that it has its own hardware that compresses the video using H.264 which in turn dramatically decreases the size of memory that each frame uses. This is very useful if we wanted to stream the camera at 30 FPS.

Decision

We decided to opt for the cheaper camera, the Logitech C270. The reason behind this is that this web camera possesses all the technical capabilities that we require. This webcam has a 60 degree field-of-view which is reasonable and with the correct positioning of the camera, we will be able to obtain a sensible view of the cart and the broom. To calculate the distance of the camera from the broom and the platform we followed this procedure:



To find x, we need to use some values that we already have. Here we assume that we require a viewing space of 3 metres. The webcam has a field-of-view of 60 degrees. If we split the triangle in half we obtain a right angle triangle with a. From this we can now calculate the length of the opposite which is 1.5m. To get the length of x, we use the formula:

$$x = 1.5/\tan 30$$
$$x = 2.6 \text{ metres}$$

This means that we will have to position the camera 2.6 metres away from the platform and the broom.

The C270 supports all video resolutions up until and including 720p HD (1280 x 720). We will be using a video resolution of 640x480 instead of 1280x720 as the USB 2.0 interface will not be able to handle the data transfer required by the 720p resolution if we have a high sampling rate. The sampling rate that we will require is 25 FPS which is supported by the camera.

8.1.4 Software

The relative merits of OpenCV, MATLAB and other 3rd party libraries were discussed in the Computer Vision literature review. We have opted to use OpenCV, since it is optimised for real-time processing and gives us a lot hardware flexibility – due to being usable with C/C++. Our initial algorithm design is based on colour thresholding in the HSV (Hue, Saturation, Value) colour base. The camera input is in RGB (Red, Green, Blue) but we have chosen to convert this into HSV. HSV is a rearrangement of RGB which separates luma (image intensity), from chroma (colour information). We have also made use of morphological processes to remove noisy artefacts from image segments – both object and background.

8.1.5 Choosing a Data Structure

In designing the software we had to consider the data structure that will store the video frames sent by the webcam. We have decided to use "Mat". Mat is a class that creates a matrix which stores a numerical value for each of the points of an image. The values determine the intensity of a pixel at that specific point. Before Mat came around, an image container called IplImage was commonly used. It is a C structure that stores the image in memory. The main issue that we had with this was managing the memory used. When using IplImage, the user is responsible for memory allocation and de-allocation.

Mat contains two data parts: the matrix header and a pointer to the matrix containing the values. The matrix header contains the size of the matrix, the method used for storing and where the matrix is stored.

In our particular example:

```
Mat hsv_img ( size (img.cols, img.rows) , 8, 3);
```

We have used Mat to declare that we will be creating a Mat structure. The name that we have assigned to this structure is "`hsv_img`". We declare the dimension of this matrix using "`size ( img.cols, img.rows )`".

Here we have used a pre-written procedure "size" which determines the number of columns and rows in matrix "`img`" and returns the `values`. "`8`" determines the colour depth of the sample and "`3`" defines that the sample will be a colour image (1 for grayscale).

8.1.6 Streaming from the webcam and storing each frame

This will allow us to get a stream from the webcam and store each frame. We can then analyse these frames to detect the two points that we have specified.

We start off by initialising capture from a camera:

```
CvCapture* capture = cvCaptureFromCAM(0);
```

Here, we assign our camera stream to pointer capture. The "0" determines which camera on our system we want to use. If we had two cameras, the second camera will be defined as "`cvCaptureFromCAM(1)`".

Now, we will need to store the each frame from the stream into our matrices. To do this we must use the function "cvQueryFrame" to allocate a frame from the capture and store this on the "img" matrix.

```
img = cvQueryFrame(capture);
```

We can change the size of the original stream by using the following function:

```
resize( img, img, Size( img.cols / 2, img.rows / 2) );
```


The function above requires us to include an input matrix, an output matrix and the dimensions that we want to resize it by. In this particular line we have divided the original size of the matrix "img" by two. This is function is not necessary although as we will have 4 windows when running the software, they may not all fit on the same screen.

8.1.7 Determining and tracking the two points

This part requires careful thinking and precise coding as we need to detect and track two distinct points. To do this we must create 3 threshold images. To do this, we will be using the HSV colour space instead of the more commonly used RGB. In HSV the first value, "hue", determines how much a colour refers to the pure colour that it resembles. "Saturation" defines the white-ness of the colour and "value" defines the brightness of the colour. We can use this to our advantage as if we are tracking, for example, a blue circle, we will be using a single number for the hue even though there are many shades of blue.

To convert the original stream from RGB to HSV we use the following function:

```
cvtColor( img, hsv_img, CV_BGR2HSV );
```

Here we have arguments in the function; the first one is the input matrix, followed by the output matrix and the colour conversion that we desire.

Now that we have our HSV stream, we can now threshold the image to isolate the two colours that will serve as our tracking points.

First we threshold the colour red:

```
inRange( hsv_img, Scalar(165, 145, 100), Scalar(250, 210, 160), threshold_img1);
```

Here we are using the function inRange. We input the matrix "hsv_img" and use that to find the value for colour red in HSV and output this into "threshold_img1". We specify the lower boundary first and the upper boundary of the colour after. This allows us to specify a small range of colours that will be identified as the colour that we are tracking.

Similarly, we do this to find the colour blue:

```
inRange( hsv_img, Scalar(105, 180, 40), Scalar(120, 260, 100), threshold_img2);
```

8.1.8 Determining the position of the two points

We will now have to determine the position of the points on the stream. First of all we must allocate memory to the Moment structures:

```
double area01 = moment01.m00;
```

We can now calculate moment01 and moment10 (the first two order moments) and the area (zeroth order moment) by using the moment structure. Next, we must find the centre point coordinates of these objects. To achieve this we divide moment01 and moment10 by the area.

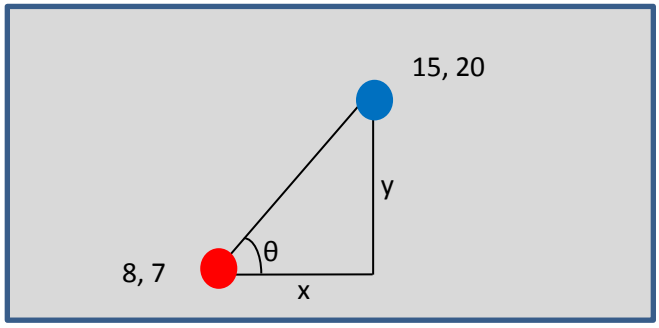
Of course, there can be noise in the camera stream that corresponds to small areas which are irrelevant. We filter these by using a conditional clause that disregards these small areas.

```
19  if (area01 > 20000 );
```

We store the coordinates using a function of OpenCV called point2d and later use this to calculate the angle created by the two points.

8.1.9 Calculating the angle

The two points that we will be tracking essentially creates a right angled triangle that we can use to calculate the angle.



First we calculate the distance between the two points, X and Y. We do this by subtracting the two coordinates of the points. Distance X = 15 – 8 and distance Y = 20 – 7.

Now that we have the length of the opposite and the adjacent, we will use trigonometry to determine the angle θ by using $\tan^{-1}$ (distance Y/ distance X). Because the code will compute the value in radians, we must also convert it back to degrees. The following snippet of code calculates the angle between the points in degrees:

```
int angle = (double)atan2( coord_list[0].y - coord_list[1].y, coord_list[1].x -
coord_list[0].x) * 180 / CV_PI;
```

8.2 Control

This subsystem requires both hardware and software elements. The controller components will be written in software.

8.2.1 Controllers

8.2.1.1 Implementing the control system in the Dancing Sweeping Brush system.

A few steps were taken to implement the controller into the system and are as follows:

1. Derive the mathematical model of the system as a whole.
2. Make assumptions to convert the non-linear mathematical model into a linear mathematical model.
3. To design the transfer function of the system and state space form.
4. Considering the types of controller to be used to determine constants which enables the stability of the inverted pendulum.
5. Utilizing the constants in a simulation to show the stability of the system.

The following steps above are further justified in the following page.

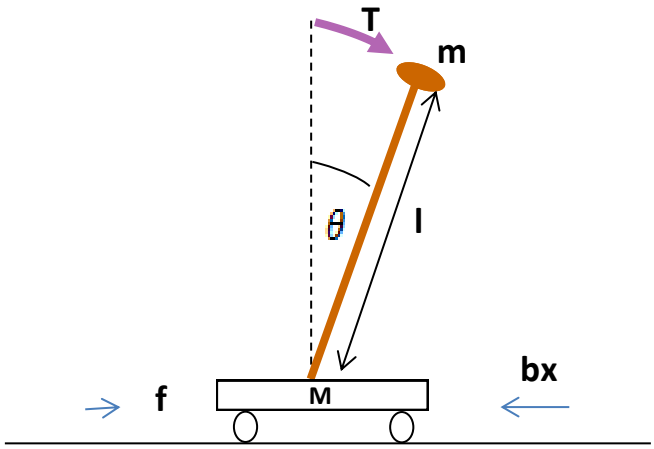
8.2.1.2 Derivation of system mathematical model

Before start designing our system, we set few requirements and goals as follows:

- Settling time to be less than 5 seconds
- The brush does not move more that few degrees away from the vertical (20 degrees)

Symbol	Description	Value
M	Mass of the cart	1.0kg
m	Mass of the brush	0.7kg
g	Acceleration due to gravity	9.8n/m^2
L	Length of the brush	1.4m
l	Length of the brush to the centre of gravity	0.7
H	Horizontal component of the reaction force	N
V	Vertical component of the reaction force	N
θ	Angle of the brush	radian
F	Applied force	N
ẋ	Velocity of the cart	m/s
Θ dot	Angular velocity of the brush	m/s
Ëθ	Angular acceleration of the brush	Rad/s^2
I	Moment of inertial (1/3 *m*l^2)	Kg/m^2
ẍ	Acceleration of the cart	m/s^2

8.2.1.3 Force analysis equation



By using newton's second law, we obtained the equations as follows:

8.2.1.4 Horizontal forces

We will start by summing the forces acting on the cart in the horizontal direction, we get

$$F = M\frac{d^2x}{dt^2} + b\frac{dx}{dt} + H \quad \text{..... (1)}$$

By summing the forces in the free-body diagram of the brush in the horizontal direction will give the expression for the reaction force H

$$H = m\frac{d^2x}{dt^2} + ml\frac{d^2\theta}{dt^2}(cos\,\theta) + ml(\frac{d\theta}{dt})(sin\,\theta) \quad \text{..... (2)}$$

Substitute H in to equation (1), will get

$$F = M\frac{d^2x}{dt^2} + b\frac{dx}{dt} + m\frac{d^2x}{dt^2} + ml\frac{d^2\theta}{dt^2}(cos\,\theta) + ml(\frac{d\theta}{dt})(sin\,\theta) \quad \text{..... (3)}$$

8.2.1.5 Perpendicular forces

If we Sum the forces perpendicular to the brush by solving along the axis we get

$$V(sin\,\theta) + H(cos\,\theta) - mg(sin\,\theta) = ml\frac{d^2\theta}{dt^2} + ml\frac{d^2\theta}{dt^2} + m\frac{d^2x}{dt^2}(cos\,\theta) \quad \text{..... (4)}$$

Moment about the centroid of the brush is;

$$-Vl(sin\,\theta) + H(cos\,\theta) - mg(sin\,\theta) = I\frac{d^2\theta}{dt^2} \quad \text{..... (5)}$$

Combine equation (5) and (4)

$$(I + ml)\frac{d^2\theta}{dt^2} - mgl(sin\,\theta) = -ml\frac{d^2x}{dt^2}(cos\,\theta) \quad \text{..... (6)}$$

Our two main equations are:

$$F = (m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta}(cos\,\theta) - ml\dot{\theta}^2 sin\theta \quad \text{..... (7)}$$

$$ml\ddot{x}cos\theta = (I+mll^2)\ddot{\theta} - mgl(sin\,\theta) \quad \text{..... (8)}$$

8.2.1.6 Making assumptions to linearize system equations

To linearize our system, we will use simple approximation around θ=0. This assumption is true due to the fact that under control, the brush will not deviate more than few degrees (20 degrees) from vertical upward position.

So for small Δθ,
Sin θ = θ and cos θ=1

By linearizing the two main equations from (7) and (8), we get,

$$F = (m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta} \quad \text{..... (9)}$$

$$ml\ddot{x} = (I+ml^2)\ddot{\theta} - mgl\theta \quad \text{..... (10)}$$

Arranging equations (9) and (10) in terms of $\ddot{x}$ and $\ddot{\theta}$

$$\ddot{x} = \frac{-ml}{M+m}\ddot{\theta} - \frac{b}{M+m}\dot{x} + \frac{1}{M+m}F \quad \text{..... (11)}$$

$$\ddot{\theta} = \frac{-ml}{I+ml^2}\ddot{x} + \frac{mgl}{I+ml^2}\theta \quad \text{..... (12)}$$

If we substitute equation (11) in (10) and equation (12) in (9) it yields,

$$\ddot{\theta} = \frac{mgl(M+m)}{I(M+m)+Mml^2}\theta + \frac{mlb}{I(M+m)Mml^2}\dot{x} - \frac{ml}{I(M+m)+Mml^2}F \quad \text{..... (13)}$$

$$\ddot{x} = \frac{-gm^2l^2}{I(M+m)+Mml^2}\theta + \frac{b(I+ml^2)}{I(M+m)+Mml^2}\dot{x} + \frac{I+ml^2}{I(M+m)+Mml^2}F \quad \text{..... (14)}$$

Since the analysis (state space model) and control design techniques that we will employ in this problem apply only to linear systems (i.e. LQR), these equations need to be linearized. We will linearize the equations about the vertically upward equilibrium position, θ = 0, and will assume that the system stays within a small neighbourhood of this equilibrium.

8.2.1.7 Designing Transfer Functions of the system and state space formation

Variable y(t) is used as an output of the system, whereby x(t) as the state of the system and u(t) as the input of the system. General representation of the space-state model is shown below

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad \text{..... State equation}$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad \text{..... Output equation}$$

The State equation shows the relationship between the system's current state and its input, and the future state of the system.

The Output equation shows the relationship between the system state and its input, and the output. From that equation we can see that, in a given system, the current output is dependent on the current input and the current state. The future state is also dependent on the current state and the current input.

After looking what state-space is, let's go back to our linear equations above and implement them in space model

Let $u = F,$

$x1 = \theta,$

$x2 = \dot{\theta} = \dot{x}1,$

$x3 = x,$

$x4 = \dot{x} = \dot{x}3$

$y1 = x,\ \ y2 = \theta,$

The four state equations will be

$\dot{x}1 = x1$

$\dot{x}2 = \frac{mgl(M+m)}{I(M+m)+Mml^2}\theta + \frac{mlb}{I(M+m)Mml^2}\dot{x} - \frac{ml}{I(M+m)+Mml^2}u$

$\dot{x}3 = x4$

$\dot{x}4 = \frac{-gm^2l^2}{I(M+m)+Mml^2}\theta + \frac{b(I+ml^2)}{I(M+m)+Mml^2}\dot{x} + \frac{I+ml^2}{I(M+m)+Mml^2}u$

8.3 Movement

This subsystem is purely hardware based. Motors must be chosen and driver circuits designed. Mechanical design of the moveable platform is also included here.

Initial ideas for the design of the moving platform varied significantly. Here we will outline the main designs we considered and our reasons for arriving at an Omni-directional wheel based free-standing platform.

The requirements of our platform are that it must move in two orthogonal directions; it must be capable of moving at speed (faster than the horizontal component of the falling brush velocity) and it must be both stable & strong enough to hold the inverted brush upright. We found the most challenging part of the platform design to be satisfying the multi-directional requirements. The final platform must be capable of switching direction quickly and moving with speed in both x and y. Our choice of mechanical platform design has had a big impact on which motors we will be using to drive it – and this will be discussed later. Firstly, let us explain the high-level options in platform design. These are:

Figure 12: Omni wheel

- **Rail based platform:** One method is to mount the platform on a

rails. Multiple rails would be needed for both the x and y direction. These rails could be in the form of tracks, as in a railway, or alternatively a linear shaft. One set of tracks would need to be moveable along the other set of tracks.

- **Wheel based platform:** Conventional wheels are not well suited to the problem as they only move in one plane, unless we were to implement a steering system. Such a system would need to operate very fast and likely add unnecessary complexity project. Alternatively, Omni-directional wheels could be used. An Omni-directional wheel looks like a conventional wheel but in addition to rolling forwards and backwards along its circumference, it also has the a ability to slide perpendicular to this on a number of rollers.

8.3.1 Unipolar Stepper Motor Test

The aim of this experiment is to test the torque and the speed of the motor. Besides, Arduino programming of stepper motor is learnt to toggle the direction of the motor and control the speed. At the end of the experiment, the most suitable type of motor is chosen for our final design.

ULN 2803 is Darlington Array which consists of 8 transistors in it. It is used to connect unipolar stepper motor by giving current to each frame so that the head of the motor is driven to move. The unipolar stepper motor has a 12V input voltage and a step angle of 1.8 degrees. Then, the motor is controlled by programming the Arduino to change the direction and speed of the motor. The next step was to have system integration between communication subsystem and the movement subsystem which includes from the computer to the Arduino via RS232 and then from the Arduino to the motors.

Procedures:

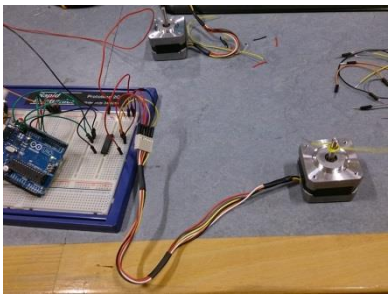


Figure 13: Testing of stepper motor

1. From the communication system using the previous 4 outputs, control signals from the PC is sent to the stepper motor to provide instruction of the direction of the motor.
2. The first step is to create a program to control the coil of the stepper motor so that high input signals provide current to the coil to stall the core.
3. There are 4 coils in the stepper motor, so If the high signals are given to pin1, pin2 then pin2, pin3 and so on accordingly, then the motor will move following in one direction.
4. Likewise, if signals are given to pin1, pin2 then pin1, pin4 and so on, then the motor will move in reverse direction.
5. The delay time set in between each set of motor instructions is to control the speed of the motor.

8.4 Communication

8.4.1 Wired Communication Methods

8.4.1.1 RS232

The first wired communication option that we considered was RS232 serial communication. The reason why we considered this option first was because we have had previous experience in creating a system that used RS232 in the module EE2A: Digital Systems & Embedded Computing that we undertook in the 2nd year of our Undergraduate degree. This gave us familiarity in this form of communication meaning that we felt confident

in reproducing a similar application for this project. Also, the option of using RS232 as a potential communication method for our project is an appealing one due to the wide availability of readily made cables (e.g. the USB to RS232 cables that are available in most computer labs) (Kramer, 2000).

Despite these advantages, the recent decline in popularity of this method means that computers often no longer contain an RS232 port. This means that whereas this option would have been more attractive when these ports were more commonly available, at present this holds no advantage over other methods of wired communications. Further, as discussed by Kramer (2000), a potential problem that could arise from using RS232 as our communication method is that it will only work over short distances (~30ft). At this stage in the project we were unsure of the maximum distance needed between the source of information and its destination meaning that if we were to consider this option, then we would be limited to a maximum distance of 30ft. As well as this, another potential problem could be that RS232 does not support multi-drop communication, where multiple receivers can be connected to the same data line and transmitter. This means that by using RS232 we would only be able to deal with point to point communication, limiting us to just one transmitter and one receiver. This would restrict our design as we would only be able to have one source of data and one destination of data.

8.4.2 Wireless Communication

8.4.2.1 ZigBee

The wireless communication method we considered was ZigBee. It is a "wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks" (Alliance, 2013). The main advantage that we found for ZigBee; aside from it needing no wires between the data transmitter and receiver is that it can support a very large maximum range of 3.2km. This range is larger than those offered by RS422 and RS485 and was the largest range offered by any communication method that we researched.

Unfortunately, it appeared that the increased communication range that ZigBee could offer us reduced the maximum data rate we could achieve by using this method to 250Kbits/s. This is less than both Bluetooth and RS422/485 wired communication. As a result, we felt that unless we were planning on having large distances between our transmitter and receiver then the advantages of this communication method would be wasted.

8.4.3 Verdict: Wired Communication vs. Wireless Communication

To conclude, it can be seen that wired communication offers larger data rates whereas wireless communication offers the convenience of having no wires linking the source and the receiver. By considering the applications of several of the communications surveyed it was found that RS232 has been used in a similar project to that we are working on. Are final decision is to initially use an RS232 serial Communication channel to send data. The main reason for this is due to us having prior experience to this communication method. If enough time is available, we would consider upgrading our

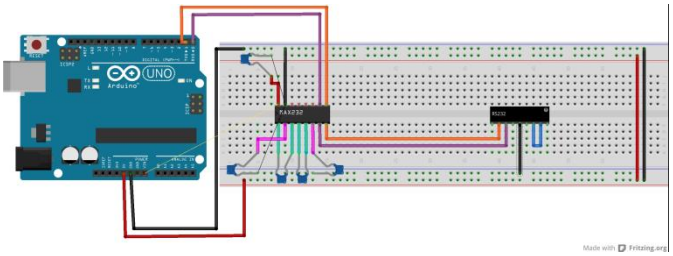


Figure 14: Breadboard layout for communication subsystem

communication method to a wireless option, with the benefit of less external wires being attached to the platform.

8.5 Microcomputers and microcontrollers

As a group, we looked into microcontrollers such as the PIC microcontrollers and Arduino microcontrollers, and microcomputers such as the Raspberry Pi and Beaglebone Black. Starting with the microcomputers, the Raspberry Pi and Beaglebone Black have larger processing speeds than a typical microcontroller, and with the components to output video and to connect directly to the internet, these can be treated as mini PCs. However, although they are relatively cheap to order, the downside with these were that that they were programmed using Linux operating system. No one in the group is familiar with Linux, meaning that we focused on microcontrollers, as they can be programmed using programming languages we were familiar with.

Most of us have experience with the PIC microcontroller prior to this project because of modules in digital logic and project work such as robot races. However, with the PIC, we would need to consider external clock and also the circuit that needs to be built around the microcontroller. The Arduino is prebuilt, meaning to connect to the microcontroller; it is just by simple plugging in wires into the sockets on the Arduino board. In the end, we have chosen to use the Arduino UNO microcontroller.

9. Prototype

We originally decided to build a prototype system, which balances an inverted pendulum in just one plane (a 2-dimensional problem). The idea here is that this simpler specification will reduce the complexity in the design and build of every subsystem. The theory will be similar to that of a two plane system. Hence, the prototype sub-systems can be developed into working sub-systems that could be enhanced for the final build. There was discussion within the group about whether to build a scaled down prototype which would as a result deal with a smaller pendulum, or whether to construct a full-scale prototype.

Before we considered the details of which approach to use, we first discussed the main points that we would try to achieve with a proto-type and why we wanted to build one:

We needed to present our progress by week 11 of Semester 1. A working prototype would be an impressive way to show good progress

A prototype is a chance to make mistakes and to learn from them. For example say we were to do some initial calculations which indicate that the platform should be three times the length of the brush, but then we could find that this is too large when building the prototype. We would be able to re-calculate for the final build

One of the difficult parts in any team engineering project is the integration of sub-systems. If we were required to integrate prototype subsystems early on in the project, we should be able to resolve this issue during the final build.

We can use our prototype to check whether the ideas about the calculations behind the physics of the system were correct in practise.

Appendix 2 addresses the advantages of both approaches.

As a result of this, we all created separate designs for what we felt that prototype should look like, after we all had presented our designs, we voted on the design we felt was the most practical. It was as follows:

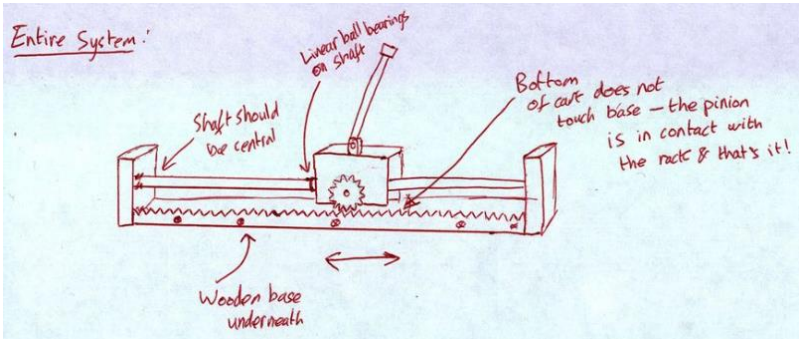


Figure 15: Single Axis Mechanical Rack & Pinion Design

The idea behind is that the platform would only move along one axis. To achieve movement, a rack and pinion would be used to allow for precise movement of the platform. The pinion would be connected to a motor which would be connected to the platform. The platform would run along a shaft. To reduce friction the platform would be encompass a set of linear ball bearings. The shaft would also help to provide stability to the platform.

Once we were confident with our design we scheduled an appointment with the Metallurgy and Materials Workshop to discuss the feasibility of our design. A summary of the meeting can be found in Appendix 3.

To conclude, our initial prototype design was considered to be very large (2m in length). As a result, we were informed that by using a rack and pinion, the project would be very expensive (≈£500 for a suitably sized rack and pinion). As we were advised neither acrylic nor wood would be suitable materials and that only metal would be sufficient to provide as much resistance against wear and tear. This posed the issue of added weight to our prototype design. Further to this, the workshop technicians stressed that time is valuable in the workshop and that if we chose to construct a prototype that it should be one that is of similar dimensions to that we were planning for the final build. This would mean that we would have to integrate a second dimension to the above design.

As a consequence of this, we decided that the idea above would not be possible. This meant that we had to evaluate our initial prototype design and instead consider what the final build should be made to look like. The result of this is that we considered integrating two dimensions into our design. The result of this is that we had two conflicting possible methods of platform movement: Rails vs. Omni wheels.

We concluded from research undertaken into Omni-wheels, Mecanum wheels and rails that Omni-wheels were the best option to help move the platform. Our reasons for this are that:

Omni wheels would be able to make the robot move in any direction

Unlike Mecanum wheels (where one dimension would be given preference over speed of movement), omni wheels, if placed correctly would give both x and y directions of movement equal speed.

Would reduce the amount of material we would require if we were to use rails as our primary movement option. If we were to use rails then similar issues could arise (see appendix 3) for summary of meeting with Warren Haye.

10. Time Management

Completion for the deadline a mandatory requirement for projects, this is no different for our project. We have till week 11 of semester 2 to produce a system that is capable of balancing a vertically upright sweeping brush on a moving platform using a real-time active vision system. In order to do this we needed to plan our time effectively. This is so we ensure time is utilised.

Our first method for doing this is to hold regular team meetings. These were scheduled every Tuesday 4-6pm and Friday 1-2pm. This has enabled us to discuss matters as a team. Although meetings were essential to discuss current information, they were aided by the creation of a project plan. The project plan consisted of the following documents:

- ***Task List*** (see Appendix 8: Figure 33) – This document was created to show the predicted tasks that needed to be completed and give estimates on the start time and finish time of these. During the project details about what had been completed each task were added.
- ***Gantt chart*** (see Appendix 8: Figure 34) – This document aimed to show a visual representation of the task list. This was used to illustrate which tasks were active at the same time. This meant that team members could be assigned to different task. The Gantt chart showed tasks on a month by month basis.
- ***Task Calendar*** (see Appendix 8: Figure 35) – Its aim was to show a week by week breakdown of the tasks that needed to be completed during the 11 week semester. This document proved important during this half of the project as we were able to all 11 weeks and the tasks that needed to be completed during this time.
- ***Timeline*** (see Appendix 8: Figure 36) – The aim of the timeline was to show the key milestones in our project. A milestone was used to represent the start of a new stage in our project or a key date. Examples of this were: "The start of proto-type design" and "1st Demonstration and Presentation".
- ***Completion Chart*** (see Appendix 8: Figure 37) – This document was very similar in style to the Gantt chart, by displaying a visual representation of the task list. The difference between the Completion chart and the Gantt chart was that it was used to indicate whether a task had been completed or otherwise indicate the amount of progress that had been made on a task.

10.1 Predicted Project Plan vs Actual Project Plan

To review our performance at this stage in the project we must compare our predicted time plan (see Appendix 8: Figure 38) with our actual time plan (see Appendix 8: Figure 39). To do this we will consider the key milestones and whether we met them. From the timelines (see Appendix 8: Figures 38 and 39) it is clear to see that we had a greater number of milestones/key events that actually did occur compared to the ones that we had planned for. These are for a multitude of reasons. Firstly, the Household Brush Experiment. We had not originally planned to undertake any experiments when we considered the tasks that we would need to complete in the first semester. However, during the first few weeks of the project it was decided to carry out this task. We found that this task became our first stepping stone into analysing the theoretical aspects of the system. As a result this task pushed back the build of all sub-systems. The reason for this being that we wanted to ensure that hardware was considered before purchase, thus ensuring that we managed our budget effectively. Another key event that had not been planned for was to "meet with the workshop technicians to discuss the feasibility of our proto-type design". This was key as the outcome of this meeting resulted in us

altering our proto-type design and eventually deciding that redesigning and building a proto-type in the amount of time remaining would not be feasible. As we had not planned for this idea, it changed the project plan for the second half of semester 1. Thus requiring quick alterations to the direction our project took.

However, we feel that the extra key events that occurred, although time consuming will save us time later in the project. For example the change in design was from feedback given from the workshop regarding build time, budget and material. We felt that it was better to receive this feedback earlier on in the project and to spend the necessary time changing our project plan and design rather than purchasing the materials needed to build the design and then finding out that it would not be feasible, thus wasting time and resources.

10.2 Semester 2 – Planning for the future

Appendix 8: Figure 40, shows the planned task list that we have created for semester 2. As we aim to build upon the progress that we have made in semester 1 we are aiming to have a completed system that is capable of balancing a vertically upright sweeping brush on a moving platform using a real-time active vision system by the 2nd demonstration and presentation on week 11 of semester 2 (26/03/14). In order to reach this goal we need to consider a set of achievable milestones for each sub-system. The aim of these will be to provide a clear set of criteria/instructions that each sub-system will aim to complete by this date.

10.2.1 Mechanical

- Model of platform using Solidworks – by integrating the frame with the motor housing, the motors and the base of the platform.
- Choose suitable materials -> materials will be considered dependent on price, strength of material, availability.
- Decision on dimensions for platform – stress test analysis based on material to be used, to ensure that we choose the optimal dimensions of the (frame, motor housing, size of holes)
- Estimate mass of platform – based upon decision of dimensions of platform and material to be used -> can be given to control sub-system for accurate value in equations
- Build and assemble the parts of the platform -> also attach hardware to frame

10.2.2 Control

- Finalise system equations in MATLAB -> implement this in C program
- Take an example angle based upon the maximum angle that can be inputted to the control algorithm ($\pm 20°$), calculate the value of force that will be outputted.
- Compare this output force to the expected force -> this can be found by entering the example angle into the calculations derived to model the physics of the system
- Integrate Vision sub-system and Control Sub-system – test that they are working by creating an angle that will match the example angle. This should then provide an output from the control sub-system that will match the value of force calculated.
- Integrate with combined communication and movement sub-systems. To test that this would work. First, apply the output force that was calculated by applying an example angle to the control system, to the motors -> check for correctness of movement and response. Secondly, by generating the same value of force by creating this example angle again using the inverted brush and the vision sub-system. Correctness of movement and response will indicate that integration has been achieved.

10.2.3 Vision

- Expand system to take inputs from two cameras. Program code will need to be adapted so that information from each camera is read in simultaneously. The greatest angular deviation of the two planes will be outputted to the control system, together with the position of the cart in that plane and a one bit flag indicating the plane.
- Improve the system to cope better with complex backgrounds. We have researched but not yet implemented background subtraction techniques. These techniques will eliminate issues with noisy backgrounds. A background model can be built by the system before the brush is introduced and this data will then be discarded later.
- Improve the system to cope better changing lighting conditions. We have chosen to use the HSV colour space in preparation for these improvements. Our background model will be based on a Gaussian distribution with mean and variance of colour. Therefore, background subtraction will still with in varying light.
- We will integrate outputs of the vision subsystem with inputs of the control subsystem. We must do this early on in the semester to account for any unforeseen integration problems. Values will be sent from the vision C++ program to MATLAB using MEX.
- If we still have time when the system is working, we may shift the processing from PC to a Micro-computer.

10.2.4 Communication

- Establish a link between the computer and microcontroller -> to further develop the connection demonstrated in the 1st presentation and demonstration. We plan to send serial data from both a C program and MATLAB to replace the current method of using Accessport. This will demonstrate that the communication sub-system can link the control sub-system and movement sub-systems together.
- Send output force from control system to movement sub-system -> aim of this is to check for the platform to respond and move as expected, to test that the RS232 communication link is working and that integration between multiple sub-systems have been achieved
- Replace the existing RS232 circuit with an Xbee Series 1 802.15.4 wireless link. The aim of this will be to get the control sub-system and movement sub-systems communicating wirelessly.
- Send output force from control system wirelessly to movement sub-system -> aim of this is to check for the platform to respond and move as expected and that the wireless communication channel is functional

10.2.5 Movement

- Perform stress analysis on platform design and update design iteratively based on results.
- Develop detailed CAD drawing of platform design to minimise the likelihood of mistakes in the build.
- Finalise choice of DC motors; purchase these and integrate with driver circuitry.
- Integrate motor control with Control subsystem outputs.
- Integrate motors with platform and conduct thorough testing.

11. Conclusion

Summary of Main Conclusions We have met the majority of our objectives for the first semester of this project. We set out to implement mock-up subsystems for each of the four areas: vision, control, communication and movement. Our initial implementations of these systems have been simplified to one plane of movement, but we have designed each in such a way that expansion to a two-plane system will be straightforward. In the vision subsystem, we have implemented a system which successfully outputs the angular deviation of the broom from vertical. Its limitations relate to vulnerability to noise and changing lighting conditions. However, we have outlined our plans to tackle these issues in this report. For the control subsystem, we have simulated control equations to tackle a similar problem to the final aim – with two different methods, LQR and PID. From this design and testing, we have concluded that PID will provide a better response time for our system. We have developed a communication system for our project, using the RS232 protocol. This will serve to integrate the other subsystems and shares theoretical principles with the wireless protocol, IEEE 802.15.4, we hope to implement next semester. With regards to movement, we have come to a design decision for the build of the mechanical platform. This has not been implemented as quickly as first planned, however this has given us the opportunity to refine the design to meet the requirements fully. The build of the platform will be a sizeable component of next semester work. A number of different motors have been considered and also tested practically, giving us a strong knowledge base for our final motor choice. We will be using DC motors to drive the Omni-wheels on our final platform. We are happy with the progress we have made in developing theoretical models to date and look forward to implementing these fully in the New Year.

12. Works Cited

- Alliance, Z. (2013). *802.15.4 – ZigBee Physical Layer*. Retrieved 11 27, 2013, from Digi: http://www.digi.com/technology/rf-articles/wireless-zigbee
- Bruce, R. (2009, August). *Communication Part 1: Communication between Computers*. Retrieved 12 02, 2013, from University of North Carolina: http://www.cs.unca.edu/~bruce/Fall09/255L/CommunicationPart1.pdf
- Bryds, D. M. (2013). Computer Control Systems: The Swinging Pendulum Case Study. *EE2C2: Control Systems*. University of Birmingham, Electronic, Electrical and Computer Engineering.
- Digi. (2008). *Demystifying 802.15.4 and ZigBee.* Retrieved December 13, 2013, from Digi - XBee: http://www.digi.com/pdf/wp_zigbee.pdf
- Digi. (2013). *XBee wireless RF modules*. Retrieved December 13, 2013, from Digi : http://www.digi.com/xbee/
- Hans-Dieter Schutte, H.-J. N. (2004). Substitution of RS422 navigation sensor networks using Bluetooth communication for integrated bridges on vessels. *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on (Volume:4)*, 3036-3039.
- Haoping Wang, A. C. (2008). Stabilization of a 2-DOF inverted pendulum by a low cost visual feedback. *American Control Conference*, 3851 - 3856.
- Holzapfel, M. E. (May 1998). Fuzzy-Logic Control of an Inverted. *IEEE TRANSACTIONS ON EDUCATION, VOL. 41, NO. 2, MAY 1998*, 165-170.
- Jia Hui-juan, G. Z.-h. (2010). Research on the technology of RS485 over Ethernet. *2010 International Conference on E-Product E-Service and E-Entertainment (ICEEE)*, 1-3.
- Kazi Mahmud Hasan, A.-A.-N. A. (2012). Implementation of vision based object tracking robot . *IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision*, 860 - 864 .
- Kentaro Hirata, Y. K. (2007). Visual Feedback Control of Cart-Pendulum Systems with Webcam. *Proceedings of International Conference on Mechatronics*, 1-6.
- Kramer. (2000). *Data Communication and RS-232.* Retrieved 10 17, 2013, from Alpha Sound & Lighting Co: http://www.aslgc.com/aslcgc_downloads/understanding_RS232.pdf
- Lei Shi, B. G. (17-19 Oct. 2009). RS485/422 Solution in Embedded Access Control System . *Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference*, 1 - 4 .
- Lei Shi, B. G. (2009). RS485/422 solution in Embedded Access. *Biomedical Engineering and Informatics*, 1-4.
- Lenka, N. (2011). *Modeling and Controller Design for an Inverted Pendulum System.* Rourkela, National Institute of Technology.

- Sharifi, M. A. (2010). *Design, Build and Control of a Single Rotational Inverted Pendulum.* University of Tehran, School of Electrical and Computer Engineering.

- Conradt, J., Cook, M., Berner, R., Lichsteiner, P., Douglas, RJ. & Delbruck, T., 2009. A Pencil Balancing Robot Using a Pair of AER Dynamic Vision Sensors. Institute of Neuroinformatics UZH-ETH Zurich, Switzerland. [Accessed: 2 December 2013].

- Bjarenstam, MJ. & Lennartsson, M.. 2012. Development of a Ball Balancing Robot with Omni Wheels. Department of Automatic Control, Lund University.[Accessed: 2 December 2013].

- Sharifi, M. 2010. Design, Build And Control Of A Single Rotational Inverted Pendulum. [online] Available at: http://ece.ut.ac.ir/SMRL/members/Sharifi/Inverted%20Pendulum.pdf [Accessed: 28 Nov 2013].

- White, W., Wagner, J., Blankenau, B., Wang, Z. and Salazar, V. 2013. Design, build, and test of an autonomous inverted pendulum cart. American Control Conference (ACC), 2013, pp. 5893-5898. Available at: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6580762&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6580762 [Accessed: 28 Nov 2013].

- Jiang, Z. and Kamise, K. 2007. Unstable Object Stabilization and Control Using a DD Robot Manipulator. Networking, Sensing and Control, 2007 IEEE International Conference on, pp. 273-278. Available from: doi: 10.1109/ICNSC.2007.372790 [Accessed: 13 Dec 2013].

13. Appendix

Appendix 1 – Table of Possible Camera Choices

Name	Logitech C270	Logitech C525	Microsoft Lifecam Studio	Logitech C920
Price	£17.00	£32.12	£53.29	£59.55
Image Quality	720p (1280*720 Pixels)	720p (1280*720 Pixels)	1080p (1920*1080 pixels)	1080p (1920*1080 pixels)
Frames per Second	30	30	30	30 (at highest resolution)
Field of View	60°	69°	75°	78°
Cable Length	1.5m	1.5m	1.8m	1.8m
Connection	USB 2.0	USB 2.0	USB 2.0	USB 2.0/USB 3.0
System Requirements	Vista, 7, 8 with 2.4GHz Core 2 Duo & 2GB RAM	Vista, 7, 8 with 2.4GHz Core 2 Duo & 2GB RAM	XP, Vista, 7 with 3.0GHz Core 2 Duo & 2GB RAM	Vista, 7, 8 with 2.4GHz Core 2 Duo & 2GB RAM
Additional Features	-	-	-	Comes with H.264 compressor hardware

Appendix 2 – Small Scale vs Full Scale Prototype

Small Scale Prototype

1. **Quick:** We would have been more likely to complete it for week 11.
2. **Cheap:** Any mistakes we would have made in the prototyping stage would cost us less as we would have been using fewer materials due to the reduced scale
3. **Simpler:** By using a mock-up pendulum (as opposed to the real sweeping brush), we can initially get our system working for a pendulum with a symmetrical weight and then build upon this.
4. **Scalable:** We would have been able to set the ratio for the size and weight of the prototype pendulum to the actual sweeping brush.
5. **Mistakes:** With a small-scaled prototype, any mistakes that we would have made could have been identified and then learned from to improve the design. We would not be restricted in our final design by what we had already created in the prototype.
6. **Do Calculations twice:** There would have been parameters we would need to recalculate after finishing the prototype but the theory behind both the prototype and the final build would remain similar

Full Scale Prototype

1. **Re-usable:** A large prototype could have been physically incorporated into the final system by adapting it to deal with a second plane (the brush being able to fall in both the x and y directions). Parts bought with the prototype budget would be recycled and used again in the final build
2. **Only need to do calculations once:** Parameters such as the gain of the feedback system should remain the same. The flip side of this is that they would be more complex to begin with.
3. **Restricted:** If we were to have built a full scale prototype, we would be restricted from making major design changes due to the full scale prototype using more materials and more budget than the small scale prototype

Appendix 3 - Meeting about the feasibility of our prototype design with Warren Haye from the Metallurgy and Materials Workshop 14/11/13

Items Discussed

1. Showed Rack and Pinion Design -> He asked "How large would the rack be?"
2. I answered that the brush is 1.3m in length so the length of the rack would be around 2m at least.
3. His initial response is that a rack and pinion for that length would be expensive ≈ £500 (for one) [He checked RS, Farnell – all examples appeared expensive]
4. He then mentioned using a ball screw -> a motor would be used to turn the screw. The problem with this is that it would not provide the required response times -> of 1.4s max.
5. I asked whether we could use the CNC Miller to make a rack out of Acrylic Plastic. =>Acryllic is not suitable, it would wear away. CNC machine can only fit a sheet of acrylic up to 1m inside of it. So a 2m rack is not possible
6. If we are to use a rack and pinion then we would need to add a guard, to protect from the rotating pinion.
7. The structure would need to be strong, metal = advised. Wood is not advised. (Precision Equipment)
8. I asked the question to Warren "If he were given the same specification as us, how would he think about designing the mechanical aspect of the project". He looked at the specification and noted that the Spec only states for the brush to **"balance"** so won't need to correct the brush at **large angles**. Could hold the brush upright and then initiate control. After a certain angle (no return angle) the platform would stop trying to correct the brush and let it.
9. This approach would mean that our platform would not need to move as much as we had planned for. As a result the platform would be smaller, thus lighter and cheaper to make.
10. An example would be to reduce the size of the movement area (rack) to say 0.5m. The result of this is that the materials would be cheaper to purchase, would be lighter. This would help during testing as we will be able to test our "Dancing Sweeping Brush" in a smaller area. Also the platform would be lighter. We need to think about the size of the build once you have added the second dimension.
11. He also noted that the prototype should be less of a prototype and more of a continuing build, meaning that it should be able to be integrated into the final build. This would save workshop time, budget and effort. So = Full sized prototype
12. Linear Ball Bearings are recommended but one shaft will not be enough to prevent the platform from rotating. Therefore using two shafts with two sets of linear ball bearings provide us with a much more stable platform.

Alternative Drive Methods

- **Winch Servo** – Continuous rotation servo with a pulley. Platform would be attached to string. The servo would pull the platform when it rotated. Servo will allow fine small movements.
- **Ball Screw** – Similar Idea to Rack and Pinion – Motor would turn the screw forward or backwards. The platform would act as a bolt and would move in accordance to the motion of the screw.
- **Toothed Belt** – Using a motor with a timing pulley gear to drive it. We could use multiple gear sizes/number of teeth to affect how fast the belt will turn -> how fast the robot responds

Figure 20 - Winch Servo

Figure 21 - Ball Screw Nut

Figure 22 - Toothed Belt

Figure 23 - Timing Pulley

Alternative Platform Methods:

- Robot Platform (Car, would have wheels, would not be able to handle diagonal movement well)
- Robot Arm (Model our platform on what a human would do to balance the brush)

A Previous Idea – Example of using a toothed belt

Whilst I was in the workshop, I was able to see a previous project that had tackled a similar method of movement:

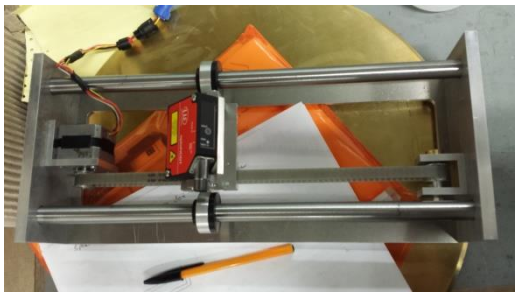


Figure 24 - Plan View, two sets of linear ball bearings

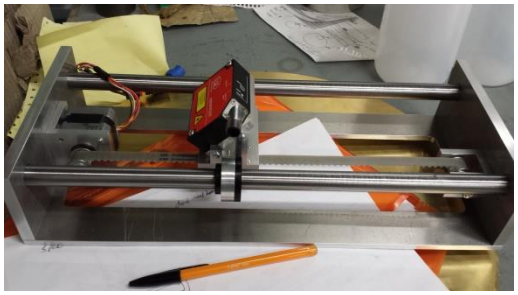


Figure 25 – Overview of previous project

How does it work

A stepper motor is used to control the movement of a platform between two points. The stepper motor is connected to a timing pulley; a second timing pulley is located at the other support. A toothed belt is connected to the platform and between both pulleys. To move the platform in a given direction the motor rotates and to move in the other direction the motor moves in the counter direction. The speed that the platform moves by is determined by the torque of the motor being used and the gear ratio between the

timing pulley and the toothed belt. Two shafts are connected between both supports to provide stability to the platform. Linear ball bearings run over the shaft providing minimal friction for the moving platform.

I thought this idea was great, by having the motor not on board the platform, then it means that the weight of it is no longer a concern. This would mean that more powerful but potentially heavier motor could be used. Idea is very similar concept to Jonas's prototype design.

Figure 26 - Timing pulley connected to Y129 stepper motor (same as robot project)

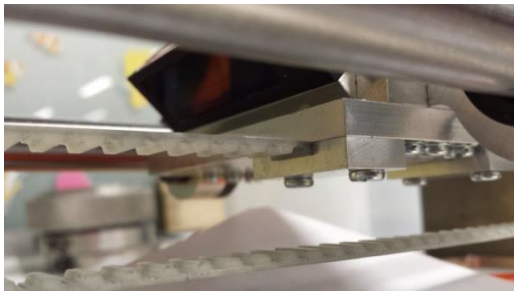


Figure 27 - Tooted belt connected to platform

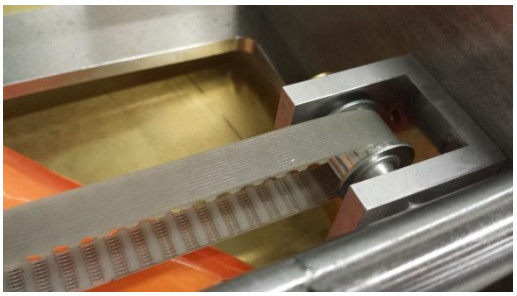


Figure 28 - Toothed belt connected to timing pulley

Appendix 4 - Possible Omni/Mecanum Wheel Choices

Difference between Omni and Mecanum wheels:

An **Omni wheel** is the same as a conventional wheel but it has small **rollers around the circumference** which are **perpendicular to the rolling direction**. The effect is that the wheel will roll with full force, but will **also slide laterally** with great ease.

The **Mecanum wheel** is similar, in that it is a conventional wheel with added rollers. However, in this case, the **rollers are placed at an angle around the periphery of the wheel**. The angled peripheral **rollers translate a**

portion of the force in the rotational direction of the wheel to a force normal to the wheel direction

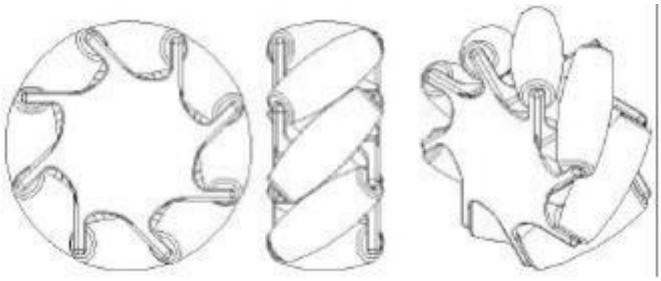


Figure 29 - Mecanum Wheel

If we use Omni wheels, the wheels must face in different directions. Omni wheels facing the same direction, like the four wheels on a conventional car, are useless – as there is no motor force to slide the device laterally. Mecanum wheels, however, are designed to be set up in the same direction – as rotation of the main wheel also results in some force normal to the wheel. The resultant vector of the devices motion can be altered by adjusting the speed of each wheel independently.

Appendix 5 – Summary of Meeting Minutes

Meeting Number: 1
Date: 15/10/13
Attendees: All
Key Topics Discussed: Team Roles, Project Aims, Personal Objectives, Risk Assessment, Ethical Review, Gantt Chart Design
Key Outcomes: Sub-System Breakdown – Assign roles
Work Set: Research into sub-systems
Deadline: 18/10/13

Meeting Number: 2
Date: 18/10/13
Attendees: All including Supervisor
Key Topics Discussed: Individual sub-system research conducted
Key Outcomes: Idea of simple inverted pendulum prototype
Work Set: Continue research into specific areas
Deadline: 22/10/13

Meeting Number: 3
Date: 22/10/13
Attendees: All
Key Topics Discussed: Risks of project, Prototype
Key Outcomes: Gantt Chart, Risk Assessment, Ethical Form, Group Contract, Video Contract, Critical Path Analysis, Objectives/Specification/aims
Work Set: Continue research into specific areas
Deadline: 25/10/13

Meeting Number: 4
Date: 25/10/13
Attendees: All including Supervisor
Key Topics Discussed: Aims, Risks, Sub-system integration
Key Outcomes: Experiment with Brush, Inverted Pendulum Lab
Work Set: Designs for Prototype
Deadline: 29/10/13

Meeting Number: 5
Date: 28/10/13
Attendees: All
Key Topics Discussed: Broomstick Stability, Critical Angle, Estimation of Mass, Maximum Response Time
Key Outcomes: Values for Broomstick Stability, Critical Angle, Estimation of Mass, Maximum Response Time
Work Set:
Deadline:

Meeting Number: 6
Date: 29/10/13
Attendees: All
Key Topics Discussed: Forces and Speed, Maximum Response Time
Key Outcomes: Benchmark system response time
Work Set: Why carry out "Household Brush Experiment", What did we learn from it, Processing time of different Microcontrollers/computers, Why are we looking at transfer times, Transfer time/data rates/ data speeds of cameras, Transfer time/data rates/data speed of communication methods
Deadline: 1/11/13

Meeting Number: 7
Date: 01/11/13
Attendees: All including Supervisor
Key Topics Discussed: Critical Angle, Group Research – work set from 29/10/13
Key Outcomes: Possible to speed of the robot to respond to change, can use to estimate kind of pc and/or microcontroller to use
Work Set: Produce sketches for idea of platform of the prototype that we intend to build
Deadline:

Meeting Number: 8
Date: 05/11/13
Attendees: All
Key Topics Discussed: Sketches for idea of platform
Key Outcomes: Decision on proto-type design, Rails vs wheels
Work Set: Vision choices on camera, computer vision, Control choices on pseudo code, software, Movement choices on motors, Communications choice on communication method, AutoCad drawing of prototype design, Neat design of prototype design, Physics of the System, Complete Electronic Circuit
Deadline: 08/11/13

Meeting Number: 9
Date: 08/11/13
Attendees: All including Supervisor
Key Topics Discussed: Discussion on Rack and Pinion Design, National Instruments Measuring Equipment
Key Outcomes: Change of future meetings on Fridays
Work Set: Block Diagram for entire system and electronic circuit, Choice of Microcontroller, AutoCad / 3D model of prototype, Updated Gantt Chart
Deadline: 12/11/13

Meeting Number: 10
Date: 12/11/13
Attendees: All
Key Topics Discussed: Proto-type Sketch, Possibility of using LabView
Key Outcomes:
Work Set: Compile parts list, Meet with Warren Haye in Workshop to discuss feasibility of design
Deadline:

Meeting Number: 11
Date: 15/11/13
Attendees: All including Supervisor
Key Topics Discussed: Feedback from workshop about feasibility of design,
Key Outcomes: Whether the prototype should be full scale or reduced scale, Need a change of design
Work Set: How will we integrate the robot design, final design, parts list
Deadline: 19/11/13

Meeting Number: 12
Date: 19/11/13
Attendees: All
Key Topics Discussed: Ideas produced in regards to the motion of the robot, Physics of the system
Key Outcomes: Omni wheels considered as movement possibility for system
Work Set: Research into Omni wheels
Deadline: 20/11/13

Meeting Number: 13
Date: 22/11/13
Attendees: All including Supervisor
Key Topics Discussed: Orders placed, Omni wheel research, literature reviews
Key Outcomes: Decided not enough time to build prototype
Work Set: Individual Sub-System Literature Reviews
Deadline: 27/11/13

Meeting Number: 14
Date: 26/11/13
Attendees: All

Key Topics Discussed: Items obtained from orders, fuzzy logic, what to produce for the upcoming presentation, interim report, control – data processing, motors
Key Outcomes: Omni wheels considered as movement possibility for system
Work Set: Meet in lab to work on build of sub-systems
Deadline: 27/11/13

Meeting Number: 15
Date: 29/11/13
Attendees: All including Supervisor
Key Topics Discussed: Discussion of the control simulation, movement, literature review
Key Outcomes: Research into LabView alternatives, MATLAB – inputs and outputs, need to work on integration of sub-systems
Work Set: Individual Sub-System Literature Reviews modifications
Deadline: 03/12/13

Meeting Number: 16
Date: 03/12/13
Attendees: All
Key Topics Discussed: Presentation plan, details of what each topic will talk about: vision, control, communication, vision, team leader,
Key Outcomes: 4 people to present: Matthew, Thomas, Jonas, Kok Li. 4 people to demonstrate: Adesegun, Raymund, Faisal, Julian
Work Set: Sub-system build, presentation slide completion, continue to write interim group report
Deadline: 06/12/13

Appendix 6 - Using MATLAB to solve the equation

MATLAB CODE

```
% Implement state space model
M = 1.0;                % Mass of the cart
m = 0.7;                % Mass of the brush
b = 0.1;                % Frictional force of the cart
I = 0.114;              % Moment of inertia
g = 9.8;                % Gravitational Acceleration
l = 0.7;                % Length of sweeping brush with respect to the centre of gravity of t he brush

p = I*(M+m)+M*m*l^2; % Denominator for the A and B matrices
A = [          0              1        0        0              ;
       (m*g*l*(M+m)/p)        0        0      (m*l*b)/p        ;
               0              0        0        1              ;
       (-g*m^2*l^2)/p         0        0      b*(I+(m*l))/p]   ;
B = [  0; (m*l)/p; 0; (I+(m*l^2))/p];
C = [  1       0       0       0;
       0       0       1       0];
D = [0; 0];
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'U'};
outputs = {'x'; 'phi'};
% Find space state model here
system = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
Gs=tf(system)    % transfer function for both output
```


After running above code we got the transfer function of the cart as

$$x = \frac{0.9128\ s - 0.025}{s\text{^}3 - 0.1125\ s\text{^}2 - 15.21\ s + 2.111}$$

and the transfer function of the brush is

$$\theta = \frac{0.8513\ s\text{^}2 - 1.89e{-}014\ s - 16.95}{s\text{^}4 - 0.1125\ s\text{^}3 - 15.21\ s\text{^}2 + 2.111\ s}$$

Illustrating equations in MATLAB and SIMULINK representations and implementing controllers

In this section, equations are both represented in MATLAB and SIMULINK functions in order to first test the stability of the system with the use of controllers.

Utilizing MATLAB representations

The following shows the modelling equations of the system using MATLAB with the implementation of the Linear Quadratic Regulation Controller

```
%--------   System Modelling (LQR)          ---------%
% Defining variables using real parameters of the system
M = 1.0;                                  % Mass of the cart
m = 0.7;                                  % Mass of the brush
b = 0.1;                                  % Frictional force of the cart
I = 0.114;                                % Moment of inertia
g = 9.8;                                  % Acceleration due to gravity
l = 0.7;                                  % Length of the brush to center mass

d=I*(M+m)+M*m*l^2;                        % Denominator of the transfer function
% State-space model
disp('state space model of the system is:')
A=[0 1 0 0;m*g*l(M+m)/d 0 0 m*l*b/d;0 0 0 1;-g*m^2*l^2/d 0 0 -b*(I+m*l^2)/d]; B=[0;-m*l/d;0;(I+m*l^2)/d];
C=[0 0 1 0;1 0 0 0]; D=[0;0];
system=ss(A,B,C,D)
% Transfer function of the system
inputs = {'u'}; outputs = {'x'; 'tetha'};
G=tf(system)
set(G,'InputName',inputs)
set(G,'OutputName',outputs)
% System Analysis of an Open Loop System
inputs = {'u'}; outputs = {'x'; 'theta'};
set(G,'InputName',inputs)
set(G,'OutputName',outputs)
figure(1);clf;subplot(221)
t=0:0.01:1;
impulse(G,t);
grid;
title('Open-Loop Impulse Response of the system')
% Open-Loop step response
subplot(222)
t = 0:0.05:10;
u = ones(size(t)); [y,t] = lsim(G,u,t);
plot(t,y)
title('Open-Loop Step Response of the system') axis([0 2 -50 50])
legend('x','theta')
grid;
%  Poles of the system and pole-zero mapping
Poles=eig(A)
subplot (2,2,3:4) pzmap(system);
title('Pole-zero mapping of inverted pendulum')
%  Checking controllability
CO=ctrb(system);
Rank_CO=rank(CO)
% Checking Observability
OB=obsv(system);
Rank_OB=rank(OB)
```

```
%  Closed Loop Step – Response
%  Controlling using LQR Method (Consider case when: Q(1,1) = 80, Q(3,3) = 400)
Q = C'*C;
Q(1,1)   = 80;                  % Increasing the weight on the cart's position
Q(3,3)   = 400                  % Increasing the weight on the pendulum's angle(theta)
R = 1
K = lqr(A,B,Q,R)                % State-feedback control gain matrix
Ac = A-B*K;                     % Control matrix
system_c = ss(Ac,B,C,D);        % The controlled system state space model
t = 0:0.01:10;
r =0.2*ones(size(t));
figure(3);clf [y,t,x]=lsim(system_c,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot'); set(get(AX(1),'Ylabel'),'String','cart
position (m)') set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
title('Step Response using LQR')
grid
poles=eig(system_c)
```

The following shows the modelling equations of the system using MATLAB with the implementation of the Proportional, Integral, and Derivative (PID) Controller

```
%--------   System Modelling (PID)        ---------%
% Defining variables
M = 1.0;                        % Mass of the cart
m = 0.7;                        % Mass of the brush
b = 0.1;                        % Frictional force of the cart
I = 0.114;                      % Moment of inertia of the sweeping brush
g = 9.8;                        % Acceleration due to gravity
l = 0.7;                        % Length of the brush to centre mass
d = (M+m)*(I+m*l^2)-(m*l)^2;    % Denominator of the transfer function
s = tf('s');
% Transfer function of both brush and the cart
Tf_brush = (m*l*s/d)/(s^3 + (b*(I + m*l^2))*s^2/d - ((M + m)*m*g*l)*s/d - b*m*g*l/d);
Tf_cart=(((I+m*l^2)/d)*s^2 - (m*g*l/d))/(s^4 + (b*(I + m*l^2))*s^3/d - ((M + m)*m*g*l)*s^2/d -
b*m*g*l*s/d);
system = [Tf_cart ; Tf_brush];
inputs = {'f'};
outputs = {'x'; 'phi'};
set(system,'InputName',inputs)
set(system,'OutputName',outputs)
figure(1);clf;
t=0:0.01:1;
impulse(system,t);
title('Open-Loop Impulse Response without PID')
grid
```

```
% Open loop impulse response of the system
figure (2); clf;
t = 0:0.05:10;
u = ones(size(t));
[y,t] = lsim(system,u,t);
plot(t,y)
title('Open-Loop Step Response without PID')
axis([0 3 0 50])
legend('x','phi')
grid
%  Closed Loop Step – Response
%  Design with PID control
Kp = 120;
Ki = 1;
Kd = 30;
C = pid(Kp,Ki,Kd);
T = feedback(Tf_brush,C);
figure(3);clf;
t=0:0.01:10;
impulse(T,t)
axis([0, 2.5, -0.2, 0.2]);
title(' Response of brush Position to an Impulse Disturbance under PID controller'); grid
on
```


Utilizing SIMULINK representations

The following shows the open loop and closed loop system using the Proportional, Integral and Derivative Controller using the Simulink representation of the system.

Open Loop System (Without the implementation of controller)

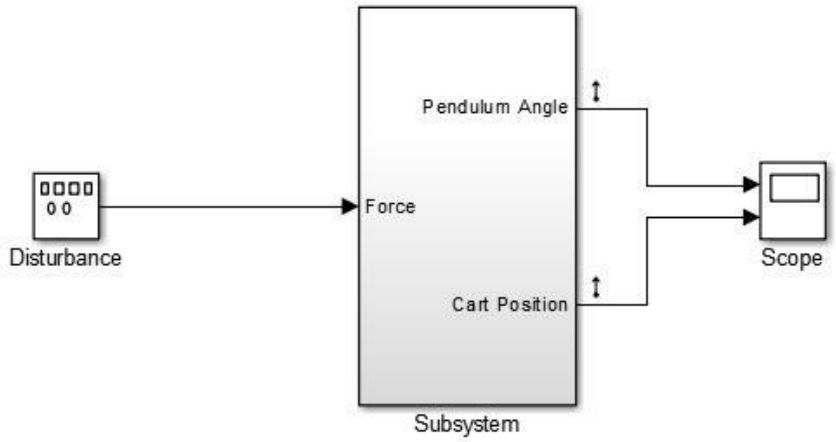


Figure 30: Open Loop System with SIMULINK representation

*The force impulse block simulates the force which is applied to the cart.

The subsystem component consists of real parameters of the system which shows the blocks which makes up the dancing sweeping brush system which consists of the brush and the moving cart.

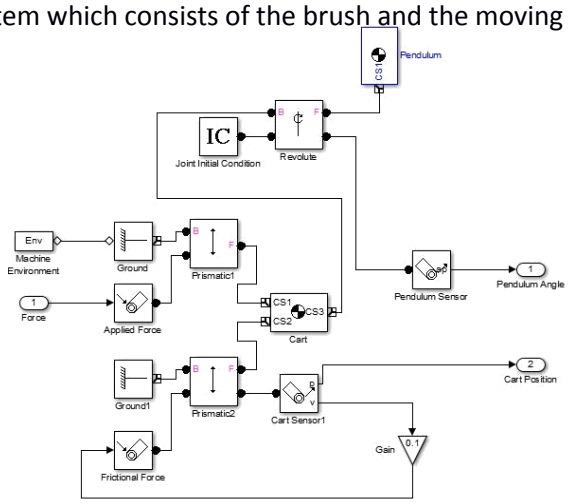


Figure 31: Subsystem block diagram

The pendulum and the cart blocks are used to simulate the sweeping brush and the moving platform of the dancing sweeping brush system. The parameters of each of the blocks are fitted with parameters in table

Similarly, the closed loop of the system is also considered with the inclusion of the PID Controller as a negative feedback into the force impulse which is placed both at the angle of the pendulum and the position of the cart in order to obtain a stabilized system. The figure below illustrates the closed loop system of the dancing sweeping brush in a closed loop system.

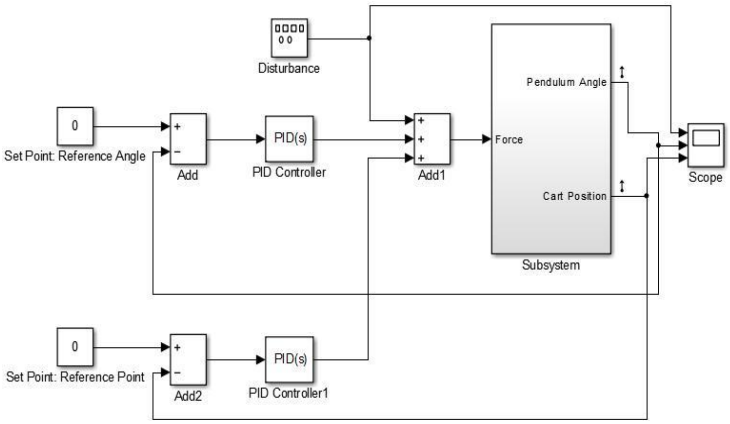


Figure 32: Closed Loop System with SIMULINK representation

At this point, the parameters of the pendulum and cart blocks placed in the subsystem block are maintained. Referring to the figure 32, two PID controllers are implemented into the system as a negative feedback. The set point of the Reference Angle and the Reference Point refers to the desired values at which the dancing sweeping brush is stable. The disturbance block shown above is represented by a signal generator which produces random signals to the system. This is purposed to simulate the random disturbance which is applied to the system and how the system reacts in order to stabilize the pendulum (sweeping brush) with respect to the parameters of the PID gains.

Appendix 7 – Background Subtraction and HSV (Hue, Saturation and Value)

As a result of previous research, we looked further into colour tracking. Colour tracking provides a basis for our object tracking purpose. By tracking a coloured object, we are able to determine its centre point and from there we are able to take measurements that will potentially enable us to calculate an angle. Video frames that are processed by the camera are of the RGB colour space. It is possible to track colour in the RGB colour space although it is hard to define a specific colour. Illumination also presents a huge factor as the shade of the colour will differ under different levels of light. We have learned that in the HSV colour space, a colour is represented by the "hue", its whiteness is represented by the "saturation" and the brightness of the colour is represented by the "value". By converting the image from RGB to HSV, we are able to make colour tracking more achievable. The reason for this is that we can define a the colour that we want to track by specifying a range of values for the "hue" and then we can define a range of values for the "saturation" and the "value" to compensate for varying levels of lighting. A disadvantage of this method is that any colours in the image that are similar to the colour that we intend to track will also be recognised by the software. A method to counter this will be to create a constraint where any objects that have the same colour but are less than a certain amount of pixels are ignored.

Our research also led us to background subtraction. Background subtraction is an image processing method in which we take a model of the background of an image and from there we can produce a foreground mask which is essentially an area in the current video frame which is subtracted from the original model of the background. Background subtraction requires a static camera. The reason for this is that if the background is constantly changing along with the frames, we wouldn't have a reference image to subtract incoming frames, rendering this method unusable. There are two main parts in background subtraction; these are background initialisation and background update. As mentioned, the result of this method gives us the foreground mask which for our purpose will be the platform and the broom. By subtracting these objects from the background image we are able to disregard any other objects in the background, which in turn (if used with colour tracking) will reduce the possibilities of incorrectly tracking other similar colours. The source (Gary Bradski et al., 2008) states that although background subtraction works fairly well in simple scenes, there is a disadvantage where it suffers from an assumption that is often violated. Background subtraction assumes that all the pixels are independent. It learns a model that does not take into consideration any neighbouring pixels. To take this into account we need to create and learn a multipart model. This model will take into consideration the brightness of any neighbouring pixels. From this we can distinguish when a surrounding pixel is dim or bright. By doing this we effectively create two models; one when the surrounding pixels are dim and one when they are bright. This solution however requires more memory which may slow our image processing depending on the hardware that we use.

Another technique that we came across is by using morphological operations. Methods such as erode and dilate allow us to produce a more accurate video tracking software. Erode enables us to remove any areas that are under a specified amount of pixels, essentially removing any noise in the images. Dilate on the other hand enlarges an area of pixels that are equal to or above a specified area. By incorporating these functions into our software we are able to enlarge the areas that we intend to track and remove any areas that are irrelevant to us.

Appendix 8 – Time Management

MEng Team A Project Gantt Chart

ID	Task Mode	Task Name	Duration	Start	Finish	Completed By
1		Distribution of Team Roles and Sub-Systems	1 day	Tue 15/10/13	Tue 15/10/13	All
2		Discussion of Project Aims, Specifications and Personal Objectives	1 day	Tue 15/10/13	Tue 15/10/13	All
3		Independent Research into Sub-System Research	4 days	Tue 15/10/13	Fri 18/10/13	All
4		Both Members in each Sub-System Compare Research - Highlight Key Item	3 days	Fri 18/10/13	Tue 22/10/13	All
5		Risk Assessment	4 days	Tue 22/10/13	Fri 25/10/13	Thomas McKay-Smith
6		Ethical Review Form	4 days	Tue 22/10/13	Fri 25/10/13	Lee Kok Li
7		Creation of Group Contract	4 days	Tue 22/10/13	Fri 25/10/13	Raymund Lagua
8		Gantt Chart	2 days	Tue 22/10/13	Wed 23/10/13	Matthew Cottrell
9		Critical Path Analysis	3 days	Wed 23/10/13	Fri 25/10/13	Ade Adepegba
10		Consent for Accidental Filming whilst using USB HD Cameras	4 days	Tue 22/10/13	Fri 25/10/13	Jun Hoong
11		Objectives/ specification / project aims documentation	4 days	Tue 22/10/13	Fri 25/10/13	Faisal Ahmed, Sean Jonas
12		Order Brush	1 day	Mon 28/10/13	Mon 28/10/13	
13		Order Chosen USB HD Cameras	1 day	Mon 28/10/13	Mon 28/10/13	
14		Start of Proto-Type Design	0 days	Tue 29/10/13	Tue 29/10/13	
15		Design of Proto-type for Vision Sub-System	6 days	Fri 25/10/13	Fri 01/11/13	
16		Design of Proto-type for Control Algorithm Sub-System	6 days	Fri 25/10/13	Fri 01/11/13	
17		Design of Proto-type for Communication Sub-System	6 days	Fri 25/10/13	Fri 01/11/13	
18		Design of Proto-type for Movement Sub-System	6 days	Fri 25/10/13	Fri 01/11/13	
19		Mechanical Design of Proto-Type	3 days	Fri 25/10/13	Tue 29/10/13	
20		Final Design for Proto-Type	1 day	Tue 29/10/13	Tue 29/10/13	
21		3D Model of Proto-Type Design [AutoCad]	4 days	Tue 29/10/13	Fri 01/11/13	
22		Complete Electronic System for Proto-Type	4 days	Tue 29/10/13	Fri 01/11/13	
23		Start of Proto-Type Build	0 days	Fri 01/11/13	Fri 01/11/13	
24		Compile Parts List for Proto-Type	6 days	Fri 25/10/13	Fri 01/11/13	
25		Order Parts for Proto-Type	1 day	Mon 04/11/13	Mon 04/11/13	
26		Build Sub-Systems seperately for Proto-Type				
27		Vision	5 days	Wed 06/11/13	Tue 12/11/13	
28		Control	5 days	Wed 06/11/13	Tue 12/11/13	
29		Communication	5 days	Wed 06/11/13	Tue 12/11/13	
30		Movement	5 days	Wed 06/11/13	Tue 12/11/13	
31		Combine Vision and Control Sub-System Proto-Types	4 days	Tue 12/11/13	Fri 15/11/13	

Page 1

Figure 33: Predicted Task List for Semester 1

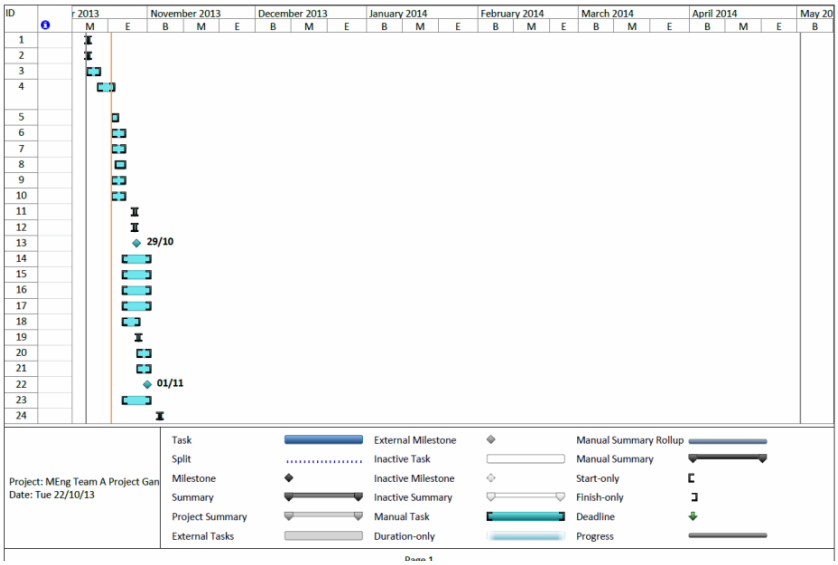


Figure 34: Predicted Gantt chart for Semester 1

Figure 35: Predicted Task Calendar for Semester 1

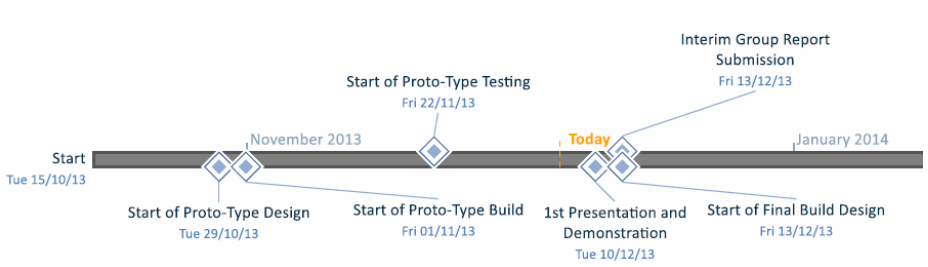


Figure 36: Predicted Timeline for Semester 1

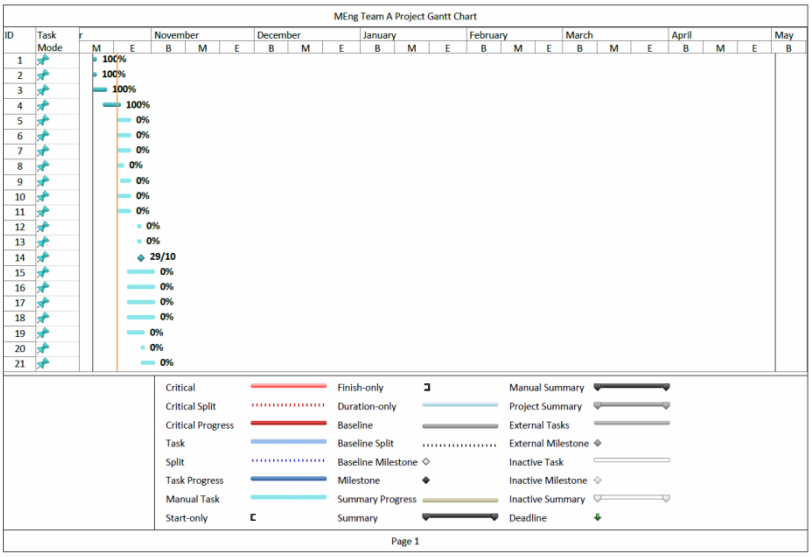


Figure 37: Task Completion Chart for Semester 1

Figure 38: Predicted Timeline for Semester 1

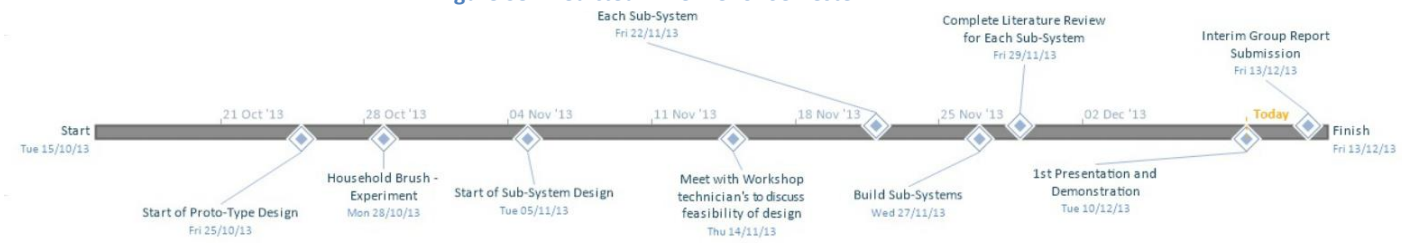


Figure 39: Actual Timeline for Semester 1

41	Start of Final Build Design	0 days	Fri 13/12/13	Fri 13/12/13
42	Propose Designs for Final Build (Mechanical)	23 days	Fri 13/12/13	Tue 14/01/14
43	Design of Vision Sub-System Final Build	6 days	Tue 14/01/14	Tue 21/01/14
44	Design of Control Algorithm Sub-System Final Build	6 days	Tue 14/01/14	Tue 21/01/14
45	Design of Communication Sub-System Final Build	6 days	Tue 14/01/14	Tue 21/01/14
46	Design of Movement Sub-System Final Build	6 days	Tue 14/01/14	Tue 21/01/14
47	Final Design of Final Build	1 day	Tue 21/01/14	Tue 21/01/14
48	3D Model of Final Build Design [AutoCad]	6 days	Tue 21/01/14	Tue 28/01/14
49	Complete Full Electronic Circuit for Final Build	6 days	Tue 21/01/14	Tue 28/01/14
50	Start of Final Build Production	0 days	Tue 28/01/14	Tue 28/01/14
51	Compile Parts List for Final Build	4 days	Tue 28/01/14	Fri 31/01/14
52	Order Parts for Final Build	1 day	Mon 03/02/14	Mon 03/02/14
53	Build Sub-Systems seperately for Final Build			
54	Vision	6 days	Fri 31/01/14	Fri 07/02/14
55	Control	6 days	Fri 31/01/14	Fri 07/02/14
56	Communication	6 days	Fri 31/01/14	Fri 07/02/14
57	Movement	6 days	Fri 31/01/14	Fri 07/02/14
58	Combine Vision and Control Sub-System Final Build	6 days	Fri 07/02/14	Fri 14/02/14
59	Combine Communication and Movement Sub-System Final Build	6 days	Fri 07/02/14	Fri 14/02/14
60	Integrate all Sub-Systems into Full System	8 days	Fri 14/02/14	Tue 25/02/14
61	Start of Final Build Testing	0 days	Tue 25/02/14	Tue 25/02/14
62	Project Testing of Final Build	22 days	Tue 25/02/14	Wed 26/03/14
63	Comparison of Build to Project Specification, Aims and Objectives	14 days	Wed 26/03/14	Mon 14/04/14
64	Prepare for 2nd Demonstration and Presentation	22 days	Tue 25/02/14	Wed 26/03/14
65	2nd Demonstration and Presentation	0 days	Wed 26/03/14	Wed 26/03/14
66	Prepare and Write Final Group Report	35 days	Tue 25/02/14	Mon 14/04/14
67	Final Group Report Submission	0 days	Mon 14/04/14	Mon 14/04/14
68	Poster Presentation Preparation	14 days	Mon 14/04/14	Thu 01/05/14
69	Poster Presenation on Project Open Day	0 days	Thu 01/05/14	Thu 01/05/14

Figure 40: Predicted Task List for Semester 2

Appendix 9 – Simulation Results

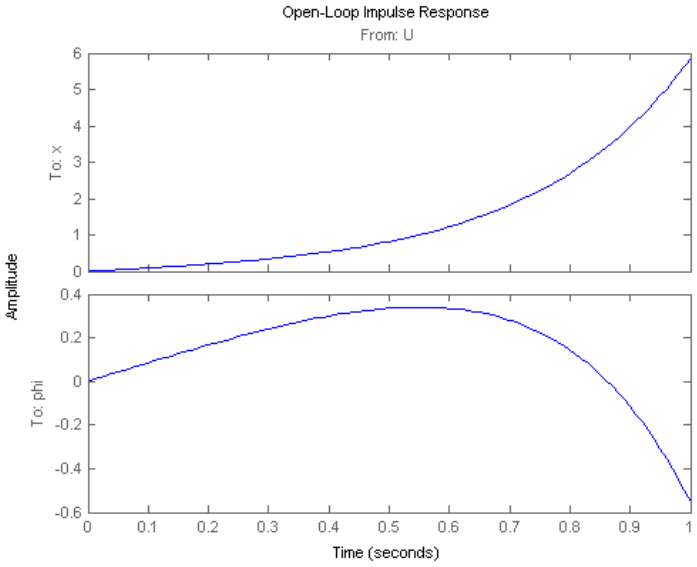


Figure 41: Open Loop Impulse Response

Open Loop Impulse Response

From figure 41, we can see the response is unsatisfactory. Both outputs never settle and the angle of the brush goes to several radians in a clockwise direction though it should be less than 0.35 rad (20 degrees). And the cart goes to the right infinitely.

This shows that the system is unstable in an open loop condition when there is a small impulse force applied to the cart.

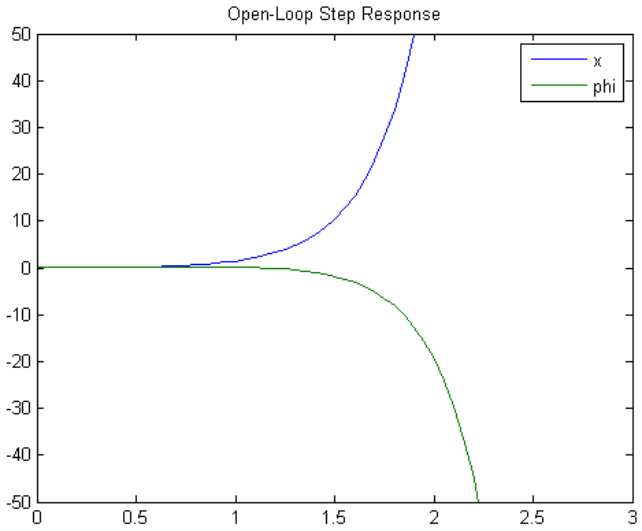


Figure 416: Open Loop Step Response

Open Loop Step Response

Based on figure 42, the step response is unsatisfactory due to the fact that the angle of the sweeping brush is not parallel to the direction of movement of the platform. In addition to that, the figure 43 below shows that the system is unstable.

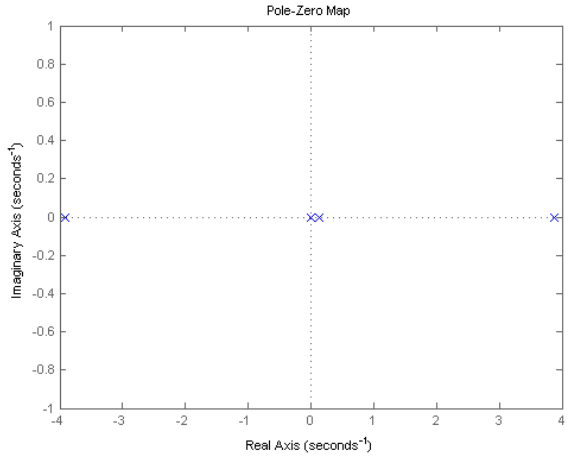


Figure 43: Pole - Zero Graph

Figure 43 on the other hand shows that the system is unstable. We can also see that three poles on the right side of the pole – zero graph confirms that the system is unstable in open loop.

Closed Loop Step Response using LQR controller

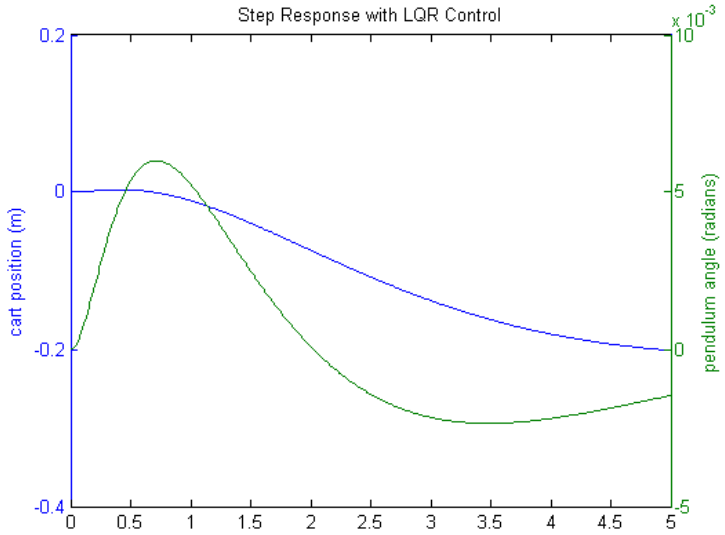


Figure 44: Closed Loop Step Response with LQR Controller

The plot above is not satisfactory. The overshoot of the both pendulum and cart appear fine, but their settling times need improvement and the cart's rise time needs to be reduced.

By increasing the Q(1,1) and Q(3,3) elements as shown in Appendix 7, the settling and rise times decreases and lowers the steady state error of the angle of the pendulum. By modifying the values of Q(1,1) and Q(3,3) to 1000 and 200 respectively, the responding step response is shown below.

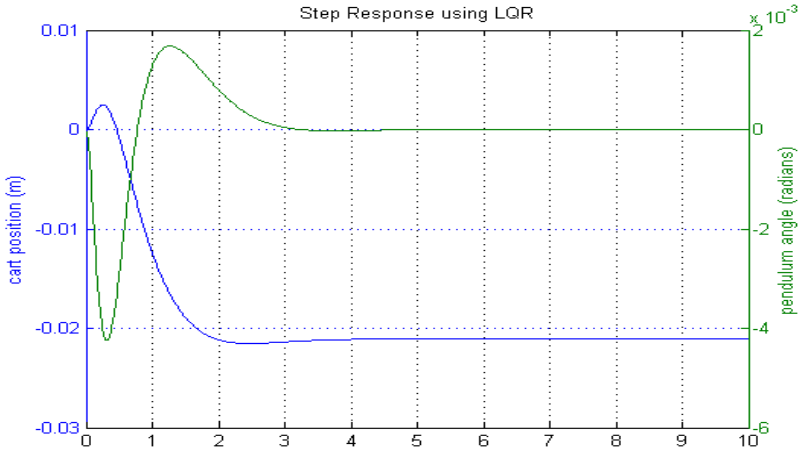


Figure 45: Closed Loop Step Response with LQR Controller with Q(1,1) = 1000 and Q(3,3) = 200

The values of Q(1,1) and Q(3,3) are chosen to be 1000 and 200 respectively because it satisfies the design requirements in which the settling time is less than 5s and the angle of the brush does not exceed 0.35 radians. Further increasing the magnitude of Q would improve response time of our system. However, this would require greater control force which would then require more energy to provide the force needed to move the cart.

PID design

Open loop impulse response of the system

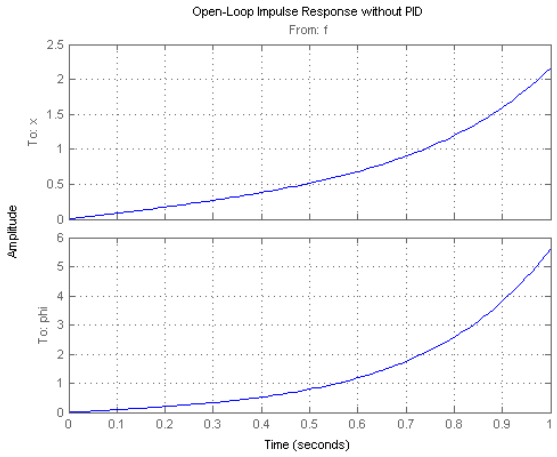


Figure 46: Open Loop Impulse Response without PID

Figure 46 shows that the open-loop response is unsatisfactory. This is due to the fact that both outputs never settle and the angle of the brush and the cart goes to the right infinitely.

Open loop step response

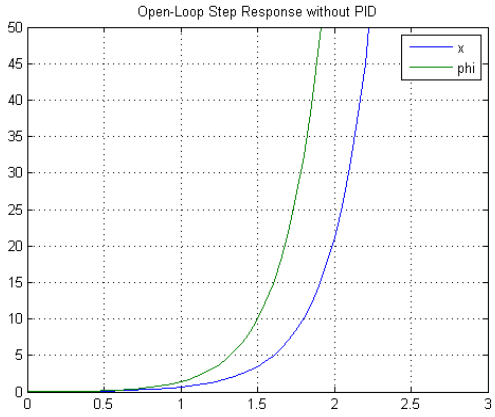


Figure 47: Open Loop Step Response without PID

The open loop step response of the system is completely unstable. The direction of both the cart and the brush goes infinitely and never settles.

Closed Loop Step Response

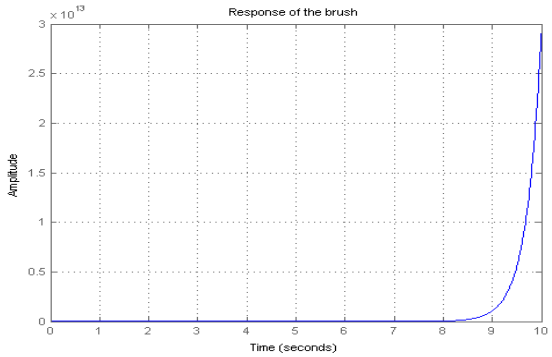


Figure 48: Closed Loop Step Response with PID at Kp=1, Ki=1, Kd=1

Considering gain values $Kp = 1, Ki = 1 \; and \; Kd = 1$

It can been seen that the system is unstable as the response time of the system is low.

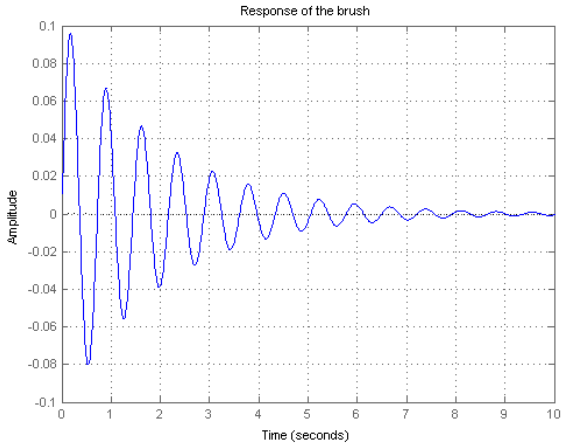


Figure 49: Closed Loop Step response with PID at Kp=100, Ki=1 and Kd =1

Considering gain values $Kp = 100, Ki = 1 \; and \; Kd = 1$

By changing the proportional gain, Kp from 1 to 100, we increased the system response but at the same time, the system becomes unstable which is shown by the increased in the number of oscillations.

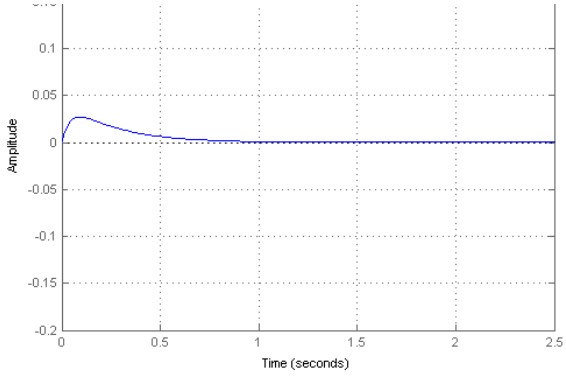


Figure 50: Closed Loop Step response with PID at Kp=130, Ki=1 and Kd =30

Considering gain values $Kp = 130, Ki = 1 \; and \; Kd = 30$

By increasing the value of Kd from 1 to 30, we can see that the system now becomes stable as the settling time of the system decreases with low overshoot and zero steady state error.