Mapping a Physical Environment
CM0343 - Interim Report

Tom Hancocks - 0930818 - BSc Computer Science
Supervisor - Frank C Langbein
Moderator - Dave Marshall

December 16, 2011

Abstract

The goal of this project is to explore the possibilities of building autonomous exploration robots from readily and cheaply available components rather than the incredibly high precision and largely expensive components typically used.

Is it possible to produce a robot cheaply that can effectively map an unknown physical environment?

This report focuses on the hardware side of the project, exploring the different types of locomotion, sensing and existing robots. From their the report moves on to designing a basic chassis and motor control circuits for the robot and producing a working prototype of these circuits.

This report will not expand to the software aspect of this project.

Contents

List of Figures

1 Introduction

It is not always possible for humans to enter an environment to find out information about it. The environment may simply be too small for a human to successfully navigate, or it may be exposed to high levels, or even lethal levels of radiation that may have long lasting adverse health effects. Whatever the reason, it is sometimes necessary to look for alternative solutions to exploring unreachable locations.

There have been numerous methods of exploring unknown environments such as remotely controlled drones like UAVs or buggy's which could have a mounted camera on it which could then send back information and receive instructions over radio frequencies. This type of exploration and navigation requires that a person or several people control it from a safe place and required many hours of work.

With the development and advancement of robotics, computers and technology through the 20th century, this kind of exploration is now turning to the use of robotics.

Perhaps most famously in the realms of exploring inhospitable places are the Mars Exploration Rovers. These robots take exploring unknown, hard to reach and dangerous locations to a new level; they are operating on a completely alien planet. The robot itself contains an impressive selection of sensors and camera's for analysing its environment and finding out as much as it can about the planet Mars (NASA 2004).

Robots such as this are very expensive. The cost of its initial 90 Day mission (including construction) weighed in at around $820Million, with costs climbing beyond this (AssociatedPress 2007). Suffice to say, not many institutions or organisations can throw this kind of funding towards exploring an unknown environment.

The same goes for many other scenarios; organisations frequently fall back on remotely operated drones in order to find out information in unreachable locations rather than looking to autonomous systems.

Increasingly however, robotics is beginning to find its way into the home in much more affordable applications. One example of this is *Samsung's NaviBot Robotic Cleaner* (Samsung 2011), which uses a camera to capture information about a persons home and devise the optimal route for cleaning the floors. Additionally the robot is able to return to a base station to automatically recharge and after recharging return to its prior task, thus requiring only minimal amounts of input from the user. This shows that it is possible to produce a robot capable of exploring an environment at comparatively low costs when compared to the Mars Rovers and other such robots.

This project will look at the creation of an autonomous robot that has the task of mapping and unknown physical environment, whilst keeping the project costs to a minimum. Is it possible to attain usable results from such a robot, and if so, why aren't they utilised?

1.1 Aims & Objectives

To make the most informed decisions possible the basic aims & objectives of the project will be outlined here. By outlining these here it will be possible to focus research into key areas of interest, rather than looking at many unrelated topics across a much broader spectrum.

* **Move under its own power**
  The robot will need to be able to move itself and not rely on a person moving it from point A to point B.

* **Avoid obstacles / Collision detection**
  As the robot moves through its environment it will need to be able to avoid obstacles and avoid collisions with them.

* **Sufficient resources**
  The robot will almost certainly produce a lot of data. The system needs to be able to handle computationally demanding situations. Will certain functions need to run offline or can the system be optimised to run online?

* **Constructing the map**
  How can the large amount of the data be turned into meaningful information?

* **Positioning**
  How will we know the position of the robot? What techniques and methods exist to determine position in the physical world?

* **Cost**
  The purpose of the project is to find out if such a robot can be built for a very low cost.

These six aspects are the main areas that I have identified as being important in the development of the robot.

Once these aspects of the robot have been developed and decided upon I will test the robot by positioning it in a sufficiently complex environment that has numerous unknowns, such as the corridors of the school in the North and Central buildings. By allowing the robot to move around and explore the environment here it will allow the software and the hardware to be truly tested. Would it be possible to have the robot retrace its path and return to its starting point?

1.2 Overview

From this, I will be exploring existing solutions for robots, including types of navigation and how to carry out any computation needed to navigate and explore the environments. I shall also determine any limitations and advantages to different solutions, and what work arounds exist to these problems.

From this background information it should then be possible to begin designing the basic hardware required for the robot and to produce a working prototype of the robot.

This report will not focus on the software side of the robot at this stage of the project.

2 Background Research

In this section I will outline and show the information that has been collected during the initial background research.

2.1 Existing Robots & Solutions

In order to get an idea of the types of problems that arise in small, cheap robots I have explored a number of different robots, but ultimately decided to focus on the following two. These robots both look at land based navigation and do not have any prior knowledge of their environment.

2.1.1 Maze Solving Robot

Maze Solving Robot was created by (patrickmccb 2011) and detailed instructions on its creation published on Instructables, and has the basic goal of being able to find the shortest path in a given maze after giving it one initial attempt at solving. The robot itself is powered by a cheap open-source prototyping solution, Arduino.

The robot addresses several issues that I will need to consider within the initial design stages of this project; controlling of the onboard systems of the robot, locomotion of the robot and in storing and remembering a given path.

Maze Solving Robot uses a clone of the Arduino which is named, RBBB or *Really Bare Bones Board*. This solution is cheaper than a standard Arduino *Uno*, but does not offer the same USB interface that the Uno does.

The robot uses a simple two wheel differential drive system, where each wheel is driven by a small DC motor.

2.1.2 Wall Avoiding Robot

Another basic robot that considers the problem of avoiding walls (and by extension object avoidance), is the aptly named *Wall Avoiding Robot* (Brandon 2007). Like with the Maze Solving Robot, this robot employs differential drive, and is also powered by Arduino.

In order to 'see' objects ahead of itself, the robot uses an ultrasonic sensor which allows it to pinpoint objects ahead of itself, along with IR Range Sensors. These forms of sensors allow for basic feedback in order to have the robot react to obstacles around itself.

2.1.3 Versatrax 300 VLR

The Versatrax 300 VLR (Inuktun 2011) is a robot designed specifically for remote inspection of pipelines, and whilst not directly mapping a physical environment, it is going into hard to reach locations in order to give people a better understanding of them.

Versatrax is simple in its design; it uses a set of simple treads to move through a pipe, following the path of the pipe in order to navigate.

Due to the nature of the design and the purpose of the robot, it has little to none onboard processing and instead transmits captured images back along a cable to a computer and thus human operator. This presents a possible means of dealing with high quantities of data which maybe too much for an onboard computer to handle; transmitting data back along some medium (Wireless or Wired) for processing.

2.2 Locomotion

I decided to focus my research purely on several types of basic land based locomotion. This decision has come from looking at the types of locomotion used in the two prior robots and exploration robots created by NASA, such as the Mars Rovers. These basic types of locomotion include:-

* **Differential Drive**
  With this method robots will use a number of wheels (typically 2) which are attached to DC motors. Each motor (and thus wheel) is independently controlled.

* **Car-like**
  This method utilises 3 or 4 wheels, with two wheels being powered by a single motor and the other 2 being used to steer the robot.

* **Walking**
  Unlike car-like or differential robots, walking robots do not rely on DC motors but instead make use of servos. Walking robots introduce many challenges to even basic locomotion.

There are numerous other methods of locomotion (i.e. Articulated Body)

2.2.1 Differential Drive

Differential Drive is the idea that each wheel of the robot can be rotated independently. By being able to rotate each wheel at varying speeds it is possible to make one side of the robot travel faster than the other side, and thus turning the robot. The difference in speeds of each gives us varying sizes in the turning circle of the robot.

Out of the methods of locomotion that I have looked at, Differential Drive, is unique in that it allows the robot to rotate about a single point without the need to translate as it does so. This ability makes Differential Drive a prime candidate for low-budget robotics as it does not require any complex mechanical engineering.

Figure 1 highlights three different configurations of wheel rotation. The first configuration, a, is the simplest to understand. Both wheels are rotating in the same direction at the same speed and thus cause the robot to move forwards in a straight line.

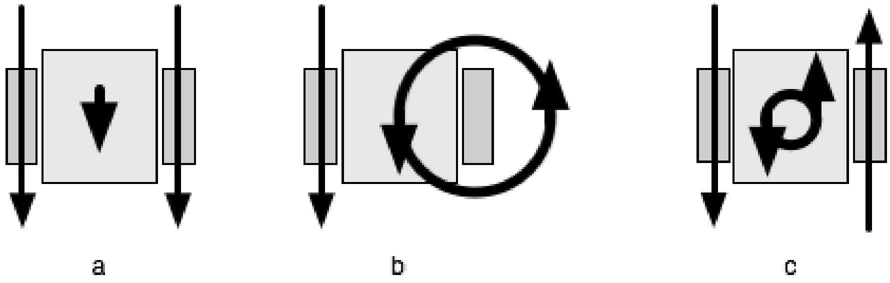

Figure 1: The effects of wheel rotation speed on turning circles and movement

By stopping the rotation of one of the wheels, as shown in b, the robot will acquire a turning circle. The size and centre of the turning circle is determined by the rotational speed of both wheels. If both wheels are rotating in the same direction, but at different speeds the turning circle will lay outside of the robot itself, at the side of the wheel with the slower rotation. If one wheel has no rotational speed then the centre of the turning circle will lay at the centre of the stationary wheel.

Finally, both wheels can have the same rotational speeds but in different directions, as shown in c. This leads to the robot being able to turn about a single point. By varying the speeds of each wheel, the centre point of the turning circle can again, be moved. However due to both wheels rotating in different directions the turning circle will always be centred somewhere on the robot.

The simplicity of Differential Drive also introduces several issues. Differential Drive is completely none mechanical (it doesn't rely on moving parts to twist or change the positioning and rotation of the robot) and thus a set velocity can never be guaranteed. When travelling in straight line both motors will have to get the exact same level of current to ensure that both wheels rotate at the same speed. In practice this is simply not possible, which causes the robot to drift to one side.

2.2.2 Car-like

As the name suggests, *car-like* locomotion uses a method of movement similar (if not identical) to that of a typical car. Although many different types of locomotion exist in cars, one typical and common design is the method of powering the two rear wheels using the same engine and then steering the car by rotating the two front wheels. This method relies on a more mechanical way of controlling the direction of movement.

It is important to note that this method of control and locomotion requires at least 3 wheels to operate correctly.

Figure 2 illustrates how the rotation of the two front wheels can effect the direction in which the vehicle travels. The two front wheels can be rotated through a mechanically determined and fixed range of angles it order to guide the direction of the vehicle. The

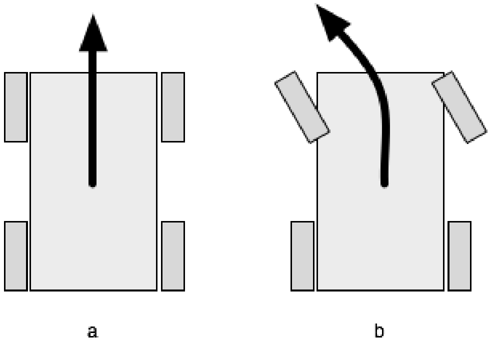

Figure 2: The effects of wheel rotation on the direction of movement

powered wheels at the rear of the vehicle push the rest of the vehicle forwards, in the direction that the front wheels point. By rotating the wheels to an angle of 0° the vehicle will move forwards.

The fact that both rear wheels are powered by the same engine means they are both physically guaranteed to rotate in sync, and thus given smooth terrain, will both deliver the same amount of drive through each wheel. This means that in theory the vehicle can be guaranteed to move in a straight line unlike the Differential Drive.

However in practice imperfections in even smooth terrain mean that each wheel will deliver two different levels of drive at any given point in time. There is also the possibility of the combined power from the forward momentum and friction that the steering wheels experience, the steering wheels will be slightly turned away from 0° and thus cause a gradual drift in the path of the vehicle similar to what is seen in Differential Drive.

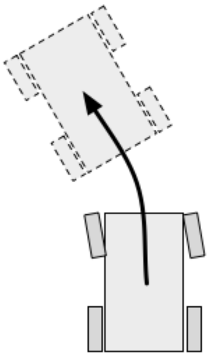

Figure 3: Car-like steering results in translation of the vehicle whilst it turns.

Car-like steering presents one major drawback which could prove to be problematic. As illustrated in *Figure 3* when a vehicle utilising a car-like steering system turns it also has to translate from one position to another. Whilst on a network of roads on macro scale this does not present much of an issue and is preferable to being able to turn on the spot. However at a much smaller level, for a robot that will have to navigate obstacles, corridors and doors this form of steering presents more problems than it solves.

Some of these problems could be, when does the vehicle need to begin turning in order to successfully avoid an obstacle and not bump in to it? What angle do the wheels need to be rotated to in order for the path to successfully pass between obstacles or around them? When travelling in reverse what angle do the wheels need to be rotated in order to turn in the correct direction?

The size of the turning circle is also an important consideration in this matter. If the vehicle can not successfully turn 180° within the confines of a corridor then it will be unable to successfully retrace its path should it meet a dead end whilst exploring its environment.

2.2.3 Walking

Walking as a form of locomotion in robots is the result of observations in the natural world and a desire to make robots more 'natural' looking. Walking robots present many challenges and have many advantages and disadvantages associated with them.

The idea of a walking robot is one that is viewed by many people as what a robot actually should be. This image comes from many popular and well known robots from *Science Fiction*, such as Maschinenmensch, one of the earliest depictions of a bipedal robot in film in 1927. More recently however research in to Artificial Intelligence and human like robots has led to developments in bipedal locomotion for robots.

One of the most well known bipedal robots is Honda's ASIMO, created at the *Wako Fundamental Technical Research Center* by Honda in Japan. ASIMO is a fully autonomous robot with ability to observe its environment and successfully navigate it, able to walk around and step over obstacles.

Locomotion based on walking is not limited to Bipedal Locomotion however. Robots can also be created with 4 legs, 6 legs or more.

Walking locomotion does present many disadvantages however.

Balance

Balance is an issue that is not really present in wheel based robots. Wheel based robots generally, by design, carry much of their weight close to the ground, thus meaning their centre of gravity is close the ground. This makes it difficult for them to capsize when they hit an obstacle or make too sharp of a movement.

However walking robots do have the issue of balance. Most of the electronics and weight are generally carried at a greater height to give the robot a greater degree of movement and freedom in the movement of its limbs.

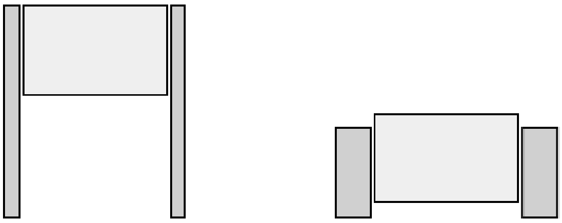

Figure 4: In walking robots the centre of gravity is elevated higher than in wheeled robots. This causes a greater degree of instability in them.

Servo's

Unlike wheel based locomotion methods, walking does not make use of motors, but instead makes use of servo's. These are a higher precision type of motor that can be rotated to a specified angle.

The ability to control and specify the ranges and angles that the servo will rotate to is essential in leg based locomotion as it allows the vehicle to know with a reasonable degree of accuracy, where its legs are at any given moment. It also allows the vehicle to move the legs in to the exact positions it needs to be able to move.

Unlike with wheel based vehicles, walking vehicles have legs, which have joints. These joints are what allow the leg to move and operate freely. However, each of these joints requires its own servo. This can rapidly increase the costs of the robot as the number of joints, and thus servo's increases.

2.3 Calibration

As previously stated, almost all forms of locomotion have some type of error discrepancy and inaccuracy. In some forms of locomotion this discrepancy is more exaggerated than in others. Due to being unable to eliminate these inaccuracies in the hardware of the vehicle, it is necessary to find ways in which to reduce these inaccuracies through software.

The inaccuracies come from trying to determine the exact location of the vehicle within its environment. As the inaccuracies of the hardware cause the vehicle to drift from its calculated path, or be either further ahead or behind the expected point of that path. Over time these errors and inaccuracies can add up and form a large margin of error.

Numerous methods exist to try and combat these errors, and although it is extremely difficult to eliminate these errors completely, they can be reduced significantly.

I will be looking at a selection of different techniques that can be used to reduce the error discrepancies.

2.3.1 UMBmark

UMBmark is a method for measuring errors related to dead-reckoning in mobile robots (J.Borenstein & L.Feng 1994). The **U**niversity of **M**ichigan **B**ench**mark**was devised by Dr. Johann Borenstein and Dr. Liqiang Feng as means of discovering certain *systematic* errors that may counteract each other and remain mostly undetected in other tests that are less rigorous.

Systematic errors are the result of kinetic imperfections such as a difference in the diameter of the wheels on the vehicle and stay constant over long periods of time. *Non-Systematic* errors however, may be caused by wheel slippage or irregularities in the floor.

By using the measurements acquired through the UMBmark it is possible to calibrate the robot in order to reduce the size and rate of errors caused by dead-reckoning.

Dead-Reckoning

It is important for any autonomous vehicle to be able to accurately determine its position within its environment at any given point in time. Low cost mobile robots make use of dead-reckoning due to its simplicity and inherent robustness.

Historically dead-reckoning has been used to great effect in maritime vessels and in wheeled vehicles.

Dead-reckoning attributes its low cost to being able to determine the location of a vehicle without the assistance of external sensors and observation (Roston & Krotkov 1991).

By using dead-reckoning techniques and formulas it is also possible to make predictions on where a vehicle will be after a certain period of time. However since the formulas and equations of the system are in-essence, making predictions the system can not truly be accurate. Whilst these errors are relatively minor when considering a large, or even, global scale, at a micro scale with a small vehicle travelling small distances these errors are much more severe.

It is because of this relative increase in the severity that calibration of the vehicle is required.

The UMBmark Procedure

(J.Borenstein & L.Feng 1994) refer to the UMBmark procedure as being the *Bi-Directional Square Path* test, a development on the older *Uni-Directional Square Path* test developed by Cybermotion in 1987.

The original uni-directional square path test was determined to be inadequate for vehicles that utilised a differential drive system. In order to overcome these drawbacks Borenstein & Feng developed the bi-directional square path test.

Following the same basic procedure as the uni-directional test, the UMBmark expands upon it by requiring that the test be performed in both clockwise and anti-clockwise directions. The reason for this is that in each direction a different group of errors becomes the dominant force in causing drift in the vehicle. Figure 5 illustrates the gradual drift of the vehicle throughout its path in the test.

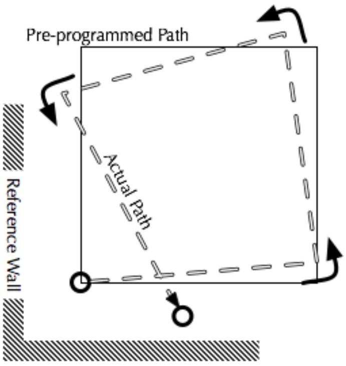

Figure 5: The solid black square represents the pre-programmed path to be travelled. The dashed line shows the actual path travelled by the vehicle.

The test requires that the vehicle complete a 4m x 4m square first five times in the clockwise direction and then in the anti-clockwise direction.

The result of this test when plotting the stopping positions for both clockwise and anti-clockwise in to a graph produces a result similar to the one in Figure 6.

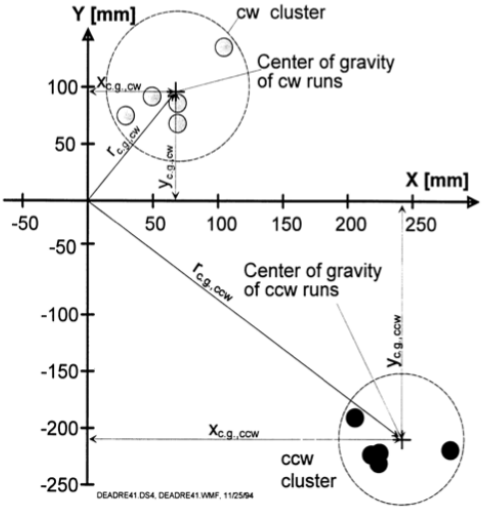

Figure 6: Typical results of the UMBmark test, show clustering of errors for both clockwise and anti-clockwise tests. Graph produced by (J.Borenstein & L.Feng 1994).

The clustering of the results for both clockwise and anti-clockwise stopping positions demonstrates that there are two distinct types systematic errors occurring in differential drive vehicles. The scattering of the individual points within the cluster is the result of non-systematic errors.

From the distribution of the clusters and individual points in this graph, it is shown that the contribution of systematic errors is greater than the contribution of non-systematic errors. In this particular example, the greatest systematic errors come from movement in the anti-clockwise direction.

2.4 Sensors & Visual Feedback

An important aspect of any robot is its ability to determine the state of the environment around it. There are a number of methods that can be employed and utilised in order to achieve this.

2.4.1 Range Sensors

One of the quickest and computationally cheapest methods of determining the robots position in relation to it's surroundings is by the use of range sensors. Range sensors work by determining the distance from the sensor to the object directly in front of it. These sensors vary in quality and accuracy based on their price.

There are 2 main types of range sensors.

- **IR Range Sensors**
  This type of sensor sends pulses of infrared light and based on the period of time it takes for the light to return it can determine the distance. A cheap and low-end range sensor in the £5 - £15 price bracket can be expected to be effective between 0.2m and 0.8m.

- **Ultra-Sonic Range Sensors**
  Ultra-Sonic sensors typically provide much better range that IR can offer, but due to the use of sound as a measuring device, it can not offer the same level as accuracy as IR sensors can.

2.4.2 SLAM

Simultaneous Localisation and Mapping or *SLAM* is a technique that allows the mapping of a physical environment through a selection of different input devices, whilst keeping track of its actual location as accurately as possible. SLAM is designed for use in unknown environments, and thus makes it an ideal candidate for collecting data for the project.

SLAM algorithms have utilised a number of different types of sensors in order to capture data that can then be used to determine the position of a vehicle. (H.Strasdat & W.Burgard 2007) states that some of these sensors include

- 1-Dimensional (single beam) or 2-Dimensional (sweeping) laser rangefinders.

* 3-Dimensional Flash LIDAR (Light Detection and Ranging), which is a process that is akin to radar, differing in the area of the electromagnet spectrum that they observe. LIDAR offers a very large detection range, spanning between 5cm to 5km.

* 3-Dimensional SONAR

* 2-Dimensional Cameras, such as webcams or other basic cameras.

Each of these input mediums carry advantages and disadvantages. Sensors such as LIDAR, whilst extremely accurate, do carry a price tag in excess of £5000. This is violates one of the goals of the project, by a considerable a margin, which is to determine the feasibility of robot that can map its environment without the requirement of a huge budget.

More in line with the minimal budget are sensors such as 1D laser rangefinders and 2D Cameras, and of these 2D Cameras provide more data and can be found within the same price bracket as laser rangefinders.

When implementing SLAM with information from a 2D Camera it becomes known as *VSLAM* or *Visual*SLAM. VisualSLAM uses imagery captured by the camera and identifies landmarks within its "sight" to help it build a map of its environment as well as to try and pinpoint its location.

VisualSLAM carries 2 major issues which if used must be addressed. The imagery captured from a 2D camera such as a webcam contains absolutely no information on distances. A landmark in the image might be a small toy 0.5m away or a massive tower 2km away. If the algorithm is to determine an object that is considerably distant from the robot then it will have difficultly in determining it's position. This is because of distant objects appearing to move slower than those objects that are less distant. This is known as *Parallax*.

The other issue is that VisualSLAM algorithms require substantial computational power in order to run. Assuming a basic RAW image is taken from the camera following a format where each pixel is stored as 3 bytes (RRGGBB) then a basic 640x480 pixel camera from a low-end camera will require

$$\begin{aligned} RAM_{usage} &= 640 \times 480 \times 3 \\ &= 921600 \\ &\approx 920KB \end{aligned}$$

In order to process this image in real-time the clock speed of the CPU will need to be sufficiently fast enough to be perform the required number of operations on a dataset of the above size in approximately 1 second.

3 Robot Hardware

3.1 Specification & Requirements

Through the course of researching different types of motion, sensors and other robots, an in-depth look at the advantages of each was overlooked. Before proceeding with any design, the advantages and disadvantages of each must be addressed.

One of the main considerations will be that of cost. As mentioned in the original Aims & Objectives, the goal of the project is to determine how accurate and reliable a robot that was been constructed on a very small budget can be.

3.1.1 Locomotion

In the background research, 3 main types of locomotion were looked at; Differential Drive, Car-like and Walking. Each of these types of locomotion has its own advantages and disadvantages.

	Differential Drive	Car-like	Walking
Motor Type	**DC Motor**	DC Motor & Servo	Servo
Number of Motors	**2 or more**	1 of each	Many
Turning Circle	**On the spot**	Large	**On the spot**
Mechanical Structures	**None**	Steering system	Legs
Minimum Wheels/Legs	**2 or more**	3 or more	**2 or more**
Obstacle Handling	Poor	Poor	***Can* step over**
Stability	**Good**	**Good**	Poor

Figure 7: This table shows different attributes to different forms of locomotion, and indicates which form has the biggest advantage.

Figure 7 lists a selection of considerations, along with typical values for each type of locomotion. From this figure we can see that Differential Drive offers us the best form of locomotion, whilst Car-like locomotion would be more complex, and lose the on the spot turning that Differential Drive offers. Walking based locomotion system have a large level of mechanical complexity and offer very little (and lose more) over Differential Drive.

Both Differential and Car-like locomotion have no means of dealing with obstacles once they meet them, short of turning and attempting to move around the object. Walking based locomotion can allow the robot to step/climb over the obstacle that blocks its path, however this is largely dependant on the size and shape of the obstacle as well as the strength of the servo's powering the robot and its stability.

This leads on to the next consideration of stability. All wheel and track based forms of locomotion such as, Differential Drive and Car-like steering are inherently stable due to their relatively low clearance from the ground and their wheels producing a large contact area on the ground. Walking robots do not have this inherent stability.

In order for a walking robot to be stable, its centre of mass needs to be lowered to the ground, in which case its ability to step over obstacles is diminished. If its centre of mass is left high above the ground then its ability to step over obstacles may be improved, but its stability is reduced.

The final consideration (as previously mentioned), cost is key factor in this project. The price brackets which I shall be working with are:

* Basic DC Motor < £10

* Basic Servo £10 - £20

Already its possible to see that any project involving servo's has the potential for higher cost, and considering the requirement of *many* servo's in walking locomotion we can immediately state that it violates the budget constraints. Differential Drive can be taken to be the cheapest (overall) and with the simplest setup.

Therefore the robot will use **Differential Drive**.

3.1.2 Sensors & Feedback

In order for the robot to be able to be aware of its surroundings and environment it will need to be a collection of sensors. In the background research I identified a number of different sensors. The table in figure 8 shows the characteristics of each of these sensors.

	Range	Connection	Computation Needs	Price Bracket
IR Range Sensor	0.08m - 0.80m	Analog 1 Pin	Very Low	£5 - £15
US Range Sensor	0.01m - 4.00m	Analog 2 Pins	Low	£10 - £20
LIDAR	0.05m - 100.0m	Ethernet	Moderate	£1000+
Web Cam	Line of Sight	USB	High	£5 - £30

Figure 8: This table compares a few characteristics of different types of sensor.

Immediately we can rule out LIDAR as an option due to its high price bracket, and violating the cost constraint of the project.

Range sensors such as Infra-red and Ultra-sonic both have the limitation of been single point sensors; that is if they detect that an object is 0.5m away, they have no way of knowing how wide or high that object is. If the object happens to be just out of the path of their detection beam then it will not be detected. This characteristic makes them ideal for last resort collision detection, but not for mapping out an environment.

This leaves Web Cams as the means of finding out about the environment. Web Cams provide a continuous stream of image data with can be analysed to determine objects in the immediate surroundings, through the use of Visual SLAM techniques. However it does have computational requirements and can not determine distances.

This will need to factored into how computation is handled by the robot.

Therefore the robot will use **Web Cams (Visual SLAM)**, along with an **IR Range Sensor** to assist with last resort collision detection.

3.1.3 Computation & Control

The most important aspect of the robot is its control systems. The bulk of the computation required will be carried out on a laptop computer which can be linked to robot through a physical connection and carried around after the robot. The sole reason for creating a physical link to a laptop is due to high computational requirements of processing and analysing image data.

The laptop computer can then issue commands to the robot, such as move forward, turn left, turn right and stop, based on the images it sees and map it builds.

However the motors and range sensors onboard the robot will require an intermediary step between them and the computer. The 'bridge' is required due to many of the components that reside on the robot directly using raw electrical signals which laptop computer do not offer ports for. An ideal bridge for this would be a prototyping system such as Arduino.

Arduino offers a built in USB connection to the laptop as well as an array of analog and digital pins which can be used to read sensor values and control motors.

3.2 Circuit Design & Layout

Using the factors decided in the specification & requirements the following components will be needed in order for the robot to correctly function.

- 2 DC Motors
- 1 H-Bridge Motor Controller (SN75441)
- 2 Batteries (At least 5V each)
  Alternatively 1 of the batteries can be substituted with the USB power when the Arduino is connected to a computer.
- 1 IR Range Sensor
- Arduino

The H-Bridge Motor Controller (SN75441) is required as it allows control of 2 DC Motors bidirectionally. It requires an additional power source to that of the Arduino board however (TexasInstruments 1986).

The H-Bridge Controller requires 6 direct digital connections to the Arduino (as shown in Figure 9). 3 of the connections control the left motor, and the other 3 control the right.

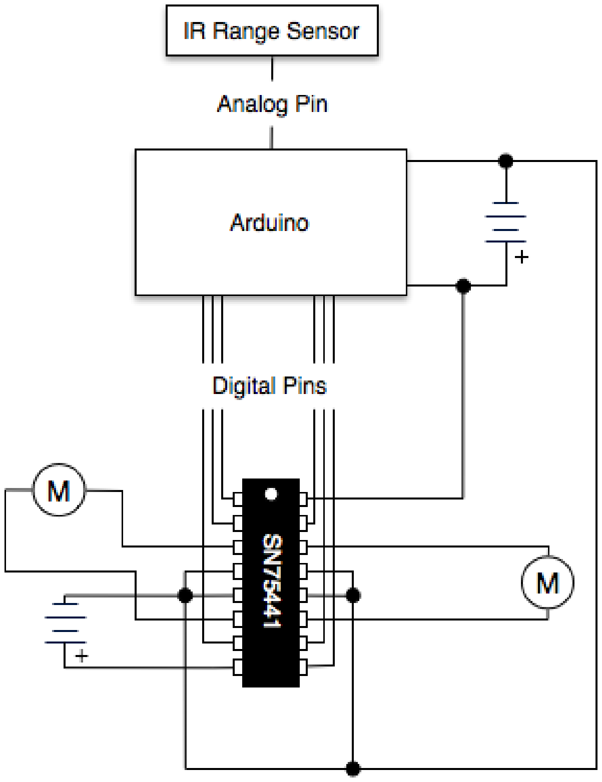

Figure 9: This circuit diagram illustrates the layout of the circuitry required on the actual robot.

Additionally the IR Range Sensor will require an analog connection to the Arduino. Figure 9 shows the circuit diagram for the robot. This diagram excludes any LED Power indicators that may be added to the robot.

3.3 Robot Chassis

In order for the robot to be able to do anything useful, it needs to have a rigid structure upon which its numerous components attach and support themselves; i.e. a chassis. To keep things simple and the costs low using a simple, single circular sheet (of perspex, metal or wood) will act as the sole aspect of the chassis.

Onto this the Arduino and Webcam can be mounted along with the motors and wheels. Finally any additional components such as batteries and stabilising components can then also be attached.

Figure 10 shows the basic concept and design of such a chassis, and keeps the components to scale. The positioning of the motors beneath the *plate* prevents the surface of it from becoming too crowded and unnecessary strain being placed on cables. Free space at either side of the Arduino and Breadboard allows room for additional components such as Webcams, Sensors and Batteries to be added without struggling to find room.

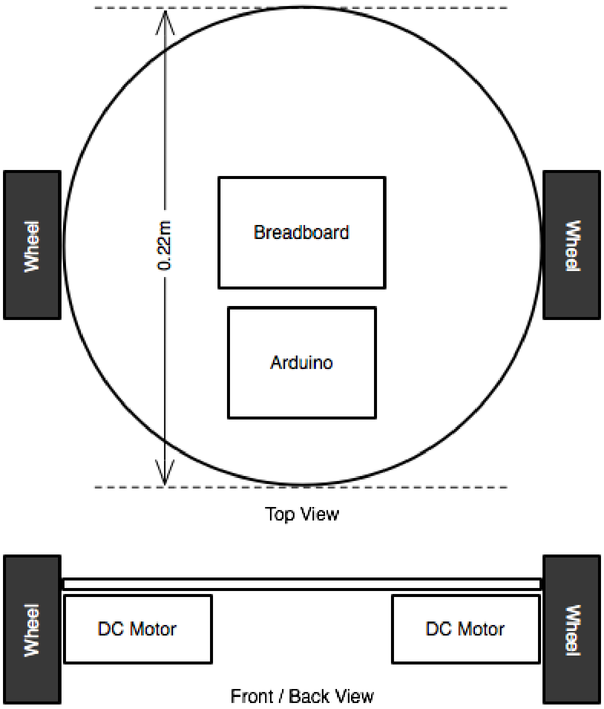

Figure 10: Basic design of the robot chassis, utilising a single 'plate' upon which components then attach.

3.4 Prototype

Before proceeding to develop the software for the robot it is necessary to determine if the basic circuits and hardware that has been designed actually works. By producing a small working prototype in which the motors can be controlled without constructing the full robot allows basic testing to find out if the circuits do what they are intended to do.

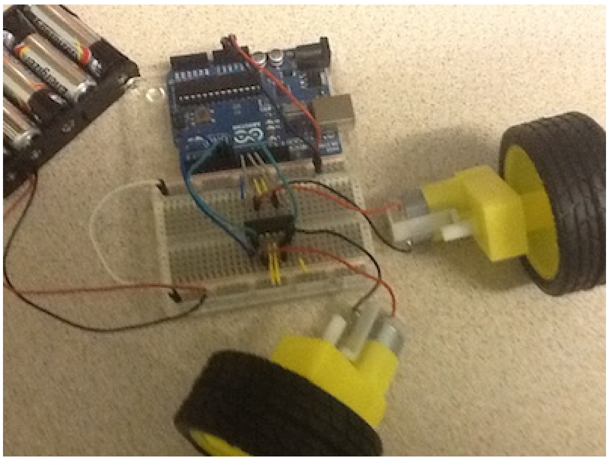

Figure 11: This a prototype of the designed circuit using the SN75441 H-Bridge chip With this both motors could be controlled at varying speeds and bidirectionally.

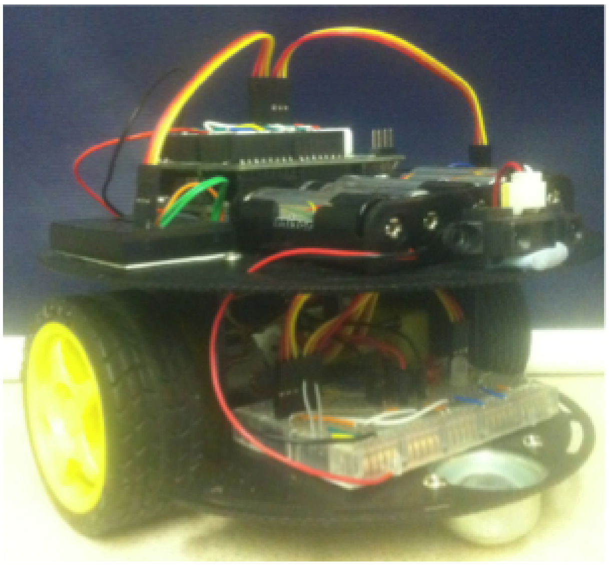

Figure 12: This a prototype of chassis with fully functional motor setup and IR Range Sensor.

4 Conclusion

With the completion of a working prototype of the motor control circuits the project can now move on to accomplishing its primary goal; exploring an unknown environment.

So far it has become apparent that although the concepts that will allow for exploration robots to be made cheaply, the accuracy of these robots is still very much a substantial barrier and one of the major reasons why such robots still have substantial budgets spent upon them.

One major factor in this is down to the quality and accuracy of the sensors used. There is a direct correlation between the size of the error margin on these and the price. Production of the chassis and implementing locomotion is relatively cheap, and is generally not viewed as a limiting factor.

Furthermore, computational power on a robot is very often limited when dealing with cheaper hardware. These cheaper Single Board Computers, such as the Arduino, do not offer high levels of precision, or the ability to analyse the high volumes of data that techniques such as Visual SLAM produce.

The next stages of the project will involve the construction of the robot, designing and developing the software systems and finally testing and evaluating the robot to determine its effectiveness and how well it meets the initial goals (if meets them at all).

References

AssociatedPress (2007), ‘Nasa extends mars rovers’ mission’.
URL: *http://on.msnbc.com/rNvhRn*

Brandon (2007), ‘Make a wall avoiding robot’.
URL: *http://www.societyofrobots.com/member_tutorials/node/45*

H.Strasdat, C.Stachniss, M. & W.Burgard (2007), Visual bearing-only simultaneous localization and mapping with improved feature matching, Technical report, Computer Science Institute, University of Freiburg, Germany. Last accessed: 15-12-2011.
URL: *http://www.doc.ic.ac.uk/ strasdat/website/strasdat07ams.pdf*

Inuktun (2011), ‘Versatrax 300’.
URL: *http://www.inuktun.com/crawler-vehicles/versatrax-300/VT300.pdf*

J.Borenstein & L.Feng (1994), Umbmark: A method for measuring, comparing and correcting dead-reckoning errors in mobile robots, Technical report, University of Michigan.
URL: *http://deepblue.lib.umich.edu/bitstream/2027.42/3753/5/bac6477.0001.001.pdf*

NASA (2004), ‘Mars exploration rover mission: Overview’.
URL: *http://marsrover.nasa.gov/overview/*

patrickmccb (2011), ‘Maze solving robot’.
URL: *http://www.instructables.com/id/Maze-Solving-Robot/*

Roston, G. P. & Krotkov, E. (1991), Dead reckoning navigation for walking robots”, Technical Report CMU-RI-TR-91-27, Robotics Institute, Pittsburgh, PA.
URL: *http://www.ri.cmu.edu/publication_view.html?pub_id=271*

Samsung (2011), ‘Navibot robotic cleaner’.
URL: *http://www.samsung.com/uk/consumer/home-appliances/vacuum-cleaner/robot/VCR8855L4B/XEU*

TexasInstruments (1986), ‘Sn754410ne - quadruple half-h driver - texas instruments’.
URL: *http://pdf1.alldatasheet.com/datasheet-pdf/view/28616/TI/SN754410NE.html*