Wireless Control of Seekur Jr Using Smartphone Application

Alec T. Beerens*

University of New South Wales at the Australian Defence Force Academy.

The application of robotics as research platforms is becoming more useful. These robots can be controlled by physical tethers, wireless capability or autonomously. The technology of mobile devices such as smart-phones is becoming increasingly popular and easier to use. This includes the open-source development of applications for various functions such as remote controls. This report looks at the project of developing a smart-phone application to be integrated with the Seekur Jr mobile robot to control its movements using wireless communication in order to provide a cheap, portable, producible product that can be used for future development of the Seekur Jr platforms. The report details achievements of the project and the future work which may be done to further develop the project.

Contents

*SBLT, School of Engineering and Information Technology, ZEIT4500

Nomenclature

OS	Operating System
DGPS	Differential Global Positioning System
GPS	Global Positioning System
AVD	Android Virtual Devices
SIG	Special Interest Group
IEEE	Institute of Electrical and Electronics Engineers
WECA	Wireless Ethernet Compatibility Alliance
SD	Secure Digital
SDK	Software Development Kit
IDE	Integrated Development Environment
LED	Light-Emitting Diode
GPIO	General-Purpose Input/Output
WRAP	Wireless Router Application Platform
WLAN	Wireless Local Area Network
TCP	Transmission Control Protocol
IP	Internet Protocols
RF	Radio Frequency

I. Introduction

The use of robotics in modern society is becoming more prevalent with increasing technological advances. These robots can have many critical real-life functions, such as detecting bombs in war zones and assisting in recovery efforts at disaster areas. In order to achieve these tasks whilst minimising the risk to human life, the platforms must be able to be controlled from safe distances. To achieve this wireless communication between control units used by humans and the robots is used. This wireless communication can be achieved through various methods, a transmitter and receiver unit connected to the controls of the robot, through the use of RF communications, Bluetooth Technology or through TCP/IP using WiFi to name a few.

With the vast improvement of mobile devices such as smart phones and tablets, the ability to create applications to assist in conducting tasks is becoming easier. These applications can utilise the hardware of an already owned device and create a device that can provide the functionality of many other devices. One such application which could be made is a remote control which could communicate with a robotics platform through wireless means to control its operation.

I.A. Project Relevance

The control methods offered by Adept MobileRobots are a tethered control or a wireless joystick unit which is pictured in Figure 1. This project is important as it will provide an improved method for users to control the movement of the Seekur Jr easily. A comparison between the MobileRobots wireless control and the mobile device with application installed can be seen below.

Figure 1. Wireless Joystick Unit (Left) and Receiver Unit (Right)

From the comparison table it can be seen that the developed application surpasses the OEM wireless control unit in many areas. The project will deliver a smaller, longer lasting, economical and easily produced product which provides more functionality than the product supplied by MobileRobots.

Table 1. Comparison of MobileRobots Wireless Joystick and Seekur Jr Wireless Control Application

	MobileRobots Wireless Joystick	Seekur Jr Wireless Control Application
Dimensions	360 x 238 x 152 mm	188 x 111 x 10 mm
Weight	3.194 kg	0.304 kg
Battery	10 x 2300 mAh AA Batteries	Non-removable Li-Ion 4000 mAh battery Standby time up to 408 hours
Charging	Batteries charged external to unit	Battery charged within device Usage during charging permitted
Control Methods	Joystick control	Accelerometer and directional pad control
Communication		Wi-Fi 802.11 a/b/g/n
Cost	$7000 for unit	$200 for tablet. Application can be installed on multiple devices

II. Aim

The aim of this project is to develop an application to be used on a smart-phone or tablet to be integrated with the Seekur Jr mobile robot platform in order to provide a cheap, portable, producible method of wirelessly controlling the motion of the robot which will replace the OEM wireless console. The requirements of application that will be developed can be broken down into three main objectives.

1. Control the robot through wireless communication using a mobile smart-phone.

2. Avoid collisions through use of the laser range finder.

3. Be completely mobile and easily transferral between devices.

III. Background Information

III.A. Seekur Jr. Mobile Robot Platform

The Seekur Jr is an intelligent mobile platform produced by Adept MobileRobots. The Seekur Jr is a four-wheel, skid-steer differential drive mobile robot. It is able to carry a payload of up to 40 Kg. The Seekur Jr is used as a reference platform in a range of academic and industrial research projects.[1]

Figure 2. Seekur Jr Research Robot[1]

The Seekur Jr platform comes with a an Outdoor Navigation Package which contains a differential global positioning system (DGPS), a laser range-finder, as well as a Wireless Router Application Platform (WRAP) and tethered joystick. A wireless joystick is also available and uses a receiver unit which connects to the platform through the joystick port. The platform operates using a 2.26 GHz MAMBA Computer running a Linux OS.

III.B. Raspberry Pi

The Raspberry Pi is a microcomputer hardware platform that will be used as the prototype device to be controlled. It is a cheap, compact and computer released in 2012 and easily obtained on the internet.[2] The model used for this project is the model B variant and uses a Linux based OS which is installed on the SD card and booted up on power up.

III.C. Mobile Device

There are a number of mobile platform possibilities to develop an application on – Apple's iOS, Google's Android, Microsoft's Windows Mobile, Nokia's Symbian OS or RIM's BlackBerry to name a few.[3] Of these listed, to date the two leading platforms to be considered would be Apple's iOS and Google's Android.

Application development for iOS requires the use of a Macintosh computer running Mac OS X 10.6 (Snow Leopard).[3] The products which can use the application whilst popular are limited. Android apps however can be developed using any of the major operating systems, being Windows (XP and higher), Mac OS X (10.5.8 and higher), and Linux systems (running kernel 2.6 and higher).[3] This allows a greater flexibility in development as almost any computer can be used to develop an android application. The Android OS is run on a variety of mobile devices produced by a number of companies allowing for a broader array of devices to choose from. For these reasons an android device will be the target device for the application development.

The device that is being used for the project was a Samsung Galaxy Tab3 7.0 Lite 8GB WiFi. This device has 802.11 b/g/n WiFi and Bluetooth capabilities giving it the ability to connect wirelessly to another device. It also has the accelerometer sensors inbuilt giving it all the desired capability for the application.

III.D. Wireless Communication

The two main methods of wireless communication are through Bluetooth technology and WiFi. Bluetooth is useful when transferring information between two or more devices that are near each other when speed is not an issue. It is best suited to low-bandwidth applications like transferring sound data with telephones (i.e. with a Bluetooth headset). WiFi, however, is better suited for operating full-scale networks because it enables a faster connection, better range from the base station, and when correctly configured better security than Bluetooth.

Each method has its advantages and disadvantages, a table showing the comparison chart between the two methods can be seen below. From the values for range, bit-rate, latency and bandwidth it can be seen that WiFi would be a better option for use in this project. This is also confirmed by both the Seekur jr and the smart-phone having WiFi capabilities where only the smart-phone has Bluetooth.

Table 2. Comparison Chart of Bluetooth and WiFi[4]

	Bluetooth	**WiFi**
Frequency	2.4 GHz	2.4, 3.6, 5 GHz
Bandwidth	Low (800 Kbps)	High (11 Mbps)
Specifications authority	Bluetooth SIG	IEEE, WECA
Range	5-30 meters	802.11 b/g 32 meters indoors
		95 meters outdoors
		802.11 n has greater range
Power Consumption	Low	High
Latency	200 ms	150 ms
Bit-rate	2.1 Mbps	600 Mbps

IV. Project Development

IV.A. Approach

The approach taken to achieve the aim of the project of mobile control of the Seekur Jr can be divided into two phases. The first phase was to establish wireless communication between the Seekur Jr and the mobile device. An explanation of how this will occur is found in section IV.B. The second stage of the project was to create an application that can be used to send commands to the Seekur Jr in order to control its movemens. The user interface of the application is detailed in section IV.C.

IV.A.1. Software Engineering Model and Best Practices

Software Development follows standards such as the IEEE 12207 "Standard for Information Technology - Software life Cycle processes". The waterfall development model will be used for the project. This process goes through its stages linearly and each phase only begins at the conclusion its predecessor. The first stage is where the requirements of the system are established. From the requirements a design of the system can be formed. The design for this project can be seen in sections IV.B and IV.C. The code required for the design can then be written before each individual subsystem is tested before the whole system being tested. The subsystems for this project can be defined as the accelerometer reader, the prototype receiver, the Seekur Jr receiver and the laser range finder proximity system where the entire system is defined as the combination of Seekur Jr platform with the program and the mobile device with the application.

In order to avoid incurring software development problems, best practices will be followed. Best practices are a set of proven approaches to software development. When used in combination they help to prevent the causes of software development problems. They are called "best practices" not because their value can be quantified but because they are observed to be commonly used in industry. The practices of iterative development, requirements management and change control were used. Iterative development was used to ensure subsections of the code were written and confirmed to be correct before moving to the next part of the program. This was most prevalent in the development of the application. Requirements management was used to ensure that the meeting of certain requirements did not result in the diminished system operation. Change control was used with versions being created in the development process to ensure that the modification of code did not result in a loss of capability and a previous version could be used to move back to a known baseline. When code was improved and deemed correct it became the new baseline for the program.

IV.B. System Operation

The control of the Seekur Jr will be completed using a smart device wirelessly communicating movement commands. The block diagram seen below shows the process in which a user changes the movement data on the smart-device by passing through the stages of the system the resulting movement of the Seekur Jr is achieved.

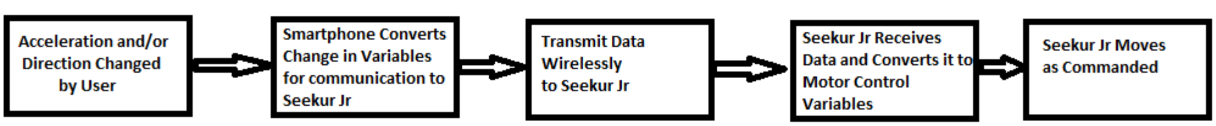

Figure 3. Flow Diagram of Communication of Control Commands from Smart-phone to Seekur Jr

The control of the platform begins with the user changing the desired acceleration or direction using the application in either accelerometer or directional pad control mode to move the robot. The acceleration and steering requirements taken from the user are translated within the application to data that can be transmitted through a wireless communication line from the mobile device to the robot.

Upon receiving the transmitted data, the Seekur Jr interprets the data to determine which direction the user wishes the platform to move to as well calculate the desired speed at which it should move. This speed value is then used manipulate the robots motors in order to move the robot in the desired direction and at the correct speed, resulting in movement of the platform.

IV.C. Application User Interface Layout

The application will be designed to connect to the Seekur Jr's host network and the control the movement of the platform through one of two methods, using accelerometer sensors which will vary direction and speed or a directional pad with fixed speeds. The below figure shows the flow diagram of the application menus that a user will progress through in order to operate the Seekur Jr.

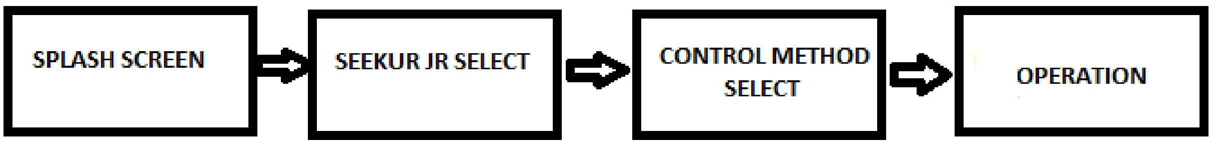

Figure 4. Flow Diagram of Application Menus

The splash screen is where the application initializes. The UNSW logo is displayed at this time. From the completion of the splash screen the next activity is where the Seekur Jr Platform will can be selected by the user. This will set the relevant wireless network connection settings within the application. Upon selection of the platform, the control method is the next activity will be selected which will determine if the user will use the accelerometer control or the directional pad control. Depending on which mode is selected will depend on which activity the application will go to. If the accelerometer is chosen then the application will display a screen with a 'GO' button and an indicator of which way the robot is being told to move. If the directional pad is chosen then the screen will show buttons for 'forward', 'backward', 'left' and 'right' as well as various speed buttons.

IV.D. Seekur Jr Receiver Program

Upon development of the application, a program must be written to be integrated with the Seekur Jr in order to control the platforms motors and sensors. This program will receive data from the mobile device and interpret the data in order to manipulate the Seekur Jr motors to meet the users requirements. A prototype program will be written and used to confirm correct operation prior to the implementation on the robot.

IV.E. Laser Range-finder Safety Distance

In order to allow for a safety control system to avoid collisions between the Seekur Jr and obstructions. The on-board laser range-finder provided with the outdoor navigation package will be used to gather the closest distance to an object and use this information to limit the ability to control the robot.

V. Work Completed

V.A. Familiarisation with Android Application Development

As no previous application development had been conducted. The need to gain an understanding of the language of Android application development as well as to learn to write it was evident. This meant the first task completed was familiarisation with the Android SDK. This involved reading of manuals as well as the information provided on the Android Developer website. This website also supplied a free download of the SDK which was based on an Eclipse IDE.

Tutorials were followed to build basic applications were created such as 'Hello World' and components such as splash screens.[5] These tutorials taught the use of the SDK as well as how to simulate using AVDs as well has how to activate the developer mode of an android device (Samsung S4 used for project) to allow an application to be tested physically.

V.B. Raspberry Pi Configuration

The Raspberry Pi being used as the prototype unit when arrived required to be booted and initially configured. This was achieved by inputting the SD Card with a preloaded copy of the New Out Of Box Software (NOOBS) to install onto the computer. NOOBS contains multiple OS that can be installed.[6] The Raspian OS was installed which is a version of the Debian Linux OS modified to suit the Raspberry Pi. Much of the configuration of the Pi was completed using the inbuilt command line where the WiFi dongle was configured to connect to a router set up as a host network. The Secure Shell (SSH) was then enabled and Putty installed on a computer so that the Pi could be controlled and worked with through the SSH in order to reduce equipment used. The Pi was pinged through the SSH to confirm operation of SSH.[7]

V.C. Accelerometer Reader Application

The first stage of the Seekur Jr remote control application was to be able to obtain and display the values for the devices accelerometer. This was achieved by finding the three accelerometer variables for the X, Y and Z planes. The values from the accelerometer in each plane were recorded into variables which could then be linked to the graphical layout of the application. These values would then be displayed on the devices screen.

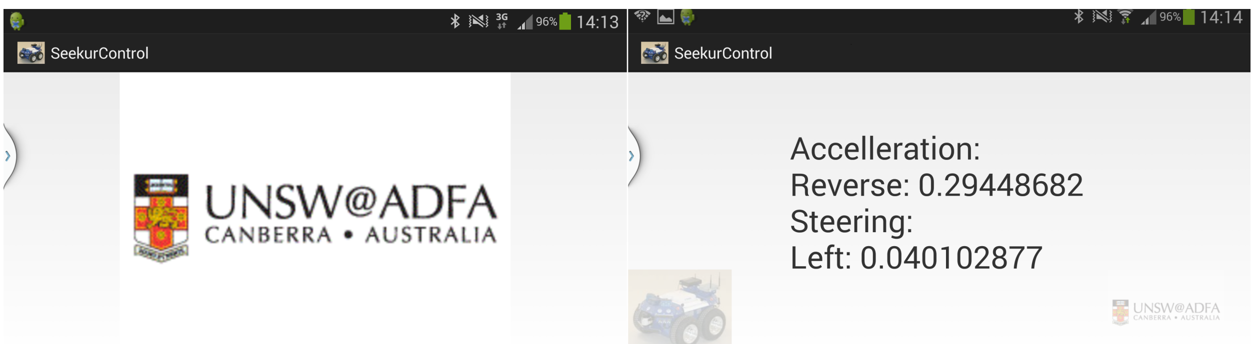

Figure 5. Splash Screen (Left) and Accelerometer Display Screen (Right) for Seekur Jr Control Application

As there are currently three planes shown but the acceleration and steering only uses two dimensions the application was loaded onto the test device and it was examined which of two of the three planes related to the orientation required for the application. The code was then rewritten to display the two planes to show acceleration to be forward or reverse(tilt forward or backward) and steering to be right or left (tilt side to side) as well as the corresponding magnitude provided by the accelerometer. The splash screen was also added with the UNSW logo upon boot-up of the application as well as on boot-up having the device connect to a WiFi network. This will be expanded on so that a user can select one of 3 networks being one of the three Seekur Jr platforms.

It was noted that if being used whilst walking the accelerometer readings would rapidly shift at times. This was due to the changes of force on the sensors as the force due to the tilting of the device is counteracted by the movement of the user. This is not an issue as functionality was still available however for better use the user should remain in one place. For the projects intended purpose of being used by a user a safe distance away there should be little requirement to move whilst driving the platform and this effect should cause little issue.

V.D. Prototype Socket Communication

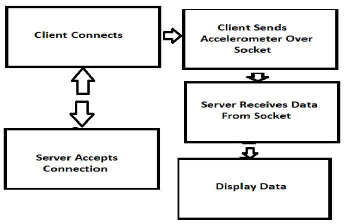

Figure 6. Socket Communication Prototype Flow Diagram

In order to utilise the accelerometer readings to control the Seekur Jr, they must be sent wirelessly from the mobile device to the robot. To achieve this socket communication is utilised. In this method each of the devices is a socket and the two sockets are connected to one another over a network, being the wireless network of the Seekur Jr's WRAP. For this project the Seekur platform will act as the socket server and the mobile device will connect to it as the socket client. To test that the socket communication between client and server will work correctly the Raspberry Pi will be used to act as a prototype server. The client device will connect to the server and then send its accelerometer data. The server will receive the data and display it.

The android application was expanded on so that it created a socket and as the client connected to the server using the known port and IP address of the server. The variables for acceleration and steering were sent from the mobile device to the server each time the accelerometer sensors changed. The Socket server code would accept the connection and then receive the data sent from the client and display it on screen so that it could be verified that what was being sent was the same as what was being received. The server code was written in python and implemented onto the Raspberry Pi.

V.E. Directional Pad Operation

Upon completion of the server code for the Seekur Jr, the directional pad control mode was written into the application. Initially designed with three levels of control in each direction, this allowed users to control the robot using a fixed velocity. The directional pad method was ideal for use when the user was following the robot as there was no effect on the operation of the application by the users movements.

The control was then modified to allow easier use by having one directional pad and speed select buttons. this made it easier to drive with one hand and adjust the speed with another as opposed to the previous method where a user would use both hands to reach the buttons.

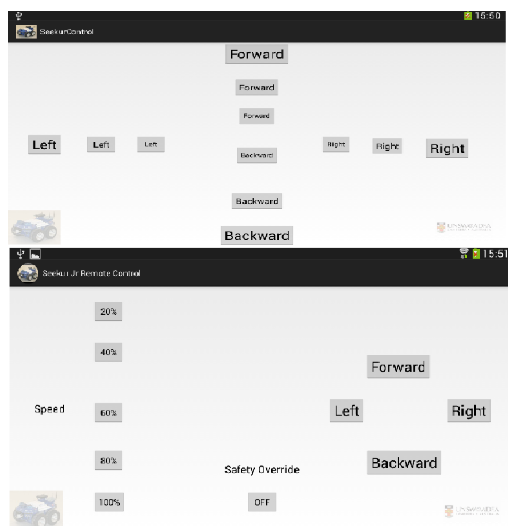

Figure 7. Initial Directional Pad (Top) and Modified Control (Bottom)

V.F. Seekur Server Python/C++ Program

Putty was used to access the on-board computer of the Seekur Jr where various examples were available to be run and viewed to learn the various methods written specifically for the Seekur Jr which involved using various components of the platform. From the examples a base algorithm was formed which utilised the accelerometer data sent from the mobile device to be used to find set the velocity and rotational velocity of the robot. This was done by calculating what percentage of the total accelerometer reading that was being sent in order to determine the velocity to be set in millimetres per second for forwards (positive velocity) or backwards (negative velocity). The same was done for the rotational velocity in which the velocity could be set to turn in degrees per second to the left(positive rotational velocity) or the right (negative rotational velocity).

Initially the code was written in python to be run on the Seekur Jr but it was noted that when the code was to be implemented on the robot that the python wrapper file which was written by MobileRobots to take the python code and utilise it in the C++ system of the Seekur Jr was malfunctioning. When tested, the

C++ examples were fully functional and the decision was made to write the code to be implemented on the Seekur to act as the socket server and drive the robot's motors in C++. This involved learning the basic principles of C++ and converting the algorithm desired to C++ from the python code.

V.F.1. Control Mode Swapping

In order to allow for the swapping of control modes between directional pad and accelerometer, code was edited within the Seekur Server program so that the program looked for a client when it had none connected. Upon gaining the client the program would receive the data and from this data set velocities for the robot. When the client socket is closed the server would look for a new client, then once a client has been found and connected repeat the process. Code was added to the application so that when one mode was exited in order to swap to the other mode, the client would close its socket and then reopen it in the new control mode. This also allows for the application to be closed and reopened a short time after and control of the Seekur Jr to be re-established.

V.G. Seekur Jr Control

The Seekur Jr receiver code was compiled and implemented on the Seekur Jr. The application was run and movement of the robot as achieved. Both the directional pad and accelerometer control methods operated as expected. A test was conducted to determine the velocities that were being input to the robot for the different commands. The velocities were calculated using the values transmitted to the robot by the mobile device and with the maximum velocity of 1200 millimetres/second and rotational velocity set to 100 degrees/second. Table 2 below shows the velocities calculated for forwards and backwards momentum and the left and right rotational velocities when the directional pad control was used for its five different speed settings. The table shows that the buttons set the velocities to the correct percentage of their maximums allowing users to utilise the robot to perform duties at a fixed velocity.

Table 3. Accelerometer Calculated Velocities

Set Speed (%)	Velocity (mm/second)	Rotational Velocity (degrees/second)
20	239.97	20
40	479.95	40
60	719.92	60
80	959.90	80
100	1199.88	100

The accelerometer was tested in a similar fashion. The mobile device was set to the angles of 0, 10, 20, 30, 40, 50, 60, 70, 80 and 90 degrees and the velocities calculated from the accelerometer values. Table 3 shows the calculated velocities for the varied angle on the accelerometer control method. The results obtained for the velocities of the accelerometer were seen to follow a logarithmic pattern. The velocity increases quickly as the device is initially tilted forward and tapers off as it reaches 90 degrees where the velocity is at it's maximum.

Table 4. Accelerometer Calculated Velocities

Angle (degrees)	Velocity (mm/second)	Rotational Velocity (degrees/second)
10	288	24
20	504	42
30	672	56
40	816	68
50	960	80
60	1056	88
70	1128	94
80	1164	97
90	1188	99

The final test conducted on the remote control application was to test the maximum distance able to be reached prior to the wireless connection breaking. To conduct this test the Seekur Jr platform was taken to the ADFA AFL Oval where a large open area could be utilised to drive the platform. The user stood at the flag poles on top of the hill on the border of parade ground and piloted the Seekur Jr across the oval in the attempt of finding the maximum possible range of the application and the distance measured using a trundle wheel. The largest range achieved was 180 m at which time testing was concluded due to a loss of sight and lack of available flat ground. Upon reaching this distance of 180 m the platform was able to be turned around and driven back to the user.

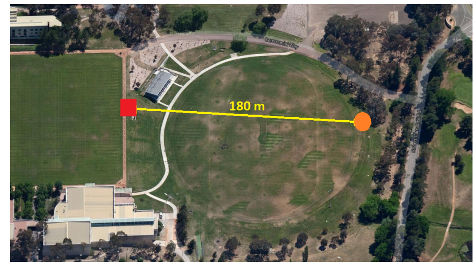

Figure 8. Application Range Testing Showing User (Square) and Seekur Jr (Circle) At the End of Testing

The range of 180 meters achieved far surpasses the expectation of approximately 95 m that can be achieved with the 802.11 g WiFi capability of the Seekur Jr's WRAP and the mobile devices wireless capability. This test has shown that the application gives the capability for a user to manoeuvre the platform from a safe distance in order to conduct a task that could potentially put human life at risk.

V.H. Laser Range-finder Proximity Safety Mechanism

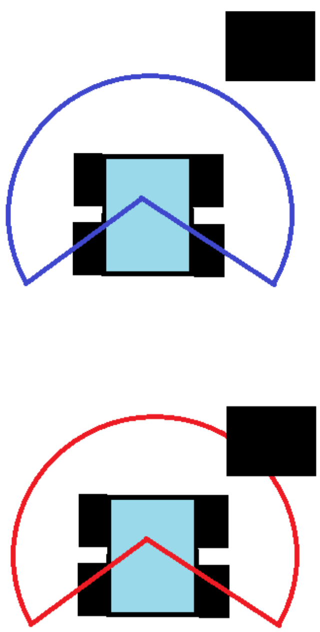

Figure 9. Laser Range-finder Safety Example Without Obstruction (above) and With Obstruction (below)

A flaw noticed in both the OEM wireless and tethered controls is that there is a lack of safety implementation to avoid running into an obstacle that appears in front of the robot, be it an inanimate obstruction or a living thing. For the project this will be achieved through the use of the on-board laser range-finder that came with the Seekur Jr Outdoor Navigation package.

Code is available from MobileRobots that uses the laser range-finder to sweep from one heading to another. This code was utilised to sweep the lasers full spectrum and return the closest distance to an object. If this distance was found to be closer than a set threshold (set to 900 mm), as seen in the bottom half of Figure 9, the robot would be restricted in its movements to be only allowed to move backwards. This is so that if an obstruction is found in the robots field of view the robot cannot turn into or drive forwards into the obstruction.

This code was implemented into the Seekur Server program and tested where it was established that a need an override of the safety system was required as the range-finder proximity distance made it difficult to navigate through tight spaces such as doors and hallways where the robot would detects the door-frame or wall as an obstruction and prevent the user from driving forwards. Similar issues were found in the OEM MobileEyes Software Supplied with the Seekur Jr platforms in which during simulations at times the path mapping algorithms which incorporated a safety distance refused to allow the robot to pass through a doorway.

The application was modified to allow a user operated button that when clicked the safety range was changed to be much larger, allowing the user to pilot the Seekur Jr through tight spaces then when required again the safety override can be turned off again but re-clicking the button.

The proximity safety functionality of the application could be further expanded on by implementing a

'context aware system' within the Seekur Jr. Context-aware systems are systems that are aware of their situation in their physical, virtual and user environment.[8] These systems use this context to adapt the system to this in some way. An example of this in the scenario of this project is that the context aware system can be aware of what a doorway or hallway looks like to the laser range-finder and can adjust the distance at which the laser range-finder proximity triggers for the duration of the platform passing through the doorway.

A current shortfall in the proximity safety mechanism is a 90 degree blind spot at the rear of the platform. The laser range-finder has a 270 degree field of view which is effective for front-on or side-on obstructions but does not cater for the robot reversing into obstructions. The current system requires some reliance on the user to be watching the rear of the robot when reversing or to avoid reversing where possible. This could be rectified by introducing sensors such as parking sensors to the platform.

Parking sensors use technology such as ultrasonic proximity sensors for operation. Ultrasonic proximity sensors emit acoustic pulses and measure the time it takes for the pulse to return to a control unit. This interval is used to calculate the distance from the object from which the pulse was reflected. The ultrasonic sensors could be mounted to the rear of the platform and integrated within the Seekur Jr platform. The distances calculated from the data obtained from these sensors could be used in conjunction with the laser range-finder to further enable the proximity safety mechanism to operate. With separate sensors the ability to more specifically prohibit the platform from moving depending on which sensors had been triggered.

VI. Conclusion

This project has been a success in the software integration within the Seekur Jr platforms in order to their increase capability as research platforms. The aims of this project have been met in the development of an application for a mobile device which controls the movements of the Seekur jr Platforms. The application was compatible with android devices and when installed on the Samsung Galaxy Tab 3 Lite mentioned in section III.C of the report the developed control application and device was smaller, lighter and cheaper than the OEM wireless joystick unit. The application was tested and was able to reach a range of 180 meters without losing connection allowing a large distance between user and the platform to be maintained without loss of control. The laser range-finder functionality was integrated into the project in order to enable a proximity safety mechanism which reduces the chances of collisions with living things or inanimate objects.

VI.A. Future Work

There are currently shortfalls in the proximity safety distance, being the 90 degree blind spot at the rear of the platform as well as the lack of ability to distinguish areas such as doors and hallways in which the platform may pass through. Further research into 'context aware systems', including the development of an algorithm which can utilise these systems in conjunction with the laser range-finder to be able to identify objects as known objects which can be passed by at closer distances would prove beneficial to future capabilities of the Seekur Jr.

In order to resolve the blind spot of the robot. Work into fitting the rear of the platform with ultrasonic sensors or similar technology to have distance data for this area would allow a safer system by reducing the possible collisions.

VII. Acknowledgements

I would like to thank my supervisors A/Prof Charles Harb and A/Prof Valeri Ougrinovski for their support, advice and for allowing me the opportunity of doing this project where I have been fortunate enough to build upon the skills learnt whilst at UNSW. I would like to thank Nathan Softley for always being available to assist in brainstorming sessions to overcome problems and most importantly, I would like to thank my partner Carlee for putting up with me over this period whilst at times I have been on another planet. I owe you my greatest thanks.

References

[1] Adept MobileRobots. Seekur jr. manual, 2013.

[2] Ankur Tomar. The raspberry pi single board computer, February 2012.

[3] Mark H Goadrich and Michael P Rogers. Smart smartphone development: ios versus android. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 607–612. ACM, 2011.

[4] Diffen.com. Diffen LLC. Bluetooth vs wi-fi, May 2014.

[5] C. Delessio, L. Darcey, and S. Conder. *Android Application Development in 24 Hours*. Sams Teach Yourself in 24 Hours. Pearson Sams, 2013.

[6] Eben Upton and Gareth Halfacree. *Raspberry Pi User Guide*. John Wiley & Sons, 2013.

[7] Adafruit. Learn how - raspberry pi, 2014.

[8] Stefan Poslad. Context-aware systems. *Ubiquitous Computing: Smart Devices, Environments and Interactions*, pages 213–244, 2009.