UNIVERSITY OF CAMBRIDGE
Department of Engineering

Mobile Manipulator UROP Project

Author Name: Xinyi Jiang, University of Cambridge

Supervisor: Fumiya Iida

Introduction

As life expectancies increase and birth rates drop worldwide, it is no surprise that many countries face a rapidly ageing population [1]. Although this trend is usually seen as a problem for economics and healthcare, it should be rebranded as an opportunity for technology. Demand for caregivers already exceeds supply [2]; this shortage can be met if we embrace robotic appliances into homes requiring assisted living.

Most seniors in good health conditions prefer to age at home [3]. However, natural deterioration in mobility and fitness from ageing means they require more assistance in their living. Furthermore, seniors who live alone also tend to experience social isolation and lower quality of life, worsening their loss of independence [4]. Assisted living robots can ease the burden on physical tasks, provide company, and preserve one's standard of living, thereby allowing aged individuals to age with dignity in the comfort of their own home.

Despite the pressing need for more caregivers, human or not, robotic helpers are not widespread in homes. One reason may be the public's lack of awareness of such technology [5]. Concern about technical faults, negative preconceptions of robots, and eerie human-like attributes are all reasons that challenge its introduction [6]. Thus, acceptance of such assisted living robots would depend on how safe, reliable, practical, affordable, and intuitive the user interface is [7]. It is easy to miss the mark and create an unsuccessful product.

This UROP project explores possibilities of how a mobile indoor robot can support an aged individual to live independently by performing tasks typically carried out by their caregivers.

Application scenarios

There are four categories in which assisted living robots can be classified: domestic, social, medical and protection. Some applications may embody more than one category, and some applications are not limited to elderly assistance.

Indoor plant watering

From reducing stress [8] to improve air quality [9], the plethora of health benefits from keeping indoor plants makes a convincing argument for homeowners to have them in their houses. However, plants are also living organisms and need to be cared for; a common cause of plant demise is when their owners forget to water them. This problem may be exacerbated in cases involving age-related memory loss. Household robots can be adapted to take over the role of watering plants, so people no longer need to worry about dry plants.

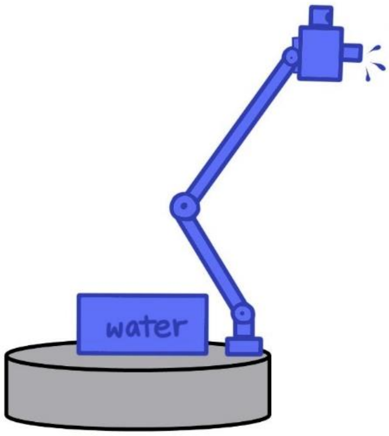

Figure 1 Sketch of concept - indoor plant watering

Category	Domestic
Suggested functions	• Self-charging and refilling • Autonomously navigates to a thirsty plant • Dispenses required amount of water for each plant • Users can take over with teleoperation
Implementation challenges	- Builds a map of its environment using SLAM which it can use later on for navigation - Have a water dispensing arm moves in 3D space to direct a stream of water while avoiding obstacles - Global and local path planning to navigate to plant but not collide with furniture - Object recognition of electronic devices, high accuracy of the water stream to avoid water damage to sensitive equipment - Livestream from the camera with decent fps, allows the user to take over via teleoperation
Market presence	- Various DIY Arduino-based solutions for simple autonomous plant watering devices, examples here and here

Away-from-home delivery handling

This application is not limited to older adults. Online shopping has become increasingly popular. However, sometimes people cannot be at the door when their orders are delivered, resulting in the failed delivery attempts for signed parcels or risk having unattended parcels stolen from the doorstep.

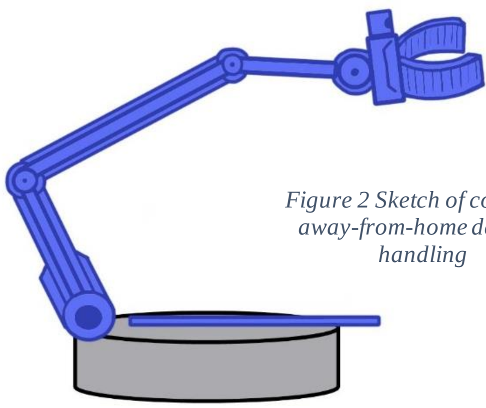

Figure 2 Sketch of concept - away-from-home delivery handling

Wouldn't it be nice if someone was always at home to receive packages? The menial task of waiting at home can be fulfilled with a robot dedicated to handling deliveries. That way, homeowners no longer have to bother with the hassle of rescheduling deliveries, going to pick-up lockers, or dealing with lost parcels.

Category	Protection, domestic
Suggested functions	• Greets the delivery man/woman at the door • Transports packages from the door to an allocated space inside the home • Ward off break-in attempts by delivery person • Report the status of delivery with a remote monitoring option
Implementation challenges	- The mobile base needs to be sturdy enough to handle heavy packages - Awareness of the additional challenges when manoeuvring through tight spaces with large payloads - Dextrous arm capable of skilful manipulation to unlock and open doors - Adaptive to handle different door types without any modification to its environment - Predict and recognise hostile body language using deep learning - Livestream from the camera with decent fps (similar to plant watering) for real-time monitoring of delivery status
Market presence	- eDOR (by eDOR delivery), the smart door delivery & security system - Amazon Key (by Amazon), an in-home delivery service - PR2 (by Willow Garage), a research robot that can open doors

Pet entertainer

To combat loneliness, seniors sometimes choose to have a pet as a companion through their older years. Dogs and cats are often treated like family members; their owners want them to be happy and stimulated. However, when it comes to physically demanding activities like throwing a ball or playing tug of war, their limited mobility means they have to give it a pass. Families away from their homes for long hours also do not have much time to play with their pets. The lack of stimulation may cause a pet to get into mischief to curb their boredom. Mobile robots may be the solution to preoccupy our four-legged friends.

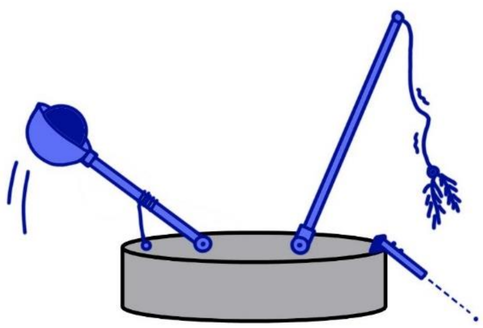

Figure 3 Sketch of concept - pet entertainer

Category	Social
Suggested functions	• Throw a ball, play tug of war – dogs • Point a laser, wave a tassel wand – cats • Camera to monitor pet's activity • Owners can speak to their pets remotely • Users can take over with teleoperation
Implementation challenges	- Remote audio and visual communication (similar to telepresence platform) - Rugged; built to withstand damage from curious pets, no parts that pose as choking hazards - Highly adaptive path planning to avoid collision with erratic movements of pets
Market presence	- iFetch, an automatic ball launcher for dogs - Ebo, a Kickstarter campaign for a smart robot companion designed to entertain cats - Furbo, a dog camera that can also give treats

Kitchen helper

It has been shown that adopting a healthy diet and active lifestyle is beneficial for ageing [10]. The responsibility of ensuring proper nutrition usually falls on the caregiver, but they are not always around and sometimes lack adequate nutrition knowledge [11]. Seniors can rest assured that their mealtime needs are met with a robot caregiver programmed to give the best possible nutrition advice. They can also be adapted to help prepare a meal in the kitchen. Moreover, seniors often need to take medication to treat age-related diseases. A robot helper can remind them to consume their medicines at the appropriate time (e.g. before or after meals).

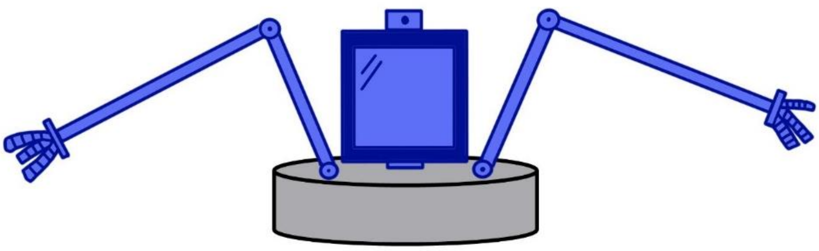

Figure 4 Sketch of concept - kitchen helper

Category	Domestic, medical
Suggested functions	• Assist in food preparation • Provide nutrition advice • Keep track of food intake • Suggest meal ideas • Medication reminder

Implementation challenges	-	Be up-to-date with the latest scientifically backed nutrition recommendations; requires a professional consultant to get the facts right
	-	Extensive database of recipe ideas with nutritional information
	-	Anthropomorphic manipulator arms to adapt to kitchens designed for human use
	-	Integrate the latest conversational agents and web services (similar to social companion)
	-	High-precision hand-eye calibration (similar to picker upper)
	-	Awareness of drug interactions
Market presence	-	Moley (by Moley Robotics), a fully automated robotic kitchen unit
	-	Mykie (by Bosch), a personal kitchen assistant for recipe inspiration

Picker upper

Lower back pain is a common complaint amongst seniors [12]; the disabling condition makes tasks requiring them to bend over particularly straining. Instead of further aggravating the pain, let a household robot do all the bending over instead. After all, robots should be of service to humankind. This feature can be used by anyone living with mobility impairment.

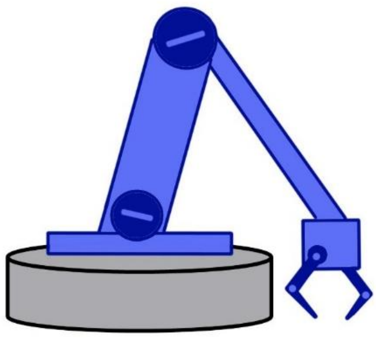

Figure 5 Sketch of concept - picker upper

Category	Domestic	
Suggested functions	•	Pick up objects that dropped on the floor upon verbal request
	•	Retrieve newspaper & letters from the front door
	•	Tidy away shoes at the doorway
	•	Users can take over with teleoperation
Implementation challenges	-	Dextrous arm capable of skilful manipulation to pick up objects (similar to delivery handling)
	-	Adaptive to handle different objects (similar to delivery handling)
	-	High-precision hand-eye calibration
	-	Overcome challenges in correspondence matching caused by texture, occlusion, and non-Lambertian surfaces
	-	Decisional autonomy between cage-closure vs form-closure grasping that best fits given object
Market presence	-	Bot Handy (by Samsung), a conceptual robotic partner to help with house chores
	-	Ugo (by Mira Robotics), a remotely-controlled robotic maid with a pair of height-adjustable arms

Targeted disinfectant

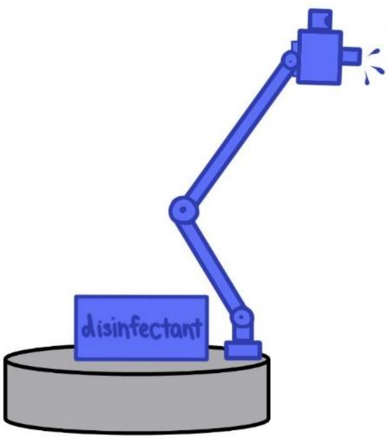

Figure 6 Sketch of concept - targeted disinfectant

As people age, their immune response weakens significantly [13], making seniors more vulnerable to harmful bacteria and viruses infections. Since these pathogens are invisible to the naked eye, they are impossible to detect. So how can homeowners keep pathogens at bay if they cannot see the enemy? The solution is to keep track of surfaces that are touched. People unintentionally harbour all kinds of microorganisms on their hands as they interact with things throughout the day (e.g. uncooked food, pets, bathrooms). If they do not wash their hands regularly, they will contaminate more surfaces with pathogens. This presents an opportunity for domestic robots. Although UV disinfection solutions can be used, their large footprint makes them better suited for spacious commercial venues. Consumers will feel more comfortable around a compact mobile robot with high specificity in cleaning. It will also provide peace of mind that their house is clean.

Category	Domestic
Suggested functions	• Monitor surfaces that people touch without being intrusive • Identifies surfaces that need to be cleaned and autonomously navigates to it • Disinfect high-touch areas with a sanitising solution • Manoeuvres around the insides of homes • Self-charging and refilling
Implementation challenges	- Method to detect occurrences of touching surfaces, either by latent heat using sensitive thermal cameras, or recognise object collision between hands and surfaces using depth cameras and pose estimation - Have thresholds for what constitutes a high touch surface, rejecting noise and false positives - Have a sanitiser dispensing arm moves in 3D space to successfully direct a stream of disinfectant while avoiding obstacles (similar to plant watering) - Global and local path planning (similar to plant watering) to trail behind a human so can monitor where they touch but not collide with them - Builds a map of its environment using SLAM for navigation (similar to plant watering)

	-	Image object recognition of electronic equipment to avoid spraying them
Market presence	-	UVD robots (by Blue Ocean Robotics), a robot that irradiates hospital rooms with UV-C to disinfect surfaces from pathogens

Social companion

As the great roboticist Cynthia Breazeal once said, "Relational AI is not just a tool that we use, but an empathetic, collaborative ally." Rapid progress in social robotics research means social robots are no longer science fiction. Robotic companions can improve mood, promote sleep, reduce anxiety, and support psychosocial needs [14]. The future is now; time to fill the void of loneliness with intelligent social robots.

Figure 7 Sketch of concept - social companion

Category	Social	
Suggested functions	•	Verbal and/or non-verbal communication capabilities
	•	Holds intellectually engaging conversations
	•	An expressive, friendly personality that grabs user's attention
	•	Well-being check-ins
	•	Provides emotional support in stressful situations
	•	Entertains by offering music, riddles, fun facts, etc
Implementation challenges	-	Share human-like qualities and follow social norms (e.g. maintain eye contact during conversation) so the robot is more comfortable to be around
	-	Should have expressive emotions and a focal area to direct conversation at, but not necessary to look like a human
	-	Manage expectations for learning and memory skills due to shortcomings in general-purpose artificial intelligence
	-	Requires an extensive repertoire of behaviours, so users do not get bored with repetitive responses
	-	Ability to read non-verbal cues, detect changes in tone and other unspoken signals
	-	Integrate the latest conversational agents and web services (e.g. Alexa, Siri)
	-	Convincing perceived sociability, ability to mimic natural human connection
	-	Well-placed haptic sensors to respond to touch sensations
Market presence	-	Mabu (by Catalia health), a wellness companion that holds tailored conversations
	-	Pepper (by Softbank Robotics), a humanoid robot optimised for human interaction

	- Palro (by fujisoft), a small humanoid robot that intellectually engages the user - AIBO (by Sony), a dog-like robot that interacts with users - Paro (by Sense Medical), an interactive therapeutic robot - Jibo (by NTT Disruption), a friendly robot assistant - Buddy (by Blue Frog Robotics), an emotional robot

Telepresence platform

Telepresence robots can move around a room and interact face-to-face with an individual. They can be a stand-in for healthcare consultants, providing diagnosis at the comfort of a patient's home, saving travel costs and time. These robots also offer the benefit of continual monitoring, as opposed to periodic check-ups at hospitals. In addition, video conferencing has become the norm of late. If loved ones want to contact the older person, they can call in via the telepresence platform and interact as though they are in the same room.

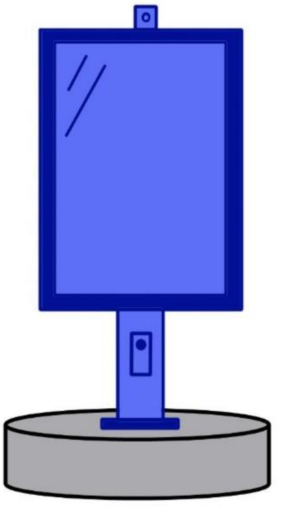

Figure 8 Sketch of concept - telepresence platform

Category	Medical, social
Suggested functions	• Tablet display with video and audio capabilities • Provide follow-along physical therapy sessions • Self-driving human follower • Healthcare worker check-ins • Fall alert
Implementation challenges	- Human gesture recognition (e.g. follow me signal) for intuitive user command input - Local path planning (similar to plant watering) to stay in front of a human without collision - Global path planning based on predictions of where humans will move next - Builds a map of its environment using SLAM for navigation (similar to plant watering)
Market presence	- Temi, the autonomously navigating robot with Alexa built-in - GEMS (by Samsung), a conceptual exoskeleton for muscle strengthening and posture training - Bot Care (by Samsung), a conceptual robot that specialises in helping users manage their daily health routine

House guard

Robots can give homeowners peace of mind with an extra set of eyes and ears to watch over their abode when no one is at home. This machine, equipped with an array of monitoring and defensive capabilities, will protect the house with devotion and loyalty, keeping it safe from intruders.

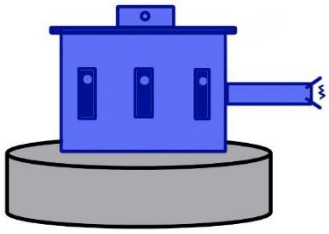

Figure 9 Sketch of concept - house guard

Category	Protection
Suggested functions	• Regular patrols around the house • Sensors (e.g. temperature, noise, motion) to detect the state of the house • Give house status updates, reports changes in its environment • Make home appear to be lived in when the family is away on vacation to deter break-ins • Non-lethal weapon to intimidate unwanted guests
Implementation challenges	- Global and local path planning (similar to plant watering) for patrol path - Builds a map of its environment using SLAM for navigation (similar to plant watering) - Predict and recognise hostile body language using deep learning (similar to delivery handling) - Awareness of room occupancy using motion detectors or heat signature of thermal sensors - May have legal repercussions if a fatal injury to an unwanted guest caused by a defensive robot
Market presence	- Carl (by design3), a conceptual robot guard - Ring, a smart security doorbell - JetBot 90 AI+ (by Samsung), an autonomous vacuum cleaner with home monitoring capabilities. - Kuri (by Mayfield Robotics), a now-defunct home companion with a personality

Technical details

Overview

Figure 10 Front profile of turtlebot

Turtlebot2i is an affordable open-source robot kit developed by Interbotix. The mobile base, manipulator arm, and array of sensors make it a versatile research and development platform.

It uses the Robot Operating System (ROS) for its communication tools and libraries. ROS is not a real operating system; it is a framework that allows abstraction of the hardware from the software and provides a standardised method for robotics software development. The turtlebot uses the Kinetic ROS distribution on the Ubuntu 16.04.7 LTS.

ROS architecture

Master –allows nodes to be named and registered so they can locate and communicate with other nodes in the ROS system.

Nodes – an executable program that performs the computation within a ROS package. They can be written in any language (e.g. C++, Python).

Topics – a channel over which nodes exchange unidirectional data streams (i.e. messages). Nodes can publish or subscribe to messages on a topic.

Services –a synchronous client/server architecture that allows one node to call a function executed in another node. It contains a pair of messages: one request and one response.

Actions - an asynchronous non-blocking client/server architecture used for client requests that take longer to complete.

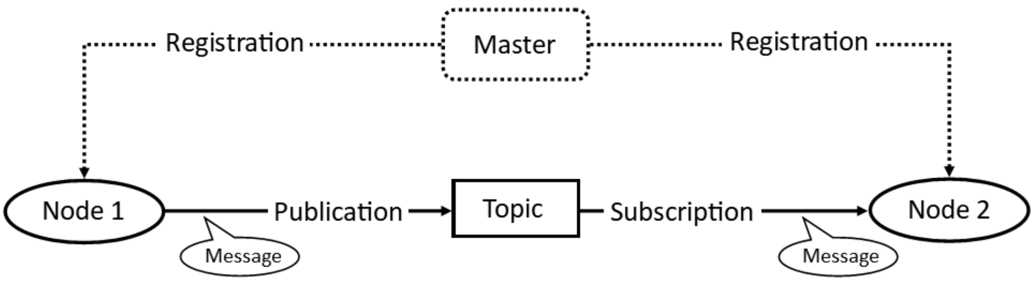

Figure 11 Example of communication between nodes

Components of the turtlebot2i

Figure 12 Side profile of turtlebot

- Kobuki YMR-K01-W1 – *mobile base*
- Intel NUC8i3BEH – *mini PC*
- Orbbec Astra – *long-range depth camera*
- Intel Realsense D415 – *short-range depth camera*
- PhantomX Pincher AX-12 – *robotic arm*
- Trossen Robotics ArbotiX-M – *microcontroller*
- Mini-box OpenUPS – *power regulator*
- 4400 mAh Lipo battery – *additional power source*
- Sabrent HB-BUP7 – *USB 3.0 hub*

Manipulator modification

The original turtlebot arm was located near the base of the robot. This location was disadvantageous as its vertical reach was limited to the length of the arm. In some situations, it would also obstruct the view of one of its cameras.

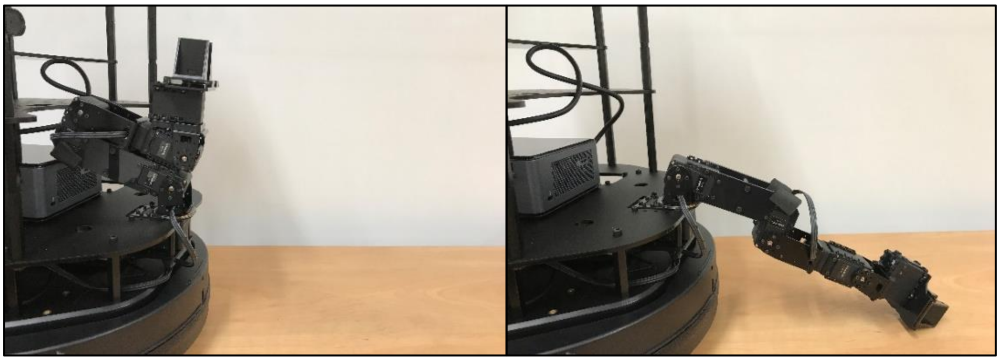

Figure 13 Retracted arm position (left), extended arm position (right)

Hence, the arm was redesigned to improve its location and simplify the build. The new setup reduced the number of servos from five to three, sufficient for the task required.

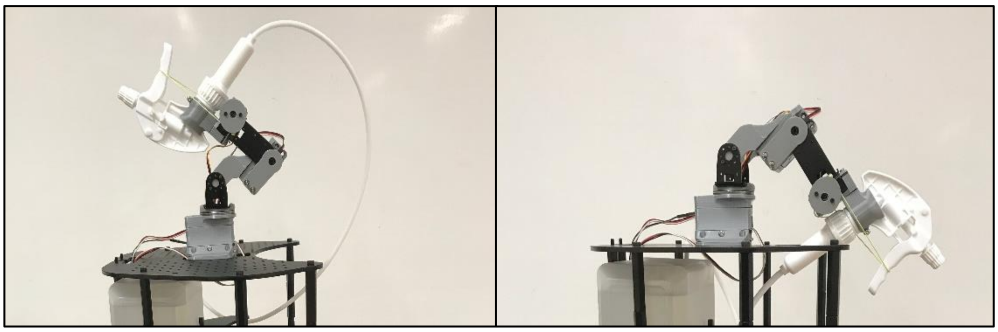

Figure 14 Range of motion in the pitch direction

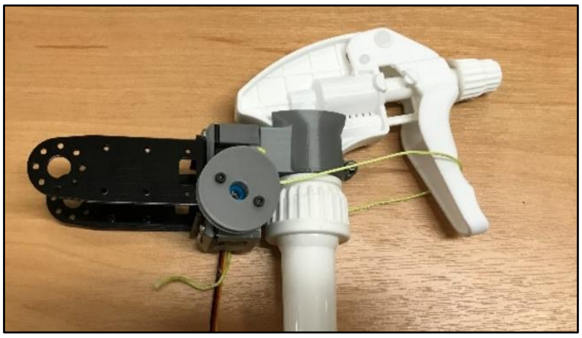

Figure 15 Modified end effector

Two hobby servo motors are used to control the pitch and yaw of the end effector. The final micro servo is used to squeeze a spray pump. The pump has an elongated tube that reaches into a water reservoir on the main body of the turtlebot.

Figure 16 New look of the turtlebot

This setup has the advantage of relieving load from the arm, which can only carry a maximum of 500g. In addition, the water reservoir is easily accessible to be refilled.

All three servos are connected to an Arduino Uno fixed onto the lowest platform, connected by USB to the NUC. The servos obtain their power supply from a 5V, 1A port on the Kobuki base.

The overall height of the turtlebot was also slightly reduced to lower the centre of gravity and make the robot more stable. Since the second depth camera (Realsense D415) will not be used, it was removed to make space for the manipulator.

How to operate the turtlebot

Starting up

1. Ensure the turtlebot is charged before operating. It is a good idea to leave the turtlebot plugged into a power source overnight so it will be fully charged when one uses it.
2. There is a black rocker switch on the side of the Kobuki base, next to the charging port. Make sure it is flipped to "On".
3. Next, locate the two toggle silver switches on the first platform of the turtlebot. Ensure both switches are flipped towards the label (see figure).

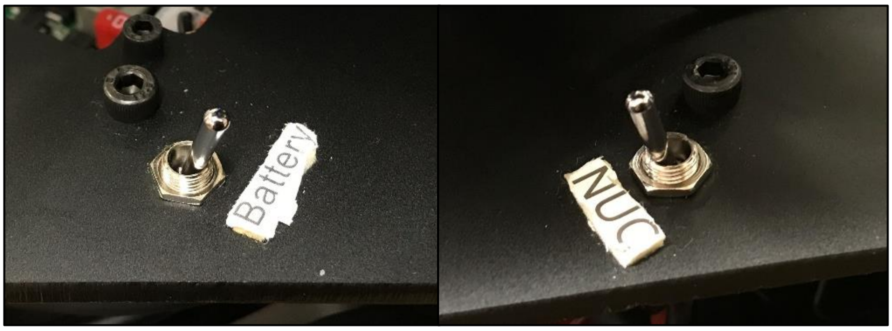

Figure 18 Position of toggle switches

4. Power on the NUC. If the blue power indicator does not light up, check that the barrel power connector is inserted correctly as it tends to come loose.
5. Finally, check that the switches on the USB hub are pressed in for all the USB ports that are in use.

Figure 17 Powered on NUC

Remote access

1. The robot's onboard PC can be accessed by a remote laptop via SSH. Both computers should be connected to the same network - eduroam[1]. You may need to use a monitor, keyboard, and mouse on the NUC to ensure the turtlebot is connected.
2. Open a terminal window (Ctrl+Alt+T), then type in *ssh turtlebot@robot.ip.address -X* where robot.ip.address is replaced with the turtlebot's IP address. When prompted for a password, use *turtlebot*.

Commands

- **To start all ROS systems:**

    roslaunch turtlebot2i_bringup turtlebot2i_demo1.launch [2]

    Optional arguments:

    - *new_rtabmap:=true* builds a new RTAB (Real-Time Appearance-Based) map of the environment; erases data from the previous mapping session
    - *localization:=true* if a valid map was built in a previous session, it can be used to determine the robot's position with respect to its environment

[1] The IP address of the turtlebot is known to change every time maintenance work is carried out on the server. In this case, use the command *sudo gedit ~/.bashrc* in the turtlebot's terminal and update the line *export ROS_MASTER_URI=http://my.new.turtlebot.address:11311*

[2] This must be running at all times when the turtlebot is in operation. Each new command should be run in a separate terminal.

- **To visualise the generated map and see the camera input using rviz[3]:**

  roslaunch turtlebot2i_bringup remote_view.launch

- **To control the turtlebot using teleoperation:**

  roslaunch turtlebot2i_teleop teleop.launch

- **To set the turtlebot in wall following mode:**

  roslaunch turtlebot2i_nav wall_follower.launch

- **To initiate autonomous plant watering sequence:**

  roslaunch turtlebot2i_water water.launch

- **To initiate autonomous plant watering sequence:**

  roslaunch turtlebot2i_water water.launch

- **To instruct the turtlebot to water a specific plant (e.g. plant 3):**

  rostopic pub /plant_destination_request std_msgs/String "plant 3" –once

Demonstration of autonomous plant watering

From the list of application scenarios, the plant watering ability was explored further in this UROP. The goal was to extend the capabilities of a turtlebot2i mobile robot to water plants on its own. All plants in this demonstration are stationed on the floor.

Teleoperation

A setup step is required before the autonomous plant watering feature can be used; the turtlebot must physically explore all areas of its environment to build an accurate map and find the houseplant later on. This exploration is achieved using teleoperation.

Teleoperation mode gives the user remote control of the turtlebot for the setup sequence, allowing them to drive the mobile robot around the house using a keyboard. The turtlebot can be monitored with rviz, which displays the camera input and RTAB map. The user can also take snapshots from the POV of the turtlebot and stored them on the drive.

[3] rviz (ROS visualisation) is a graphical interface that displays various information about the robot's state and can show how the robot perceives its 3D world.

```
Control Your Turtlebot!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
p   : take a snapshot
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit
```

Figure 19 Keyboard controls for teleoperation

If mapping was done diligently, all plant markers are guaranteed to have been registered and stored in its map. The map information is used during the navigation sequence to locate a given plant.

Plant markers

Figure 20 The plant markers

The turtlebot is trained to remember up to four plant markers. April tags were initially suggested as placeholders for plants but later changed to depictions of plants to preserve similarity to the actual object. The markers were placed around the room adjacent to potted plants, at roughly the same level as the turtlebot's camera. Characteristics like points, lines, edges, colours, and their relative positions help with their identification.

Recognition is handled by a ROS package called find_object_2d. Desired images are registered and saved using a setup wizard. When the turtlebot is running around, its cameras constantly process every frame, extracting image features and comparing them to the set stored in its

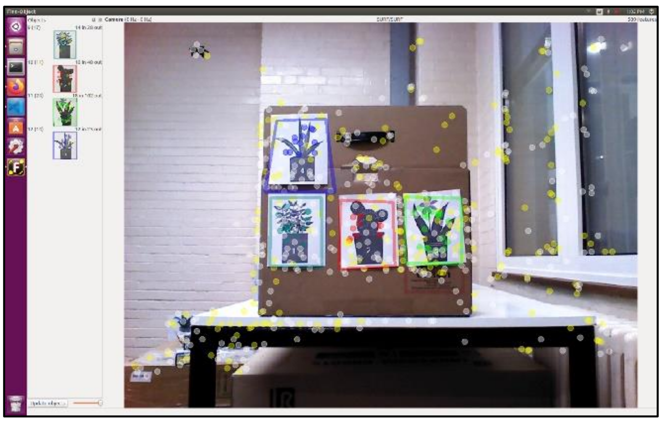

Figure 21 Registering markers using a setup wizard

memory. If enough features match, the marker is identified. The package publishes a homography matrix for each recognised marker, giving its position on the map.

Navigation

The ROS Navigation Stack is a collection of software packages that drives a mobile base from its starting position to a goal location while avoiding obstacles. It uses information from odometry, sensors, and the environment map to guide its decisions.

The turtlebot uses an RGB-D based SLAM[4] approach to build its RTAB map. When launched, the navigation stack generates a dense 3D point cloud of the environment and creates a 2D occupancy grid. Coordinate frames are also transformed to put them into perspective relative to the global frame to be tracked over time.

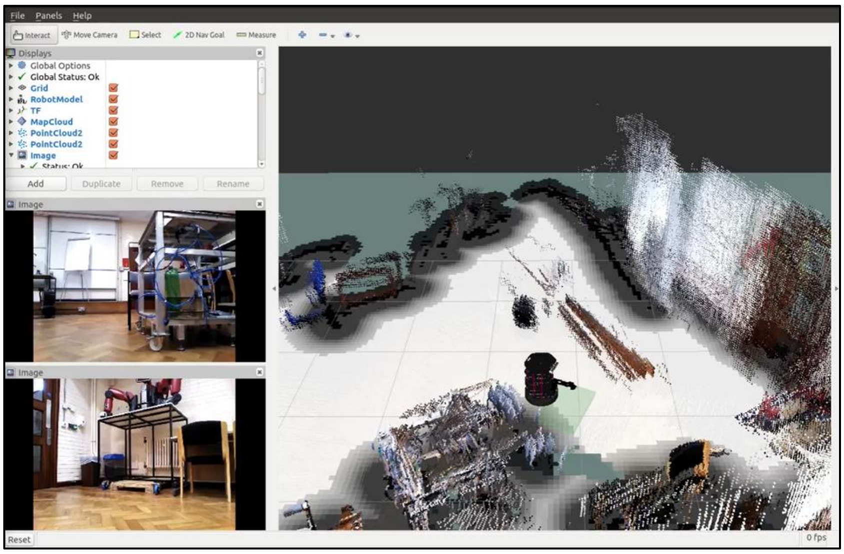

Figure 22 Screenshot of a rviz window on the remote PC

Both global and local costmaps[5] are used. The global planner generates the shortest path from start to goal, while the local planner avoids obstacles in its immediate environment. When the turtlebot receives a plant destination goal, it will calculate a trajectory based on the desired goal and planner information. Velocity commands are then sent to the mobile base as instructions for how the robot should move to reach the goal.

[4] Simultaneous Localisation and Mapping (SLAM) is the process of constructing a map of an unknown environment while at the same time navigating through the environment using the map. It is often known as the "chicken and egg" problem, because the robot needs a map of the environment to estimate its own location, but it needs to know its own location to generate the map.

[5] costmaps are 2D maps that reflect the difficulty in traversing certain areas of the environment due to obstacles. The obstacles usually have their radii inflated to prevent the robot from getting too near and colliding.

Watering

Once the turtlebot has navigated to the plant, it would be in close proximity (~ 0.5 m) to its watering target. The arm needs to be positioned from its retracted state to aim at the plant.

To improve the aim, OpenCV was used to detect and draw a bounding box around a green object in the scene. Since the turtlebot is very close and already facing the plant, it is reasonable to assume that the largest green object in the scene is the plant.

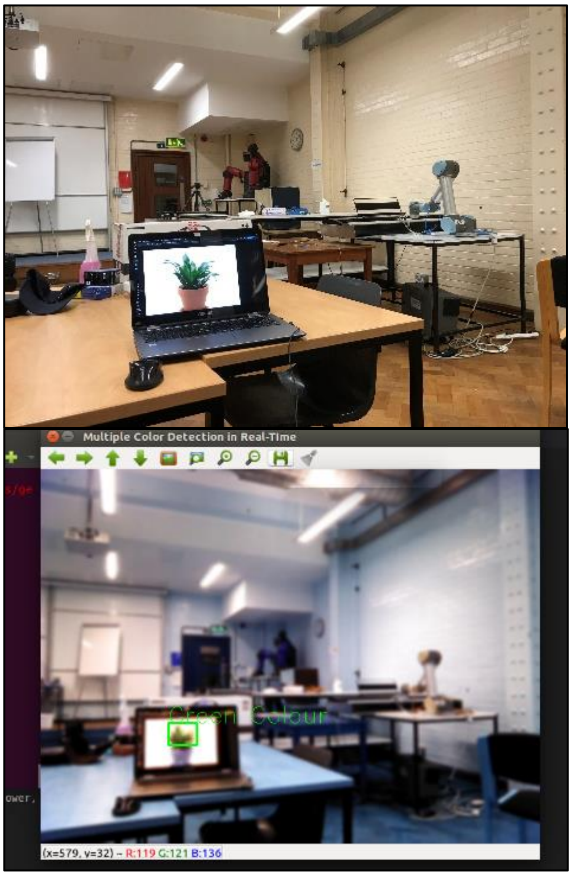

Figure 23 Picture of scene (top) vs what the turtlebot sees (bottom)

The depth and location of the centre of the bounding box are used to guide the positioning of the pitch and yaw servos, increasing the likelihood that a stream of water from the pump will reach the pot.

Once the pump dispensed a fixed amount of water, the arm is retracted to keep it out of the way. Then, the turtlebot moves on to the next plant until all plants have been watered.

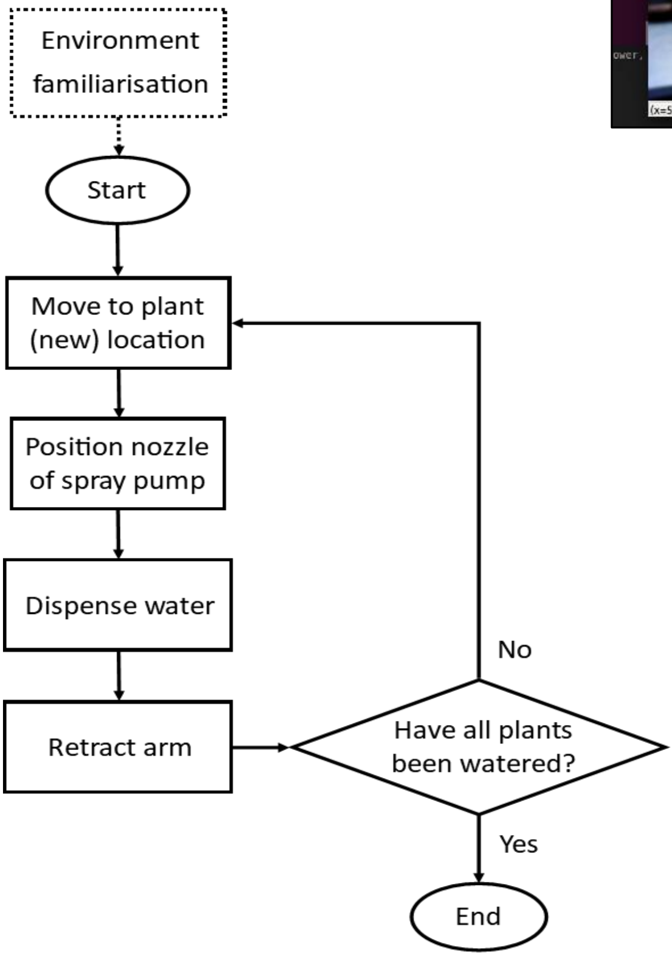

Figure 24 Flowchart of plant watering sequence

Discussion

Challenges faced

The biggest hurdle was to get the arm to move. The original pincher arm made from five Dynamixel servos is controlled using the ROS MoveIt package. The motion planning framework creates necessary trajectories to move the end effector to a goal position. However, MoveIt is notorious for being difficult to use. After spending weeks of fruitless effort trying to control the arm using MoveIt, it was still not working; thus, an alternative was sought. A simplified arm was rapidly prototyped from widely available components. The reduced complexity made it much easier to control.

Initially, an attempt to create a navigation method from scratch was made based on the occupancy grid map and Dijkstra's algorithm. However, time constraints, improved functionality, and convenience made using the ROS navigation stack more practical. This example highlights the merits of ROS; standard packages exist for common robot functions, so one does not have to reinvent the wheel each time.

Areas for improvement

When the arm was moved to the top of the turtlebot, it was overlooked that the spray pump would be immediately above the exposed circuit boards. As a result, any dribble from the nozzle can risk permanent damage to the electronics. A simple solution would be to build a cover that shields the sensitive components from accidental water splashes.

One limitation of the mobile base is that it can only climb thresholds of 12 mm or less. Thick rugs and carpets are out of the question since the turtlebot will get stuck on them. Stray power cords can also obstruct the robot's path. Since these height changes are small, it is difficult for the camera to pick up the obstacle. Better algorithms that look for texture changes on the floor surface can reduce this problem. A dedicated obstacle sensor positioned near the floor is another possible solution.

The turtlebot does not perform well in narrow spaces. This trait can be attributed to the generous inflation of obstacle boundaries, which is great for preventing collisions, but overlap in tight spaces. The radius of obstacles can be fine-tuned to traverse narrow spaces but still avoid collision with obstacles.

When the turtlebot approaches a plant marker, it is indiscriminate and treats the largest green object in its view as the watering target. However, other green coloured objects can be in the scene (e.g. green sofa, green wallpaper), or the plant can have non-green leaves. A more sophisticated method to differentiate potted plants from their environment would be using OpenCV and YOLO (You Only Look Once), a real-time object detection algorithm trained on the COCO dataset.

Future work

The current plant watering sequence is executed on command by the user. A fully developed smart plant watering system would be capable of monitoring the state of the plant so that the plants would be tended based on their condition. Each potted plant will have a module containing sensors that measure soil moisture, temperature, and humidity. Information about the plant's state can be transmitted to the mobile robot, which decides when and how much it should water depending on sensor data and the type of plant involved. In addition, a docking station can be used to make operations more hands-off where the turtlebot can charge itself when the battery is running low and refill its water reservoir.

If human involvement can be removed from the setup sequence, it would add another layer of autonomy. A wall following algorithm was considered as an alternative method to teleoperation to perform autonomous exploration of the environment during the setup sequence. The assumption was that indoor potted plants are usually located along the walls, not in a room's centre. However, when testing the algorithm, the robot would not face the plant markers head-on while following the wall. As such, it often missed the markers and failed to identify their location on the map. Therefore, other methods should be explored.

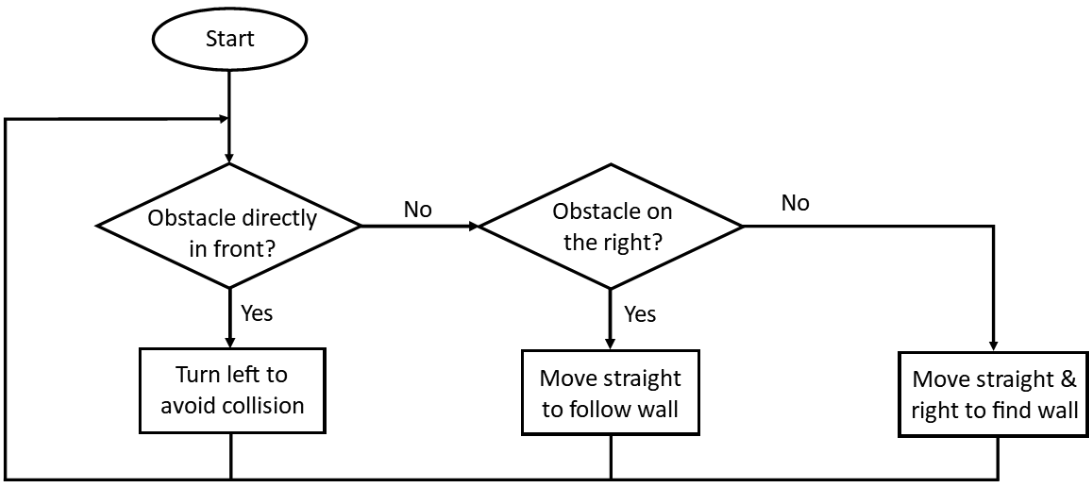

Figure 25 Wall following algorithm

References

[1] U. N. Department of Economic and Social Affairs, “World Population Ageing 2019,” United Nations, 2020.

[2] K. C. Fleming, J. M. Evans and D. S. Chutka, “Caregiver and Clinician Shortages in an Aging Nation,” *Mayo Clinic Proceedings,* vol. 78, no. 8, pp. 1026-1040, 2003.

[3] C. Fernández-Carro, “Ageing at home, co-residence or institutionalisation? Preferred care and residential arrangements of older adults in Spain.,” *Ageing & Society,* vol. 36, no. 3, p. 586–612, 2016.

[4] E. Escourrou, M. Cesari, B. Chicoulaa, B. Fougère, B. Vellas, S. Andrieu and S. Oustric, “How Older Persons Perceive the Loss of Independence: The Need of a Holistic Approach to Frailty,” *J Frailty Aging,* vol. 6, no. 2, pp. 107-112, 2017.

[5] A. Smith and M. Anderson, “Automation in Everyday Life,” Pew Research Centre, 2017.

[6] I. Papadopoulos, C. Koulouglioti, R. Lazzarino and S. Ali, “Enablers and barriers to the implementation of socially assistive humanoid robots in health and social care: a systematic review,” *BMJ Open,* vol. 10, no. 1, 2020.

[7] R.-M. Johansson-Pajala and C. Gustafss, “Significant challenges when introducing care robots in Swedish elder care,” *Disability and Rehabilitation: Assistive Technology,* vol. 0, no. 0, pp. 1-11, 2020.

[8] M.-s. Lee, J. Lee, B.-J. Park and Y. Miyazaki, “Interaction with indoor plants may reduce psychological and physiological stress by suppressing autonomic nervous

system activity in young adults: a randomized crossover study," *Journal of physiological anthropology,* vol. 34, no. 1, p. 21, 2015.

[9] B. Wolverton, A. Johnson and K. Bounds, "Interior Landscape Plants for Indoor Air Pollution Abatement," National Aeronautics and Space Administration , 1989.

[10] N. Davies, "Promoting healthy ageing: the importance of lifestyle.," *Nursing standard ,* vol. 25, no. 19, pp. 43-50, 2011.

[11] C. Ryan, "Caregivers of the Elderly, Lack of Nutrition Knowledge," *Journal of Nutrition For the Elderly,* vol. 17, no. 2, pp. 35-44, 1998.

[12] A. Wong, J. Karppinen and D. Samartzis, "Low back pain in older adults: risk factors, management options and future directions," *Scoliosis and spinal disorders,* vol. 12, no. 14, 2017.

[13] S. A. Katharina, H. Georg A. and A. McMichael, "Evolution of the immune system in humans from infancy to old age," *Proceedings of the Royal Societ B: Biological Sciences,* vol. 282, no. 1821, 2015.

[14] L. Hung, C. Liu, E. Woldum, A. Au-Yeung, A. Berndt, C. Wallsworth, N. Horne, M. Gregorio, J. Mann and H. Chaudhury, "The benefits of and barriers to using a social robot PARO in care settings: a scoping review," *BMC Geriatrics,* vol. 19, no. 232, 2019.

Appendix

ROS resources

ROS tutorials - Official: http://wiki.ros.org/ROS/Tutorials

ROS tutorials - Husarion: https://husarion.com/tutorials/ros-tutorials/1-ros-introduction

ROS tutorials - The Construct: https://www.theconstructsim.com/about-ros-robot-operating-system/

ROS tutorials – Clearpath Robotics: http://www.clearpathrobotics.com/assets/guides/melodic/ros/

ROS tutorials - YouTube Robomechtrix playlist: https://www.youtube.com/channel/UC5ZQinPsJ4C8YiauoT8xZUg

ROS kinetic cheat sheet: https://clearpathrobotics.com/ros-robot-operating-system-cheat-sheet/

Find_object_2d package tutorial: https://discourse.world/h/2018/06/26/Detection-and-recognition-of-objects-from-the-camera-in-ROS-using-the-find_object_2d-package

Arduino & ROS interfacing tutorial: https://maker.pro/arduino/tutorial/how-to-control-a-robot-arm-with-ros-and-arduino

Turtlebot2i resources

Github: https://github.com/Interbotix/turtlebot2i

Wiki: https://github.com/Interbotix/turtlebot2i/wiki

Body assembly guide: https://learn.trossenrobotics.com/projects/189-turtlebot2i-assembly-guide.html

Arm assembly guide: https://learn.trossenrobotics.com/projects/193-turtlebot2i-pincher-arm-assembly.html

Custom packages made for this project: https://github.com/J-248/UROP-2021.git