PO Box 1127
Rotorua 3040
Ph: + 64 7 921 1883
Fax: + 64 7 921 1020
Email: info@ffr.co.nz
Web: www.ffr.co.nz

Theme: Harvesting

Task No: 1.2
Milestone Number: 5.3

Report No.: H021

Teleoperated Steep Slope Harvester: Semi-Autonomy and Haptic Feedback

Author:
Bart Milne
University of Canterbury

This document is Confidential to FFR Members

Date: 19 December 2014

Leadership in forest and environmental management, innovation and research

TABLE OF CONTENTS

1. INTRODUCTION

Forestry is an important primary industry in New Zealand. While mechanisation of harvesting is well advanced on flat land and shallow slopes, on steep slopes (beyond 22°) forest harvesting is still performed using manual labour, which is very dangerous. The scope of this project is to outline the software concepts for implementing teleoperation of a steep slope harvester including semi-autonomy and haptic feedback. However, in terms of the greater goals of the Future Forests Research Ltd (FFR) Teleoperation project, such a report would be incomplete without recommendations for how teleoperation and semi-autonomy would be implemented on a full size machine.

Early in the project an assessment was made on whether to buy and modify a commercial off-the-shelf teleoperation system or develop a teleoperation system in-house. It was decided to pursue in-house development since no specialised vendors existed for teleoperated forestry equipment suitable for New Zealand conditions. It is possible to use off-the-shelf parts for remote control but doing in-house development allows for a much greater scope for experimentation. Apart from worker safety, a core goal of semi-autonomy is to increase worker productivity. At present semi-autonomy has not been tested in the field, but the Robot Operating System (ROS)-based software architecture has plenty of scope for increasing worker productivity, including incorporation of real time sensor data and machine vision.

This project concentrates on hydraulic machinery, but two related projects included in the FFR Objective 1.2 Teleoperation Project, the John Deere 909 Remote Control Project (909 Project) and the 'Stick Insect' project have significant crossover with this project. The 909 Project has an implementation of remote control using ROS, and the Stick Insect uses ROS and also incorporates real time sensor information. A possible future development path would be to use the remote control architecture and visual feedback of the 909 Project with the real time sensing of the Stick Insect.

Overall, the choice of in-house development is well justified due to the much greater flexibility when developing and evaluating the system components, especially the control components. Even if a commercial off-the-shelf teleoperation system becomes available in the near future the findings from the in-house development will provide background and guidance of what is required of such a system. It also provides good experience into what would is required of off-the-shelf components for smaller sub systems such as hydraulic valve control.

This report is organised into the following chapters:

- Chapter 2: A description of the hydraulic system used in the laboratory.
- Chapter 3: Information about ROS, the platform for developing and testing the semi-autonomy.
- Chapter 4: Discusses the results of the semi-autonomy testing.
- Chapter 5: Discussion of the experimental outcomes and how they affect implementation on a full size machine. Also includes recommendations on how a teleoperated machine should be integrated into an existing work place.
- Chapter 6: A discussion of how the work in this report can be applied to future projects.
- Chapter 7: Conclusions.

1.1. Report Objectives

The primary objective of this report is to answer the following two questions:

- Where is the greatest potential for application of semi-autonomy in a logging operation?
- What are the projected benefits of applying semi-autonomy in this way?

The objectives of this report are to make recommendations on the future use of semi-autonomy and haptic feedback in forestry operations, including steep land harvesting. The type of machine semi-autonomy covered is the autonomous avoidance of obstacles. A further objective of this report is to comment on some of the possible crossovers with the two related FFR projects (the Stick Insect and the John Deere 909 Remote Control).

This report is completes Objective 1.2 Task A Milestone 5.3 of the FFR Harvesting & Logistics Annual Plan 2014-2015. The objective of this project milestone is to complete development and testing of software concepts that provide semi-autonomous operation of some of the functions of the steep slope harvester by 31 December 2014.

2. ROBOT DESCRIPTION AND SETUP

The research robotic system is a small scale hydraulic arm as pictured in Figure 1. The hydraulic setup is similar to a full size excavator except that the slew is implemented with a hydraulic ram as opposed to a hydraulic motor. The machine is immobile so there are no hydraulic controls for the tracks, and these have not been investigated.

Figure 1: Hydraulic test system

2.1. Hydraulics

Each of the three rams in the hydraulic test system is connected to four-way three-position closed centre electro-hydraulic proportional valves. The pressure compensator valve acts to match the system pressure to ram load and bypass the relief valve so it does not operate when all three valves are closed. The ram load is sensed using a system of sensing lines that detects the highest load pressure on any given ram. The pressure control valve works by being blocked unless the supply pressure is at least 110 psi (7.58 bar) higher than the pressure on the sensing line. Table 1 shows the specifications of the hydraulic test system. Figure 2 is a diagram of the hydraulics in the hydraulic test system.

Table 1: Specifications of hydraulic lab system

Part	Description
Prime Mover	Electric Motor 4 pole induction, rated output 0.75 kW (1 HP) at 1410 rpm
Pump Flow	5.78 L/min at 1410 rpm (0.0041 L per revolution)
Pump	Constant Displacement
Valves	Electrically operated proportional valves
Relief Pressure	800 psi (55.2 bar)
Maximum Relief Pressure	1130 psi (77.9 bar) at 5.78 L/min pump flow and 0.75 kW Prime Mover output
Pressure Compensator Bias Spring	110 psi (7.58 bar)

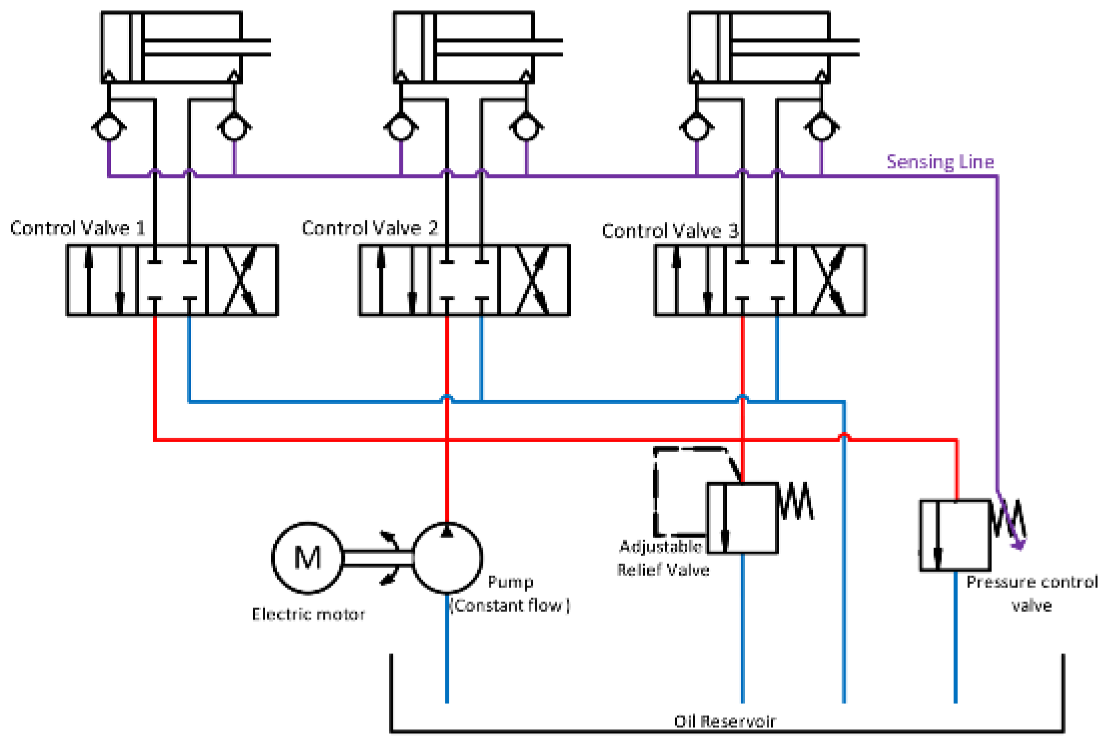

Figure 2: Diagram of hydraulic test system

2.2. Input and Output

The hydraulic test system uses a Programmable Automation Controller (PAC) and amplifiers to control the valves. The PAC is a SNAP PAC R1 model manufactured by Opto 22, originally chosen because the original target of the teleoperation project was the Trinder ClimbMAX steep slope harvester. The present set up runs off the 230 V AC mains supply, but a different power supply can be substituted so it can run off 12 V or 24 V as would be found in an excavator. Figure 3 shows the control cabinet with its associated circuitry.

Figure 3: Hydraulic arm control cabinet. From top: power supply, PAC, protective circuit breaker and three PWM generators (lower right)

The PAC is programmed using the Opto 22 ‘Pac Control’ software. The PAC is programmed using control charts, which are very similar to flow charts. Figure 4 shows the control charts that have been programmed into the PAC: the Powerup, WatchDog Timer and Control Charts. The three charts run simultaneously. The Powerup chart runs when first powered to launch the other two charts.

The WatchDog_Timer (which is a one second timer and is reset every time a new control packet is received), and the Control charts run indefinitely. If the watchdog timer expires, the PAC shuts the valves down. The watchdog timer resets automatically on the reception of a command.

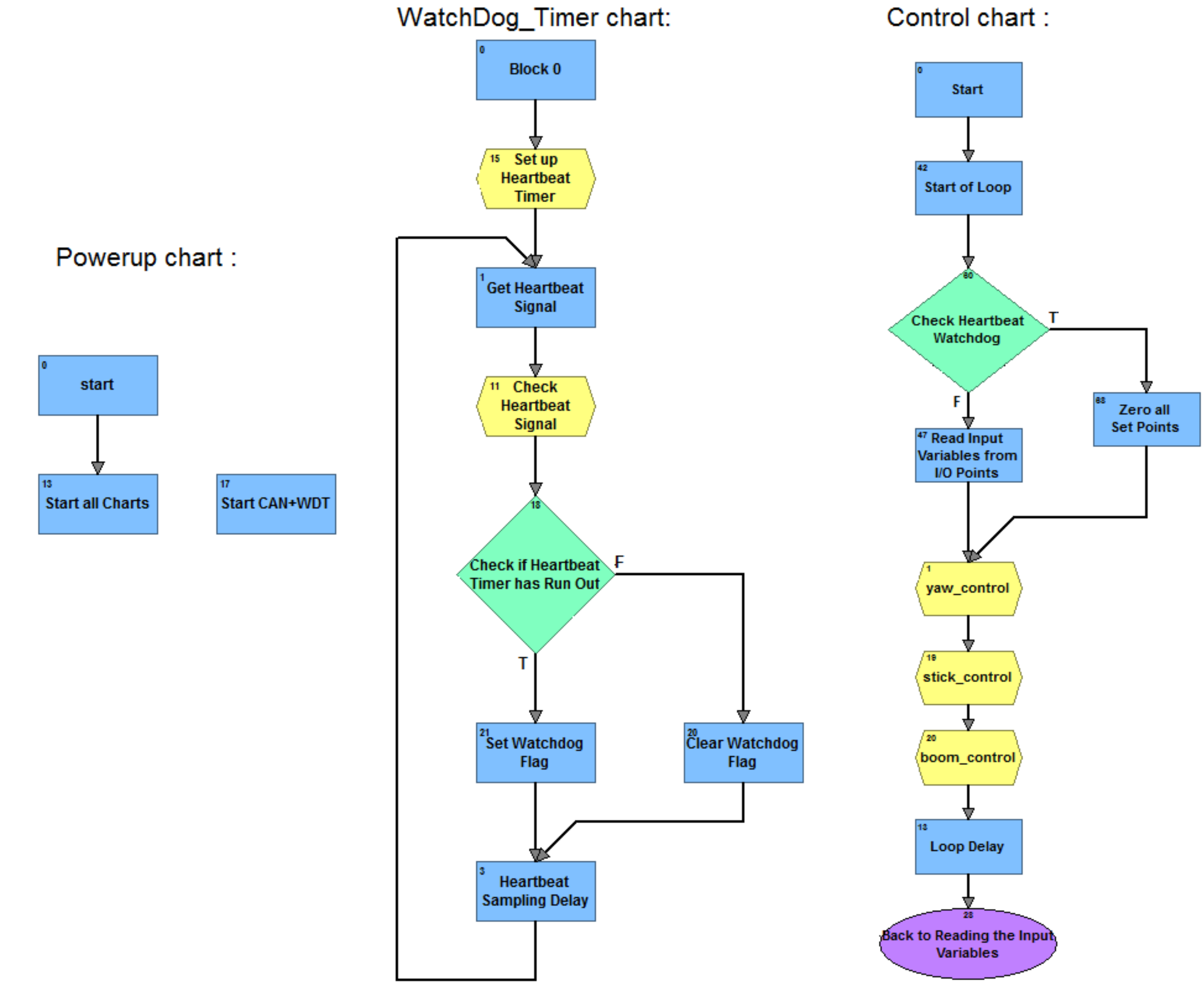

Figure 4: PAC control charts

The Control chart sends the received valve commands to Digital-Analogue Converters (DAC) to output a control voltage from 0V to 10V for the valve amplifiers. The DAC outputs are sent to the Pulse Width Modulation (PWM) generators (boom and stick rams) or valve current controllers (base ram). The M171 PWM generators (Figure 3, lower right) pictured are actually intended for driving motors, and so facilities like dither are not built-in and must be implemented in software. The base ram has controllers that control the valve coil directly, but the other two rams have motor controllers that generate a PWM voltage to drive the valve coils. Tests of the resistance and inductance of the valve coils confirm that the response of the coils (time constant 140 µs) is fast enough to assume that valve current changes instantaneously at the 100 Hz frequencies of PWM operation.

Valves 2 and 3 have a single PWM generator each. The PWM output is switched between the valve coils using relays depending on whether the valve command is for the ram to extend or retract. This arrangement saves on output modules on the PAC, but if there are enough output channels this is not necessary, and omission simplifies the output hardware. Communication with the PAC is done through an Ethernet connection. The PAC listens for incoming packets, and once a command packet is received it loads the packet content into the designated input array. The PAC has been programmed to take valve commands as 32-bit floating point values in the range -100 to +100.

2.3. Robot Sensing

The control algorithms and kinematics libraries for position control work with Cartesian coordinates and joint angles. There are two types of sensors in use for sensing the hydraulic arm's angles: slope sensors and string encoders. The sensor data is received over a USB to Controller Area Network (CAN) bus adapter connected to the local control computer and processed for both feedback about the machine state and control. Figure 5 shows a diagram of the kinematics of the hydraulic test system with joint angles labelled.

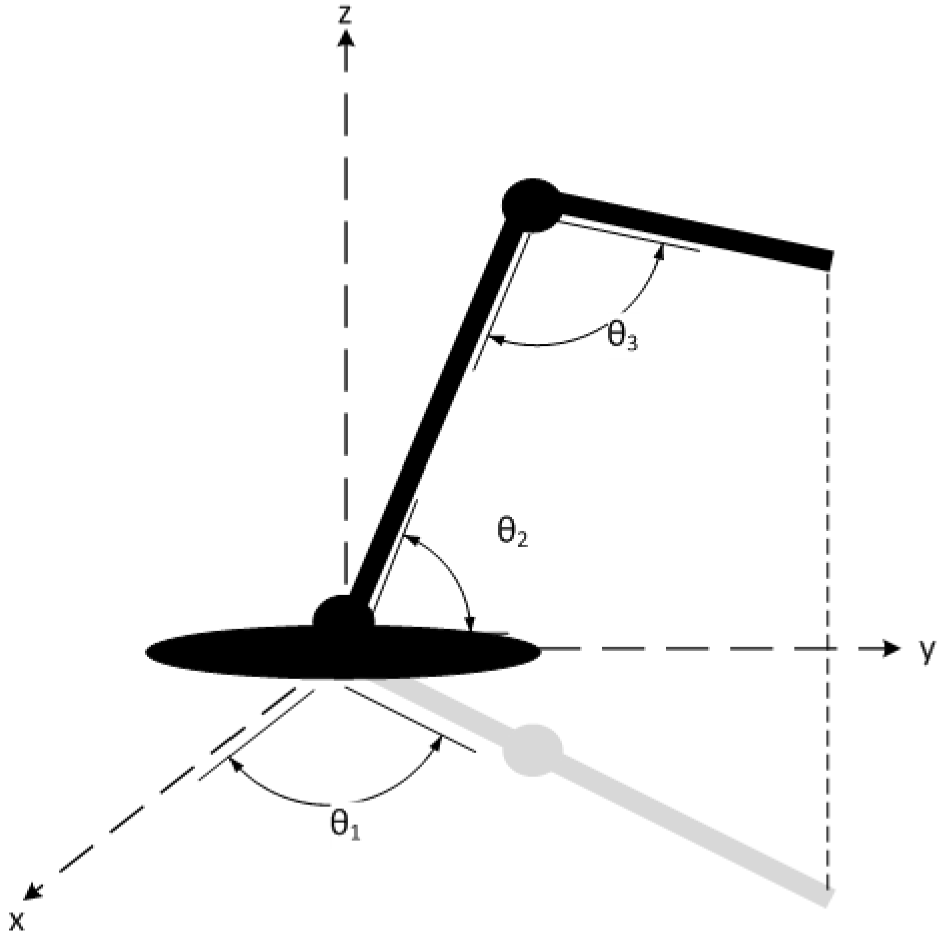

Figure 5: Diagram of hydraulic arm kinematics with joint angles labelled

Figure 6 shows the hydraulic test system with the rams and arm links labelled. In Figure 6, Link 1 can be thought of as the boom and Link 2 as the stick.

Figure 6: Hydraulic test system with links and rams labelled

2.3.1. Joint Angle Sensing

Joint Encoders

Joint encoders were considered but were not used, as the joints in the robot have not been engineered for inclusion of joint encoders. This is also a likely issue with excavators. Instead solid state slope sensors and calculation of joint angles from ram length measurements were investigated.

Slope Sensors

Slope sensor angle sensing calculates joint angles by detecting differences in the gravity vector between different arm segments (Figure 7). The slope sensors use an Inertial Measurement Unit (IMU) to generate a roll and pitch angle output. The main advantage of the IMUs is that they are small in size, self-contained, do not require calibration and can be bolted or stuck onto the arm at any convenient point that gives the correct data. The slope sensors used were supplied by Trimble, and one is shown in Figure 8. There was a slope sensor fitted to the base, boom and stick. The accelerometer and gyroscope (angular rotation) measurements are processed to synthesise a slope in the roll and pitch axes. The slope sensors cannot reliably detect slew angle because in general the gravity vector is parallel or near parallel to the axis of rotation for slew, so the gravity vector does not change with changing slew angle.

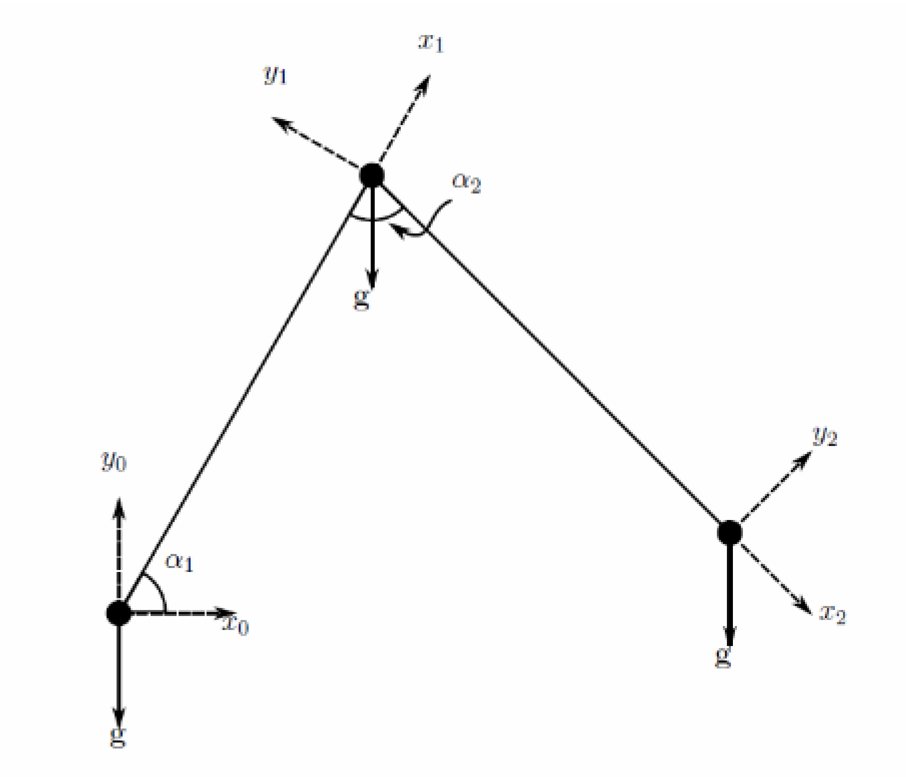

Figure 7: Principle of IMU-based angle measurement

Figure 8: Trimble GS Inertial Measurement Unit (IMU)

The main shortcoming of the slope sensors is that cancellation of the non-gravitational accelerations requires complex algorithms. Any imperfection in the correction causes the joint angles to appear to change when in fact they are not changing. This effect was noticed on the hydraulic arm, and caused instability with some control regimes, particularly with the haptic feedback. The haptic feedback would respond to movements 'seen' by the slope sensors which did not actually happen.

The slope sensors supplied were two-quadrant and so could not determine if a joint had an angle of 70° because it had been rotated from horizontal 70°anti-clockwise (90 minus 20) or 110° anti-clockwise (90 plus 20). This would not normally be a problem on an excavator on level ground but if it is on a 45° slope the boom could easily become vertical, giving incorrect readings. The two-quadrant limitation is firmware based, as the accelerometers and gyroscope sensors work in all four quadrants.

2.3.2.Ram Length Sensing

The main alternative investigated for joint angle sensing is ram length measurement. In some cases, such as the base of the arm, ram length sensing is the only way to obtain the joint angle due to the axis of rotation of the base being vertical. The devices used were Trimble CE21 string encoders. Figure 9 shows a string encoder set up to measure the length of Ram 2, which controls the angle between the base and the boom. The situation shown is an example of where the slope sensors give incorrect readings, since the joint angle shown is greater than 90°. The ram length derived angle may read 96° but the slope sensor would have read 84°.

Figure 9: String encoder installed to measure ram length

Figure 10, 11 and 12 show the principles behind measuring the joint angles. The joint angle is directly calculated with the cosine rule and the measured ram length. The cosine rule also requires the lengths of the other two sides of the ram triangle and any extra angle offsets if the ram location is not on the axis of the arm segment. This data can be entered manually but is actually derived from information in the Unified Robot Description Format (URDF) file of the arm. The advantage of using the URDF data is that all of the measurements needed to calculate are included automatically, and update automatically if the robot model is changed. String encoders were used in the laboratory because they are non-invasive and easy to set up. They are not appropriate in the forest because the string is vulnerable to damage and inaccuracy if debris gets caught in them. Using ram length also requires accurate measurements of the dimensions of the triangle formed by distances from the ram pivots to the arm pivot. The smaller this triangle is relative to the arm, the more accurate the measurements must be. The CE21 encoders required calibration after every power cycle. In the lab, the string encoders were calibrated by driving all the rams to their shortest lengths and setting that reading as zero from the string encoders. The string encoder reading was added to a 'zero length offset' distance to obtain the true length of the ram for calculations. If this is inconvenient the string encoders could be calibrated against the slope sensors.

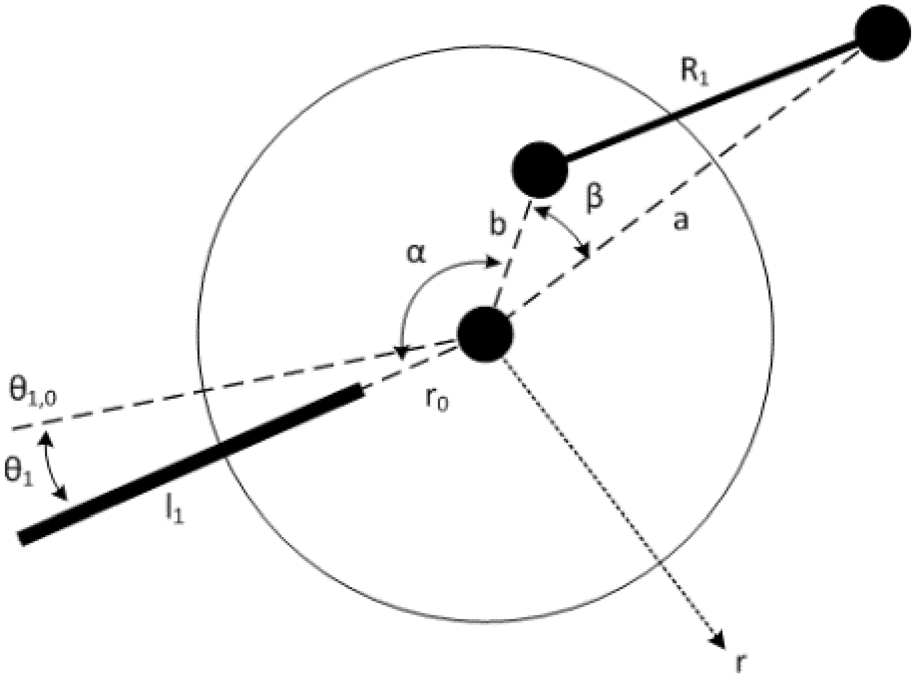

Figure 10: Arm base angle calculation diagram

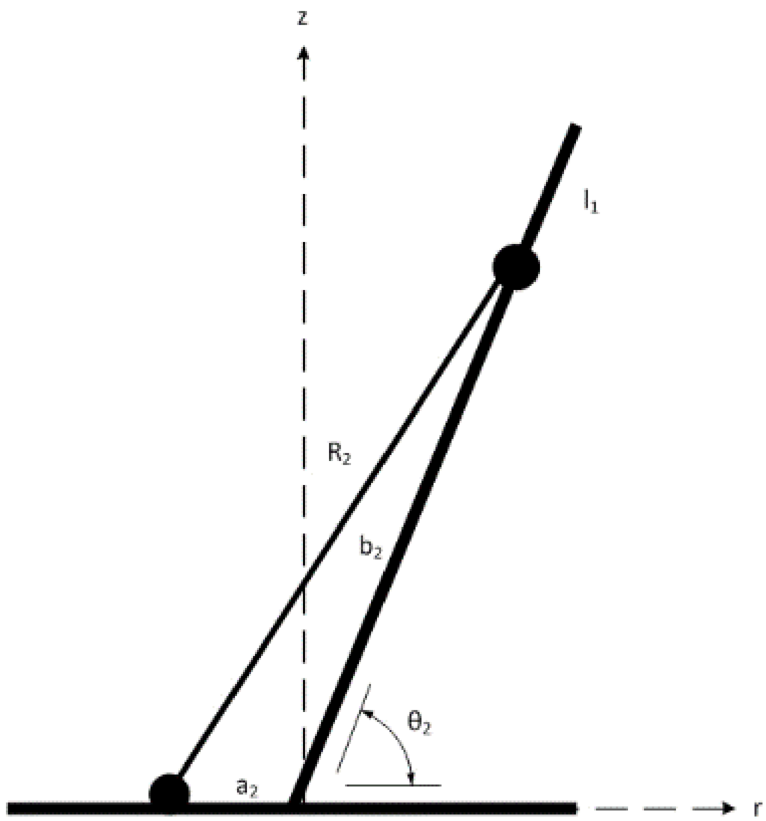

Figure 11: Shoulder joint angle calculation diagram

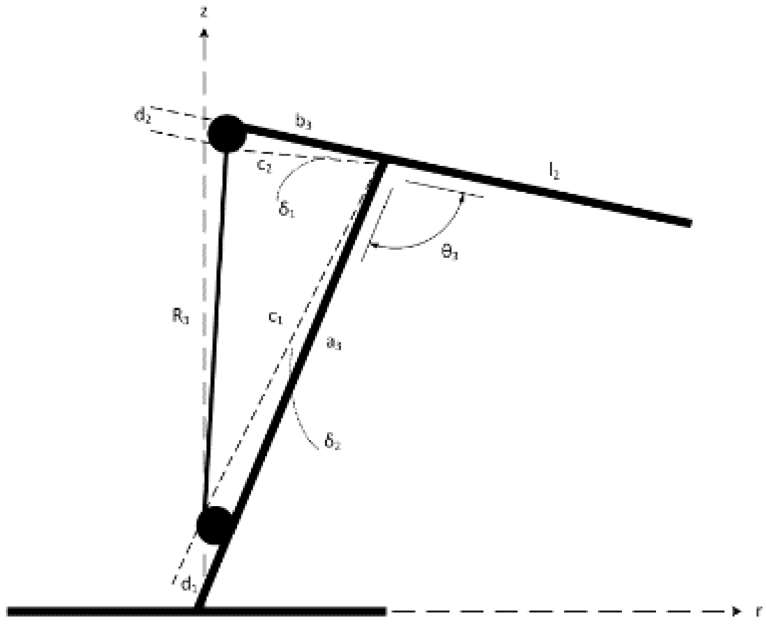

Figure 12: Elbow joint angle calculation diagram

2.4. Visual Feedback

Visual feedback comes from an Axis M2014 Internet Protocol (IP) camera mounted on the turntable as shown in Figure 13. The IP cameras can stream directly to an Internet Browser window, so ROS is not strictly required. However it is a good idea to integrate the camera with the rest of ROS, as the browser interface provides no direct means of integrating extra information such as camera status. The software uses a modified version of the Axis camera drivers developed for ROS.

Figure 13: Turntable camera location

The camera data is streamed over ethernet, received by the computer and converted into a ROS camera image message. The data is used by the display programme to produce operator's output as seen in Figure 14. If no signal from the camera is received for one second, an annotation ‘Stream is not Live' appears on the display to warn the operator that the picture being seen is not being updated.

Figure 14: Operator's view from turntable camera

The turntable camera is mounted on a special pole to give it a field of view approximately the same as an operator in a full size excavator would see. On a full sized harvester there are two main positions for mounting the operator's camera: inside the cab in the approximate position of the operator's head, or outside the cab just above the front windscreen. The main advantage of having the camera in the operator's cab is that it requires less protection from the weather and the viewpoint is more similar to an operator's. The main advantage of mounting above the front window is that the camera is always out of the way and the picture is not affected by a dirty windscreen and protective bars.

2.5. Controller Setup

The controller runs on a PC using ROS. The valve commands from the ROS PC are sent over an Ethernet link to the PAC which generates the drive voltages for the coil controllers.

All the controllers used are Proportional-Integral-Derivative controllers (PID controllers), a control loop feedback mechanism widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point.

Figure 15 shows the fundamental structure of the control architecture used for the hydraulic test system. An explanation of the quantities in Figure 15 is given in Table 2. The quantities in Figure 15 are linked by transfer functions, some of which are linear (such as the PID controller), but others (e.g. Non-Linearity Compensator and Length Sensor) are non-linear, but the overall goal is to provide a linear response of the system as a whole, from r to x^*.

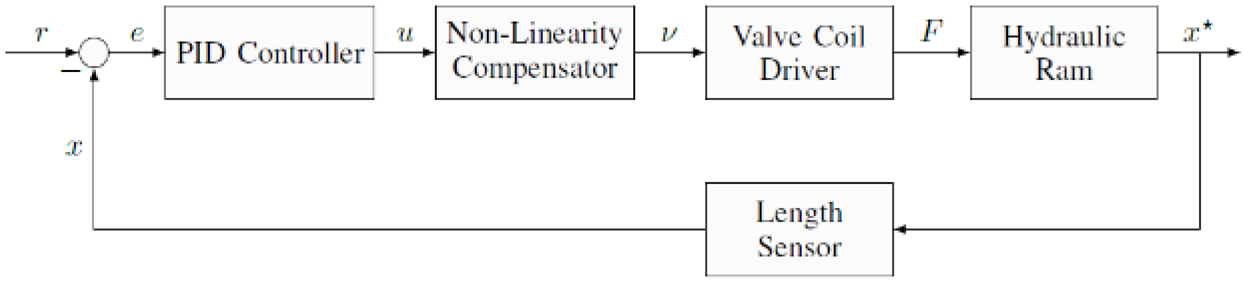

Figure 15: Low level control loop architecture

Table 2: Description of quantities used in Figure 15

Quantity	**Description (unit)**
r	Input position (m)
e	Error (m)
u	PID Command, normalised so that $\lvert u \rvert \leq 1$ (no unit)
v	Converted Valve Command, normalised so that $\lvert v \rvert \leq 100$ (no unit)
F	Force on the ram (N)
x	Position of the ram as measured by the sensors (m)
x^*	Actual position of the ram (m)

More detailed modelling such as model predictive control is a possibility, but PID control has been found to have adequate performance. Proportional position controllers have been used because the valve setting controls ram speed, so the controller has a natural integrating action built in. The controller architecture is designed to separate the command generation from the compensation for valve non-linearities. Valve non-linearities are determined from data sheets or experimental measurements (Figure 16 and Figure 17).

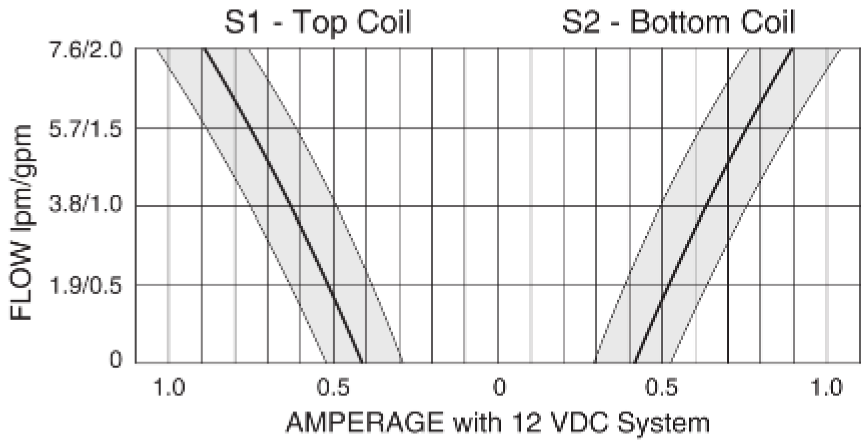

Figure 16: SP08-47CL Flow-current characteristics

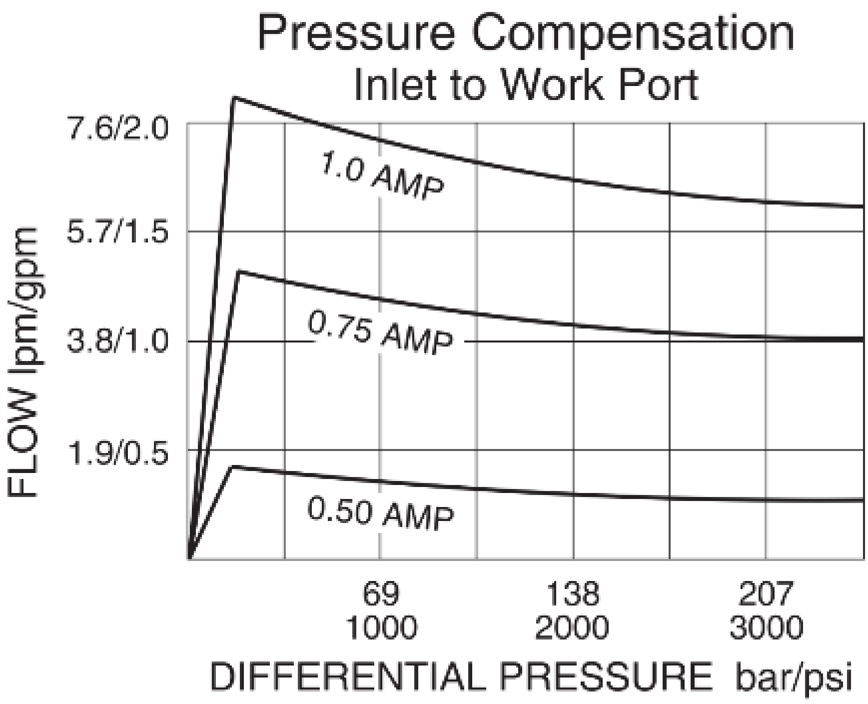

Figure 17: SP08-47CL Flow-pressure characteristics

3. ROBOT OPERATING SYSTEM (ROS)

The Robot Operating System (ROS) is used to implement the control system, including receiving sensor data, calculating control commands and sending control commands to the PAC. ROS has been selected and used because it is specifically designed for robotics development. ROS works by separating the data handling from the data processing. A ROS control system will work with a large number of small programs (nodes). The nodes communicate with each other using messages. A sending node publishes a message with a named topic (e.g. /node/position) and another node can receive this data by subscribing to the topic. The advantage of using ROS is that it has its own event queueing and message handling libraries built in. All the programming and development of message queues and other message handling is handled by ROS, so the development effort can be concentrated on developing the control algorithms themselves as opposed to the means of shifting data from one place to another. A full description of the capabilities of ROS is beyond the scope of this report, but more information may be found at the ROS Website (http://ros.org). ROS is designed to run on Ubuntu Linux, and the control computers in the laboratory run ROS Hydro on Ubuntu Linux 12.04 'Precise Pangolin'.

At the highest level, teleoperation is implemented in ROS by splitting the programs it runs between a 'local' machine (the machine connected to the robot) and a 'remote' machine (the machine the controller interacts with). In reality, with the exception of hardware inputs and outputs there is no fundamental boundary between the two machines. In fact, the 'local' and 'remote' machine can be the same. The flexibility of the machine arrangements is useful for testing the teleoperation against running the teleoperation program on a single machine.

The ROS computers in the laboratory interact with each other over Ethernet, though the existing control system can easily be extended to control over an internet. The ROS computer connected directly to the PAC is designated the 'local' computer. The local computer is designated the master, and is the control computer that interacts with the hydraulic test system directly. The present control system is set up as a 'single-master' system. The primary advantage of a single master system is its ease of setup – its main drawback is that if the master computer fails the entire control system stops working.

3.1. Software Interface with the Robot

The main design philosophy is to maintain ROS's goal of having small nodes that do a few things well and keep coupling between different nodes to a minimum. It is also useful to maintain a hierarchy of different software functions in line with the ROS Control architecture diagram given in Figure 18. Doing this allows decoupling the design of the ROS control system from the design of the sensors and reducing the interdependency between different algorithms, which may have been written by different people over different times. The PAC is accessed from ROS through code supplied by Opto 22. The ROS interface takes data in ROS topics, translates it into the form used by the PAC, and then despatches it to the PAC over Ethernet using the Opto 22 Application Programming Interface (API) functions.

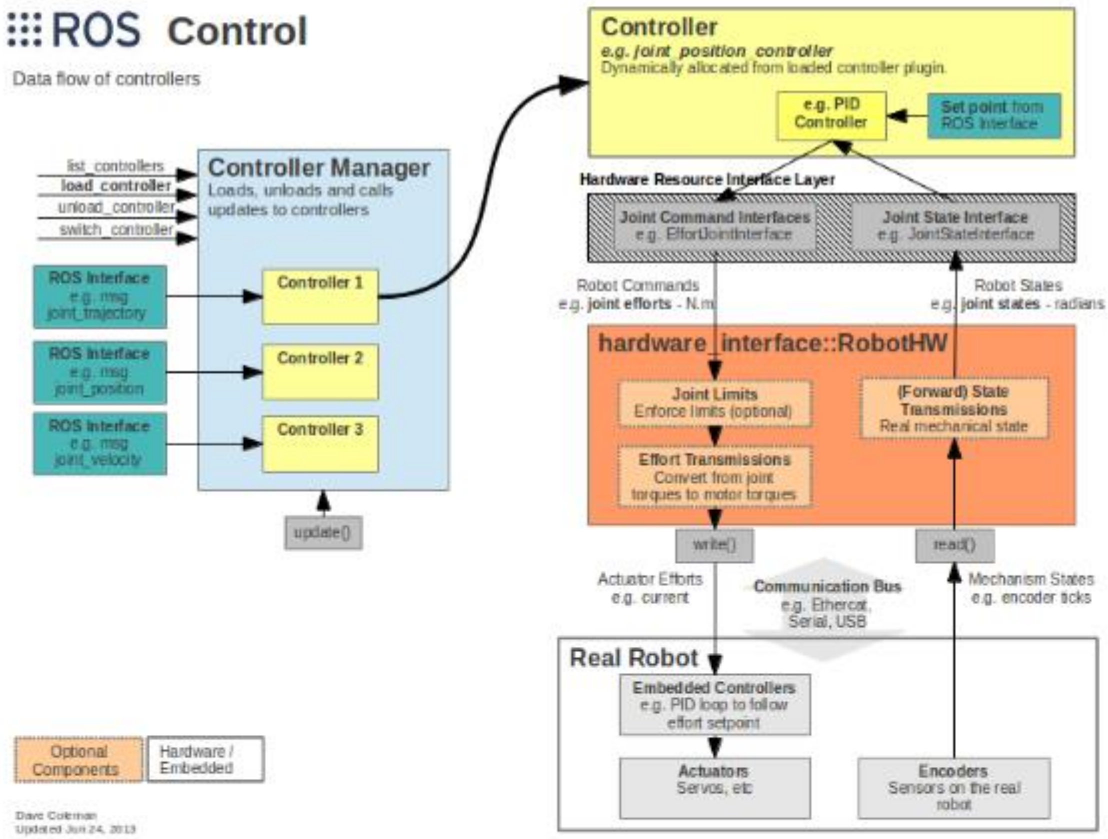

Figure 18: High level ROS control architecture

3.2. Control System Safety

The two main safety levels in the hydraulic test system are an Emergency Stop (E-Stop) rated to Safety Integrity Level (SIL) 3, and a watchdog timer that is programmed into the PAC (as detailed in Section 2.2 Input and Output). The E-Stop disables the pump if the button is pressed. The button may be pressed at any time, and must be reset before attempting to start or restart the pump.

The PAC itself is not turned off if the E-Stop is activated, so it will continue to issue the same command to the valve. So while the hydraulic pump will no longer be providing power, the valves may still be open to some degree. This scenario can still cause ram movement, but any problems with this can be eliminated by providing an input to the PAC from the emergency stop and cutting off the circuit to the valve coils.

Another important influence on safety is the computer and network security. Security issues that are important include authenticating commands and making sure that a given harvester or robot is controlled by a single computer. In the laboratory only one controller computer was used at once. Scenarios in which more than one computer is trying to communicate with it at once have not been tested.

3.3. Kinematics Model

Although it is possible to program the robot kinematics directly into ROS, the control system architecture uses ROS's MoveIt! libraries to perform kinematics (forward and inverse) and path planning. MoveIt! has the advantage of having complete integration with ROS with a wide range of kinematics and motion planning libraries, and once the robot model has been created all of these libraries are available. The robot model is based on a URDF files and Semantic Robot Description Format (SRDF) files. The URDF file has information about the joint locations and axes in the robot, and does not require detailed information about the size and shape of the robot or its parts. This is a beneficial feature for existing equipment that lacks a detailed solid model.

The hydraulic test system also has a solid model which was developed in SolidWorks from direct measurements on the hydraulic test system. The solid model includes the arm, table and legs. In particular, it allows the derivation of a model of arm inertia that is useful for path planning. The extra information about moments of inertia is included in the SRDF file and is used along with the URDF file by MoveIt!. In the case of the hydraulic test system, the solid model and the assumption it was constructed from mild steel was also used to calculate the mass and rotational inertia of each arm segment. The rotational inertia from the SRDF file and joint limits from the URDF file are used by the path planner to limit acceleration, speed and joint angles to what the robot is physically capable of, thereby providing an achievable path plan.

The coordinate systems specified in the URDF are generated from SolidWorks by specifying reference coordinate systems. Ram lengths in the solid model are calculated by placing reference coordinates at each end of the rams. The ram length can then be calculated from the distance to the two coordinate system origins. This feature is used to measure ram lengths, which otherwise would not be directly possible with a URDF model, as URDF models do not allow closed paths in a kinematic chain. A more advanced robot format SRDF is used by Moveit! as it is able to store extra information such as end effectors and closed kinematic chains.

3.4. Open Loop Joystick

The Open Loop joystick is the simplest of the interfaces. The joystick controls the rams directly with no sensor feedback. While it is not autonomous, its main use is for sensor calibration and testing if the rams are controllable or not. For example, the hydraulic test system requires the rams to be driven to their shortest length before calibration. The open loop joystick interface is the only way to do this, as any other one that uses sensor feedback will not work because the sensors are not calibrated or are not working.

3.5. Implementation of Teleoperation

Teleoperation is the primary goal of the project. Some justification needs to be given as to why it is not mentioned more specifically in conjunction with the different interfaces and controls. The reason is that ROS is extremely flexible with the layout of the control system, so apart from any changes due to the properties of the communication medium itself (delay etc.), no specific changes are required to the source code.

In ROS, implementation of teleoperation is a matter of determining which programs should run on the local computer and which should run on the remote computer. The lab implementation has identical code bases on both computers, with different launcher files for the local and remote machines. Some programs must run on the remote computer, for example user interface and input, and others must run on the remote the computer, such as sensor and machine control interfaces. For example, the autonomous robot control has the trajectory generator running on the remote computer, and the real time feedback controllers run on the remote computer.

All of ROS's communications run over TCP/IP (Transmission Control Protocol / Internet Protocol) or UDP (User Datagram Protocol), so it never actually sees the medium over which it is transmitted. The only effects that are seen in ROS are the latency and data rate. As long as the latency and data rate are within requirements the system will work. As a consequence of using TCP/IP and UDP, the teleoperation can be extended to operations over the internet quite easily.

The key to easy teleoperation with ROS is to separate out the different software components to run on the local computer and the remote computer. Ideally the remote computer should generate high level commands and the local computer should generate all the low level real time control signals. Autonomy works extremely well with this type of teleoperation due to its ability to separate high level and low level commands.

Implementing teleoperation in this way allows the same code base to be used on both the local and remote computers. While in the longer term specific code bases can be developed, the use of identical codebases eases research and development significantly.

4. RESULTS

4.1. Autonomous Path Planning

The trajectory planner based on ROS's MoveIt! libraries uses inverse kinematics, joint dynamic limits (speed, acceleration, etc.) and knowledge about obstacles to plan trajectory that gets to the desired point while meeting all of the constraints. The planner works with a planning scene consisting of a computer model of the robot and its surroundings. In the case of the hydraulic test system, a computer model of a post was made and this was placed in the planning scene to reflect the set up shown in Figure 19. The robot is moved around by dragging the model to a different point, then executing the path. Figure 20 shows a planned path for a scene with no obstructions. The starting point is on the right and the end point is on the left at the location of the arm (orange) and head (blue ball) with marked coordinates. As would be reasonably expected, the path planner shows a 'straightforward' path between the start and end points.

Figure 19: Obstacle avoidance physical setup

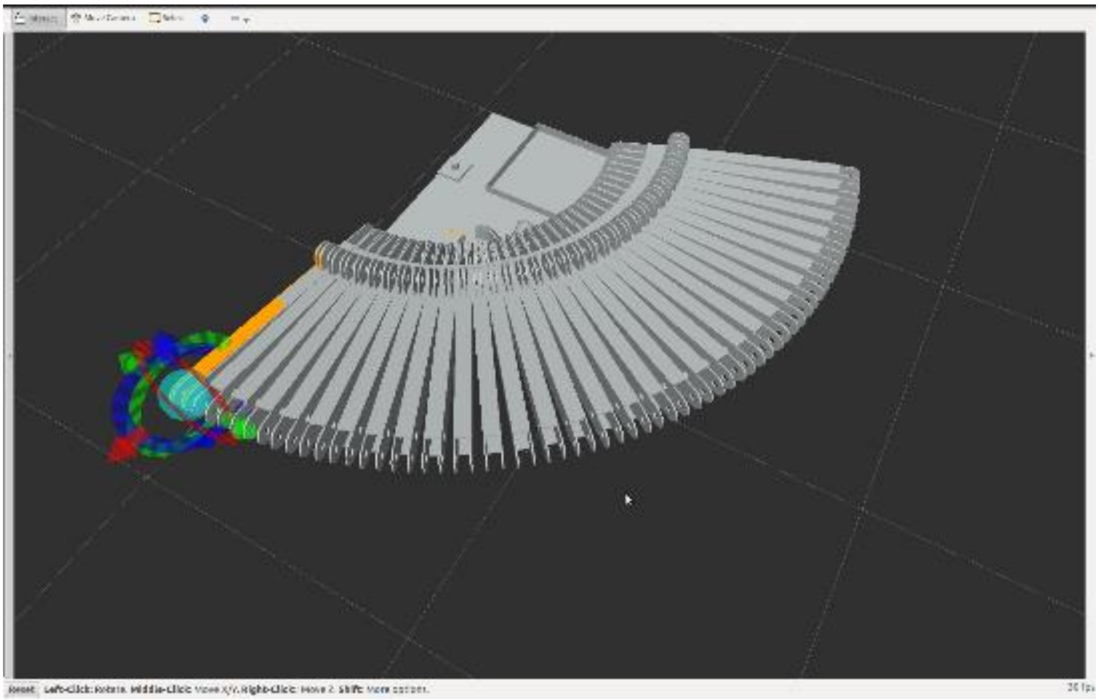

Figure 20: Planned path - no obstructions

Figure 21 shows the planned path with the post model loaded into the planning scene but without the planning scene published to ROS. Since the planning scene has not been published, the path planner does not recognise that the post is there. The post model in the planning scene is actually larger than the real post because the planning scene post model has a 150-mm buffer zone added to it to prevent the robot unintentionally contacting the post during a planned path. An allowance of this type is important because sensor and model errors will cause the arm to not conform exactly to its planned path.

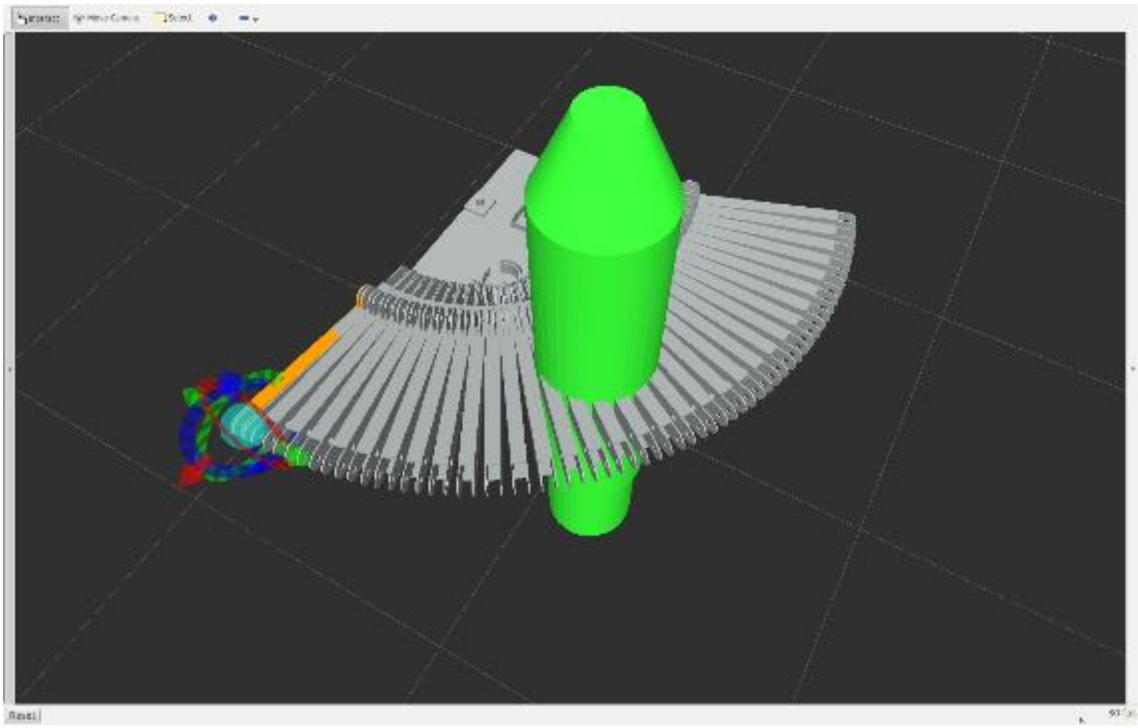

Figure 21: Planned path - with collision

The planned path shown in Figure 21 is clearly unsuitable due to the collision with the post. However, once the planning scene has been published, any subsequently planned paths will be devised to avoid the post. Figure 22 and Figure 23 show the planned path to avoid the post.

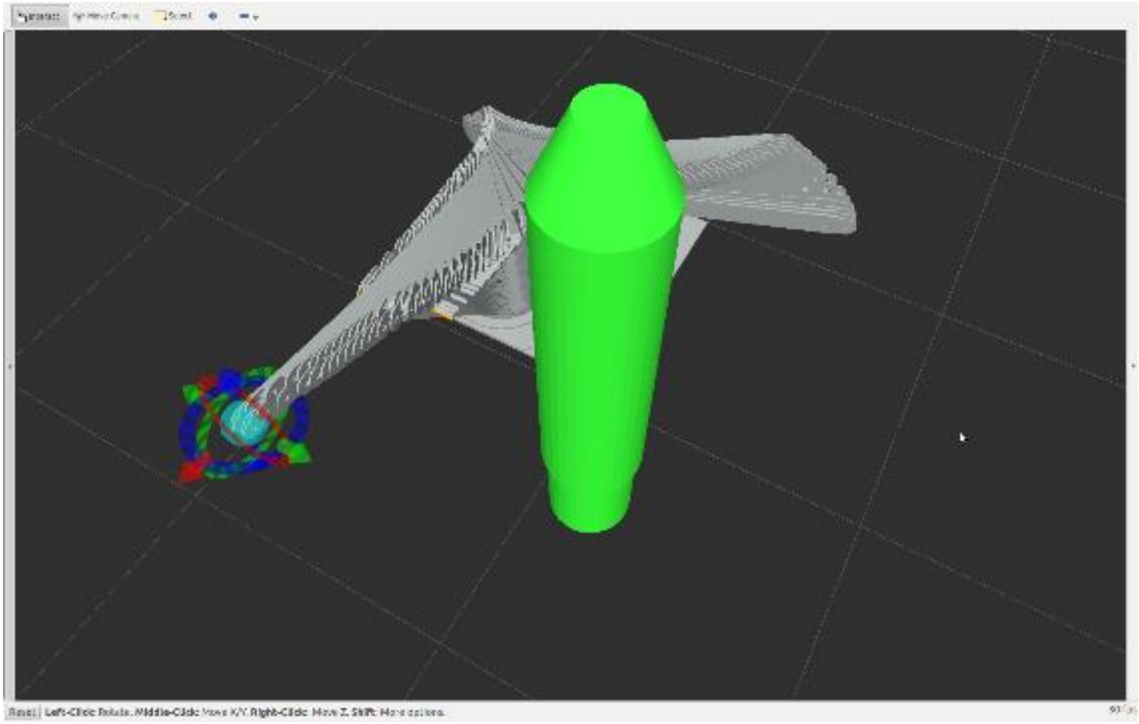

Figure 22: Planned path - view from front of hydraulic test system

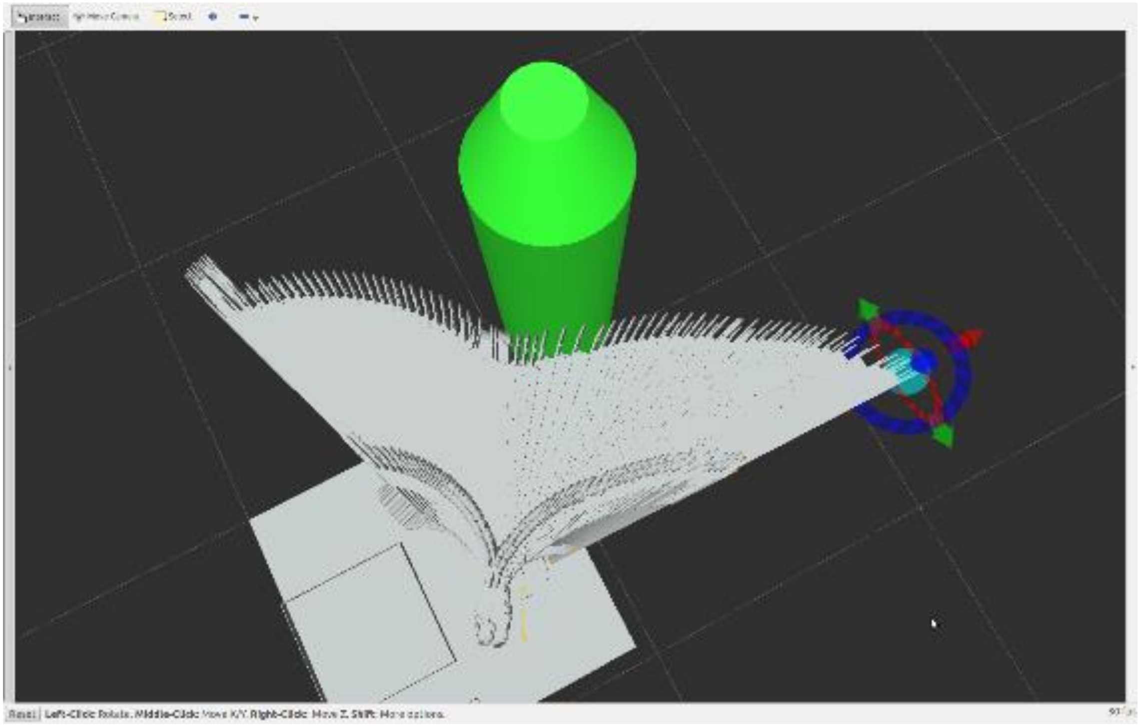

Figure 23: Planned path - view from rear of hydraulic test system

Sometimes a particular state is not allowed due to a collision with a scene object (Figure 24 with post actual size). If this is the case the colliding part will turn red. Collisions can occur between any two solid bodies, but the MoveIt! Setup Assistant is normally set to ignore collisions between adjacent machine segments. If collisions between adjacent machine segments are taken into account, the path planning can be unreliable because the machine might appear to be colliding with itself even though such a position is physically impossible and only represents an error in the solid model. Ignoring these collisions is useful because model and sensor inaccuracies can cause the machine to seem to collide with itself even though it is not physically possible. This is especially noticeable at shallow joint angles – the solid model of the robot arm appears to have a collision between the boom and the stick at shallow joint angles ($\theta_2 \approx 180°$ in Figure 5) despite it being physically impossible in real life.

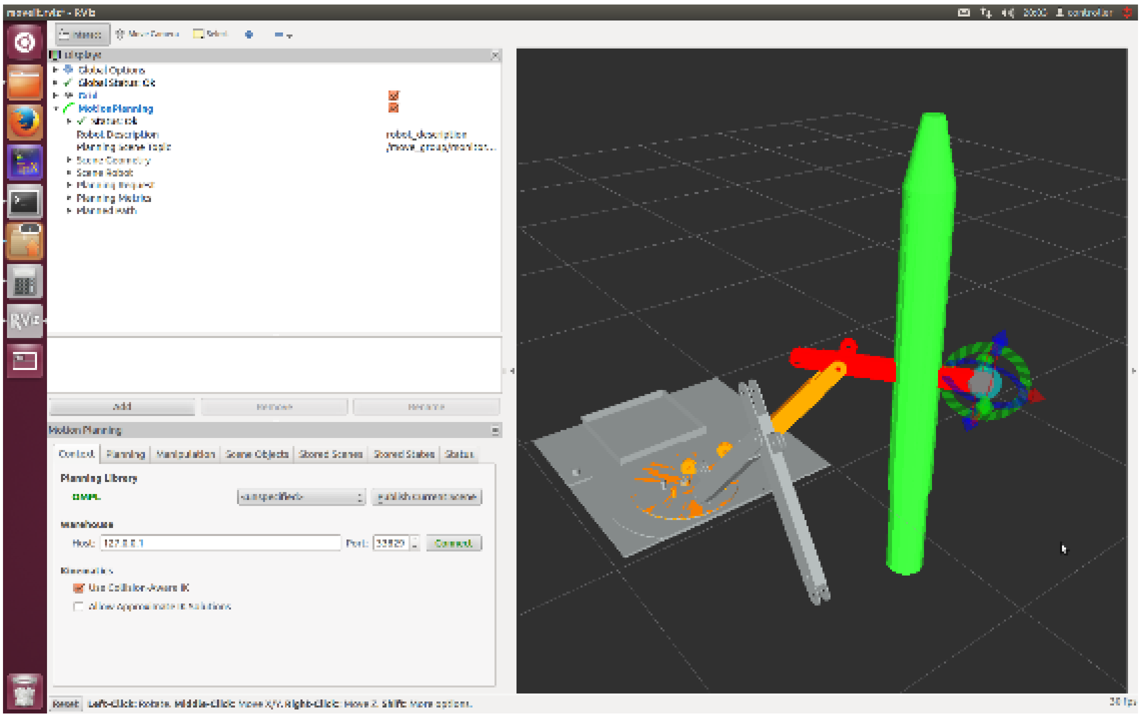

Figure 24: MoveIt! trajectory plan - disallowed position with colliding part shown in red

4.2. Haptic Feedback

Haptic feedback has been trialled using a Sensable / Geomagic Phantom Omni haptic feedback arm. The Phantom Omni is designed to be held in the hand, so the operator controls the arm by holding the stylus of the Phantom Omni and changing its position. In this usage scenario the Phantom will give resistance to movement in proportion to the error in joint angles between the hydraulic test system and the Phantom Omni. As the hydraulic arm starts to move towards the Phantom Omni's position the force will decrease. The operator will feel this as a decrease in resistance and the feeling that the hydraulic arm has moved to the desired position.

Figure 25: Phantom Omni

The Phantom Omni is also able to operate in non-haptic feedback mode. In this mode the joint angles of the Phantom Omni are used to control the joint angles of the arm, but there is no force feedback to the Phantom Omni. The two grey buttons on the stylus are used to switch between the modes. Figure 26 shows the Phantom Omni arm mirroring the joint angles on the hydraulic arm. The image of the hydraulic arm also emphasises that an accurate solid model is not a prerequisite. The model has joints in the same locations with the same axes as the real hydraulic arm, but the links were modelled as simple boxes.

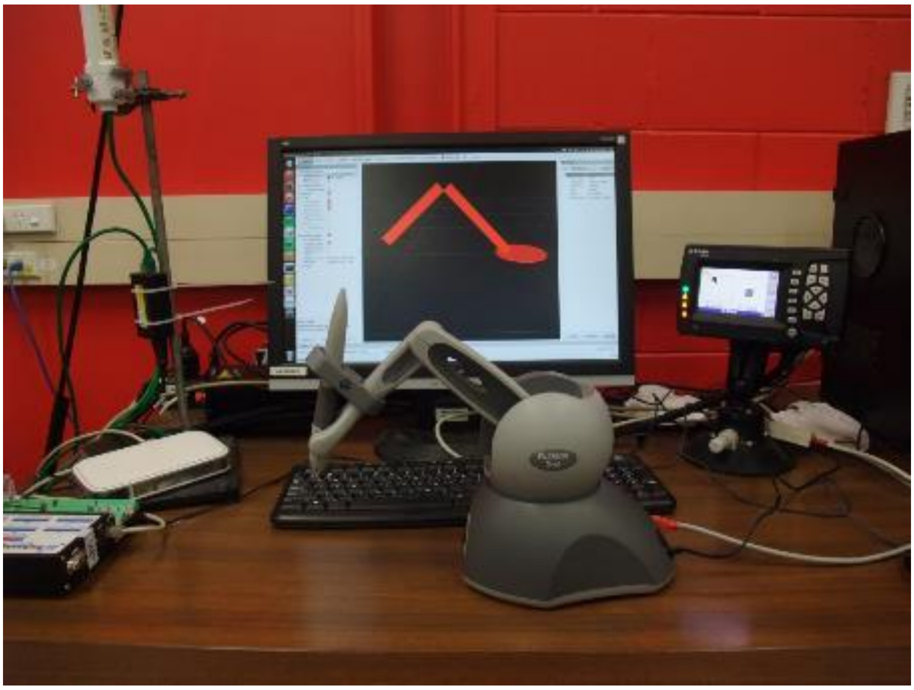

Figure 26: Phantom Omni mirroring arm position

5. DISCUSSION AND RECOMMENDATIONS

5.1. Relationship with Other FFR Teleoperation Projects

A related project in the FFR Objective 1.2 Task A is being pursued to teleoperate a John Deere 909 harvester (909 Project). The scope of this report is different from the 909 Project as it deals directly with hydraulics, whereas the controls for the John Deere are done using CAN bus commands. While the work on cameras and semi-autonomy has been done on a hydraulic machine, with a suitable interface it can be transferred to a CAN bus-controlled machine such as the John Deere.

The obstacle avoidance scenarios shown are not based on real time sensor data. However, work is being done in a related project in the FFR Objective 1.2 Task B (the ‘Stick Insect’ project) on using laser scanners to detect trees in real time. An example of tree detection from a laser scanner point cloud is shown in Figure 27. ROS includes an extremely good architecture for integrating this kind of data: the screen shot shown is exactly the same type of planning scene as shown in the previous section on Autonomous Path Planning (Section 4.1).

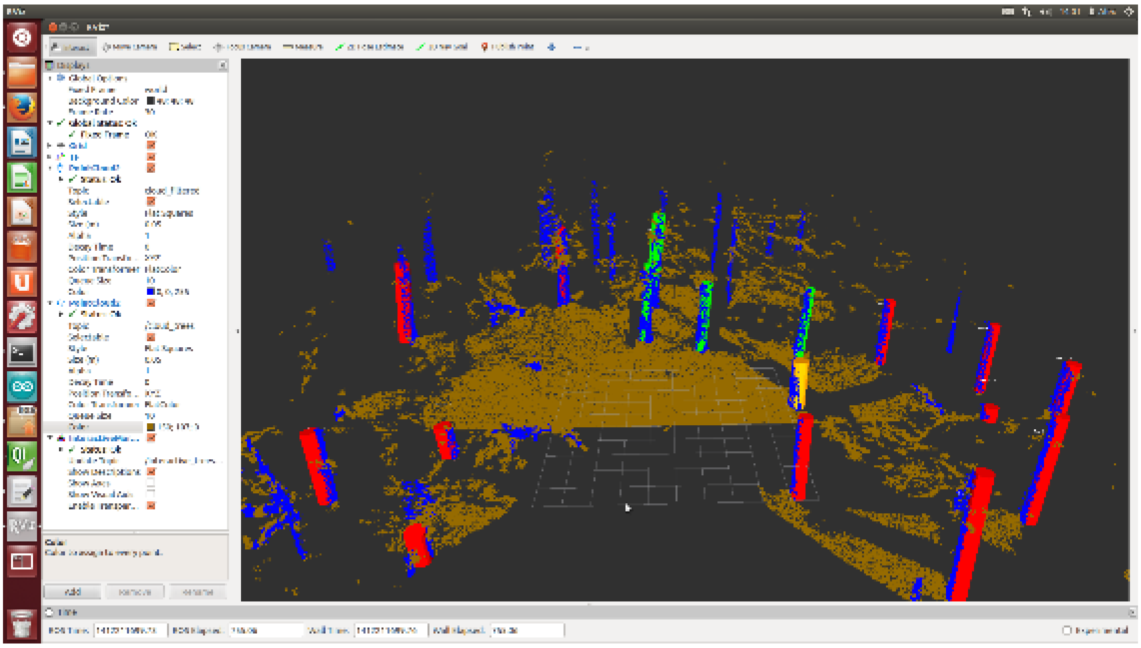

Figure 27: Example of automatic tree detection from stick insect project

5.2. Extension to Full Sized Excavator

The final application of the research is to extend semi-autonomy to a full sized excavator. There are different levels of semi-autonomy; the greater the level of semi-autonomy the more must be known about the target machine's construction and materials. Direct coordinate control requires just knowledge of joint locations and angles, whereas autonomous path planning requires knowledge of the construction of the machine so a path can be planned and executed which is within the physical capabilities of the machine.

5.2.1.Training and Integration into Worksite

Special plans need to be made for safe operation of a teleoperated machine, especially if it is the first such machine on the worksite. Operators need to be made aware that workers in surrounding areas may not always know what the machine is doing, and should be careful to ensure that they are able to get out of the way of the machine if necessary.

Conversely, personnel working near a teleoperated machine need to ensure they have a plan for avoiding hazards. These plans should include escape routes in case control of a teleoperated machine is lost. It is recommended that personnel never approach a teleoperated machine unless under instruction of the operator.

At a minimum, operators need the following knowledge about the machine and teleoperation system:

- Starting and stopping the engine.
- Determination of when the machine is in teleoperation mode.
- Emergency stops, their operation, and how to reset them.
- Start of work day, end of work day and other ‘Make Safe’ procedures.

5.2.2. Safety and Emergency Stops

It is recommended that all life safety emergency stops should be rated to a minimum of IEC61508 SIL 2 or PLd. While the ROS control system does not have a SIL rating, a separate emergency stop system can be integrated into the control system.

The ROS Control system is not designed or intended to provide life safety in the event of a malfunction or failure of the machine or control system.

If is recommended that the design of the Emergency stop be totally separated from the ROS-based control system. ROS's role is primarily as a control system, not a safety system, and emergency stop systems can be built from standard off-the-shelf parts.

5.2.3. Worksite Safety and Hazards from Semi-Autonomy

The most significant new hazard from semi-autonomy is that the machine can now move in ways that are not directly predictable by personnel in the surroundings. In an excavator this should be covered by the standard safe work practice of never being inside the radius of the boom and stick. At present this should be enough for the machine, as autonomous movement of the tracks has not been covered. A machine moving semi-autonomously may also knock over an object it encounters in its path, and this needs to be taken into account when planning out work sites.

While the machine can move semi-autonomously and can avoid obstacles, there are still limitations to how quickly and accurately the machine can avoid the obstacles due to positioning accuracy. The semi-autonomous obstacle avoidance should never be depended on to protect life and property. It is also noted that the remote operator will not be able to see the surroundings as well as an operator in the cab, so care needs to be taken to avoid any blind spots. Good communication will be essential to ensuring that no dangerous situations arise.

5.2.4. Visual Feedback

Cameras on the outside of the machine will be exposed to very harsh operating conditions. Cameras designed for excavators are generally low profile, sealed against moisture and condensation, and have hardened glass lenses to resist scratching. It is recommended that any camera system mounted on an excavator have an International Protection rating of IP69K, so it is dust tight and can withstand water blasting.

Visual feedback is recommended on all sides of the machine. In particular it should never be assumed that the only direction the machine may move is where the operator's cab camera is pointing (i.e. the same direction as the boom). In some applications such as a tail-hold/backline, the normal direction of movement will be at 90° to the boom. It must also not be forgotten that the cameras are not just for machine control – they are also useful for detecting hazards in the environment around the machine.

ROS has libraries such as OpenCV which have machine vision capabilities that could be used to perform tasks like automatic tree recognition. The 909 Project is using IP cameras integrated with a ROS-based control system. It is recommended that this course be pursued, as using IP cameras gives much more flexibility in system design than proprietary designs.

5.3. Autonomy

An example situation for semi-autonomy would be automatic grabbing of a tree from a point-and-click style interface. All path planning and obstacle detection would be handled by the machine, with the operator providing safety oversight and telling the machine where to go next.

There are two main areas that are important to semi-autonomy in the forestry machine: autonomous path planning and autonomous obstacle avoidance. The two are closely linked but the data sources are different. The obstacle avoidance testing used a model of an object whose size and location were both known in advance. In a forest this is not the case. There is plenty of scope to include this obstacle sensing with the hydraulic machine, as the Stick Insect also uses ROS for its obstacle detection system. Some different implementations of semi-autonomy along with their pros and cons are shown in Table 3. The main points to consider are the sort of operator interface required and the sort of sensing required. All types of autonomy require sensors of machine state, but not all types require sensors for the surrounding environment. This is an important point to consider, as a control system that doesn't require environmental sensing will be cheaper and possibly more robust than a control system that does.

Table 3: Summary of different types of semi-autonomous control

Type	Description	Pros	Cons
Cylindrical Coordinate Control	Excavator controls work on arm in cylindrical coordinates (radius, slew angle, height)	Sensors only required on machine and not for environment	No obstacle avoidance
'Point-and-Click' Interface with Autonomous Obstacle Avoidance	Operator specifies end point of manoeuvre, obstacles are automatically avoided	Less input required from operator	Requires accurate physical model of machine and environment which can be difficult to acquire, cumbersome for small manoeuvres
Real-time Haptic Feedback	Operator 'feels' machine movements in real time with haptic (touch-force) feedback	Helps restore feeling lost with teleoperation, does not require environmental sensing	Performance and stability heavily dependent on time delay
Workspace Haptic Feedback	Similar to 'Point-and-Click' interface except operator can use haptic feedback to feel where the machine is physically capable of going	Less dependent on network latency than real-time haptic feedback	No feel for machine movement during manoeuvres, has same sensing requirements as 'Point-and-Click' interface

5.4. Haptic Feedback

Haptic feedback has many possibilities for control. The haptic feedback that was tested was position control. There are other possibilities, such as giving the boundaries of a workspace including obstructions. The Phantom Omni is a research tool only and is not considered durable enough for daily use in a forestry environment. However, the project has possession of a Novint Falcon haptic feedback device (Figure 28) which can be used for further research on haptic feedback. The kinematics of the Novint Falcon are quite different to the Phantom Omni and an excavator, but ROS has the kinematic libraries to make the two devices useful together. One of the advantages of the Novint Falcon is that it is designed for daily use as a gaming machine, whereas the Phantom Omni is designed as a research and development machine.

Figure 28: Novint Falcon haptic feedback device

Work is currently being done on characterising the stability of position-position haptic feedback over an Ethernet link. Proof of stability is complicated by the fact that the delays over the internet are not predictable, and the control algorithm needs to be robust towards these effects. However, provable stability will put the bilateral haptic feedback system on a good and robust footing for use in the field.

6. FUTURE WORK

There is plenty of scope for convergence between the Stick Insect project and the 909 Project. The Stick Insect project includes a more in depth exploration of real time environmental sensing, and the 909 Project is a demonstration of remote control using system architecture similar to the ROS-based machine in the laboratory.

Future work on the hydraulic test system involves refining the control concepts presented in this report to the point where a direct path can be seen from the laboratory scale system to a full size excavator. The environmental sensors (laser scanner etc.) also need to be deployed and tested on the hydraulic test system.

There are significant technical risks associated with scaling up to a full size excavator. Future work will concentrate on making the control system configurable so that it can still work even if some sensors are not working. This is considered essential for the control system to be practically usable in a forest environment.

The proposed tail-hold fit out will provide a lot of useful data on the teleoperation experience that will be useful when implementing the ROS-based system.

7. CONCLUSIONS

Initially there was some uncertainty about whether to use an off-the-shelf system or develop a system in-house. There is certainly justification for such uncertainty as it is essential to ensure that one does not attempt to reinvent the wheel. However, in this case I believe that the in-house development route has been well chosen. It has provided a measure of flexibility in development that would be very difficult to obtain from a full commercial off-the-shelf teleoperation system that would have needed to be adapted to forestry harvesting anyway.

The choice of using ROS for teleoperation development was inspired, and has allowed a degree of development that would have been much more difficult any other way. Acknowledgement must also be given to Trimble Navigation for making their ready-for-excavator equipment available to the program without charge. There are technical issues to be solved to implement it on a full size excavator, but due to the research outlined in this paper several of these issues have been identified, so steps can be taken during the design to resolve them. The most important issue is scaling the performance from the laboratory system to a full sized excavator. However, a proper understanding of the physical properties of the excavator should help in this task.

The greatest potential for applications of semi-autonomy in a logging operation include modification of the excavator control style (e.g. cylindrical coordinate control) and autonomous path planning. The reason is that control in cylindrical coordinates reduces the cognitive load on the operator and is extensible to steep country harvesting because the coordinate calculations can be compensated for machine slope automatically. Autonomous path planning with obstacle avoidance means an operator doesn't have to control a machine all the time, even when there are obstacles in the environment that would normally require manual intervention from an operator.