



THE UNIVERSITY OF
AUCKLAND
Te Whare Wānanga o Tamaki Makaurau
NEW ZEALAND

Department of Mechanical Engineering

Mechatronics

Undergraduate Research Journal

Volume 7 - December 2014

Showcasing student research in the field of mechatronics



Contents

**RGB-D SLAM AND
REACTIVE NAVIGATION OF
AN OMNI-DIRECTIONAL
MOBILE ROBOT**
CHIEN-TANG TSENG.....1

**3D TERRAIN MAPPING
VEHICLE**
O. SINGH, E. TEH.....11

**ROBOTIC GAIT TRAINING
DEVICE FOR CHILDREN
WITH CEREBRAL PALSY**
CHATCHAWAN
LAKKHANANUKUN, JUN
YOUNG PARK.....26

*The University of Auckland
Undergraduate Mechatronics Research
Journal* is published annually by the
Mechatronics Engineering Group of the
Department of Mechanical Engineering,
located within:

Faculty of Engineering
University of Auckland
20 Symonds Street
Auckland

EDITOR-IN-CHIEF

Kean C. Aw
k.aw@auckland.ac.nz

EDITORS

Andrew McDaid
andrew.mcdaid@auckland.ac.nz

Shen Q. Xie
s.xie@auckland.ac.nz

Peter Xu
p.xu@auckland.a.nz

ISSN 1171-9494 © University of Auckland

Preface

Mechatronics research basically integrates mechanical, computer and electrical engineering for the innovation of smart devices, machines, systems and production processes. At the University of Auckland, we focus on systems design and integration using mechanisms, sensors, actuators, controls, computers and real-time software, with primary applications in healthcare, medicine, sports, manufacturing and agriculture.

The Bachelor of Engineering (Honours) programme in Mechatronics Engineering was launched in 2002. As part of the degree requirements, fourth year students are required to complete a Final Year Research Project (FYRP) in pairs.

FYRPs also represent our academic staff members' research activities. This journal, for students and by students, showcases a few sample FYRPs in summary. As contribution to the journal is voluntary, the publication is by no means representative of all student research programmes.

In addition to the BE (Hons), study in mechatronics engineering can also be pursued at postgraduate level through masters and PhD programmes. It is not uncommon for research topics studied in FYRPs to be further developed in postgraduate theses and beyond.

We hope you enjoy reading this issue.



Peter Xu
Professor & Chair of Mechatronics Engineering

RGB-D SLAM AND REACTIVE NAVIGATION OF AN OMNI-DIRECTIONAL MOBILE ROBOT

CHIEN-TANG TSENG

*Department of Mechatronics Engineering, University of Auckland,
Auckland, New Zealand[‡]*

The project was about implementing a navigation system into an Omni-directional vehicle with Mecanum wheels as an autonomous mobile robot. Software design and sensors fusion were the main objectives of this project, and Microsoft Kinect sensor was the core sensor for the robot. Three Kinect sensors are required to provide sufficient coverage to operate within the safety requirements but due to the time constraint, only one Kinect sensor was implemented in the system. By using API from Mobile Robot Programming toolkit, simple RGBD-SLAM was implemented. Also by using the API, a Reactive Navigation System has been implemented. In order to improve the reliability of the navigation system, path planning and virtual laser scan was implemented. The improvement had showed the robot was able to construct maps and avoided both static and dynamic obstacles. However, some teething problems such as frequent crashes and significant drift of the orientation readings were observed. Nevertheless, the 9 degree of freedom sensor had greatly increased its reliability and robustness.

Keywords: Mecanum; Omni-directional; mobile robot; MRPT; SLAM; Reactive Navigation; Kinect

1. Introduction

The Mecanum wheel (Fig. 1) is a wheel that has numbers of sub-wheels angle at 45 degree to its rotational axis. With four Mecanum wheels, combine individual motion on each wheel to result vehicle moving in any desire direction, in other words, Omni-direction.

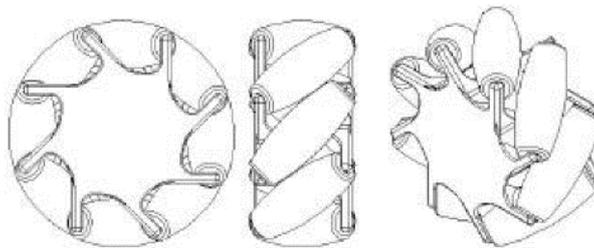


Fig. 1 Mecanum wheels [5]

2

The greatest benefit from Omni-direction Mecanum wheels is the vehicle does not require to be steered to change its direction. This allows the robot to be agile within constrained space. There were few applications that had already benefited from the advantage such as automated guided Mecanum wheel forklift that is able to carry heavy goods and to transfer in a space limited warehouse.

2. Project overview and objectives

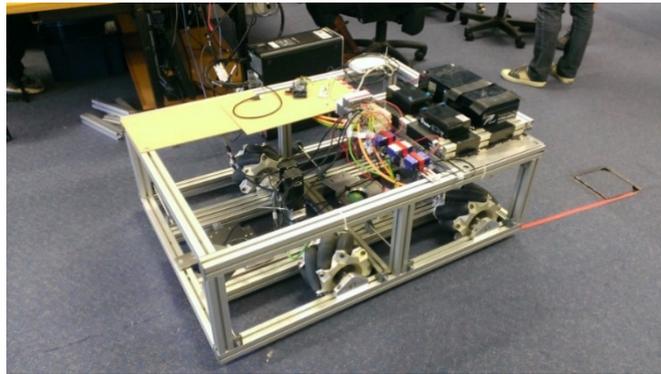


Fig. 2 Mecanum wheels Omni-directional mobile robot

A heavy duty Mecanum wheels omni-directional mobile robot was available at the beginning of this project (Fig. 2) [1]. It had a programmable logic controller (PLC) with 4 encoders, 2 track balls and a 9 degree of freedom sensor. The objective of this project was to develop an autonomous mobile robot. The critical objectives were to implement simultaneous localization and mapping (SLAM) algorithm and intelligent navigation system. The robot would need to avoid any collisions and to navigate through the environment. By achieving this, multiple sensor fusion and real-time software will be require. The design process was broken down into several stages.

- Define current problems
- Operating system choosing and sensor choosing
- RGBD-SLAM and Navigation software integration
- Experiment test for navigation
- Improving safety and robustness

This project will be focusing on SLAM algorithm implementation, navigation system implementation and sensor fusion into this robot.

3. 9 Degree of freedom sensor (9DOF sensor)

MPU-9150 is a 9DOF sensor that featuring 3 axes gyro, 3 axes accelerometer and 3 axes magnetometer. A 9DOF sensor is required to measure the orientation of the robot together with the gyro and magnetometer. Unfortunately, few problems were found. First, the micro-controller Arduino Mega that reads sensor values from 9DOF cause frequent

software crashes and stop working. The reason was found that the I²C communication between Arduino and 9DOF was very unstable. Because the Arduino logic level needs less than 3 volts as logic high, then the 3.3 volts I²C component is like living at the edge and hoping that the 3.3 volts will be recognized as a logic high. Making sure that 3.3 volt will rise fast, both SDA and SCL have a pull-up resistor. According to the datasheet [2] 2.2K ohms was used in this setup to make sure the voltage will rise fast enough to 3.3 volt (Fig. 3)

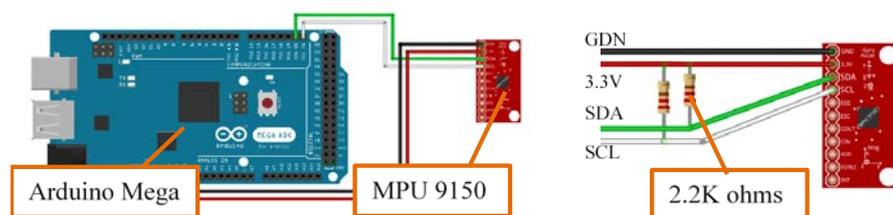


Fig. 3 left, Standard connection of I2C; right, I2C connection with pull-up resistors

Because the rate of I²C communication is about 400 kHz, it is very high frequency and very sensitive to magnetic flux. The magnetic flux from motors can easily interfere with the connectivity. This is a common problem for high frequency communication through electric wires. To overcome this, a ferrite bead was added to the I²C wires (Fig. 4) and suppressed the interference to the wires. After adding the ferrite bead, the signals became smoother and more reliable.



Fig. 4 Ferrite Bead

Also because the orientation measurement was originally determined with a gyro and magnetometer, the magnetic fluxes from the motors will also cause interference to the sensor readings. By modifying the Arduino's software, the orientation measurement was now purely based on the gyro.

4. Sensor and Operating System

In this project, creating a high quality map was essential for developing navigation system. Because of the limited fund, this project could not use an expensive laser

scanner. Instead, a Microsoft Kinect sensor was used in this project. Kinect is a powerful sensor that contain a color image camera with depth image sensor (RGB-D sensor), and it is 10 times cheaper than laser scanner.

Beside the sensor, a development platform needs to be decided to develop navigation software. Because Kinect sensor was chosen, it will not work with the PLC and needs dedicated computer operating system (OS). Two OS that had been considered, Microsoft Windows, and Robot Operating System (ROS) based on Linux. Although ROS is specific for robotics developing and research [3], the lack of dedicated computer in this project made using ROS to be relatively difficult. Therefore due to resources limitation, Windows had choose to be the platform in this project. To maintain the cross platform capability in future, Mobile Robot Programming Toolkit (MRPT) [4] was used to implement Simultaneously Localization and Mapping (SLAM) algorithm and navigation system.

5. Kinect SLAM implementation

Using the libraries provided by MRPT to quickly grab data from Kinect sensor. The features of color image were tracked with integrated API and depth information was added to each features. This converts 2D features into 3D features and tracked in 3D space to achieve 6DOF SLAM with a Kinect sensor. (Fig. 5)

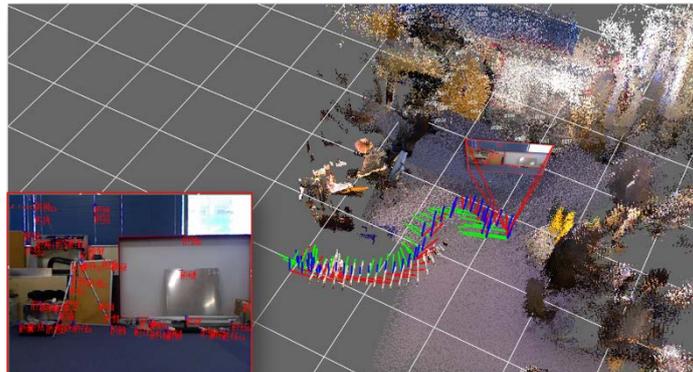


Fig. 5 Result of Kinect 3D SLAM

This algorithm has several limitations. When the sensor observes a view that is out of its maximum depth range, the 2D feature cannot be recreated in 3D space and unable to distinguish the relative position from last key-frame (new key-frame will be create when relative position was exceeding the threshold). In other words, the odometry will be lost when directing the sensor to a scene that is too far. Another limitation was the algorithm needs sufficient number of features to successfully compute the odometry information. If the sensor moved too fast and possibly caused blurry image, the number of features will drop significantly. Then the odometry information will be lost because there is insufficient number of features to track.

6. Reactive navigation implementation

MRPT provided an API to integrate Reactive Navigation System in order to control the robot and to avoid obstacles. However, the performance was not satisfactory because the Kinect's field of view was too narrow. This caused the robot to be unable to distinguish the obstacles that were out of view and collided with them (Fig. 6).

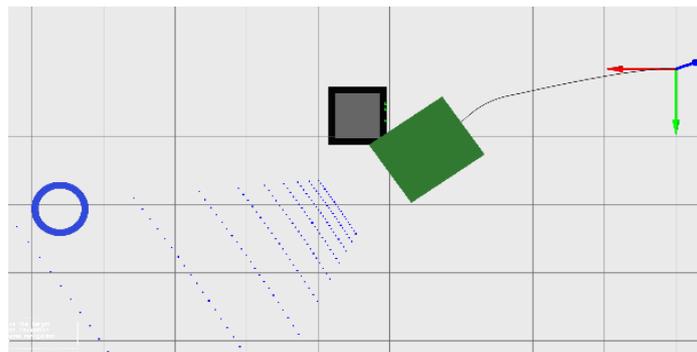


Fig. 6 Collision between robot and obstacle

6.1. Kinect 2D Scan

Instead of 3D mapping, the robot only moves within 2D. Hence, only 2D grid mapping is needed for this application. In future implementation, it will be a great advantage to use internal API to integrate new capability like ICP SLAM and grid matching. With MRPT, it supports fast conversion 3D observations into 2D laser observations (Fig. 7) and construct 2D grid map with it.

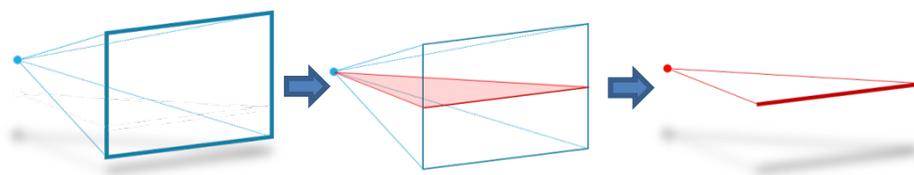


Fig. 7 Process of 3D Kinect observation to 2D laser scan observation

6.2. Path Planning

Reactive Navigation was not the complete solution for robot navigation. The purpose of reactive navigation is to allow robot to explore unknown environment and adapt to its dynamic change. It is very common when a navigation system was unable to pass

through the door and reach the target behind the wall. This was because the algorithm only reacts to real-time sensor observation.

There are two benefits from using path planning API to generate a path. It provided a path with a step size allowing navigation system to follow and remains reactive to any dynamic environmental change. Because in path planning the robot size was considered, the navigation system will not drive the robot into corners of the wall by following the path (Fig. 8). Path planning API had a limitation. It needs a known environment to be able to compute the path

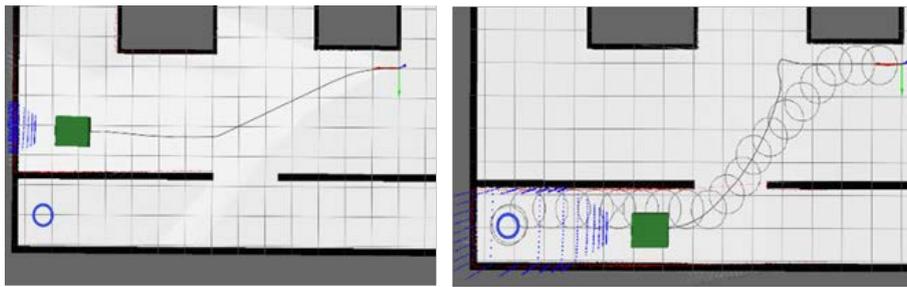


Fig. 8 left, before path planning implemented; right, after path planning implemented

6.3. *Virtual Laser Scan*

The performance of Reactive Navigation System with single Kinect sensors was very poor due to the insufficient coverage of the scene. Because the system only reacts to real-time sensor observation, a virtual laser scan was simulated from 2D grid map and fed into the navigation system. The result shows the navigation system has successfully avoided collision with the corner wall in both known and unknown environment (Fig. 9).

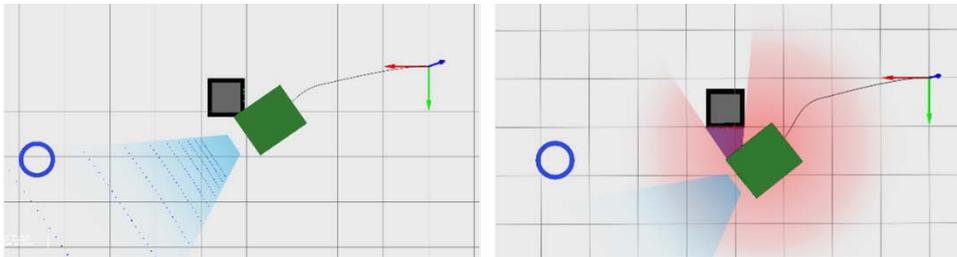


Fig. 9 left, before virtual laser scan; right, after virtual laser scan

7. Experiment

7.1. *Static obstacle avoidance test*

A test was performed in order to test the obstacles avoidance capability in practice. Two panels were set within the environment allowing sensor to pick up the image easily (Fig. 10 top, marked with red boxes). The result shows the robot has successfully avoided any obstacles and the target was reached (Fig. 10, bottom).

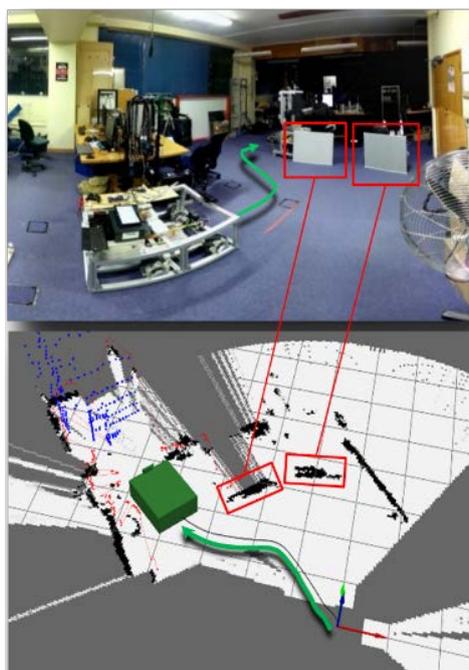


Fig. 10 Static obstacle avoidance test in mechatronic lab

7.2. *Dynamic environment test*

This test was to perform a real-time navigation in a dynamic environment. The test started with a target set right ahead of the robot to accomplish a forward motion. During the forward motion, an un-expected obstacle will interrupt the robot. As the result, a real-time reaction responded to the dynamic obstacle. Both the dynamic obstacle and static obstacles were successfully avoided (Fig. 11).

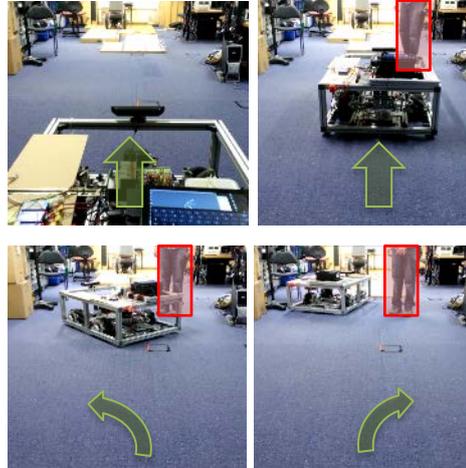


Fig. 11 Sequence of pictures show dynamic obstacle avoidance

7.3. *Narrow environment test*

A narrow environment was created by placing obstacles that surround the robot. This is to test the robot's ability to turn around within a small and limited space (Fig. 12). As the size of the lab was very limited and safety issues need to be considered, the obstacles were manually scanned first before running the navigation system. Orientation shifting was observed in this test. When the target was set right behind the robot, the robot was expected to make a 180 degree turn back. In practice, the robot over turned and ended with in an undesired position. Further tuning was needed. The result shows the robot has successfully avoided all the obstacles during the test (Fig. 12).

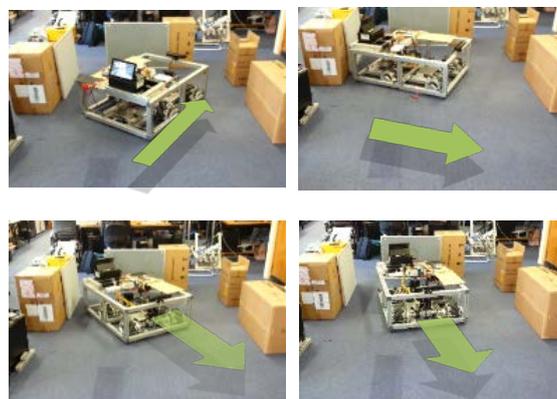


Fig. 12 Sequence of pictures show robot able to turn around from obstacles surrounded environment

8. Discussion

In this project, only one Kinect sensor was implemented due to time constraint. Thus the algorithm was able to avoid static obstacles stored in the map, but the sensor placement made the scanning very inefficient. Fig. 13 (left) illustrates the valid area of Kinect scanning. When the robot was exploring within a new environment and banking right with a trajectory (blue dash line) for obstacle avoidance, and potentially, a new obstacle with same trajectory (red dash line) will be missed out by the Kinect and collide with robot. There are several possible solutions. Implementing multiple sensors or the use of laser scanner can mitigate this problem. Alternatively, by restricting the kinematics to collaborate with the sensor's limitation, or reverse the turning sensor's heading to match the kinematic of the robot can also be considered. The easiest way is to re-allocate the sensor to the rear of the robot (Fig. 13Right).

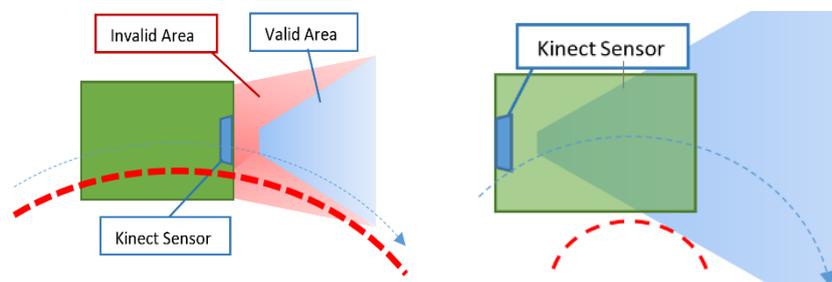


Fig. 13 Difference in placement of Kinect sensor

The navigation system in current platform has only one dimensional speed that drives robot forward and a single angular speed that steer the robot. The system did not exploit the benefit of the Mecanum wheel. Also gaining an extra degree of freedom to control the robot will directly affect the mapping capability and mobility.

Currently the project only use Kinect 3D SLAM algorithm to produce the odometry information. As a result, it was not very satisfactory. After mapping domain was transformed from 3D to 2D grid map, it directly supports ICP (Iterative Closest Point) SLAM API provided by MRPT. The algorithm aims to find the best transformation and reduces the distance to the minimum between two points. Because Kinect 3D SLAM often loses its odometry information and have serious error accumulation problem, implementing ICP SLAM could improve the odometry and reliability of the system.

9. Conclusion

- Microsoft Kinect sensor was choose for this project as it is affordable.
- TCP/IP Terminal Application was develop with joystick control integrated with it.
- A simple RGBD-SLAM has been implemented into the system.
- A better RGBD-SLAM, RTAB-Map had yet to be implemented.
- Reactive Navigation System has been implemented.

- The 9DOF sensor improved the system's reliability.
- The Reactive Navigation System had improved the reliability by adding path planning and virtual laser scan.

10. Acknowledgements

I would like to thank the following people who provided their assistance in this project. Professor Peter Xu, who as a project supervisor gave peerless guidance. Martin, who provided his support on connection issues between ROS and Kinect. Lab technicians Logan Stuart and Hossien who gave their advice and provided hardware support and resources. Finally to my project co-workers, Theng Kiat Chua and Christain Scheifele who also contributed significantly to this project.

References

1. L. Xie, "Mechatronics Design and Heavy-Duty Omni-Directional Mecanum Mobile Robot for Research Purpose," University of Auckland, Auckland, 2013.
2. Microchip, "Microchip 24AA512/24LC512/24FC512 512K I2C Serial EEPROM," 5 1 2010. [Online]. Available: <http://www.farnell.com/datasheets/575340.pdf>.
3. www.ros.org, "ROS.org | Powering the world's robots," www.ros.org, 7 12 2013. [Online]. Available: www.ros.org. [Accessed 21 3 2014].
4. mrpt.org, "MRPT | Empowering C++ development in robotics," 2014. [Online]. Available: <http://www.mrpt.org/>. [Accessed 12 4 2014].
5. L. Doroftei, V. Grosu and V. Spinu, "Omnidirectional Mobile Robot – Design and Implementation," Technical University of Iasi, Iași, 2007.

3D TERRAIN MAPPING VEHICLE

O. SINGH, E. TEH

*Department of Mechanical Engineering, The University of Auckland, 20 Symonds Street,
Auckland, 1010, New Zealand*

The emergence of urban search and rescue robotics as a field of research is largely attributed to the failure of robots in the field thus far. They fail to fulfill the task of finding survivors because they are expensive, specialized, and require a very long lead time, but they also cannot localize once a survivor is found. This project seeks to remedy the issues. An untethered ground vehicle with a multitude of sensors to map its path in 3D, coupled with a PC interface for facilitating two-way communication with the on board Raspberry Pi micro-computer has been developed. All communication is real-time and implemented through the Transmission Control Protocol over a Wi-Fi connection. Although sensor fusion and filters have not been extensively developed or tested, the primary aim of demonstrating the unification of various quantities to produce a 3D model has been fulfilled. Dead reckoning has been used as the technique to process proprioceptive sensor data, meaning the system is not reliant on external support to produce the models. Simple exponential and complementary filters clean the incoming sensor data of any anomalies or noise sufficiently to be able to demonstrate this ability. In fact, noise is attenuated three-fold while moving.

Keywords: 3D mapping; unmanned ground vehicle, search and rescue.

1. Introduction

The tragic loss of life in the Great Hanshin Earthquake in Japan and the Oklahoma City Bombings, both in 1995, heralded a new era for Urban Search and Rescue (USAR) (1). Over the past twenty years, countless research labs devoted time and resources into specialist robotics research for USAR applications. On an international level, two mobile robot competitions were conceived to engage the extended scientific community in USAR robotics research (2).

The input by the Center for Robot-Assisted Search and Rescue (CRASAR) during the 2001 World Trade Center disaster response marked the first time robots had been used for USAR (3).

The last few years has seen more and more disaster response teams using unmanned robots in efforts to find survivors, but to date, only remains have been found (4), and frequently the robots have failed for various reasons. Thus, there is still a long way to go before USAR robots prove their worth and are fully accepted in the community.

This paper details the prototyping of a simple 3D terrain mapping vehicle that is capable of mapping its own path using proprioceptive sensing. The path is then later presented as a 3D CAD model.

2. Overall System

Shown in Fig. 1 is the prototype developed in this project. The four wheeled vehicle uses a pre-assembled chassis with a Raspberry Pi as the micro-controller. Navigation and mapping were made possible with a Raspberry Pi camera, IMU, and motor shaft encoders. Two 16.8V batteries were used to power the electronics and four motors separately. A regulator and power distribution board from Homer's and Tinwal's project (5, 6) was used to step-down voltage from the battery to the micro-computer and the other electronics. A USB Wi-Fi dongle was used to establish a reliable connection, which enables a PC to interface with the vehicle in real time.

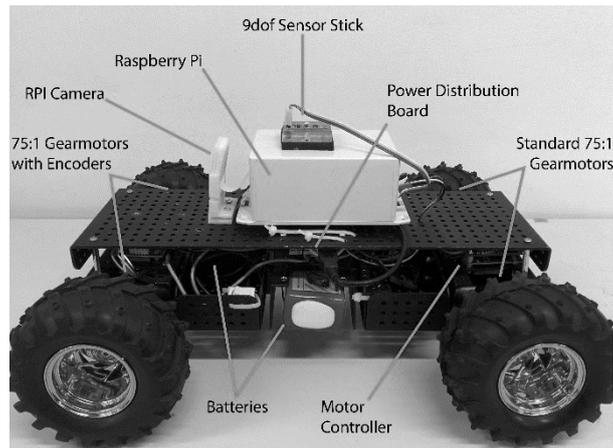


Fig. 1. The vehicle and components used in developing the prototype of a 3D terrain mapping vehicle.

2.1. Hardware architecture

The vehicle system contributes a large portion of the system's hardware architecture. Fig. 2 shows the components used in building this vehicle and the corresponding connections.

2.2. Software architecture

The software architecture of the entire system comprises of dedicated software written in Python and C# for the vehicle and the PC respectively. The applications were configured in a master/slave configuration which allows the master PC to control the slave vehicle, while receiving live data and video feed. The Raspberry Pi transmits filtered sensor data at a constant time interval to the master PC, where the data are post-processed and integrated to produce a 3D model of the path. Both application utilised multithreading to enable the processors to perform concurrent execution of a given number of tasks. An overview of the system's software architecture is shown in Fig. 3.

3. Data Acquisition

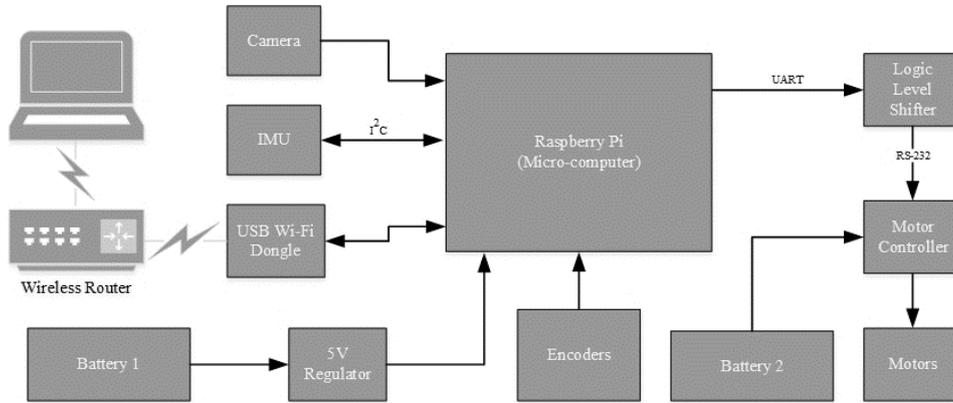


Fig. 2. System's hardware architecture.

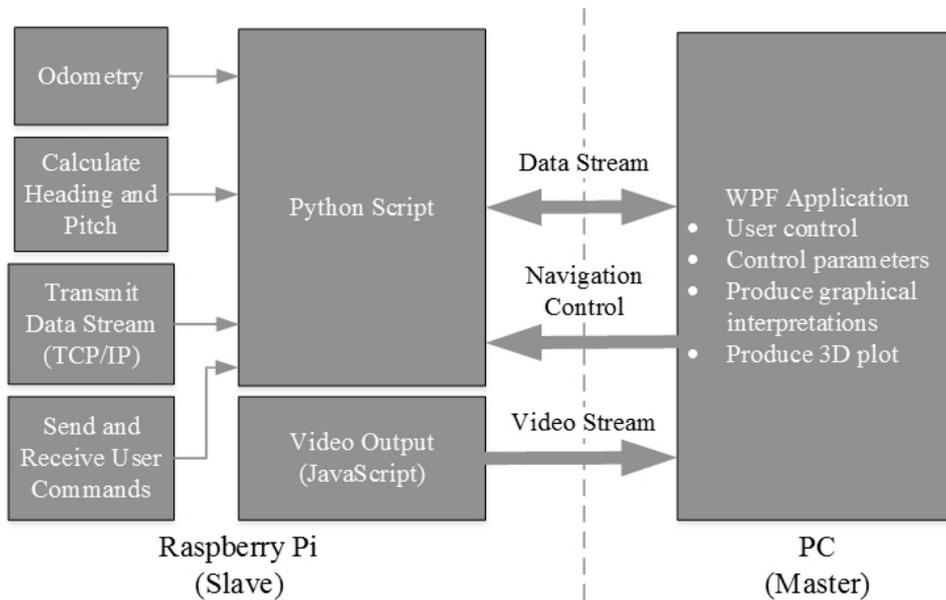


Fig. 3. System's software architecture.

Only proprioceptive sensors are used because GPS and other common tracking techniques are unsuitable for search and rescue scenarios (eg. underground). Magnetic wheel encoders track linear movement, while a 9 degree of freedom (DOF) inertial measurement unit (IMU) determines the instantaneous pose.

The IMU is susceptible to electrical and mechanical noise, thus all readings are filtered using an exponential low pass filter. A complementary filter combines the fast

response of a gyroscope and stability of an accelerometer to accurately detect pose (more specifically the tilt and roll). Meanwhile the magnetometer was used for acquiring the heading of the vehicle.

4. Dead Reckoning

In order for the vehicle to map its relative path, it has to be aware of its location relative to its starting point in the world coordinate system. The amount of data transmitted by the vehicle were kept to a minimum for maximum data transfer rate, while providing essential data for the position and orientation estimation. These data are constantly received by the back end interface of the PC application, hence allowing a real time estimate of the vehicle's position and orientation.

The position of the vehicle travelling from point A to B can be determined using simple trigonometry. The estimated pitch angle (θ_{est}), compensated heading (ψ'), and relative distance travelled (R), were expressed mathematically with the following equations. R is the projected distance of R onto the $X - Y$ plane in Fig. 4c, while P_x , P_y and P_z are the distance vectors of the vehicle at point B relative to A in the 3D space, $P = [P_x \ P_y \ P_z]^T$.

$$r' = r \cos \theta_{est}. \quad (1)$$

$$P_z = r \sin \theta_{est}. \quad (2)$$

$$P_y = r' \cos \psi'. \quad (3)$$

$$P_x = r' \sin \psi'. \quad (4)$$

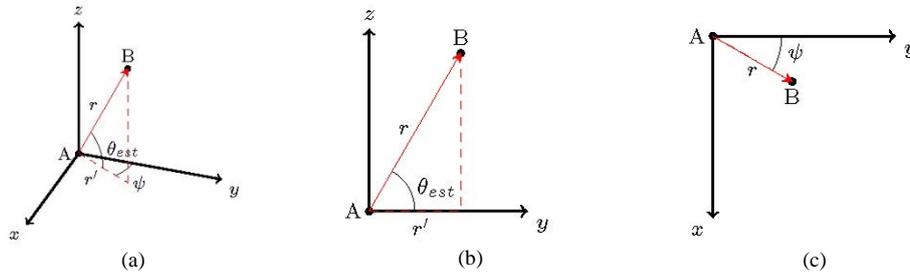


Fig. 4. Position of the vehicle at point B relative to A; (a) Three dimensional view; (b)Z plane; (c) Projection of R on the X-Y plane.

Once these distance vectors were determined, the following equation was used for updating the vehicle's relative position, P_{new} .

$$P_{new} = P_{old} + P = \begin{bmatrix} P_{old,x} \\ P_{old,y} \\ P_{old,z} \end{bmatrix} + \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}. \quad (5)$$

5. Path Mapping Performance Results and Discussion

Using dead reckoning, there are three quantities which combine to give a position in space. These are odometry (distance moved), heading (angle respective to north), and pitch (the inclination). Each aspect is tested independently and a discussion of experimental procedure with results follows.

5.1. Odometry performance

The odometry results outlined in this section simply assess the performance of the wheel encoders in determining position/distance when driving in a straight line:

- on a flat surface
- up an incline (6° , 12° , 18°)
- down an incline (6° , 12° , 18°)

A cruise controller was also developed to ensure that the vehicle travels at an almost constant speed. Therefore the tests for driving up and down an incline are performed with and without the aid of the cruise controller.

5.1.1. Experimental procedure

Fig. 5 shows the experimental setup. The aim is to determine whether or not, regardless of inclination, the measured distance (through wheel encoders) and actual distance (measured physically) varies no more than ± 50 mm/m. This is an approximation for a reasonable error per meter of forwards movement, as when looking at the scale of a building or mine, it would be unnoticeable.

Summarizing experiment details:

- 2m long ramp with 6° , 12° , 18° slope set-points
- Rough surface to represent dirt/gravel/carpet
- Multiple tests of one configuration to validate results

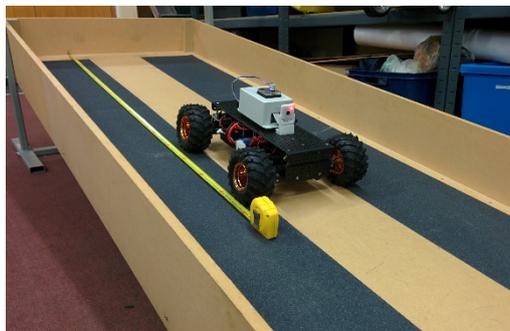


Fig. 5. Test ramp.

5.1.2. Results

For a base scenario, multiple unaided (no cruise controller) runs on a flat surface (carpet) were executed, and the distance traveled measured. The two black dashed lines in Fig. 6 visually show the $\pm 50\text{mm/m}$ error bounds. The X axis shows the distance measured by the Raspberry Pi and wheel encoders, while the error for the Y axis is calculated by comparing the measured distance to the actual distance.

The results in Fig. 6 verify the assumption of a maximum error of $\pm 50\text{mm/m}$, and the general trend shows that the error will not significantly multiply with increased travel distance.

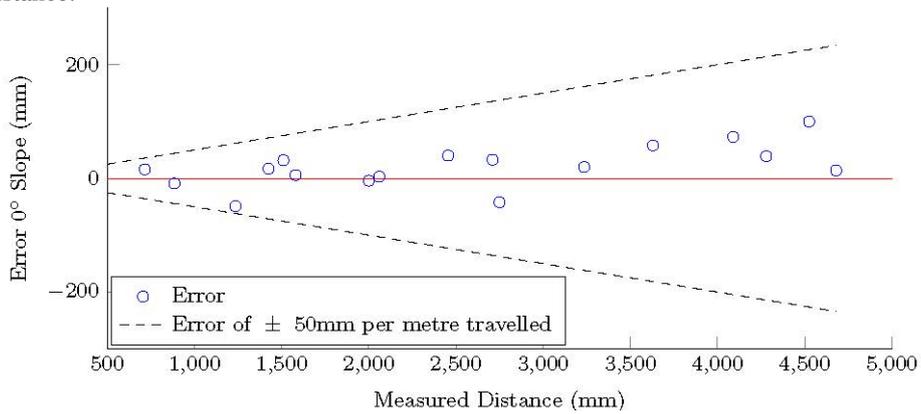


Fig. 6. Measured distance along a flat surface compared to actual distance traveled.

The results for each run driving up and down each of the inclines (angles 6° , 12° , or 18°) are presented in Fig. 7. In each plot, the dashed black lines represent the $\pm 50\text{mm/m}$ error bounds. Initial tests were executed without the use of a cruise controller, allowing gravity to have full effect on the vehicle as it drives up and down the slope, to gauge whether or not a cruise controller is even needed.

The red and blue data points represent up and down respectively. Observing all three plots (7a, 7b and 7c), traveling down a 6° incline consistently produces an error over double the threshold of 50mm . As this is a positive error, it indicates the vehicle believes it has traveled further than it really has. Similarly, although not present on a 6° incline, an out of bounds error is also present while the vehicles drives up an incline. A negative error here indicates the vehicle believes it has traveled less than it really has.

From these results, one can only expect the error to accumulate rather quickly, through combinations of flat and inclined sections of the route. This justifies the need for a cruise controller to eliminate or reduce the error integrating over distance.

Again, observing the plots in Fig. 7, however this time looking at the green and light blue data points (representing down and up respectively), it can be seen that they are consistently within the error bounds for both traveling up and down the slope. This shows the profound effect a cruise controller has on the overall accuracy of the data gathered.

5.2. Orientation performance

The orientation of the vehicle can be described by its pitch, roll, and heading. However, the roll does not affect the overall path mapped, hence its performance was not quantified.

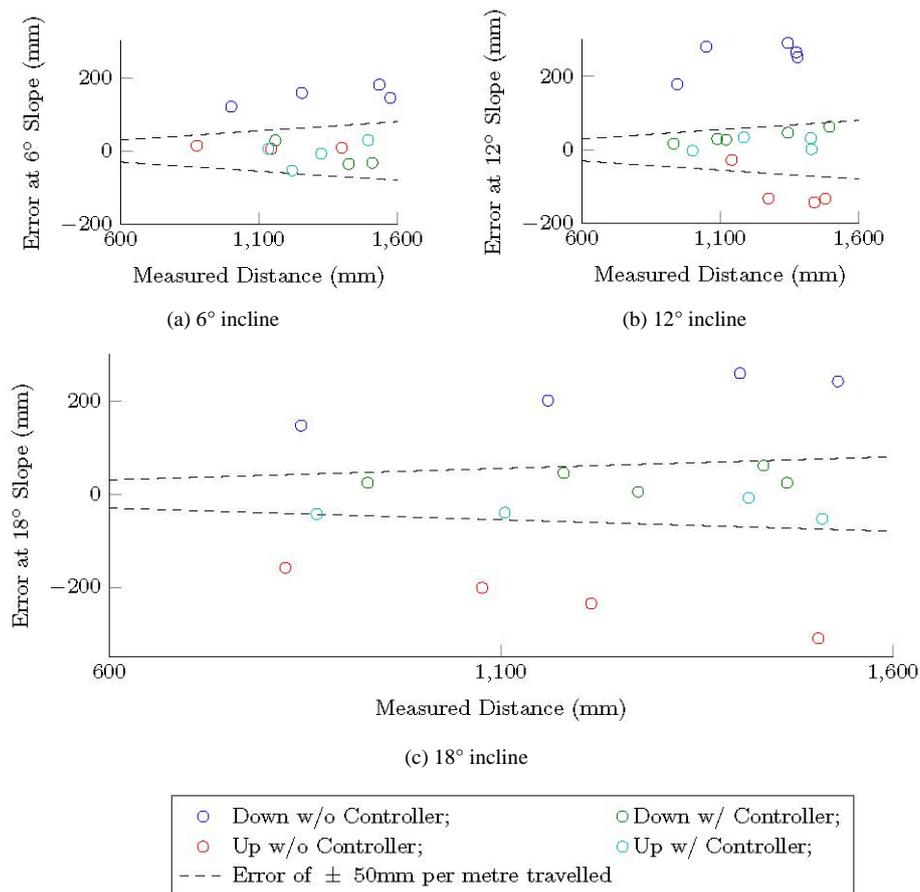


Fig. 7. Measured distance along various slopes compared to actual distance traveled.

5.2.1. Pitch

The estimated pitch is a combined result of the exponential and complementary filters. The accuracy and reliability of the estimated pitch angle affects the resulting elevation of the constructed path. Two separate experiments were implemented to test for its static and dynamic responses with five sets of data each. The first experiment tests for the responsiveness of the individual filters, from there the gains for each filter is determined.

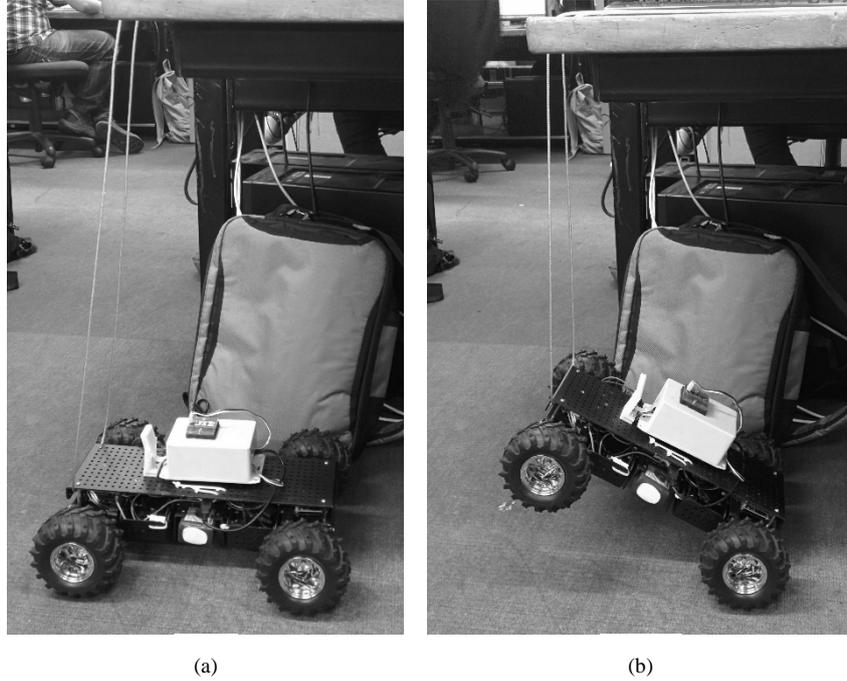


Fig. 8. Test setup for static pitch angle response.

The responsiveness of the filters can be measured by the rise time of the response. A response with a longer rise time will have little to no overshoot, thus providing a good level of noise reduction. However, a filter gain that is too large will result in a loss in accuracy and responsiveness. Individually, these rise times are expected to be significantly lower than when the filters are combined. Thus the chosen filter gains must have both a short rise time and reasonable noise rejection.

The sensor was first held on a levelled surface as depicted in Fig. 8a, then it's tilted at an angle of roughly 40° instantaneously to simulate a step input, as shown in Fig. 8b. This was repeated for a range of complementary gain (α) and exponential gain (β).

To visualize this better, the data was collected and represented as a box plot as shown in Fig. 9. Low mean rise time was found to have little to no noise rejection to the filtered response, especially with the β gain. However, a longer rise time would significantly slow the response time of the filter. Thus a compromise of $\alpha = 0.55$ and $\beta = 0.6$ was

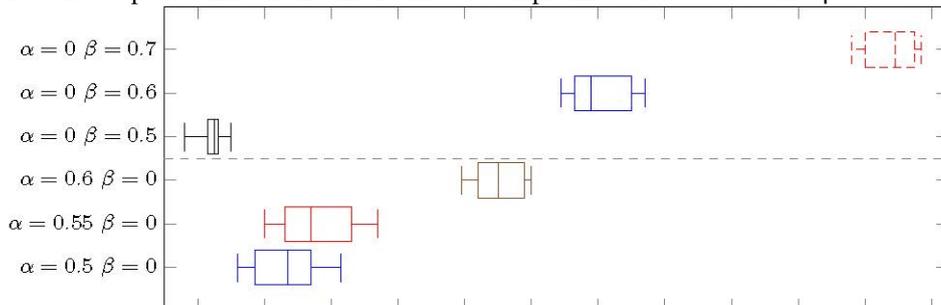


Fig. 9. Rise time comparison between individual α and β gain values.

chosen.

The combined $\alpha = 0.55$ and $\beta = 0.6$ result in a mean rise time of 0.684 s, with a typical response shown in Fig. 10. This increased in rise time was the result of combining the complementary and exponential filters. In addition, with the moving speed of the vehicle, the latency of the filtered response is still small enough to be unnoticeable.

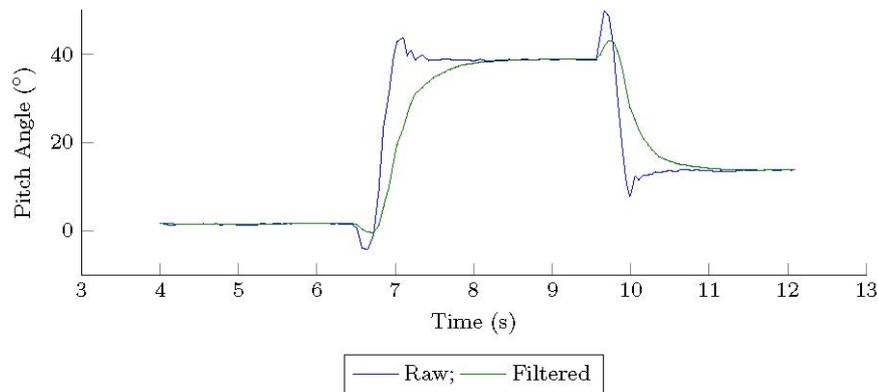


Fig. 10. Typical static pitch response with combined exponential and complimentary filter; $\alpha = 0.55$ and $\beta = 0.6$.

In the second test, the dynamic test ensures that the filters are able to perform with vibrational noise introduced while the vehicle is driven along a straight line, around 3 to 4 m in length. The raw and filtered pitch angles were then plotted together producing Fig. 11. Two large spikes in the raw pitch angle can be seen at the start and end of the figure, which indicate when the vehicle started and stopped. On the contrary, the filtered pitch angle response is clearly less affected by the sudden change in angle, as well as the high frequency noises. The filtered response rejects half the noise that was present in the raw pitch angle. This was measured using the signal to noise ratio shown in Table 1. The signal to noise ratio is the ratio of the mean and standard deviation of the response. The

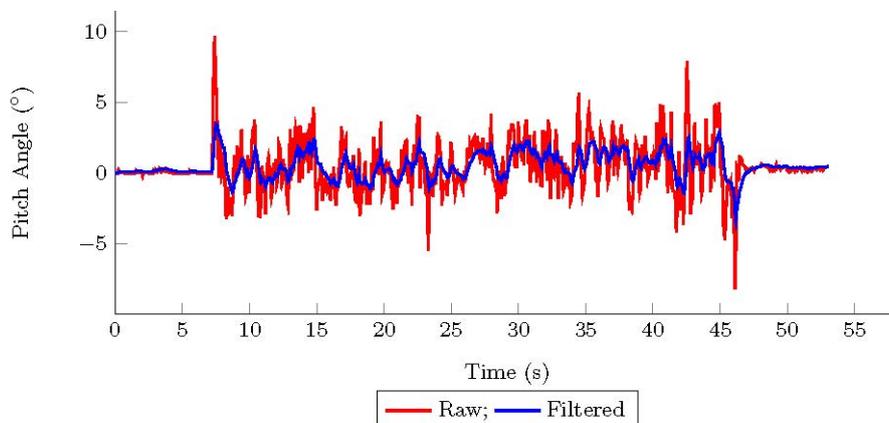


Fig. 11. Typical dynamic response of the raw and filtered pitch angles.

maximum and minimum measured angle of the response describe the amplitude of the noise. Again the amplitude of the filtered response was at least 2.5 times less than the raw response.

Dynamic Response	$\alpha = 0, \beta = 0$	$\alpha = 0.55, \beta = 0.6$
Mean angle	0.42	0.51
Minimum angle	9.65	3.46
Maximum angle	-8.19	-3.32
Standard deviation	1.69	0.857
Signal to noise ration	0.247	0.593

Table 1. Summary of the dynamic response of the raw and filtered pitch angles

5.2.2. Heading

The heading was simply tested by doing multiple calibrated and uncalibrated runs, driving in a straight line, in different areas of a room. It can be observed that although calibration essentially rotates the tracked path by a certain calibration quantity around the origin, the shape of the path remains the same.

Calibration can remove the soft and hard iron effects locally, but as the vehicle moves into a slightly different environment, the calibration will have less of an effect (and may even contribute in distorting the results further). As the tests were executed in a lab

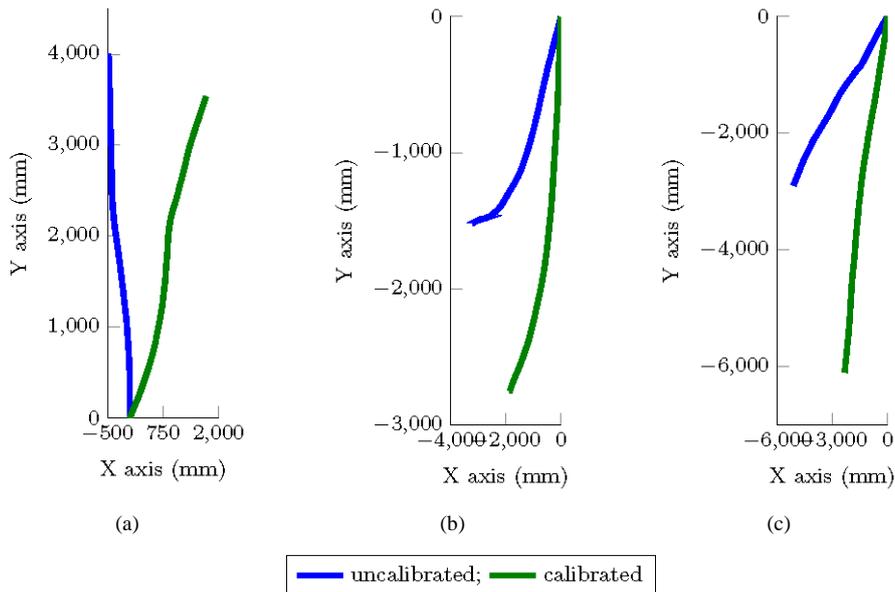
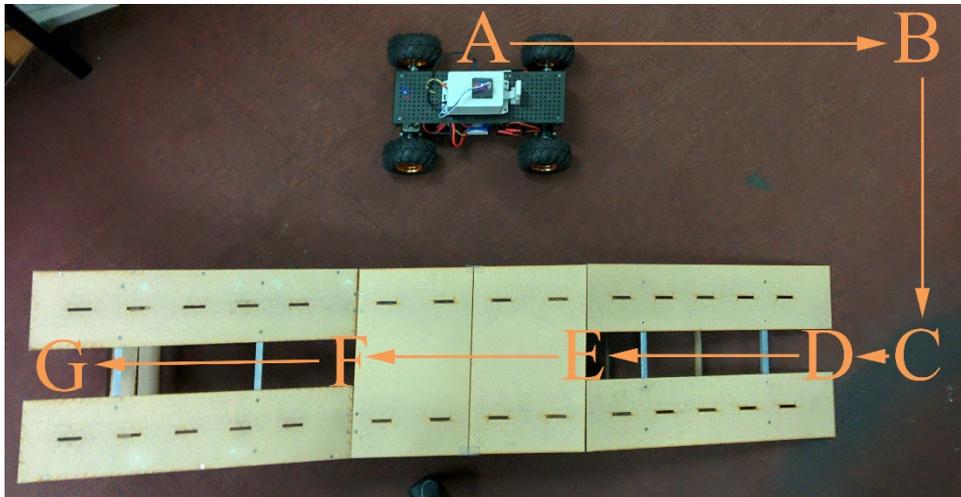


Fig. 12. Paths mapped before and after the magnetometer was calibrated.

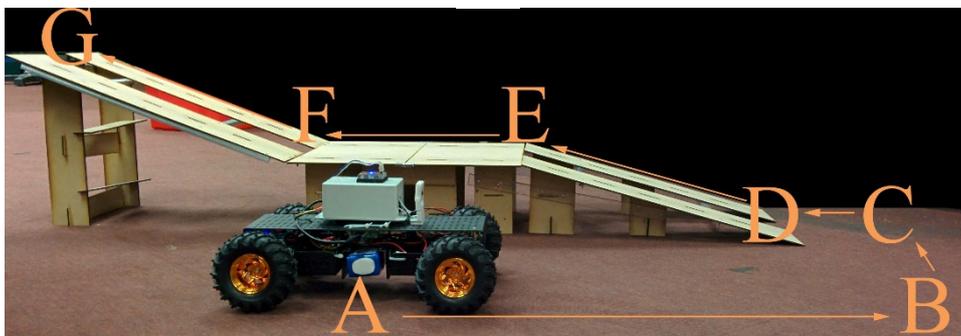
environment with many metallic objects such as tables and shelves, the effects of the distorted magnetic field around the robot are very prominent in Fig. 12a and Fig. 12b. In each case, the robot is driving straight, however the distorted magnetic field distorts the results. Evidently, the experiment for Fig. 12c was executed in a more suitable environment where interference with the magnetic field was at a minimum, although the hard iron effects are still present.

Unless operating in an environment with minimal magnetic field distortion, solely using a magnetometer for accurate heading calculation is unreliable and inaccurate. However, this issue has not been addressed to a greater extent simply because the vehicle is still a prototype, and the improvement of sensing techniques and filters is outside the scope of this project. In saying that however, the filtering and calibration implemented thus far is required to demonstrate the mapping capability of this vehicle.

5.3. Combined 3D path



(a)



(b)

Fig. 13. Actual test setup course with points A to G labelled; (a) Top view; (b) Side view.

The following test aims to demonstrate the overall performances of the combined measurements of the odometry, pitch, and heading. The vehicle is driven on a test course from point A to G shown in Fig. 13. Three test runs were conducted with the vehicle pointing due East, so that the path driven are in the plane of X-Z. This was done so that the paths of the test runs can be compared and analysed with the expected path shown in Fig. 14. Prior to this, the magnetometer was calibrated to ensure that any offsets and distortion in the magnetic field were removed. Markers were placed at point A, B, C, and G on the test course so that the results obtained are consistent.

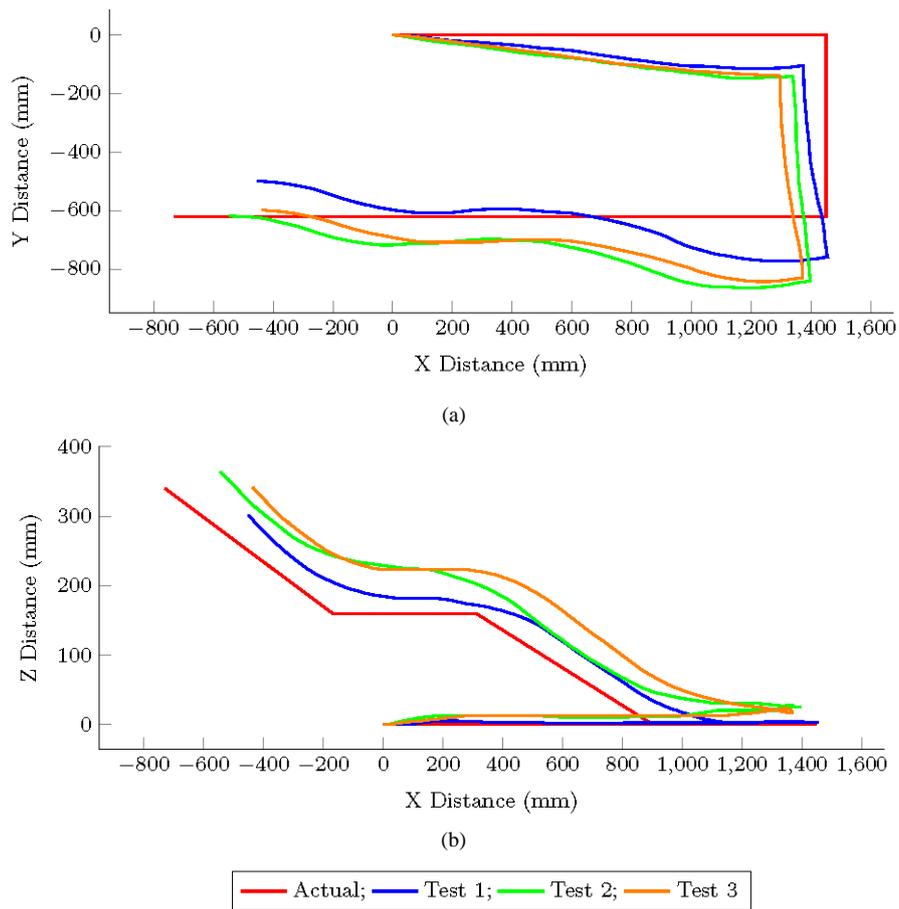


Fig. 14. Comparison between the actual path and the paths from the three test runs plotted on: (a) X-Yaxes (Top view); (b) X-Z axes (Side view)

The X-Y plot (top view) in Fig. 14a further demonstrated the issues of the magnetometer for measuring the heading of the vehicle. All three test runs displayed similar curves in the top view, which further indicate the consistent interferences due to the presence of external sources, most notably between points A to B and C to G.

Similarly the paths from the test runs shown in Fig. 14b, demonstrated that the accelerometer and odometry were able to track the elevation and the slope of the ramp. The vehicle experienced a gradual change in angle, as opposed to an immediate and sudden change, thus the corners at point D, E, and F had larger radii. This was caused by the use large wheels and the long length of the chassis. The halfway point of these arcs were used to determine the points D, E, and F.

Furthermore, the path in the X-Z plot, were also affected by the heading because the paths travelled between A to B and C to G were no longer in parallel to the plane. The interferences on the magnetometer had also caused the path between C to G to become wavy, thus the overall accuracy of the combined performance could not be commented on with these results.

Nonetheless, the estimated paths from all three test runs displayed the shape of the test course consistently. Since the paths were constructed through a series of points relative to each other, the error of the measurements were cumulative. The individual distances between two points in the X, Y, and Z direction were summarised in Table 2.

Overall, the developed prototype has shown its ability to track its path and produce a 3D model, as specified in the objectives of the project. However, these findings had demonstrated that the simple method of using the magnetometer alone for measuring the heading was unreliable, thus affecting the overall accuracy and reliability of the 3D path produced.

Table 2. Absolute difference in measured distances along different points in the X, Y, and Z directions.

Test Runs	ΔX (mm)			ΔY (mm)		ΔZ (mm)	
	A-B	C-D	D-G	B-C	Ground-E	Ground-G	
Actual	1450	550	1630	620	163	340	
Test 1	1447	411	1463	654	165	301	
Test 2	1390	397	1539	698	214	360	
Test 3	1372	343	1466	676	200	340	

6. Conclusions

The following conclusions can be drawn from this project:

- An inexpensive, fully off-the-shelf prototype vehicle capable of mapping its path over a terrain in 3D has been developed.
- It was discovered that many urban search and rescue robots are still unreliable prototypes and they have never found a living survivor, but more importantly, they cannot localize to tell rescuers exactly where they are.
- The physical vehicle is an overhaul of past years' systems. A Raspberry Pi microcomputer is used, magnetic encoders have been employed.
- A fully functional, standalone custom PC application was developed. A video stream provides perspective, arrow keys provide control, inclinometer and heading gauges

provide the vehicle status and pose, and two 2D maps provide a localization method all in real-time.

- Fully untethered communications through Wi-Fi.
- Automated generation and running of AutoCAD scripts to produce a 3D CAD model. Fully controlled by the PC application until the model is ready for manipulation. Point-of-interest markers as set by the user are also generated.
- Filtering and fusion of sensors has been implemented although the unreliability of the system can be attributed to this.

7. Future Work

From the results presented in this article, this prototype has the potential in search and rescue operations or the mining industry. However, in order to fully develop this project into a working product, further improvements and future work can be carried out:

- To develop an alternative method of measuring the heading as well as an improved sensor filtering methods used in this project so that more accurate and reliable 3D path maps can be produced.
- To develop a fully automated UGV so that the vehicle is fully aware of its surroundings and map a path given a set area.
- To combine point cloud technology to reproduce texture of the surroundings into a CAD model along with its path. Thus allow the user to visualise the surroundings of the path driven.
- To develop a solution for long range wireless transmission to increase the allowable travelling distance of the vehicle. Such as wireless transmission with lower bandwidth frequency (larger wavelength) or a series of wireless repeaters to extend the overall transmission range.

Acknowledgments

Many people were involved with making this project a success, and the authors would like to recognize them here.

Dr. Kean Aw and Dr. Kevin Wang as supervisors have been very supportive throughout the year, and their advice has been invaluable - thank you very much.

Last but not least, the unmatched technical assistance and expertise of Logan Stuart and Ken Snow has also been incredibly useful.

References

1. Davids, A. (2002). Urban search and rescue robots: from tragedy to technology. *Intelligent Systems, IEEE*, 17(2), pp. 81–83.
2. Siciliano, B., and Khatib, O. (2008) 'Search and Rescue Robotics.' In: *Springer Handbook of Robotics*, Springer, Berlin, pp. 1151 - 1173.
3. Murphy, R. R. (2004). Trial by fire [rescue robots]. *Robotics and Automation Magazine, IEEE*, 11(3), pp. 50–61.
4. Murphy, R. R. (2012). A decade of rescue robots. *Intelligent Robots and Systems (IROS)*, pp. 5448–5449.
5. Homer, N. (2013). *The Autonomous Terrain Profiling Vehicle*. Project iv report, Department of Mechanical Engineering, University of Auckland.
6. Tinwala, F. (2013). *The Autonomous Terrain Profiling Vehicle*. Project iv report, Department of Mechanical Engineering, University of Auckland.

ROBOTIC GAIT TRAINING DEVICE FOR CHILDREN WITH CEREBRAL PALSY

CHATCHAWAN LAKKHANANUKUN, JUN YOUNG PARK

*Mechanical Engineering Department, University of Auckland, Symonds St,
Auckland, 1010*

This report outlines the mechanical, software and control design of a new gait rehabilitation device (GRD) which aims to help rehabilitate and correct the gait patterns of children affected by Cerebral Palsy (CP).

An H-frame, timing-belt driven, XY positioning slide rail system has been designed and built. The end-effector consists of an orthotic foot brace attached to the slide rail via a force sensor. The system is mobile and will operate over-ground.

A mathematical model of the gait cycle as measured from the ankle was developed on MATLAB. Using this model, a LabVIEW program was created to control two sets of actuator motors based on position, force and impedance control systems. Different sensors were used for each control strategy. Magnetic sensors were used to measure the end effector position and a 2-dimensional force sensor was used to calculate the forces applied by the device user's leg.

Initial testing of the position control system was conducted on normal gait participants and the force and impedance control systems were tested by the researchers. The results gathered from this testing were used to further improve the system.

Keywords: Gait Rehabilitation Device (GRD), cerebral palsy, control systems

1. Introduction

Cerebral Palsy (CP) is the most common form of disability in early childhood, occurring in approximately 3 out of 1000 live births [1]. In the absence of therapy and appropriate interventions, the impact that it has on the functional abilities will severely limit the potential development of the child. Those affected by CP, are more likely to achieve independence in walking through high intensity, task-specific and long-term rehabilitation [2]. The need for an effective gait rehabilitation device for children is therefore critical for giving them the best possible chance to develop into independent adults.

Gait rehabilitation devices currently out in the market focus on a stationary, treadmill-based form, such as the Lokomat and LOPES [3, 4]. Although some gait trainers have been designed for over-ground use such as the Ohio State Gait Trainer [5] and the Therasuit [6], these trainers do not implement any form of systematic control on the subjects lower limbs, hence they do not contribute much

in terms of corrective gait training, merely restricting movement and providing body weight support (BWS).

The aim of this project is to develop a new method of gait rehabilitation in children with CP and to implement this into a fully functioning device. The resulting system will be an over-ground device with adaptive control over the subject's lower limbs.

2. Design

2.1. Frame

The system, shown in Figure 1, is an H-frame, parallel, two-axis positioning system consisting of two rotary drives, eight pulleys and an elastic timing belt. The frame was constructed using MiniTec EN-AW-6060 T6 aluminum extrusions. The structural integrity of the proposed final frame design was tested using ANSYS as shown in Figure 2 below. The frame material, which has a yield stress of 190MPa, has a maximum stress of 14.8MPa induced on it when a 50Kg load is placed on either sides of the top of the frame. A ministry of health study showed that the average weight of a New Zealand child between the ages of 5-14 was 40Kg [7]. The maximum stress induced in the frame is an order of magnitude below the yield point. This confirms that the frame will be more than capable of withstanding the loads placed on it during use.

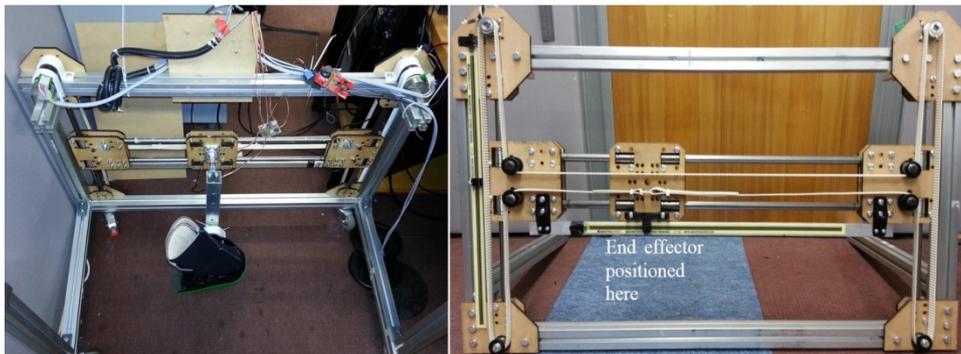


Figure 1 - Gait Rehabilitation Device (GRD) inside view (left), H-frame (right)

2.2. Transmission System and Actuators

Two Maxon brushless DC motors were used as actuators. The motors simultaneously rotate the timing belt in order to create motion at the end effector position. The motors had gearheads attached in order to reduce the high rotational speed to a more usable speed. The gear reductions in the gearheads also allowed the motors to have higher torque so that it met the requirements for moving a child's leg. An inverse kinematic model shown in Equation 1 was created that could determine the angular

displacement (ϕ_1, ϕ_2) each motor would need to rotate in order to displace the end-effector by a set trajectory (x, y) . The pulley radius (r) of the motors is a factor used in the model.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}r & -\frac{1}{2}r \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \end{bmatrix} \quad [1]$$

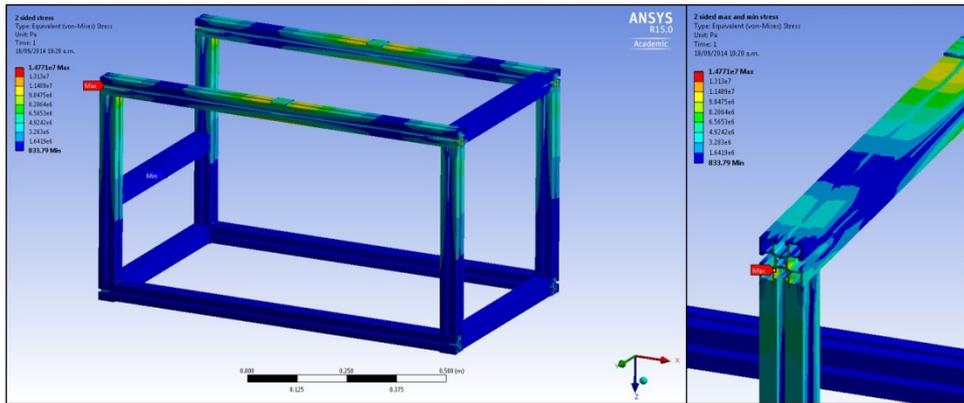


Figure 2 - ANSYS stress distribution on frame, maximum stress location (right)

2.3. Sensors

2.3.1. Position Sensors

Position is measured by two Magnetopot magnetic sensors; one in the x/horizontal direction and the other in the y/vertical direction. The benefit of this position tracking system is that it works in a contactless manner, hence reducing the frictional wear from prolonged use. In order to prevent the magnetic sensors from giving unstable position readings due to noise, 2nd-order Bessel filters were implemented in order to filter out voltage readings and place saturation limits on the reading values.

2.3.2. Force Sensors

Two sets of load cells are used to measure the force applied at the foot-brace in both horizontal and vertical directions. The design of the force sensor on which the load cells were mounted, was tested on ANSYS. The results below show that with 1000N of force

acting on the load cell, the maximum shear strain was approximately 0.001, which was deemed to be more than sufficient for use on the GRD

3. Control System

Figure 3 below shows the entire control framework of the GRD system. The control system can be divided into three groups: position control, force control and impedance control. These control methods were used to create both circular and gait motions. The speed controller uses the proportional-integral (PI) control method to apply Pulse Width Modulation (PWM) to the actuators. The entire control framework was implemented using a LabVIEW program running on the myRIO controller.

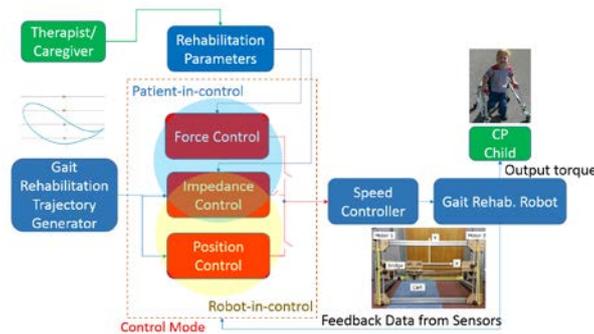


Figure 3 Control Framework

3.1. Position Control

Position modelling is based on the inverse kinematic model (equation 1) shown in the transmission section above. The magnetic sensors are used to measure the current position as well as the desired final position of the end-effector. This data is then used to calculate the required speed and angular displacements of each motor. The disadvantage of the position controller is that it is entirely “robot-in-control”, meaning that the patient does not have any control over the motions created.

3.1.1. Gait Motion

A kinematic model of gait motion was derived and used to create a model of ankle position during a gait cycle as shown in Figure 4. This model took thigh and shank lengths of the user as inputs, as well as the cycle time for one gait cycle. Due to the dimensional limits of the machine created by the limited size of the frame, the system is unable to create a gait motion for any participant with a total thigh and shank length longer than 400mm. This limitation will need to be addressed in future work on the device. The slide rail frame will need to be larger in order to fit the gait cycle of taller children. This gait motion was then mapped into LabVIEW and used as an array of desired points to control the position of the foot.

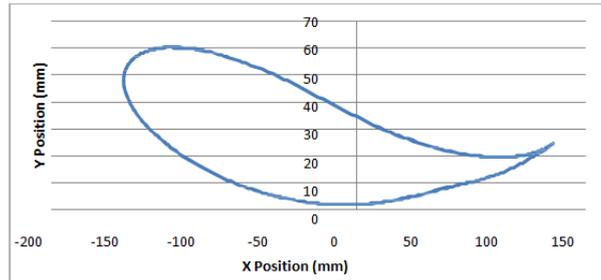


Figure 4 Gait Cycle (Thigh length = 200mm, Shank length = 150mm)

3.2. Force Control

A back-drivable, patient-controlled, force control system was created in order to allow the patient to back-drive the system's end effector to any desired position. This control system utilizes the force-feedback from the load cells to maintain a position at which the resultant force is zero. However, the force control system does not allow for any circular or gait motion control.

3.3. Impedance Control

Impedance control incorporates both the force and position control parameters in order to eliminate the disadvantages of each method. This control mode allows both the patient and the robot to have partial control of the GRD. This will allow the patient to retain a certain degree of control over the motion while the robot controlled system moves their leg in a gait motion. The key benefit of this system is that the stiffness settings of the force feedback can be adjusted in order to provide a transition from high assistance to low assistance for the patient as their training progresses. This also allows for a variety of training difficulty levels for all patients.

3.4. Safety Measures

In order to prevent the end-effector from colliding with the frame, dimensional limits were placed on the system using the magnetic sensors. The readings from the sensors are checked continuously and should they reach the boundary limit values, the system will stop the motors and return the cart back to a safe position before continuing again on its path.

4. Results

4.1. Ethics Approval

In order to test the gait rehabilitation device on children, certain standards for safety and ethics set out by the University of Auckland Human Participants Ethics

Committee (UAHPEC) had to be met. An Ethics Summary was written up, outlining the details of the project, as well as Participant Information Sheet and Consent forms for both the child and the parent/caregiver. Application for Ethics Approval (ref: 012051) was approved on 18 June 2014 by the UAHPEC for a period of three years. This allowed testing of the device to commence in the second half of the year.

4.2. Position Control

Position control was tested on two healthy children (8 years old, 127cm tall, 23Kg; 11 years old, 143cm tall, 32Kg) with normal gait, on 24 August 2014. Overall, the system performed well, moving in accurate circular and gait motions, as well as being strong enough to take the load of the child's body weight.

Circular motion and gait motion was run at cycle times of 30 seconds and 50 second respectively, for each of the participants. The results are shown in Figure 5.

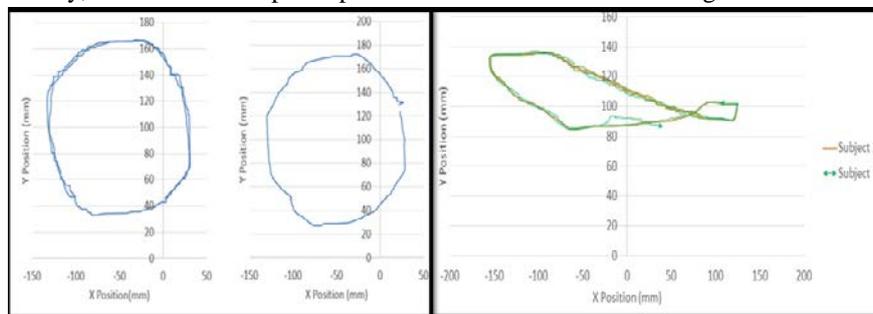


Figure 5 - Circular Motion (left), Gait Motion (right) for both subjects

4.3. Force Control

Force control was tested by the researchers. A circular motion and gait motion was manually driven by the user, with the force feedback allowing the end effector to follow the leg's motion. As shown below in Figure 6 the circular and gait motions were achieved with relative ease utilizing the force-feedback controller.

4.4. Impedance Control

Impedance control was also tested by the researchers. A desired position was set for both x and y coordinates, to which the end effector was driven. The end-effector was then displaced away from the desired position by the researchers in order to see how the system would track back to the original desired position. The results are shown in Figure 7.

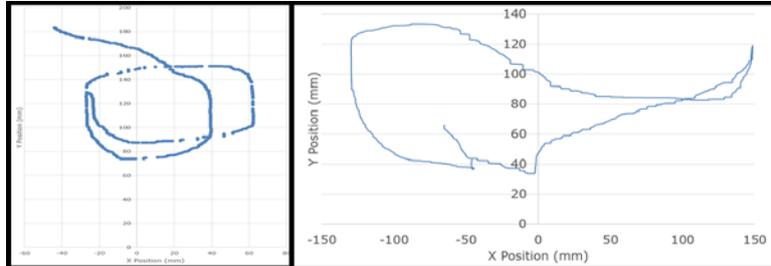


Figure 6 Circular Motion (left) and Gait Motion (right) for Force Controller

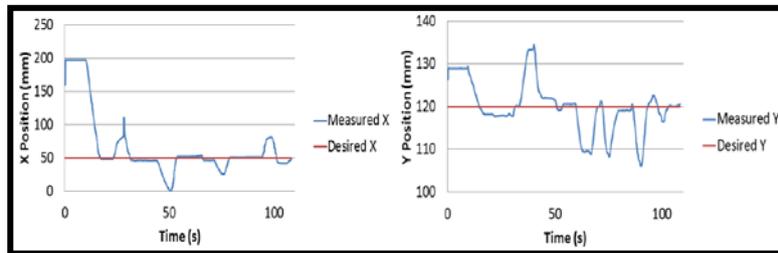


Figure 7 Impedance Control Testing for Horizontal and Vertical Motion

5. Discussions

5.1. Controller Performance

The performances of all three controllers were acceptable. The gait motion and circular motion in position control followed the original motion with minimal deviations. However, the speed of the system will have to be increased significantly in order for it to simulate the gait motion at a true walking speed. This will be discussed further in the future improvements section.

5.2. Future Improvements

5.2.1. Frame Size Increase

The current MiniTec frame dimensions for the slide rail are not large enough to map a gait trajectory for a larger child. The maximum thigh + shank length allowed for the current setup is 400mm. The frame will need to be extended along with the belt in order for the system to be used by taller children.

5.2.2. Motor Upgrade

As mentioned above, although the current Maxon brushless EC flat motors (part number 251601) generate enough torque to move the leg, they do not have the required speed to generate a gait cycle at a realistic walking speed (5 km/hr). New motors with an after gear reduction speed of 1020 rpm are required in order to produce a gait cycle at this desired walking speed.

6.0 Conclusion

- The final Mechanical design of the system consists of a two-axis, H-frame positioning system with two motors, an elastic timing belt for transmission and an end effector system.
- The end effector consists of an orthotic foot brace attached to the slide rail via load cells. The system is mobile and the end effector will operate over-ground.
- Position Control was implemented successfully and tested on normal gait children.
- Force and impedance controls were implemented successfully and tested by the researchers.

Acknowledgments

- Dr Andrew McDaid for his guidance and support throughout the project
- Logan Stuart for assistance in the lab
- Trial participants for position control (anonymous)

References

1. Beckung, E. and G. Hagberg, Neuroimpairments, activity limitations, and participation restrictions in children with cerebral palsy. *Dev Med Child Neurol*, 2002. **44**: p. 309 - 316.
2. Song, R., et al., *Myoelectrically controlled wrist robot for stroke rehabilitation*. *Journal of NeuroEngineering and Rehabilitation*, 2013. **10**(1).
3. *Hocoma: Lokomat Pro*. Retrieved 6 July, 2014 from http://www.hocoma.com/fileadmin/user/Dokumente/Lokomat/bro_L6_130225_en_A4.pdf.
4. *University of Twente: LOPES*. Retrieved 6 July, 2014 from <http://www.utwente.nl/ctw/bw/RESEARCH/PROJECTS/LOPES/INDEX.HTML>.
5. The Ohio State University College of Engineering: Cerebral Palsy Project. Retrieved March 10, 2014 from <https://engineering.osu.edu/news/2010/07/engineering-occupational-therapy-students-join-forces-cerebral-palsy-project>.
6. TheraSuit LLC:Services. Retrieved 6 July, 2014 from <http://www.suittherapy.com/pdf%20training/2011%20catalog%2002%2001%20web.pdf>.
7. Ministry of Health NZ: Food and Nutrition Guidelines for Healthy Children and Young People. Retrieved 1 July, 2014 from <http://www.health.govt.nz/system/files/documents/publications/food-and-nutrition-guidelines-for-healthy-children-and-young-people-p3.pdf>.