Low-cost mobile robot arm controlled through teleoperation server-client web system

Dominic Cassidy,
University of Plymouth,
Drake Circus, Plymouth,
PL4 8AA,
September 2016
dominic.cassidy@postgrad.plymouth.ac.uk

Abstract

Current mobile robot platforms capable of carrying weights up to 2kg and reaching heights normally faced within a home are expensive and overly complex for the task. Universities and people in need of at-home support, would benefit from a low-cost, open source robot with these capabilities. The use of a teleoperation, platform-independent control system would ensure availability without additional costs or reliance on changing operating systems.
Development will focus on minimising costs and robot complexity to ensure reproducibility for under £2000, while also creating a web server based control system.
The resulting system proved capable of lifting 2kg loads to over 1.5m. The gripper design required additional torque to ensure effective load grasp. The use of a platform-independent control system proved successful, and experimentation showed the importance of 4 degrees-of-freedom in manipulating objects.
In conclusion, a prototype system was effectively developed, with a low-cost easily controlled robot created.

Keywords

assistive robot wheelchair pulley teleoperation

1. Introduction

Access to robots for use in supporting humans is currently limited by cost. Robots are generally highly complex, and require expensive motors to enable the strength to move loads. The open source Katana robot arm used by the assistive 'El-E' robot (Nguyen et al., 2008) was reported to cost US$24,900 in 2008 (Robots.net, 2008). The Katana arm was considered open source (Robots.net, 2008), and the El-E robot aimed to be cheaper than carer staff.

In reality, this is not a financially viable option for those who need such a system. This is the primary reason why this issue must be solved. The cost of robotics must be lowered so that both the people that need them, and the research community, can access them. Through increased user experience, interaction and feedback, the pace of development of such systems can increase. Furthermore, the high cost of large scale robots limits the availability for Universities and education. A truly lost cost robot would even be capable of being introduced at earlier education levels.

By developing a tele-operated robot, complex machine intelligence systems can be avoided, keeping cost and complexity low, while giving additional control and independence to its users. Direct control was found to be preferred by users in Tsui and Yanco (2007), although also seen as more difficult. The implementation of a control system as platform-agnostic, through modern web technologies, allows hardware or platform requirements to be removed, increasing accessibility to users (Goldberg, 1995). The issue of compatibility was especially noticed in Goldberg (1995), when using image streaming within a system – with different technologies used by different operating systems. Furthermore, this removes dependencies and costs relating to the use of specific operating systems – such as development licenses.

Robots capable of effective support within the home, or within a care home require the strength and reach to access the array of objects that they may interact with on a daily basis. With this goal in mind, along with the simplicity of implementation targeted by this project, alternative methods of handling heavy weights must be examined.

Both Hillman et al. (2002) and Nguyen et al. (2008), chose to implement vertical arm traversal through a 1-DOF linear actuator. This simplified gripping motions as all objects, regardless of height, could be accessed from the same angle (Nguyen et al., 2008). However, both robots have key issues. First of all, the Wilson robot developed by Hillman et al. (2002), is designed for use with a wheelchair and lacks a manipulator capable of handling large loads. The El-E robot meanwhile (aside from cost), uses an additional 5-DOF robot arm as the gripper, also incapable of supporting large and heavy objects.

The novel approach developed by this research will focus on minimising complexity and cost. By using an omnidirectional base for XY movement, and a pulley system to enable vertical traversal (Z-axis), all possible movement axis will be achieved within the robot body, rather than the robot gripper. This design will enable the use of grippers designed to carry heavy loads, with the only required component of the gripper being a clasp to hold the item in place. A rotational ability will exist on the gripper so that items both perpendicular and parallel to a surface may be grasped.

With the robot body, not gripper, being responsible for all non-rotational movement, the control system must supply access to all three directional degrees of freedom at all times – with base rotation also beneficial. In comparison, the previously discussed robot systems could have separate controls for robot movement, and gripper manipulation. Therefore, an additional task is the development of a user interface that simplifies access to these movements, presenting an intuitive control system.

2. Method

The developed control system was created by Dominic Cassidy and centred around a Raspberry Pi and direct Dynamixel communication through two USB2Dynamixels.

Three independent sub-systems ran on the Raspberry Pi in parallel allowing full teleoperation of the robot by a user on the same Wi-Fi connection using any touch or non-touch device. The sub-systems were as follows:

- Web Server – Hosted HTML/CSS/JavaScript client teleoperation web page.
- Control Listener – Listened using a web-socket for client commands and passed these to the correct Dynamixels. Multiple conditional arguments were used in order to check the name of the incoming command and correctly process it (Figure 1 – "Control Handler").
- Stream Server – Hosted a live image stream (captured by OpenCV) from the camera mounted on the robot gripper.

The sub-systems were all built on the Python (2.7) programming language, with JavaScript used by the client for detecting user input and communicating these to the control listener.

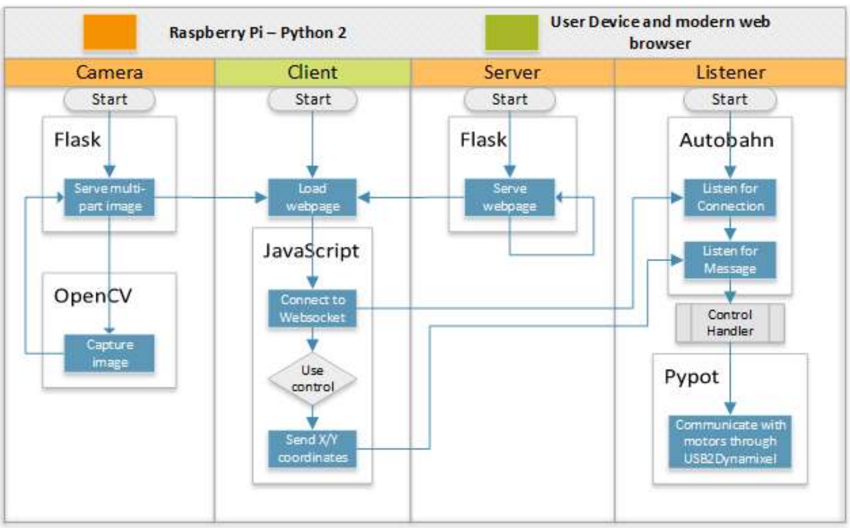


Figure 1 - Control system flowchart.

This simple system enabled high concurrence with limited bottle necks between systems. Each sub system was capable of running fully independently and simply waited for either passive usage (web and stream server), or direct inputs (control listener). As can be seen in Figure 1, multiple Python modules were used to enable this behaviour cleanly. Specifically, Flask was used for hosting the web server, while Autobahn enabled the control listener to directly connect to the client using a web socket. Finally, Pypot enabled direct communication with the used Dynamixels individually and as groups. The live web stream was accessed by the client-loaded web-page and presented on the user interface background.

The robot arm and base was controlled using dynamic controls which were only visible on the interface while in use. The user interface enabled access to 4 degrees of freedom including base rotation with only two input options. This was made possible by locking commands to specific events. Touch devices used the left and right sides of the screen for different controls, while mouse users could use left or right mouse clicks.

- Z-axis movement and base rotation – Left mouse (non-touch); left side (touch).
- XY-axis movement – right mouse (non-touch); right side (touch).

The left side control was unique as it was a vertical or horizontal slider dependant on the initial movement of the user. Movement would be checked after a short distance, then the slider would lock to this orientation until the control was released.

Gripper controls were static and remained on screen at all times to avoid confusion.

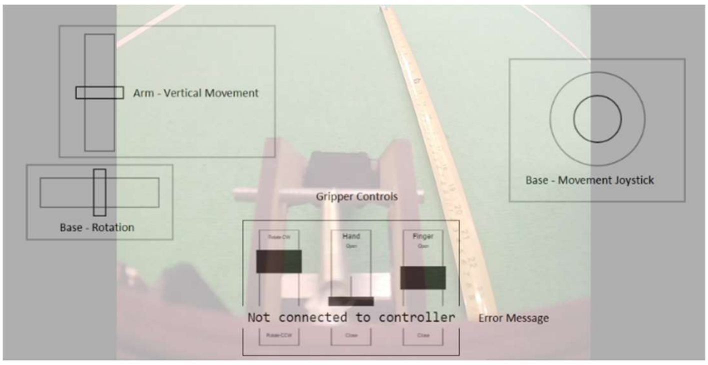


Figure 2 – Simulated Control UI.

The robot base consisted of three omnidirectional wheels equally spread around the base and was developed and constructed by Arunaganesan Swaminathan with the final design developed by Dominic Cassidy. The front wheels were 40cm apart from one another, with the rear wheel placed 40cm behind the front wheels (perpendicular).

The base was controlled by a joystick, with XY-axis movement available within a fixed radius of the user's initial touch/click. The XY values on the joystick where then sent by the client JavaScript (as seen in Figure 1) and managed by the control listener on the Raspberry Pi. The angle of each wheel towards the base centre was used in order to calculate the wheel velocities.

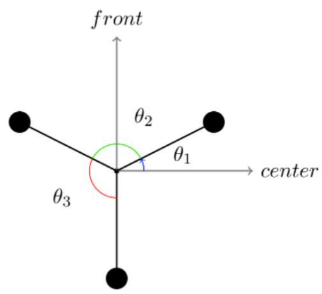


Figure 3 - Wheel positions and angles

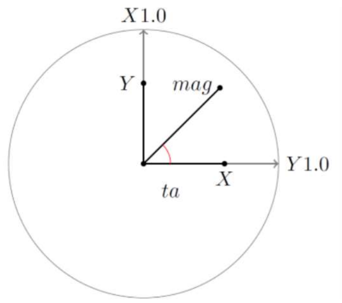


Figure 4 - Velocity calculation for wheels

Velocity was calculated for each individual wheel using the following sets of formulas, with θ equal to that of the current wheel (Figure 3), x and y being the result of user input (Figure 4):

$$ta = \arctan\left(\frac{y}{x}\right)$$
$$mag = \sqrt{x^2 + y^2}$$
$$velocity = mag \times \cos(\theta - ta)$$

The XY values were between -1.0 and 1.0. The value's angle showed the direction of travel, while the actual values were used to calculate the speed of travel through their hypotenuse.

The horizontal slider available on the left side/left click was used to apply rotation. A pre-set maximum speed was set within the control listener during development (chosen to avoid excessive speed). This maximum speed was multiplied by the slider position – passed as a value between -1.0 and 1.0 similar to the joystick.

Construction of the arm and tower system was completed by Guido Bugmann with support from Arunaganesan Swaminathan.

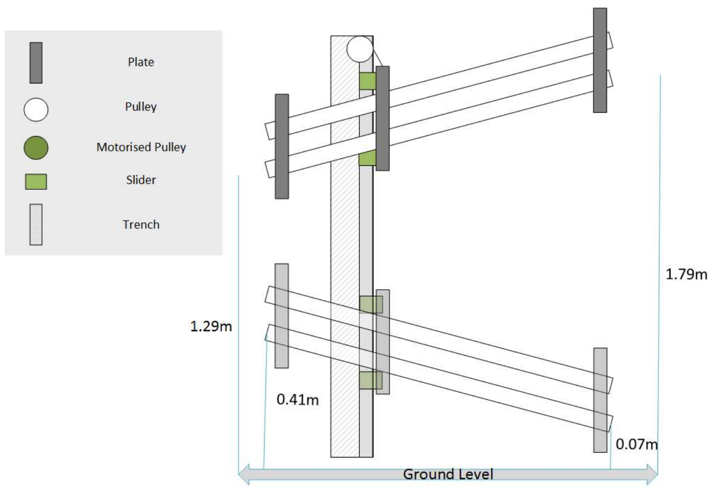


Figure 5 - Arm and tower movement concept and measurements.

Vertical velocity was achieved through the use of a double pulley system. This system applies pulling force to a cable through the use of an MX64 Dynamixel motor. This force resulted in the robot arm being raised. Control of the arm was managed by the vertical option of the 'left side' control, identical to rotation.

The gripper design, built by Kushdeep Singh Mann, also relied on simple pulley systems.

Gripper rotation was enabled through the use of a pulley, with a motor mounted on the arm. In order to control this, the Dynamixel used was given a set of minimum and maximum joint positions to enforce a maximum rotation of 90°. Moving between these positions was done by converting input slider position from the slider movement range (0-100) to the joint movement range. As a result, moving the slider to the half way point, would result in a 45° rotation.

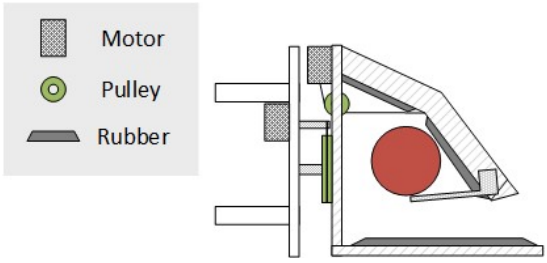


Figure 6 - Gripper design. Using both knuckle and finger components.

The final finger joint also operated in this manner, with a movement range given to ensure the integrity of the cables.

Finally, the rotation of the "knuckle" portion of the gripper also operated through a pulley, however this required the use of the speed based wheel mode – in comparison to the positional based joint mode. This was due to the pulley system loosening and resulting in the lack of fixed positions. Tightening and loosening of the robot was managed identically to that of arm vertical movement, with speed applied dependant on slider position within a range of -1.0 and 1.0.

3. Experiment

Testing of the robot was accomplished through the use of a pilot study followed by three recorded experiments.

Each participant was given a short (less than 2 minutes) visual introduction to the control system. This was in order to discover the intuitiveness of the controls.

The gripper was tested for its ability to handle objects of different shapes and sizes. Experimentation required three objects be picked up and moved a short distance.

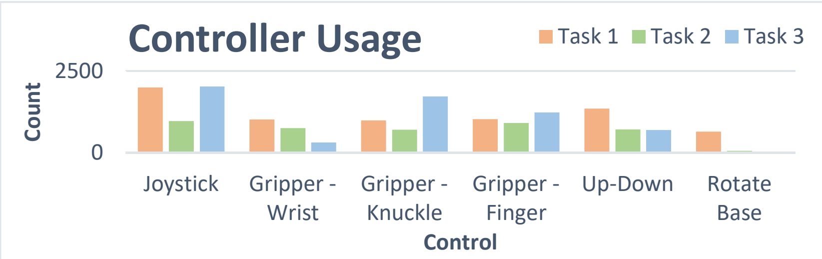
- Task 1: Roll of wire (200g)
- Task 2: 500ml bottle of water (600g)
- Task 3: Book (400g)

The three items used were of highly different shapes. This was done to test the abilities of both the gripper, and the user. The complexity of the objects required effective use of the robot's 4 degrees of freedom, and as found during the experiments, all 4 degrees were used in response. The participant was placed in a seated position perpendicular to the work area 1 meter from a table and the robot. Objects were placed on the table prior to each experiment, with the user moving them from the right side of the table to the left (with the robot starting position being perpendicular to the object). The enforced seated position was to increase reliance on the image stream, and remove advantages of improved visibility that could not be replicated between users. During the pilot study the gripper failed multiple times due to overload errors in the knuckle, and occasionally the finger. As a result, the main experiments required the researcher to be ready to power cycle the specific motors upon failure.

4. Results

Timed tasks showed mix results. With ¾ of participants failing to pick-up the book; which was a deliberately slippery surface and reflected certain weaknesses in the gripper. One participant managed to pick up the bottle within 40 seconds – compared to the other three participants who had an average time of 3 minutes 12 seconds. This result was due to superior technique (combined knuckle and finger use), and reflected the extent to which human participation may lower the effectiveness of the robot due to improper technique.

During each of these experiments no issues were found with the base or tower systems. The control system was reliable and worked as expected. The robot was capable of carrying all weights attempted, ranging from 100g to 2kg (during the pilot), and lifted these over 1.5m from ground level. Controller usage showed that all degrees of freedom were used throughout the experiments. A hardware failure (broken screws) resulted in the base being incapable of rotating, but otherwise was capable of full omnidirectional movement. No issues where found with the control system UI, although some confusion was seen in differentiating the right and left hand side controls. Controls were otherwise exact and highly responsive. The extent to which all controls were used is important as it shows that each degree of freedom was needed for object manipulation and therefore required constant access within the interface.



5. Discussion

The experiments showed strengths and weaknesses in the robot design. The base, tower and control software operated as expected, however the gripper showed areas were improvements were needed.

The key issue discovered within the gripper was that both the knuckle and finger were underpowered for the tasks, with the knuckle having the greatest need for redevelopment. The currently used MX28 is rated for a max stall torque of over 7N·m when supplied with 14V. However, at 11V this is limited to 5.3N·m, with a holding torque noted to be $\frac{1}{5}$ of the stall torque at approximately 1.06N·m. The user interface was found to be reliable with real-time responses from the hardware. The image streaming system worked as intended and reliably worked on multiple devices and operating systems. The camera placement limited visibility and would require future testing. An outdated Android tablet had issues showing the image stream within the Firefox web browser application – this issue was not replicated on the Chrome browser, or other devices.

6. Conclusion

The developed system effectively used low cost materials and motors while outputting the required lifting force to carry 2kg loads at up to 1.5m. With seven motors used throughout the arm and base, and each motor costing approximately £200, this mobile robot arm can be expected to cost, at least £1400 if constructed from scratch, and no more than £2000. The teleoperation system allowed reliable wireless control of the robot and its multiple degrees of freedom with limited hardware and software compatibility issues. It can be said that the use of web and cloud technology, rather than platform-specific and/or compiled applications can greatly simplify software systems, while allowing greater availability and lower costs.

The completed robot arm represents an effective prototype in the move towards low-cost worldwide availability of support robots. Through open sourcing this research, it is hoped that the design will be used as the basis for further research development.

7. Acknowledgements

I would like to thank Guido Bugmann, Plymouth University for all his work on this project and his support in developing my mechanical and technical knowledge throughout.
I would like to thank Kushdeep Singh Mann for developing and constructing the robot gripper.
I would like to thank Arunaganesan Swaminathan for his work in constructing the initial version of the base,

and developing the wheel control mathematics.
Finally, I would like to thank Vera Marmarova for all her support and patience during this project.

8. References

Goldberg, K., Mascha, M., Gentner, S., Rothenberg, N., Sutter, C. and Wiegley, J., 1995, May. Desktop teleoperation via the world wide web. In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on (Vol. 1, pp. 654-659). IEEE.

Hillman, M., Hagan, K., Hagan, S., Jepson, J. and Orpwood, R., 2002. The Weston wheelchair mounted assistive robot- the design story. Robotica, 20(02), pp.125-132.

Nguyen, H., Anderson, C., Trevor, A., Jain, A., Xu, Z. and Kemp, C.C., 2008, March. El-e: An assistive robot that fetches objects from flat surfaces. In Robotic helpers, int. conf. on human-robot interaction.

Robots.net. (2008). *robots.net - Linux-based Katana Intelligent Robotic Arm*. [online] Available at: http://robots.net/article/2700.html [Accessed 20 Aug. 2016].

Tsui, K.M. and Yanco, H.A., 2007, March. Simplifying Wheelchair Mounted Robotic Arm Control with a Visual Interface. In AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics (pp. 97-102).