SPEAK BOT: SPEECH TO ACTION MODEL FOR ASSISTIVE MOBILE ROBOTS

Module Code: ENGD3000

Student Name: David Jeffreys

Supervisor Name: Sunny Katyara

Student ID: P2655839

Course: Mechatronics BEng

Academic Year Completed: 2024-2025

Abstract

Considering the exponential technological advances in autonomous robotic systems along with the imperative demand for day-to-day assistance aimed at an ageing global population, the need for a system that can contribute to an improved lifestyle for the elderly and physically impaired – effectively reducing the physical labour associated with common tasks – is emphasised. This paper will aim to provide a proof-of-concept for a speech-controlled robotic manipulation system utilising OpenAI's automatic speech recognition (ASR) API, 'Whisper', to generate actionable commands that will be configured to drive a 3WD on-demand mobile robot and Open Manipulator X robotic arm using NLP toolkits on a platform agnostic framework. With the main challenge surrounding the conversion of decoded verbal stimuli into logical commands that can be compiled, this project will propose the implementation of an 'Action Verb' (AV) toolkit, capable of identifying transitive dynamic verbs categorised to domain specific semantic fields within the low-level user-inputted statement, scanning through a pre-populated register to match the verbs with pre-determined responses in compilable action script and generating an output result for the operation of the respective robot. Implementation of a summarisation layer derived from the GPT 3.5-Turbo LLM will enable user request key topic apprehension and urgency weighting. Developed within the open-source Visual Studio Code IDE prior to integration into the ROS Noetic environment, usage of the AV toolkit will enable effective communication with relevant exteroceptive and proprioceptive peripherals on the omni-directional robot. Key functionality and performance features will be addressed and discussed, with emphasis placed on the safe integration of the domestic assistance robot into the home of an elderly/disabled individual. Testing and analysis of proof of concept will be undertaken within the Gazebo simulation environment.

Keywords: Action Task; Whisper; DWA Planner; ROS Noetic; Natural Language Processing, Prompt Engineering, Python, RegEx

Acknowledgements

I would like to express my deepest gratitude to my project supervisor, Sunny Katyara, who suggested the project concept and supported me in bringing the concept to fruition. From providing guidelines and technical material, to aiding in the generation of pseudocode/actionable script and constructive criticism, you have truly played an imperative role in the creation and success of this project and the development of my technical knowledge in the robotic and artificial intelligence space.

Many thanks to Seng Chong for suggesting Sunny as a suitable supervisor after hearing my aspirations for a robotics project, and Dr Michael Oliver for proposing the use of Python instead of C++ as my OOP language due to string manipulation advantages obtained with use of the former. I am also grateful to Dr John Gow, for instilling several embedded system concepts that aided the design process for my action script, including aspects of computational latency, real time system consideration and efficient utilisation of RAM.

I'd like to acknowledge Dr Nicholas Humprey, for overseeing the individual project module for 2024-25, offering profound support in the initial stages, ensuring that the material required for success within the module was available.

I would be remiss in not mentioning my family. I am grateful to my Dad for providing technical insight, support and solutions when I was uncertain in my thought processes, and to my Mum and sister for providing emotional support and care when stress levels were high.

Contents

Table of Figures

Table of Tables

Introduction

From implementation of decentralised swarm architecture in drone control to incorporation of efficient fruit picking robots within the agricultural sector, the global transition into the robotic era is becoming inconceivably prominent [1], [2]. Encompassing most of the 21st century, the notion of 'smart' robots has evolved from a hypothetical notion to a well-grounded reality, with significant levels of investment being allocated to the research and development of robust, collaborative & assistive robotic technology. The German government provide a noteworthy example of this incentive, allocating 69 million USD to robotic research and development annually until 2026 [3], a testament to the global migration towards the fourth industrial revolution. Considering current advancements in robotic subsystems, efficient PCB topology, ever-evolving artificial intelligence (AI) API models and standardised usage of open-source development frameworks including ROS (Robot Operating System), the global drive towards a future that is not only designed and developed by the synergy between humans and robots but also nurtured and sustained, is well justified. Such synergy can only be facilitated via effective communication between both entities, a dilemma aptly addressed by the integration of natural language processing (NLP) in robotic system architecture. Fostering trust and natural interaction between the robot and human parties through usage of NLP will ensure a smooth transition of robotics into various domestic settings; most notably within the health and social care sector, wherein the scope of this project will partially be derived.

Recognising the opportunity for sustainable development in the robotics sector is paramount in ensuring long-term product feasibility – the onus is placed on product designers to derive solutions for current problems that will still serve societal constructs in the far future [4]. With primary focus on the UN Sustainable Development Goal (UNSDG) 3.4.2, the project will aim to address this responsibility by indirectly converging on reduction in suicide mortality rates. Considering humanity's rapidly ageing population, inevitable contention with medical conditions pertaining to impaired movement have become prominent - as stated by the World Health Organisation, occurrence of symptoms ranging from moderate neck and back pain to osteoarthritis are common in older age groups [4]. Conversely, a study by the Office for National Statistics highlighted that higher levels of suicidal deaths were noticeable amongst disabled people compared to other non-disabled groups, as a greater number of disabled individuals felt incapable of living by themselves leading to inflated depression levels amongst this demographic. In a similar manner, one can contemplate the report in [5], where participants of a study in 2011 that reported limitation in their day-to-day activities due to medium-long term health problems, experienced higher rates of suicide in 2021. Notably, a key denominator between both instances is a notion of independency, with elderly and disabled individuals feeling unable to complete daily tasks due to their respective ailments. Considering the United Kingdom, many disabled homeowners and their carers stated that their homes lacked adequate requirements for independent living, emphasising the significant adjustments that needed to be undertaken to facilitate accessibility requirements. 53% of the disabled participants that were surveyed in report by the UK Government Disability Unit stated that their home met their needs to live independently. Whilst the report did not factor in national metrics (highlighting the underrepresentation of older disabled population within the UK) this key metric highlights the need for improvement of assistance in domestic settings for elderly disabled people [6]. Thus, as Maresove et al emphasizes in [7], the importance of investing in smart home technology and assistive robots for improved accessibility and independence for seniors is greatly accentuated. The directive for this project will aim to achieve this as a primary goal.

Whilst investigating the broader scope of the project, a consideration will be made for the UNSDG goals 9.2 and 8.2.1, relating to the inclusivity of disabled persons within manufacturing environments and increase in the output achieved per disabled worker. With functional capabilities provided via use of a verbally controlled mobile manipulator – as will be proposed in this project – disabled individuals can systematically integrate into dynamic manufacturing lines, utilising the controlled robots as aides for acquisition of components and tools. Thus, satisfaction levels within the disabled worker demographic may alleviate, inherently reducing depression levels and improving net value generated by each disabled worker.

The principal aim for this paper is to propose a proof-of-concept for a voice-controlled mobile base and manipulator that can complete on-demand task processing and completion, leveraging an ASR API and 'Action Noun' functional dictionary for target word identification.

Existing Literature

Employment of natural language understanding (NLU) algorithms within the collaborative robot (cobot) space is imperative to building and sustaining effective human-robot communication. Iterative development of large language models (LLM) such as the GPT model series have fostered extraordinary technological progressions for modern day applications including text-generation, automatic speech recognition and system dialogue generation [8]. Moreover, the adaptability of these algorithms offers manipulation of the model performance when trained with input data unique to expected use-cases. As stated by Radford et al in the Whisper model proposal [9], robust effectiveness of the audio encoder/decoder pipeline used in the transformer-based large-language model is partially dependent on the adhered-to protocol of data-specific fine-tuning, without which the functionality of the model is mitigated, but not completely nullified. With respect to use-cases where the functionality of the robot is tested in multiple domains ranging from domestic to industrial, this factor of fine-tuning is of great importance to the safe deployment of efficient systems driven by human-machine interaction. More than a simple notion, the rationale behind the importance of fine-tuning has become a distinctive point of research. Lu et al reinforces this fact, proposing an alternative LLM development pipeline in which the base model undergoes the standardized Continued Pre-Training, Supervised Fine-Tuning and Direct Preference Optimization processes, only to be merged with other fine-tuned LLMs via Spherical Linear Interpolation (SLERP) to generate an output trained model with language processing capabilities previously unattainable by the parent models, open-source models Llama 3.1 & Mistral v0.3. Baseline model and post-merge results convey the effectiveness of the proposal distinctively, with the merged models exhibiting extensive performance gains that exceeded the baseline accuracy results. Notably, however, the recorded performance was obtained from models with modest parameter sizes, specifically in the 7-8 billion range. Implementation of SLERP within models enforcing lower parameter sizes (approx. 1.7B parameters) provided output results that were void of the distinctive performances, potentially highlighting a limitation with the SLERP merging process and its ability to extract amplified abilities when considering model size[10]. Considering the use of the Whisper-tiny.en (on the local machine as opposed to the whisper-1 API equivalent), due to reduced VRAM requirement and increased reasoning speed, further general-purpose model merging would be ineffective. Given that

Whisper's tiny model offers usage of only 39 million parameters [9] in contrast to the aforementioned 7-8B, the result of the merger may be further nullified to the point of futility. Nevertheless, a consideration can be made for the inclusion of the SLERP merger in Whisper-turbo applications (809M parameters), since it experiences minimal dilapidation in accuracy compared to the largest Whisper model: Whisper-large, which boasts 1550M parameters. Thus, whilst medium/large LLM's may produce greater distinctive performance differences post SLERP merger implementation, greater accuracy of the turbo model may counter the effect of the reduced parameter range potentially allowing similar phenomena to be recorded with use of the turbo model. Since the scope of this project focusses exclusively on the use of GPT-3.5-Turbo as the baseline LLM, resulting use of dual-model SLERP combination will be avoided.

Presently, extensive research has been undertaken to incorporate human-machine interfaces (HMI) into robotic systems for effective communication and development of an intrinsic relationship with assistive cobots [11]. Methodologies vary, due to the consideration of the most effective human-human (HH) communication method in multiple use-cases. A 2014 article by Barsics concludes on the view that semantic and episodic information can easily be obtained from faces than from voices [12], implying that a greater onus be placed on visual exteroceptive sensors for constructing unique user profiles that aid in fostering dynamic, long-lasting human-robot relationships per end-user. Whilst this may be so, results from recent papers displaying use-cases that involve computer vision interaction dispute the feasibility of Barsics's findings in real-time applications. Notably, Wang and Zhu [13] proposed a solution for a hand gesture recognition and worker tracking vision-based framework utilising YOLOv3 that enabled remote control of construction machinery. Though overall precision and recall results obtained showed great promise, one major pronounced drawback is identifiable. Due to inherent generalization issues with ML models, lack of individuality is prominent. Zero-shot evaluation imposed with YOLOv3 may be the causation for this, as lack of fine-tuning may lead to inaccurate recognition of specific gestures.

Considering high level use-case instructions for voice-controlled robotics systems and the previously mentioned importance of verbal stimuli over visual equivalents for robot control, research into the grounding methodologies used on LLM has been prominent in recent times. Such has been encompassed in [14] by Ahn et al. Proposing a 'real-world' grounding algorithm based on pre-trained, temporal-based (TD) reinforcement learning, their solution highlights remarkable usability amongst a range of domains by implementing a weighted value for the two parameters; the probability of the high-level instruction aligning with the pre-trained skillset of the robot and the probability of the robot showing capability for said allocated skill at that point in time (state). A notable factor in the findings was emphasised in the proposal of Chain-of-Thought prompting – the integration of intermediate thought procedures that can facilitate deeper reasoning and arithmetic intuition; a key facet of incorporating 'smarter' LLM algorithms[15]. Specifically on the notion of negation in high-level prompts (the user stating that they would not like a specific object) this degree of selective decision making is key in determining effective human-robot communication and high-level instruction comprehension. Additionally, with forecasts for dementia projects to increase significantly by 2040 [16], successful, non-recursive completion of 'long-horizon' manoeuvres[14] mitigates the likelihood of a previous instruction being re-administered to the planning pipeline, as would likely be the case with elderly users that suffer from dementia.

Management of Project – Scope, Breakdown Structure and Deliverables

Scope: Proposed as a human-robot communicative pipeline, SpeakBot will focus on the generation of low-level instructions from high-level user commands to enable zero-shot functionality of a mobile manipulator (post LLM grounding) – namely the Robotis TurtleBot3 Waffle Pi [19] with 4DOF Open Manipulator X manipulator arm [20]. For effective high-level user command audio transcription, OpenAI's Whisper ASR system will be incorporated using the whisper-1 model, whilst inclusion of role prompting from the GPT-3.5-Turbo model will enable input request summarization and intuitive semantic reasoning for urgency levels in the users' request – for clarity this will be undertaken in a 'Summary' level, before the user command is processed. An 'ActionTask' dictionary will be developed, highlighting the low-level skill set of the mobile manipulator in each domain. Implemented on the Ubuntu 20.04 architecture, ROS Noetic will be used as the development system. Universal ROS packages will be employed for effective peripheral manipulation and robot localisation, including the 'move_base' and 'MoveIt!' for navigation and arm motion planning, respectively. The functionality of the command processing algorithm will be evaluated within the 3D simulation environment, Gazebo. Implementation will be undertaken for a specific task use-case; navigating to a location, picking up a coloured block and bringing it back to the original location. Notably, for demonstrative purposes, this project will not focus on long-horizon task completion as undertaken by Ahn et al in [14], due to inherent hardware limitations such as the restricted reach of the Open Manipulator X. Conceptually, use-cases wherein the user exhibits restricted verbal capability due to serious presbyphonia conditions [21] will not be considered in this project (especially pertaining to complete loss of voice, though other use cases will be considered, including users who experience reduced vocal stamina and tremor/shakiness of the voice, in which the robustness of Whisper will be scrutinized within this zero-shot testing).

Deliverables: The following deliverables for the project are proposed:

- Functional toolkit for processing the low-level user commands within Python 3.9.5 interface on VS Code IDE, incorporating GPT3.5 Turbo LLM role prompting and generating 'action tasks' for robot control within system pipeline on ROS Noetic.
- Dynamic "ActionNoun" set containing 3 objects located in the environment; the positions of which are obtained via Gazebo's GetModelState() class.
- Static 'ActionVerb' dictionary containing 8 skills conveying the "Pick up" skill of the robot.
- Self-navigating, non-holonomic differential drive Turtlebot3 mobile base capable of localising and navigating the relevant domain to achieve a goal.
- A 'Wake' function, capable of allowing the robot to stay in hibernation whilst listening for high-level command stimuli. This functionality will be initiated by user-inputted salutation: "Hey SpeakBot."
- Proximity scanner functionality embedded within node capable of running concurrently when the robot is static for human safety and awareness.

Target Benefits: The concept for this project aligns with multiple UN SDGs, reinforcing the degree to which the solution contributes to sustainable development. For example, with great value being placed on intuitive technology in the assistive care for the elderly and disabled [7], the integration of a low-profile, simplistic mobile base in a domain void of much noise/distraction is justified. Successful inclusion of this demographic in the introduction of robotic systems is somewhat uncertain however, as noted in the 2024 Eurobarometer [22] where only 54% of respondents over 55 years of age viewed the use of robotic & AI driven systems as beneficial for society. Considering the psychological importance of verbal communication and unrestricted activities-of-daily-living (ADL), and the relation of these factors to depressive symptoms amongst the elderly and disabled population, proposal of domestic assistive 'carer robots' in this domain would be a viable solution [23]. Since the correlation between depression and disability is prominent [24], enabling elderly/disabled users to complete daily tasks by themselves with the aid of assistive robots would encourage individual confidence and wellbeing (thus reducing depressive episodes), whilst fostering a gradual acceptance of the technology in domestic settings. Conclusively, UN SDG 3.4.2 [25] will spearhead this project, incorporating a robotic pipeline that allows end-users to undertake their activities with confidence, mitigating any sense of inability encountered from their physical ailment.

Considering UN SDG 8.2 and 9.2 [26], [27], the implementation of verbally controlled, self-navigating robots in labour intensive domains may allow for improved productivity, especially with respect to repetitive tasks. Incorporating long-horizon instructions to the robot from high-level prompts as stated before, may enable distinctive, remote robotic system control for disabled and non-disabled users alike, within the industrial workspace. As such, an inclusive workforce can be reinforced, allowing for greater manufacturing capability. Furthermore, as deducted in [28], an underpinning disadvantage of using assistive robots in the industrial space is the training of employees. Proposing the SpeakBot overcomes this issue, with the medium for control being speech; a skill that requires little training. Thus, human resource allocation for training users of the technology can be targeted on other tasks, establishing the effectiveness and adaptability of zero-shot learning within assistive robot architectures.

Requirements: Proposals for the requirements of the robot will be declared under the MoSCoW method:

- SpeakBot must succeed in converting high-level user commands into low-level actions determined by its pre-determined skillset. In turn, this skillset must be reflective of objects located within a pre-mapped domain.
- SpeakBot must visually identify objects within a given domain, registering their I.D. and pose within the global domain frame for object localization.

- Operation parameters, including manipulator joint accelerations and mobile base velocity restrictions, must be limited for safe functionality of the robot (considering usage in domains co-habited by the elderly/disabled).
- By updating the global map for the domain on a regular basis, SpeakBot should be capable of identifying user-requested objects that were previously un-registered within the ActionNoun dataset and incorporating those into the Action Determination algorithm for skill relation.
- SpeakBot should be capable of precisely returning to an initial pose once a task has been completed, mitigating odometry drift through use of global pose reference.
- Processing latency within the 'Transcription' node should be restricted to enable real-time operation of robot; encountered latency should be approximately deterministic and measurable.
- Integrating a front-end user interface using a REST API could be feasible for enabling mobile communication with the robot. If implemented, the additional feature would facilitate data transfers on client-server protocols via JAVA HTTP requests. The need for a user-friendly UI design would be necessary for this feature.
- SpeakBot will not incorporate a text-to-speech API, eliminating conversational capability with the robot.
- SpeakBot will not curate a user-unique profile for command prompt grounding; LLM responses will be fine-tuned to a general user request.

Risk Assessment: Multiple risks may be encountered throughout the duration of this project, varying between software, hardware and administrative risks. As is inevitable with a discrete sensory system processing analog domain inputs, the likelihood of hardware error and associated repercussions due to processing latency (software risks), are high. Appendix B portrays a high-level overview of the risks that may be encountered within this project, implementing a probability-impact rating for the varying categories of risk. Weight classification is clarified in Appendix A. According to the value derivations, the greatest risk for the project is associated to its timely completion. To ensure completion of the project, adequate planning for relevant tasks is required. Appendix C showcases the task breakdown structure used to deconstruct the high-level scope into multiple low-level tasks fields, whilst the associated Gannt chart for timeline structure and milestone identification is shown in Appendix D. The bulk of the project will focus on software development, with hardware being integrated if/when the node is developed and tested successfully. The initial buy-off meeting and following progress meetings are recorded in Appendix J. Following, software development in the project, most of the meetings were ad hoc, taking place if issues arose with node development, URDF & tf transform enquiries and Gazebo simulation problems.

SpeakBot Methodology: Toolkit Development

Minor procedural differences will be implemented w.r.t Ahn et al's proposal, with the main difference being the usage of the 175B parameter GPT-3 LLM [17] as opposed to the 540B trained in PaLM [18]. Knowing this, the SpeakBot algorithm for generating suitable low-level command lines will be constructed as shown in Algorithm 1, effectively mirroring the algorithm used in [14]. Deviating from the SayCan algorithm, SpeakBot will impose prompt engineering via role prompting, declaring the system as a chat persona. Returned parameters from the model will be used to determine target objects to acquire and weighing parameters for mobile base trajectory dynamic re-configuration. Once tasks are complete, the action flag is set True.

Python Interface Realization: Data mapping will be effectively undertaken between three classes in one node for better synchronization and encapsulation of task processes. The first class – GetInstruction – will handle audio manipulation and transcription. Navigation handles will be driven by the second class – Navigate. Finally, the Open Manipulator X trajectory will be orchestrated by the ControlTrajectory class.

Human Safety Consideration: When static, given the low physical profile of the Turtlebot3, there is a great likelihood that the robot my encounter the elderly/disabled user. Risk of injury must be mitigated. To address this requires embedding a constantly proximity sensor within the SpeakBot node, realized through pre-emptive multi-threading procedure, as reviewed in Appendix E. Within the Proximity_Sensor thread, the node subscribes to the "/scan" topic, which is of LaserScan type. Upon receiving a message, the 5 smallest scanned values are obtained and averaged. This average value is then scrutinised; if it is below a certain threshold and the robot is not moving, a notice is printed out. In practice, this notice would be replaced by an alarm to notify the user of the robot's presence. Here, threading priority is assigned via the OS instead of manual priority assignment. For the minimal load application that the separate thread undertakes, this is sufficient. The main target is to ensure concurrent operation of the proximity sensor and the main SpeakBot thread.

Workspace Prerequisites, Python Script Initialization and Design Articulation: The SpeakBot workspace (SpeakBot.ws) houses the build, devel and src files for the structure of the programme, including the "speechtotext" package derived for the Transcription node. Relevant build dependencies are highlighted in Appendix E, with appropriate dependencies comprising of rospy, message_generation and gazebo_msgs (simulation purposes). Within the Python interface on VS code – assuming the current node is the main process running – the Transcription node is initialised under the node name "speakbot".

Program integrity was considered in the structuring of the system algorithm. Naturally, modular design would be favoured due to inherent debugging advantages, efficient task management and future feature updates. Division of software into easily developed "chunks" is far more effective for larger algorithms [29], and leverages object-oriented programming functionality for multi-node data parsing. However, considering performance of the real-time system, it is preferable to implement monolithic structuring for the Transcription node, effectively grounding all processes within one node to eliminate the communication latency between nodes that may be prominent with modular design.

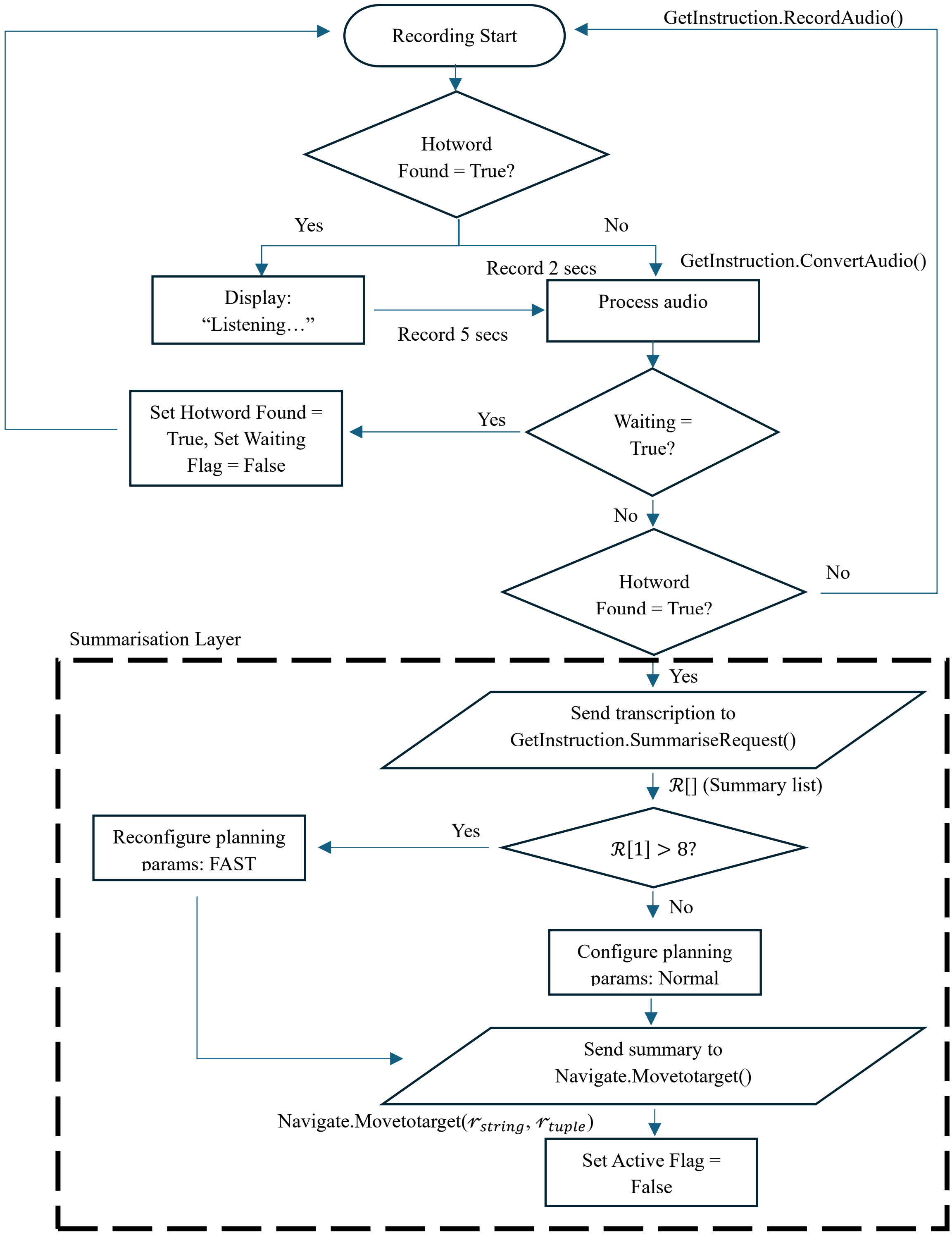
Class – GetInstruction: An initial step requires audio acquisition from the user for high-level instruction generation, highlighted in Process 1. For the project's main use case as an aide for the elderly/disabled, it is imperative to utilise lossless audio formats for effective audio identification, speech recognition and consequent transcription. Seeing as the elderly generally experience vocal hoarseness/weakness, recording the input in higher quality ensures sufficient transfer of audio data that would have otherwise been truncated [30]. As such, the audio file is recorded and processed as a WAV file, enabling lossless audio parsing and high-quality sound retention. Notice that the use of the segmented audio recording instead of live streaming is reinforced with the use of WAV files, since the lossless processing algorithm leads to a greater audio file size; streaming would be unadvisable for memory management [31]. Once the node is initialised, the class is called with the system allocated microphone as the input source. The activity state of the microphone set by the Boolean flag "hotword_detected" will be false upon initialization. Input audio chunks are forwarded to the OpenAI Whisper API, utilizing the "whisper-1" model. After recording the audio chunk for approximately 2 seconds and obtaining the transcription from Whisper, the "hotword" prompt is scrutinised. Upon identifying the prompt: "Hello Speaker", the class will enter an active listening state, setting hotword_detected as "True". The recording and transcription process is repeated, before the transcribed string is summarised in SummariseRequest(). Implementation of the summary layer is realised through role prompting with the openai.ChatCompletion() function, using GPT 3.5 Turbo for 'system'/'user' role creation. From the output of the summarization prompt highlighted in Appendix F, the high-level command (I) is determined.

In the target domain, consideration for human safety and efficient interaction is paramount, thus, SpeakBot's capability to alter trajectory planner parameters for the Turtlebot3 based on user sentiment is reinforced. To achieve this, an urgency weighting ($\mathbb{N}$) is derived from the summarisation layer (rating 1: minimal urgency, rating 10: maximum urgency). SummariseRequest() effectively alters the DWA planner and local costmap inflation layer configurations, using $\mathbb{N}$ to determine whether the plan must favour a faster, efficient trajectory or a slower, considerate path. The desired configurations were obtained via dynamic manipulation, highlighted in Table 1. Underlined parameter choices show the favourable results – and associated parameter values – for a faster journey and greater overall positional accuracy. Notably however, simulation of the SpeakBot algorithm permitted further calibration of these values. Attention can be drawn to the XY goal tolerance value, for example. Whilst a parameter setting of 0.2 initially conveys faster goal achievement, simulation with this parameter setting causes the Turtlebot to stop far from its allocated goal, impeding its ability to plan a successful path for the manipulator.

Parameters	Config Values	Journey Duration(s)	Positional Accuracy (%)	**Actual Use (Fast Config)**	**Actual Use (Normal Config)**
XY Goal Tolerance (m)	**0.05**	18	96.9	**0.075**	**0.05**
	0.1	19	96.08		
	0.15	17	94.66		
	0.2	15	90.66		
	0.25	15	85.17		
Yaw Goal Tolerance (m)	0.07	24	97.07	**0.17**	**0.07**
	0.12	18	96.32		
	0.17	18	96.9		
	0.22	26	99.66		
	0.27	21	95.57		
Path Distance Bias	8	22	97	**64**	**16**
	16	26	99.1		
	32	26	97.99		
	64	20	98.59		
	128	34	98.36		
Max Velocity Trans (m/s)	0.06	-	-	**0.26**	**0.18**
	0.11	58	96.37		
	0.16	23	96.11		
	0.21	20	97.57		
	0.26	18	96.9		
Local Costmap: Inflation Radius (m)	0.33	18	98.12	N/A	N/A
	0.66	36	94.44		
	1	18	96.9		
	1.33	17	98.49		
	1.66	20	96.9		
Local Costmap: Cost Scaling Factor	1	21	98.9	**4**	**4**
	2	31	99.26		
	3	18	96.9		
	4	17	99.08		
	5	26	98.83		

Table 1. Trajectory completion duration and positional performance (accuracy) of Turtlebot3 WafflePi within Gazebo environment based on different DWAPlannerROS and local costmap inflation layer settings. Journey durations and position values are acquired using Gazebo simulation time and Gazebo GetModelState() service, respectively.

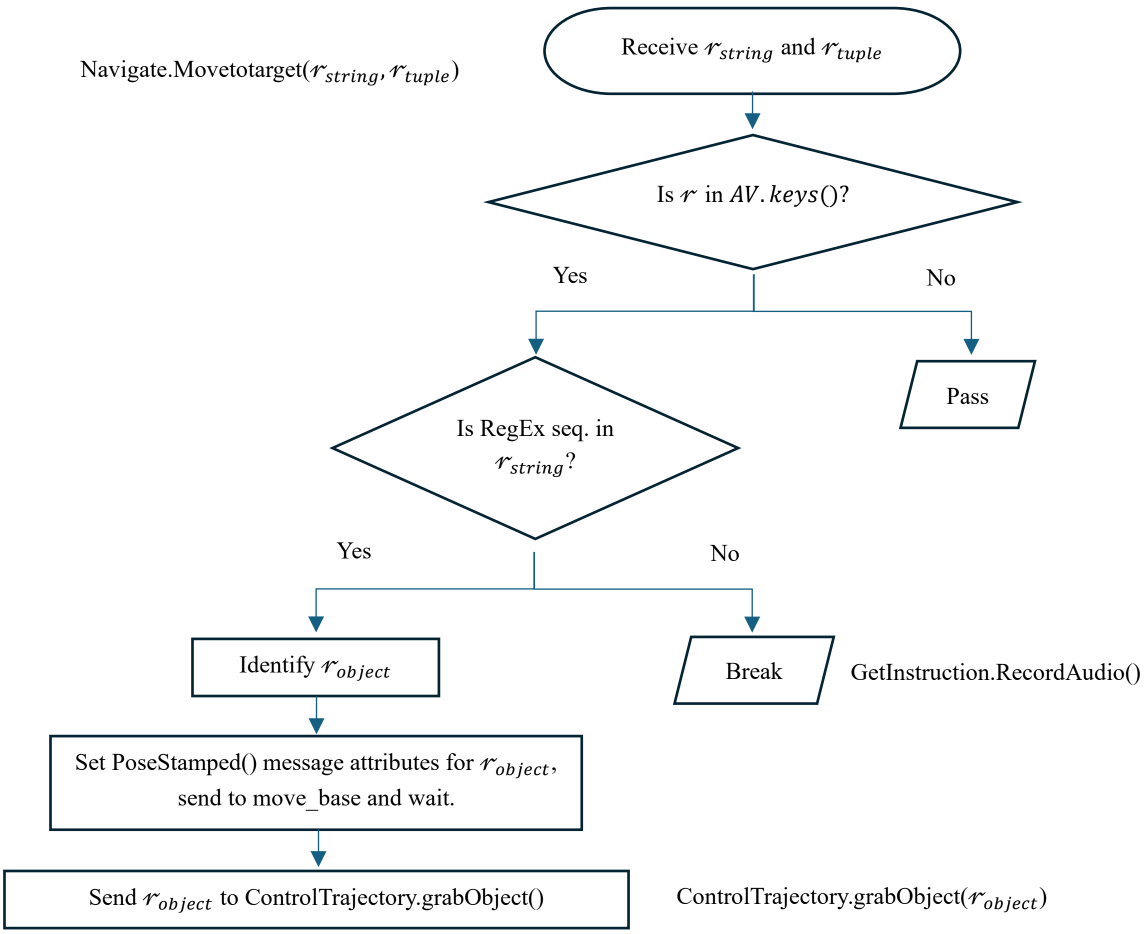
Process 1: GetInstruction


Recording Start
GetInstruction.RecordAudio()
Hotword Found = True?
Yes
No
Display: "Listening…"
Record 2 secs
GetInstruction.ConvertAudio()
Process audio
Record 5 secs
Set Hotword Found = True, Set Waiting Flag = False
Yes
Waiting = True?
No
Hotword Found = True?
No
Summarisation Layer
Yes
Send transcription to GetInstruction.SummariseRequest()
$\mathcal{R}$[] (Summary list)
Reconfigure planning params: FAST
Yes
$\mathcal{R}[1] > 8$?
No
Configure planning params: Normal
Send summary to Navigate.Movetotarget()
Navigate.Movetotarget(r_{string}, r_{tuple})
Set Active Flag = False

Class – Navigate: Housed within Navigate are three primary member functions; setObjectInfo, movetotarget and movetohome. Called prior to GetInstruction, setObjectInfo acquires the poses of the objects within the mapped domain. Object information is stored as a dictionary, with the object identifier as the key, and the object pose in a list structure, as the respective value. Within Gazebo, all object poses are relative to the map frame, acquired from the Gazebo service GetModelState(). The movetotarget member function communicates with the move_base action server, processing the high-level transcription result from the Whisper model. Considering the scope of this project, proposed objects will not deviate from the pre-determined, pre-localised data recorded in setObjectInfo. Knowing this, high-level instruction variation is limited to the pre-determined objects: the red, green and blue blocks. Therefore, since movetotarget obtains the transcribed string and equivalent iterable, one can iterate through the text tuple. Verbal commands are triggered by the recognition of an action verb, stored within the actionVerb dictionary. Shown in Appendix F, 7 synonyms are mapped to one key: "Get". Though rudimentary, this allows the lexical field of "Get" to be referenced if the input string falls under the same lexical field. From this, one can iterate through the transcribed text tuple, recognising the action verb and initiating the algorithm depicted in Process 2. Determining the target item is accomplished through use of the RegEx Python tool [32]. The string equivalent is utilized here, wherein the RegEx tool identifies the specific string pattern allocated to an actionNoun, highlighted in Appendix G. Assuming the string pattern is in the actionNoun dictionary, the pose of the target (denoted by the string pattern) is allocated to the wafflepose_msg object, and sent to the move_base client. Once the goal is achieved, the target object is passed as an argument to the Control_Arm.grabObject member function.

Process 2: Navigate

Prerequisites: $r \in r_{tuple}$, actionVerb = AV, r_{object} is the target object as a string



Navigate.Movetotarget(r_{string}, r_{tuple})
Receive r_{string} and r_{tuple}
Is r in $AV.keys()$?
Yes
No
Pass
Is RegEx seq. in r_{string}?
Yes
No
Identify r_{object}
Break
GetInstruction.RecordAudio()
Set PoseStamped() message attributes for r_{object}, send to move_base and wait.
Send r_{object} to ControlTrajectory.grabObject()
ControlTrajectory.grabObject(r_{object})

Class – ControlTrajectory: Upon acquiring the location of the target item in the global domain, this class handles the trajectory planning algorithm for the Open Manipulator X 4DOF manipulator by implementing a rudimentary state machine with the following 8 states, shown below. Using the IK Kinematics OMPL RRT Connect OMPL planning library within the MoveIt! motion planning framework, a smooth, collision free end-effector trajectory can be created. In this instance, a trajectory is derived using the adjusted waypoints, imposing the path at 0.01m steps.

1. **PREP**: Once grabObject is called, the object information is updated. In doing this, the latest information for the object is readily available for trajectory planning. Inherently, this is important for end-effector pose adjustment, as the position of the target object may have been slightly altered whilst the mobile base was navigating to the proposed target. Effective object localization requires the

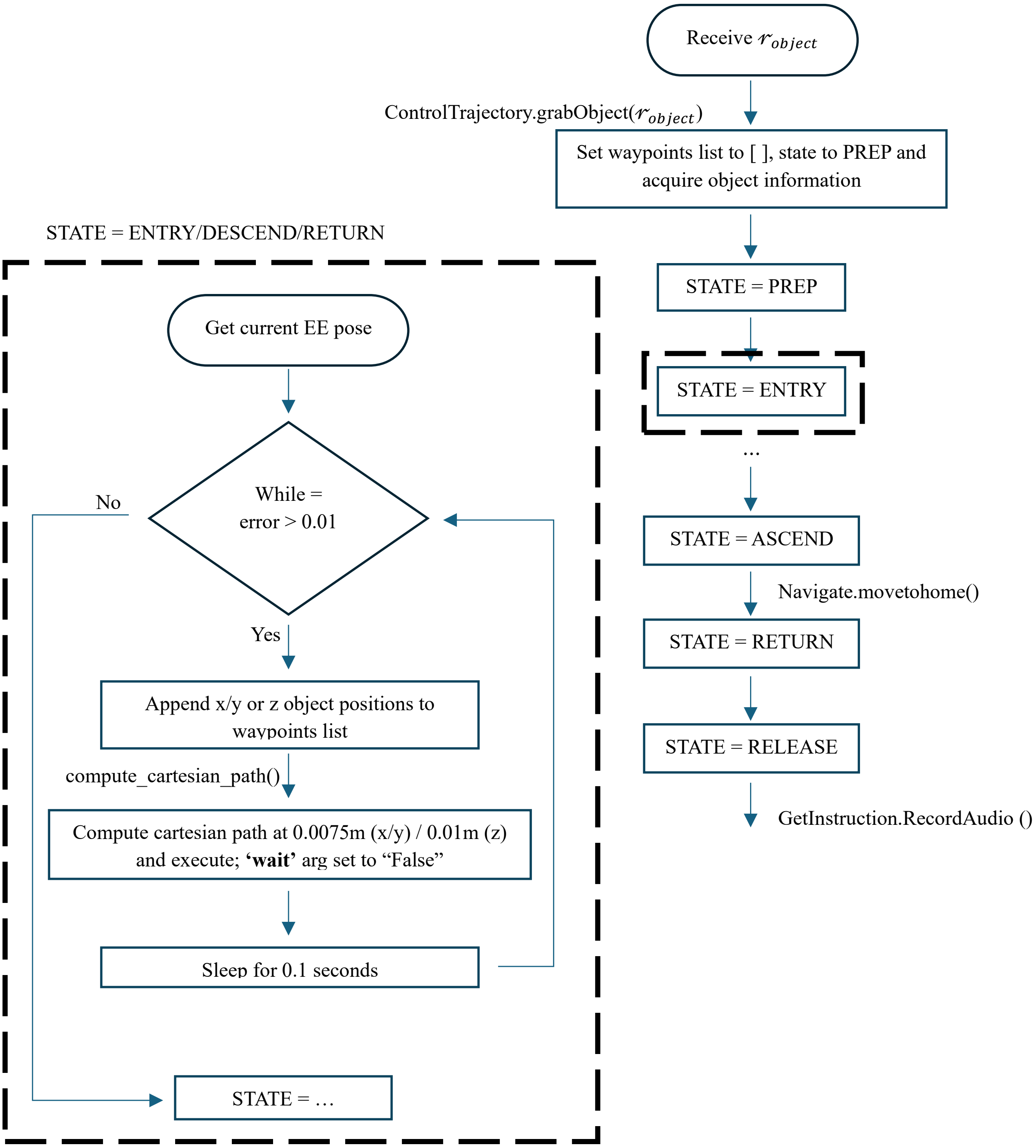
realisation of the target object pose in the Waffle Pi base frame, as opposed to the global world/map frame, since the trajectory planning for the manipulator has the base frame as its denominator. Thus, once the pose for the object – assumed static – is acquired, this pose is transformed to the base_footprint frame at runtime. Cartesian planning is implemented for intuitive end-effector pose configurations within the Open Manipulator X workspace.

2. **ENTRY**: Planar (XY) trajectory planning of the end-effector to a pose with 0.01m of the target object; the process is undertaken utilising the MoveGroupCommander() class, whereby cartesian paths are computed at 0.075, increments. Later waypoints (for the z DESCEND state) are calculated every 0.01m. The slight difference between the two cartesian path planner processes is due to minor jerking experienced near the goal pose in the ENTRY state with the waypoints calculated every 0.01m. To address this, the waypoints are calculated at slightly smaller consecutive distances to ensure the planner can produce a path when the target pose is not inherently far from the current pose.
3. **ORIENT**: Rotate revolute joint 4 to orient the end-effector towards the target object.
4. **DESCEND**: Similar to ENTRY but displacement occurs in the Z axis, until the end-effector pose is approximately less than 0.024m (clearance distance between the end-effector link and the ground plane).
5. **OBTAIN**: Direct the gripper move group commander to close.
6. **ASCEND**: Position arm group in a TRANSIT configuration, before requesting for the Turtlebot to navigate back to the initial pose - “Home”. Since this process is called as a member function of Navigate (Navigate.movetohome()), further progression within the state machine is blocked until completion of the move_base goal.
7. **RETURN**: Once Navigate.movetohome() is complete, the arm group is positioned to place the object at a specified pose. Considering simplification within Gazebo, this process was undertaken as a joint configuration goal instead of a cartesian path. Thus, collision detection during simulation is not considered, potentially leaving the arm prone to collision the mobile base. Whilst this may be a disadvantage, the main scope of this project is centred around the proof of concept for the speech to action pipeline. Aspects of collision avoidance are not primarily focussed on.

8. **RELEASE**: Direct the gripper move group commander to open, direct the arm move group commander to return to the default HOME config and clear all targets in preparation for the next instruction.

Process 3 evaluates the state machine within the ControlTrajectory class.

Process 3: ControlTrajectory



Receive r_{object}
ControlTrajectory.grabObject(r_{object})
Set waypoints list to [ ], state to PREP and acquire object information
STATE = PREP
STATE = ENTRY
...
STATE = ASCEND
Navigate.movetohome()
STATE = RETURN
STATE = RELEASE
GetInstruction.RecordAudio ()
STATE = ENTRY/DESCEND/RETURN
Get current EE pose
While = error > 0.01
No
Yes
Append x/y or z object positions to waypoints list
compute_cartesian_path()
Compute cartesian path at 0.0075m (x/y) / 0.01m (z) and execute; **'wait'** arg set to "False"
Sleep for 0.1 seconds
STATE = ...

Considering the Cartesian planning states ENTRY, DESCEND and POSE, it is evident that the "wait" argument in the MoveGroupCommander.compute_cartesian_path()

member function is set to False, implying that the algorithm does not wait for the previous waypoint to be achieved before recalculating the upcoming batch. The reason for this was identifiable from the Gazebo simulation. Setting the wait argument to "True" causes the manipulator to exhibit rough, rigid movement as it stops and starts, following each waypoint completion. Altering this to False, ensures smooth, unrestricted path completion, at the risk of overloading the communication topic for broadcasting the waypoints. To address this, a short blocking delay is implemented for 0.1 seconds, allowing the OMPL RRT Connect planner to process waypoints at a reasonable rate, whilst still portraying smooth movement of the Open Manipulator X.

For the sake of simulation, a case can be made for the pose reference frame used in the manipulator trajectory planning. Usually, it is appropriate to declare a world frame as the central frame for object pose verification and mobile robot trajectory planning. However, due to inaccuracies experienced with the world frame, the odom frame was implemented as the global pose reference. Whilst this may seem illogical due to odometer drift from the TurtleBot rotatory encoders, practice within the simulation highlighted greater accuracy of the end-effector w.r.t the final target pose. As such, the 'odom' frame was set as the pose reference frame for the end-effector.

Used within the primary domain of an elderly/disabled persons home, 2 considerations were used to determine the tangibility of the concept. A verbal adherence test, dictating how well an instruction given to the robot gets processed, and an operational test, relaying the degree to which an additional prompt, via compound request, alters the robots planning path to accommodate the intent of the user.

Verbal Adherence Test

Here, the audio processing capability of Whisper is tested under 5 varying input sound conditions, as highlighted in Table 2. This test is undertaken prior to request input to the Navigate class, focussing primarily on the response from the summarisation layer. Acknowledging 58 dbA as the sufficient benchmark for casual conversation between humans, and 42-47 dbA as the average for soft spoken vocal sounds [33], the accuracy of Whisper-1 was tested with the previously mention role prompt on GPT-3.5-Turbo. A few points must be raised regarding external variable regulation in this test, namely low-level noise and lack of precision in distance recorded from microphone. Decibel A measurements were taken during the recording of the audio, though samples were not taken in an anechoic environment, possibly introducing inaccurate initial calibration for audio levels. Additionally, distance values were estimated based on distance from hardware computer running SpeakBot, potentially varying in actual practice or showing inconsistency across the samples. The categories were chosen to reflect use-cases of vocal speech levels in the elderly or those with minor verbal impairments; differences in clarity and roughness of the input audio were adjusted to emulate these conditions. Categories without background conversation mimic scenarios where the user is alone with the robot, whilst background conversation readings mimic scenarios where user is accompanied by another human – a carer or family for example. Results from the summarisation layer are shown in Appendix I.

Input Voice Categories	Approx. Distance(m)	Avg dbA	Re-listens
Hoarse	0.3	41	2
	0.6	41	2
	1	41	2
Whisper	0.3	42	2
	0.6	41	2
	1	41	2
Normal	0.3	43	2
	0.6	42	2
	1	42	2
Background Conversation (Whisper) - underlying dbA rating: 56 dbA avg	0.3	56	2
	0.6	56	8
	1	56	2
Background Conversation (Normal) - underlying dbA rating: 56 dbA avg	0.3	56	2
	0.6	57	2
	1	57	15

Table 2. Verbal accuracy test results using whisper-1 model for audio transcription and GPT-3.5-Turbo for summarisation post few-shot role prompt engineering. Since the SpeakBot algorithm listens for the "Hello Speaker" hotword phrase, and then re-listens again for the actual request, no literal re-listens are undertaken i.e.the system recognises the user's request straight away.

Referencing to the results from the summarisation layer, whisper-1 showcases consistent summarisation levels of the input response, given varying audio sound levels and external noise inputs. The outliers in the re-listens recorded may originate from inaccurate sound level recorders or unintentional amplification of the testers voice whilst recording. Nevertheless, results from the whisper-1 model highlight the validity of the Whisper ASR and GPT3.5 Turbo LLM as input mediums for the SpeakBot pipeline, even in noisy domains.

Operational Functionality: Urgency Adherence

Utilising the semantic apprehension capabilities of the GPT 3.5 Turbo LLM, the urgency of the user request was obtained, depicted as the second element of $\mathcal{R}[]$ in the SpeakBot algorithm. Virtual completion of this test required setting up the Gazebo environment with necessary obstacles. Whilst the TurtleBot world provided suitable profiles to consider in path planning, the following setup generated a map with slightly greater complexity; higher realism compared to the target domain.

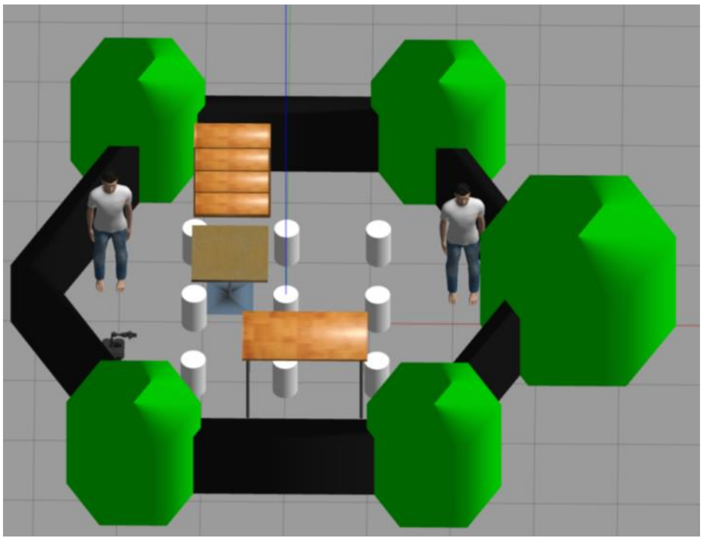


Figure 1. Initial Gazebo world setup.

Simulation issues were encountered when implementing this scene, however. Due to the pose of the LiDAR on the TurtleBot3 URDF, objects projected on the local costmap were restricted to one stagnant plane. Inherently, the laser scanner could not identify the wider geometry of the coffee table base, as shown in Figure 2, repeatedly failing to re-calculate a suitable plan for the robot to reach its required goal. Thus, to accommodate this, the coffee table was removed from the environment. As an alternative solution, generating a 3D point cloud may have been suitable. Utilising the depth camera information, sensor fusion between the two exteroceptive peripherals may provide suitable information for the local costmap to generate obstacle inflation radii with greater precision, bypassing this encountered issue.

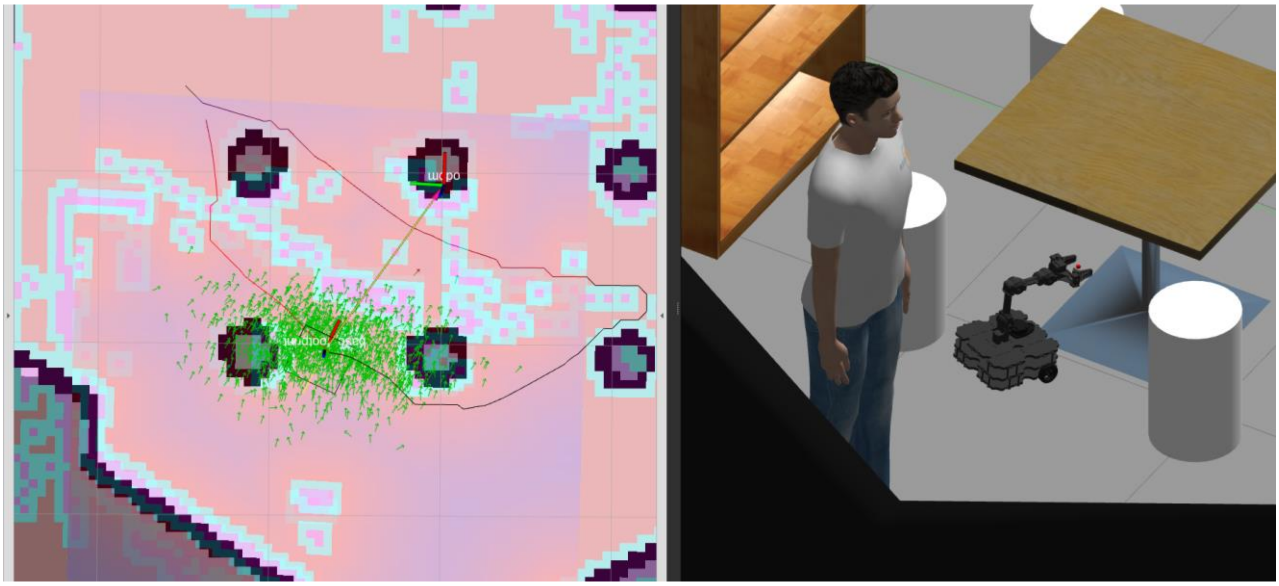


Figure 2. Failed obstacle inflation layer generation due to position of LiDAR and non-existent 3D data.

Operational functionality was tested within the finalised Gazebo environment shown in Figure 3, to record the successful identification of higher urgency requests and track task completion times for tasks varying in urgency.

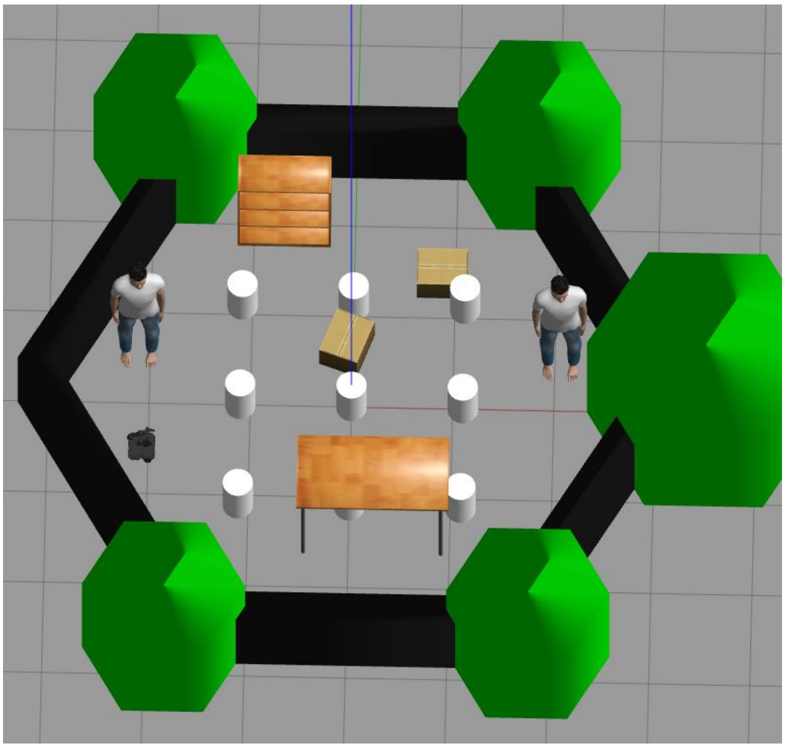


Figure 3. Updated Gazebo environment without disrupting coffee table.

Reiterating the configuration parameter setup highlighted in the SummariseRequest class, only two parameter configurations were implemented: “fast” DWA configuration for urgency weightings greater than 8, and “normal” DWA configuration for urgency weightings less than 8. Figure 4 conveys the local and global costmap visualizations in rviz at runtime for the fast configuration and normal configurations.

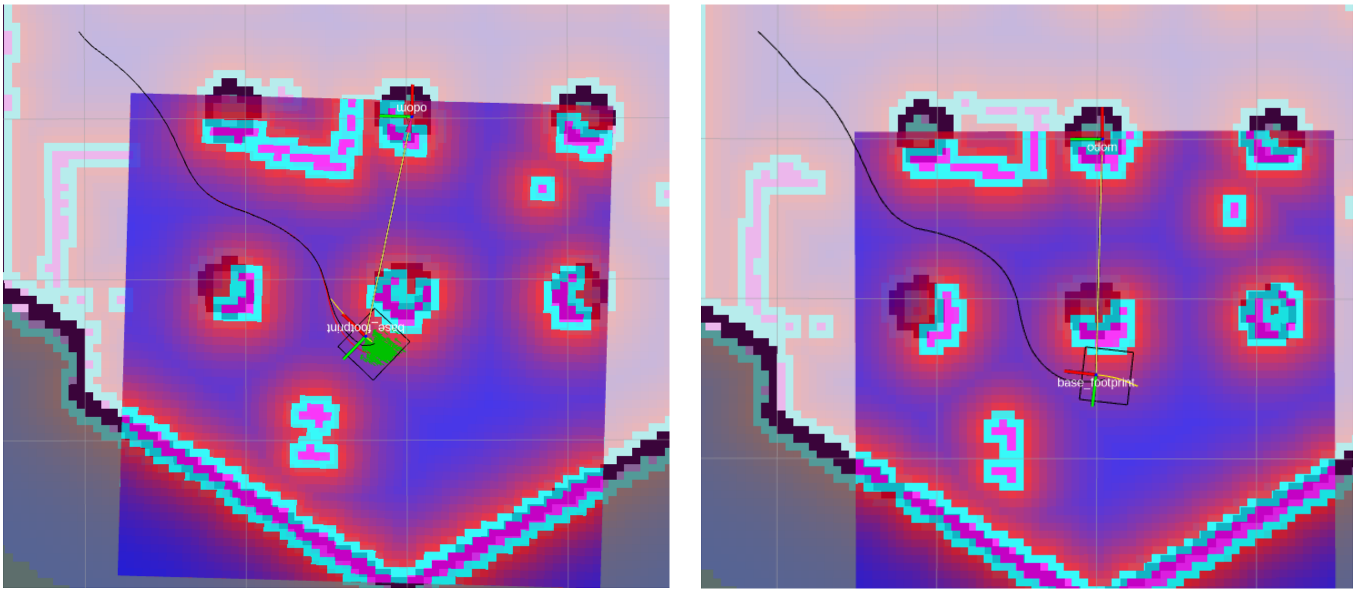


Figure 4. Comparison between planned path with DWA "fast" (left) and DWA "normal" (right).

At a surface level, there is minimal difference between the two planner configurations, though the ‘fast’ configuration portrays a slightly less aggressive turning behaviour, likely due to the increased path_distance_bias parameter which favours a greater bias towards the robot tending to the proposed path since the adjoining goal_distance_bias is retained at the default value of 20. Table 3 states the successful trajectory duration times recorded for the “fast” and “normal” parameter configurations, emphasizing the effect of the dynamic re-configuration. One must note that for the results below, the Gazebo simulation time was utilized.

DWAPlannerROS Fast Config: Trajectory Duration Times (secs)			DWAPlannerROS Normal Config: Trajectory Duration Times (secs)		
Blue Block	Red Block	Green Block	Blue Block	Red Block	Green Block
88	85	84	139	121	117

Table 3. Move_base path planner duration values using DWA Planner ROS "fast" and "normal" configurations.

Acknowledging the reactiveness of the robot, alteration of the “sim_time” DWAPlannerROS achieves the effect of improving the reaction speed of the robot. However, as shown in Figure 5, under-weighting this parameter has led to numerous scenarios where the plan is over-computed, or the robot attempts to go through a high penalty area (cost-wise). For the case in Figure 5, all other configurations were retained at the default value (not “fast” or “normal”), yet this behaviour was still exhibited. As such, this parameter shall remain unchanged, though fine-tuning of this value may provide benefit for a future use case of SpeakBot.

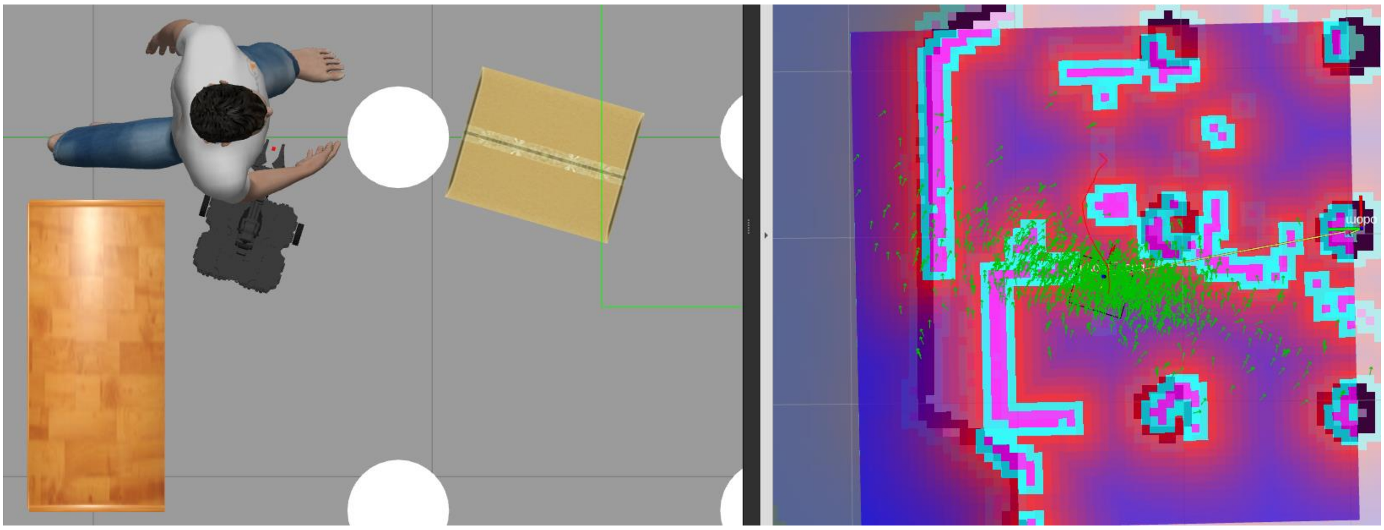


Figure 5. Sim time value: 1.0. TurtleBot3 continuously colliding with obstacle in front after attempting to find another path around the same obstacle.

Conclusion

In this project, the feasibility of the speech to action pipeline is analysed through the construction of SpeakBot. Implemented as a predominantly platform agnostic pipeline, SpeakBot leverages ROS Noetic as a baseline framework for handling topics and services broadcasting information to the TurtleBot3 mobile base and Open Manipulator X manipulator. Embedding the whisper-1 model via the Whisper ASR API for .wav file transcription and utilizing GPT-3.5.Turbo as an intermediary persona within the summarisation layer for key user request information, this paper has reinforced the notion of low-latency, assistive technology via mobile manipulators. The proof of concept for the node has been verified via simulation within the Gazebo environment, where promising results for the trajectory accuracy of the mobile manipulator and the adherence to the input request show great promise for the future of verbally controlled, low-profile mobile manipulators.

Limitations and Recommendations

Within this project, multiple points of improvement could be surmised. Firstly, the trajectory planners for the manipulator were not significantly analysed; lack of attention was given to viable alternatives including PRM, CHOMP or the slower RRT (Rapidly exploring Random Tree). A key point for future investigation w.r.t the SpeakBot implementation with the Open Manipulator X would be to consider the above planners. Additionally, whilst this project did implement prompt engineering for astute, relevant model response, future robustness will be partially dependant on Chain of Thought prompting within the summarisation layer. Algorithm X conveys a potential base frame for deriving such into the SpeakBot pipeline. Adding such reasoning to the pipeline will make long-horizon tasks feasible achievable.

Algorithm 1: Action Determination

Provided: High-level command, I, current mobile-base & manipulator state, s_i, a pre-determined skillset, Π, and their associated low-level descriptions ℓ_π

```
$\mathcal{R}[\,] = LLM_{sum}(I)$      ← Generate ‘Summary’ list from CoT prompt
$n = 0, \pi = \emptyset$              ← Zero low-level instruction set & empty skill selection
while $\mathcal{R}[n] \neq$ "Complete":
    $\mathcal{C} = \emptyset$
    for $\pi \in \Pi$ and $\mathcal{R}[\,] \in I$:
        $p_{\pi}^{LLM} = p(\ell_{\pi} | \mathcal{R}[n], \ell_{\pi_{n-1}}, ...., \ell_{\pi_0})$
        $p_{\pi}^{feas} = p(c_{\pi} | s_i, \ell_{\pi})$
        $p_{\pi}^{sum} = p_{\pi}^{feas} p_{\pi}^{LLM}$
        $\mathcal{C}\ [\ell_{\pi}] = \mathcal{C} \cup p_{\pi}^{sum}$      ← List denotes position for $skill_{\pi}$
    $\pi_n = \arg max_{\pi \in \Pi} \mathcal{C}[\,]$
    Append $\pi_n(s_n)$ for current step: $\mathcal{F}[\,]$ = append.$\pi_n(s_n)$
    Send $\mathcal{F}[\,]$ to robot controller
        Wait for completion….
    Update step value for next $\mathcal{R}[n]$: $n$ += 1
end
```

Algorithm 1 provides a high-level overview of the determination process for the action of the robot. Following on from [14], notice that the list assignment for processed data has been contextualised; it is easier to visualise certain procedures, such as storage of the low-level prompt data from the Summary level and the appending of the total probability for a given skill to the relevant data container (Line 9). Additionally, by mapping the algorithm in this manner, iteration over the elements of the low-level string array $\mathcal{R}$ has a more intuitive flow. Understanding that the sequential generation of targets (based on repeated broadcasting of planned skills) will be created after every maximum weighted value of skill is created, the algorithm appends the skill log for a given $\mathcal{R}$. This is repeated until the skills required for I are all determined. Only once this is accomplished, the algorithm will broadcast the targeted points (skill) to complete the high-level task. The bulkhead of the policy mapping is initialised by the feasibility probability p_{π}^{feas} , as the current state of the robot inevitably determines the best possible action. The token "Complete" would be appended to the Summary list from the CoT prompt to ensure that the loop is terminated once all the input tasks are completed. Akin to the methodologies used in [14], this solution incorporates Markov's decision process (MDP) to accomplish temporal-difference (TD) reinforcement learning for the completion of goals within the domain.

Finally, this project did not embed the YOLOn11 Object Detection algorithm for object pose localization in the global frame, due partially to time constraint and limited understanding, though the hardware GPU was also inadequate for effective image processing. High latency was repeatedly experienced on the /yolo/detections topic, rendering the YOLO stream insufficient for real-time updates, even from the Gazebo environment. Future improvements to the SpeakBot algorithm could focus on either optimising the YOLO model using OpenVINO, for example, or acquiring a higher quality GPU. Renting a GPU may have the same effect, but testing periods for the SpeakBot pipeline would be restricted to the amount of time that the GPU is rented for. In addition to this, the YOLO detection could be trialled with a smaller version of the algorithm, like YOLOv4.

References

[1] A. Dimakos, D. Woodhall, and S. Asif, “A Study on Centralised and Decentralised Swarm Robotics Architecture for Part Delivery System,” *Acad. J. Eng. Stud.*, vol. 2, no. 3, Oct. 2021, doi: 10.31031/AES.2021.2.000540.

[2] A. Ranjan, R. Machavaram, and P. Patidar, “Design and development of a peduncle-holding end effector for robotic harvesting of mango.,” *Cogent Eng.*, vol. 11, no. 1, pp. 1–15, Jan. 2024, doi: 10.1080/23311916.2024.2403706.

[3] I. I. F. of Robotics, “Robotics Research: How Asia, Europe and America Invest – Global Report 2023 by IFR,” IFR International Federation of Robotics. Accessed: Feb. 07, 2025. [Online]. Available: https://ifr.org/ifr-press-releases/news/robotics-research-how-asia-europe-and-america-invest

[4] World Health Organisation, “Ageing and health,” Ageing and Health. Accessed: Feb. 07, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

[5] “Sociodemographic inequalities in suicides in England and Wales: 2011 to 2021,” Office for National Statistics (ONS), Bulletin, Mar. 2023. Accessed: Jan. 30, 2025. [Online]. Available: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthinequalities/bulletins/sociodemographicinequalitiesinsuicidesinenglandandwales/2011to2021

[6] “UK Disability Survey research report, June 2021,” GOV.UK. Accessed: Feb. 07, 2025. [Online]. Available: https://www.gov.uk/government/publications/uk-disability-survey-research-report-june-2021/uk-disability-survey-research-report-june-2021

[7] P. Maresova *et al.*, “Challenges and opportunity in mobility among older adults – key determinant identification,” *BMC Geriatr.*, vol. 23, no. 1, p. 447, Jul. 2023, doi: 10.1186/s12877-023-04106-7.

[8] M. Li, “Exploring the Application of Large Language Models in Spoken Language Understanding Tasks,” in *2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, Aug. 2024, pp. 1537–1542. doi: 10.1109/ICSECE61636.2024.10729345.

[9] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision.” 2022. [Online]. Available: https://arxiv.org/abs/2212.04356

[10] W. Lu, R. K. Luu, and M. J. Buehler, “Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities.” 2024. [Online]. Available: https://arxiv.org/abs/2409.03444

[11] T. N. Canh, B. P. Nguyen, H. Q. Tran, and X. HoangVan, “Development of a Human-Robot Interaction Platform for Dual-Arm Robots Based on ROS and Multimodal Artificial Intelligence,” Nov. 08, 2024, *arXiv*: arXiv:2411.05342. doi: 10.48550/arXiv.2411.05342.

[12] C. Barsics, “Person Recognition Is Easier from Faces than from Voices,” *Psychol. Belg.*, 2014, doi: 10.5334/pb.ap.

[13] X. Wang and Z. Zhu, “Vision–based framework for automatic interpretation of construction workers’ hand gestures,” *Autom. Constr.*, vol. 130, p. 103872, Oct. 2021, doi: 10.1016/j.autcon.2021.103872.

[14] M. Ahn *et al.*, “Do As I Can, Not As I Say: Grounding Language in Robotic Affordances”.

[15] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 24824–24837. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[16] Y. Chen *et al.*, “Dementia incidence trend in England and Wales, 2002–19, and projection for dementia burden to 2040: analysis of data from the English Longitudinal Study of Ageing,” *Lancet Public Health*, vol. 8, no. 11, pp. e859–e867, Nov. 2023, doi: 10.1016/S2468-2667(23)00214-1.

[17] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” Jul. 22, 2020, *arXiv*: arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[18] A. Chowdhery *et al.*, “PaLM: Scaling Language Modeling with Pathways,” Oct. 05, 2022, *arXiv*: arXiv:2204.02311. doi: 10.48550/arXiv.2204.02311.

[19] Y. Name, “ROBOTIS e-Manual,” ROBOTIS e-Manual. Accessed: Mar. 01, 2025. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

[20] “PincherX-100 — Interbotix X-Series Arms Documentation.” Accessed: Mar. 01, 2025. [Online]. Available: https://docs.trossenrobotics.com/interbotix_xsarms_docs/specifications/px100.html

[21] “Aging Voice Problems | Presbyphonia, Presbylaryngeus | Duke Health.” Accessed: Mar. 02, 2025. [Online]. Available: https://www.dukehealth.org/treatments/voice-disorders/aging-voice

[22] European Commission, “Artificial Intelligence and the future of work,” European Union, Feb. 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://europa.eu/eurobarometer/surveys/detail/3222

[23] R. You, W. Li, L. Ni, and B. Peng, "Study on the trajectory of depression among middle-aged and elderly disabled people in China: Based on group-based trajectory model," *SSM - Popul. Health*, vol. 24, p. 101510, Dec. 2023, doi: 10.1016/j.ssmph.2023.101510.

[24] S. Meshkat, Q. Lin, V. K. Tassone, R. Janssen-Aguilar, W. Lou, and V. Bhat, "The association between depressive symptoms and limitations in disability domains among US adults," *J. Mood Anxiety Disord.*, vol. 9, p. 100103, Mar. 2025, doi: 10.1016/j.xjmad.2024.100103.

[25] "Goal 3 | Department of Economic and Social Affairs." Accessed: Mar. 02, 2025. [Online]. Available: https://sdgs.un.org/goals/goal3#targets_and_indicators

[26] "Goal 8 | Department of Economic and Social Affairs." Accessed: Mar. 02, 2025. [Online]. Available: https://sdgs.un.org/goals/goal8#targets_and_indicators

[27] "Goal 9 | Department of Economic and Social Affairs." Accessed: Mar. 02, 2025. [Online]. Available: https://sdgs.un.org/goals/goal9#targets_and_indicators

[28] "7 Challenges In Industrial Robotics - Find Out With Our Guide - Rowse." Accessed: Mar. 02, 2025. [Online]. Available: https://www.rowse.co.uk/blog/post/7-challenges-in-industrial-robotics

[29] T. Hardin, M. Jaume, F. Pessaux, and V. Viguie Donzeau-Gouge, *Concepts and Semantics of Programming Languages 2: Modular and Object-Oriented Constructs with OCaml, Python, C++, Ada and Java*. Newark, UNITED STATES: John Wiley & Sons, Incorporated, 2021. Accessed: Mar. 13, 2025. [Online]. Available: http://ebookcentral.proquest.com/lib/dmu/detail.action?docID=6690677

[30] L. Brown, "Speech Changes in Older Adults," Senior Care Advice & Caregiver Support. Accessed: Apr. 10, 2025. [Online]. Available: https://seniorcareadvice.com/health-well-being/health-safety/speech-changes-in-older-adults.htm

[31] "MP3 vs WAV File: Key Differences, Advantages & Disadvantages," Unison. Accessed: Mar. 13, 2025. [Online]. Available: https://unison.audio/mp3-vs-wav/

[32] "Regular Expression HOWTO," Python documentation. Accessed: Mar. 13, 2025. [Online]. Available: https://docs.python.org/3/howto/regex.html

[33] J. Galster, "Revisiting expectations for average and soft speech levels | Audiology Blog," Phonak Audiology Blog - Phonak Pro - life is on. Accessed: Apr. 09, 2025. [Online]. Available: https://audiologyblog.phonakpro.com/revisiting-expectations-for-average-and-soft-speech-levels/

[34] "ISO 13482:2014(en), Robots and robotic devices — Safety requirements for personal care robots." Accessed: Mar. 13, 2025. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:13482:ed-1:v1:en

[35] “Built-in Types,” Python documentation. Accessed: Mar. 13, 2025. [Online]. Available: https://docs.python.org/3/library/stdtypes.html

Appendices

Appendix A. Probability/Impact Comparison Table for SpeakBot project

Weighting classification for risks using qualitative 3x3 risk matrix; high-impact, high-likelihood risks are denoted in red. Conversely, risks processed with a moderate overall risk rating are shown in orange, and acceptable risks ratings are highlighted in green. Overall risk ratings are categorized as below:

- $\mathbf{r \leq 0.1}$ = Low risk
- $\mathbf{0.1 < r < 0.4}$ = Medium risk
- $\mathbf{0.4 \leq r}$ = High risk

Likelihood	Very Likely: > 60% (0.6)	🟧	🟧	🟥
	Possible: 10% (0.1) - 60% (0.6)	🟩	🟧	🟧
	Very Unlikely: < 10% (0.1)	🟩	🟩	🟧
		Low Impact: < 0.3	Medium Impact: 0.3 – 0.7	High Impact: > 0.7
		Impact		

Table 4. Risk matrix for the SpeakBot project. Overall risk is determined by multiplying the likelihood and impact values together.

Appendix B. Qualitative Risk Assessment Table

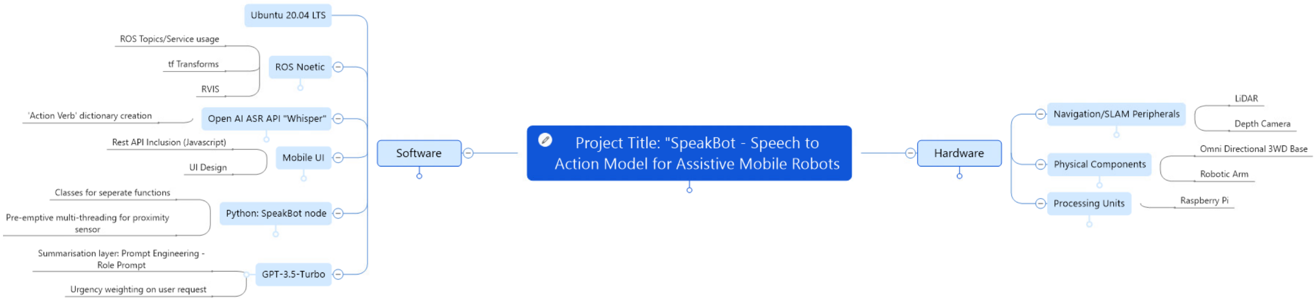
Risk assessment table highlighting categorised risks.

Risk Description	**Risk Category**	**Probability**	**Impact** (0 – minimal impact, 1 – highest impact)	**Risk Level (r)**
Product may not be completed on schedule	Control Risk	**60%** - Given the conciseness of scope and pre-developed software framework (ROS) and hardware, much of the time for the project is dedicated to troubleshooting and testing the functionality of the project. This is undertaken near the end of the project, by which point most of the software development should be complete. However, due to lack of previous experience with the ROS, and fault finding within simulation environment, there is a great likelihood that the project may not be completed within the given time frame.	**0.9** – Failure to complete the product (fully functional) will not allow for adequate testing with a wide range of operators. Thus, the result of this testing will not be recorded in the final report; this feedback is a key component for the success of the project.	**0.54**
The robot may not operate in accordance with the requested instruction	Control Risk	**20%** - Instructions that drive the robot will only be derived from the coding script generated.	**0.7** – Since the instructions for the navigation and operation of the robot will be derived from the script developed, there is a restriction on the output that can be provided to the robotic system. In case of malfunction, the robot will not be able to operate outside of this scope – given that the mapping procedures for movement and object identification are sound.	**0.14**

The robot may cause harm to the user	Compliance Risk	**30% -** The product will adhere to ISO 13482:2014, which focuses on the safe design, protective measures and information for the use of robots in personal care. Updates to next version – ISO/DIS 13482 – will also need to be considered, though standard is not presently released[34].	**1** – Failure of the robot to adhere to regulation surrounding the operational safety of the robot may lead to moderate/fatal injury, since the main target audience are the elderly and physically impaired.	**0.3**
The logic/action script driving the robot may be altered (including actionVerb function and 'Operation' class)	Control Risk	**20% -** The main functionality of the script is based on the occurrence of verbs found in the member array 'actionVerb'. Initialising this as a 'frozenset' type ensures that the contents within cannot be altered after creation [35]. Additionally, since the set array will be a private data member within a class, it is impossible for any other part of the script to access the member variable form outside of the class.	**0.5** – Each value within the 'actionVerb' set is mapped to a certain numerical output signifying an instruction to the robot. Since multiple verbs can ultimately mean the same thing ('get', 'grab', 'bring') there will be a standard output for various verbs. Under the circumstance that a value within the 'actionVerb' set is deleted, a verb requesting a similar action will be able to produce the same output.	**0.1**
Risk of electrocution from power source	Hazard Risk	**10% -** The power source for the robot is compliant will necessary electrical appliance regulations. For the duration of the project this power supply will not be tampered with.	**0.8** – Electrocution at 12V may potentially cause fatal injury to users of the product if incorrectly wired.	**0.08**
The robot may begin to process complicated instructions that relate to objects with complex geometry	Opportunity Risk	**20% -** Upon integrating the CoT procedure into the task determination pipeline for long-horizon task completion, and with additional reinforcement learning via zero-shot implementation, it is possible that the robot will begin to operate outside of the designated scope, regarding object identification mainly.	**1** – This option would allow for a greater level of integration of the robot as a truly assistive machine, capable of handling tasks that are much more complex than iterative, simple tasks (picking up a remote or bringing a cup).	**0.2**
The user interface for mobile	Opportunity Risk	**50% -** Coding of the robot and its output characteristics will take priority over the	**0.2** – The lack of the UI on the mobile application will	**0.1**

control may not be finished		development of the UI for the mobile application. If possible, the addition of the user interface will allow for better use of the robot by the target audience.	not render the robot useless.	🟩

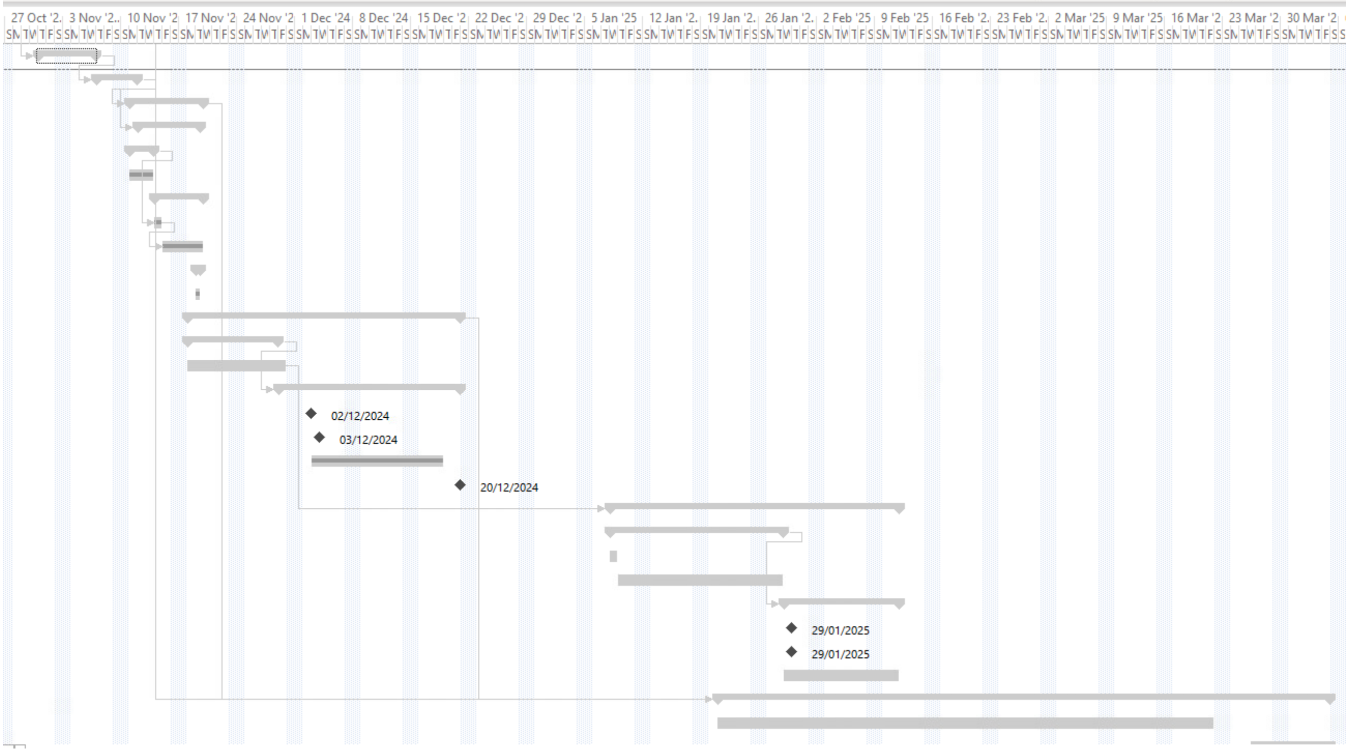
Appendix C. Task Breakdown Structure for SpeakBot

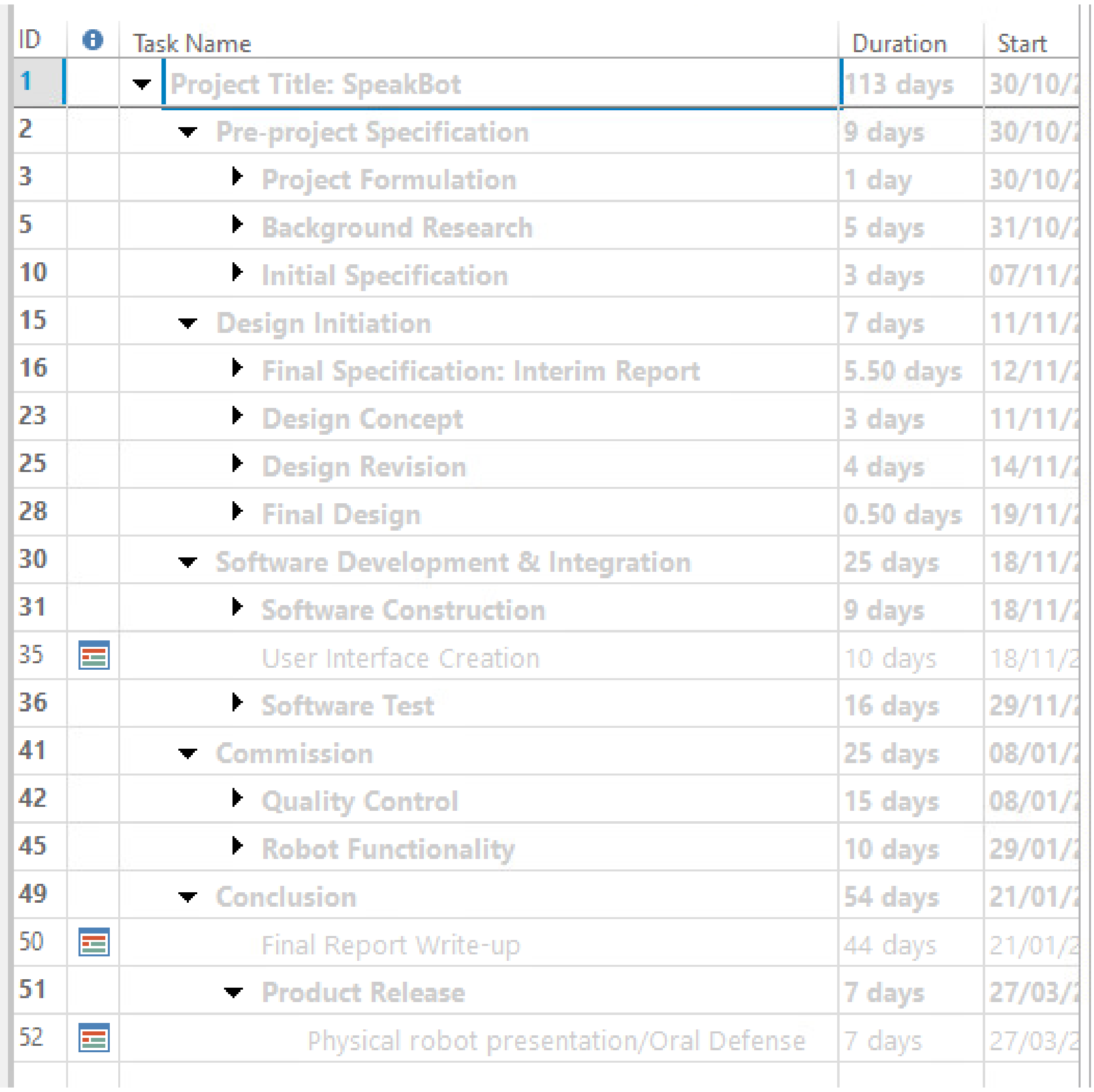


Project Title: "SpeakBot - Speech to Action Model for Assistive Mobile Robots"
Software
Ubuntu 20.04 LTS
ROS Noetic
ROS Topics/Service usage
tf Transforms
RVIS
Open AI ASR API "Whisper"
'Action Verb' dictionary creation
Mobile UI
Rest API Inclusion (Javascript)
UI Design
Python: SpeakBot node
Classes for seperate functions
Pre-emptive multi-threading for proximity sensor
GPT-3.5-Turbo
Summarisation layer: Prompt Engineering - Role Prompt
Urgency weighting on user request
Hardware
Navigation/SLAM Peripherals
LiDAR
Depth Camera
Physical Components
Omni Directional 3WD Base
Robotic Arm
Processing Units
Raspberry Pi

Appendix D. Gannt chart for project (initial)

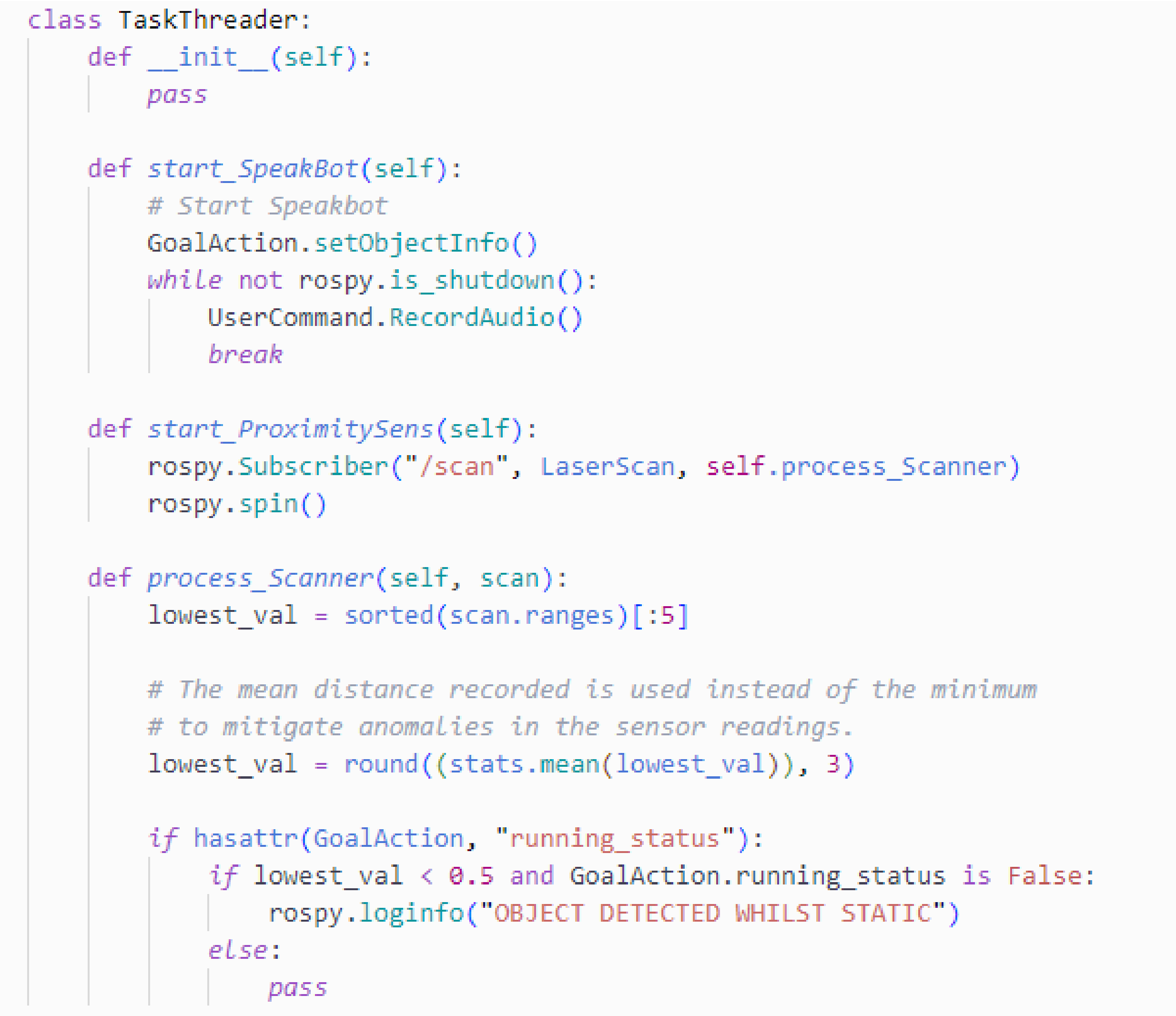
Setbacks were experienced due to lack of experience with the ROS framework, software issues and time constraints that were inadequately accounted for. Additionally, physical realisation of the product was not achieved. The advice to thoroughly learn the nuances of ROS and how it interacts on a software level is highly valued.





ID	ℹ	Task Name	Duration	Start
1		▼ Project Title: SpeakBot	113 days	30/10/2
2		▼ Pre-project Specification	9 days	30/10/2
3		▶ Project Formulation	1 day	30/10/2
5		▶ Background Research	5 days	31/10/2
10		▶ Initial Specification	3 days	07/11/2
15		▼ Design Initiation	7 days	11/11/2
16		▶ Final Specification: Interim Report	5.50 days	12/11/2
23		▶ Design Concept	3 days	11/11/2
25		▶ Design Revision	4 days	14/11/2
28		▶ Final Design	0.50 days	19/11/2
30		▼ Software Development & Integration	25 days	18/11/2
31		▶ Software Construction	9 days	18/11/2
35	▤	User Interface Creation	10 days	18/11/2
36		▶ Software Test	16 days	29/11/2
41		▼ Commission	25 days	08/01/2
42		▶ Quality Control	15 days	08/01/2
45		▶ Robot Functionality	10 days	29/01/2
49		▼ Conclusion	54 days	21/01/2
50	▤	Final Report Write-up	44 days	21/01/2
51		▼ Product Release	7 days	27/03/2
52	▤	Physical robot presentation/Oral Defense	7 days	27/03/2

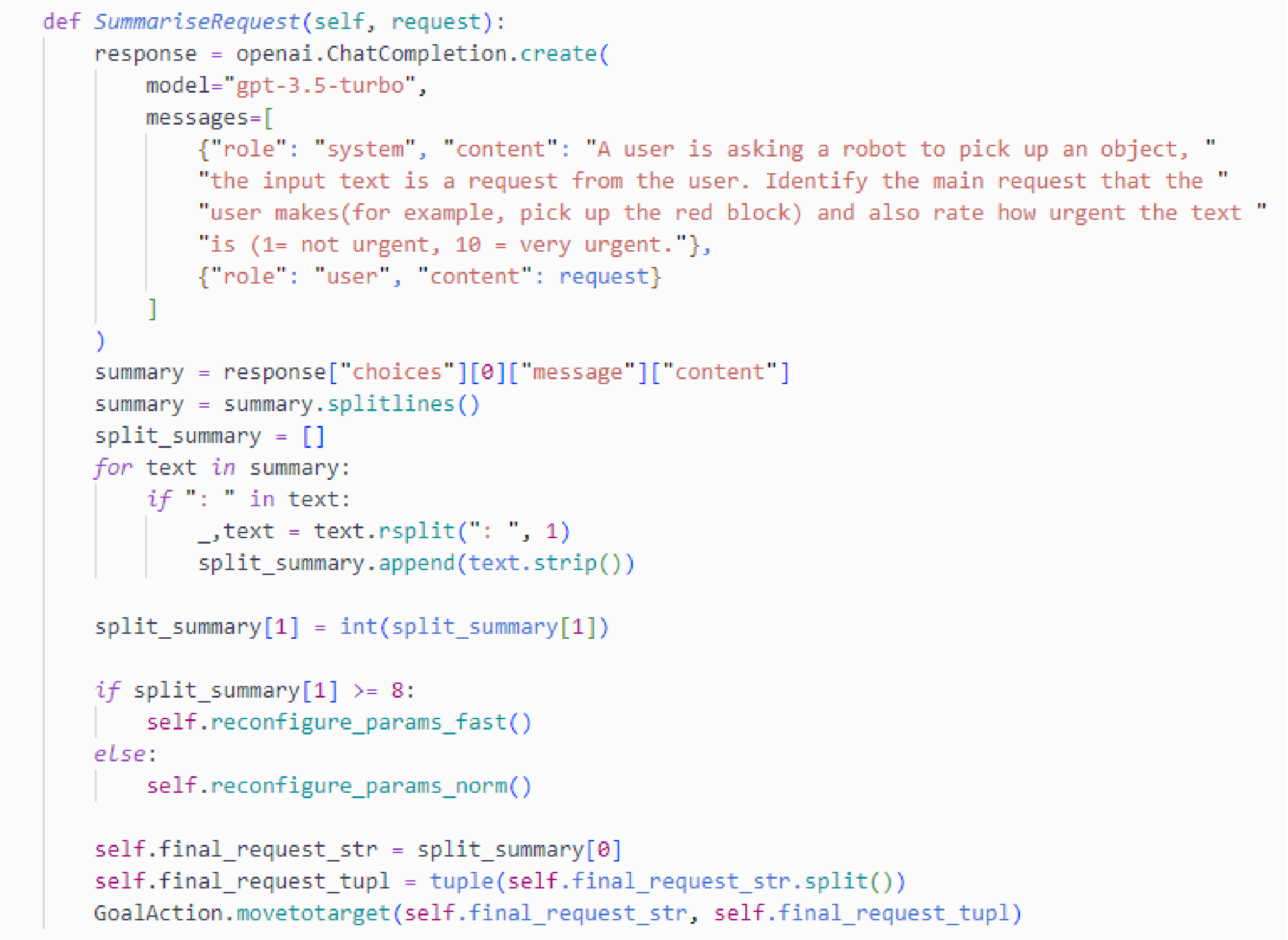
Appendix E: TaskThreader for Proximity Scanner



```
class TaskThreader:
    def __init__(self):
        pass

    def start_SpeakBot(self):
        # Start Speakbot
        GoalAction.setObjectInfo()
        while not rospy.is_shutdown():
            UserCommand.RecordAudio()
            break

    def start_ProximitySens(self):
        rospy.Subscriber("/scan", LaserScan, self.process_Scanner)
        rospy.spin()

    def process_Scanner(self, scan):
        lowest_val = sorted(scan.ranges)[:5]

        # The mean distance recorded is used instead of the minimum
        # to mitigate anomalies in the sensor readings.
        lowest_val = round((stats.mean(lowest_val)), 3)

        if hasattr(GoalAction, "running_status"):
            if lowest_val < 0.5 and GoalAction.running_status is False:
                rospy.loginfo("OBJECT DETECTED WHILST STATIC")
            else:
                pass
```

Appendix F. Summarisation Layer Prompt

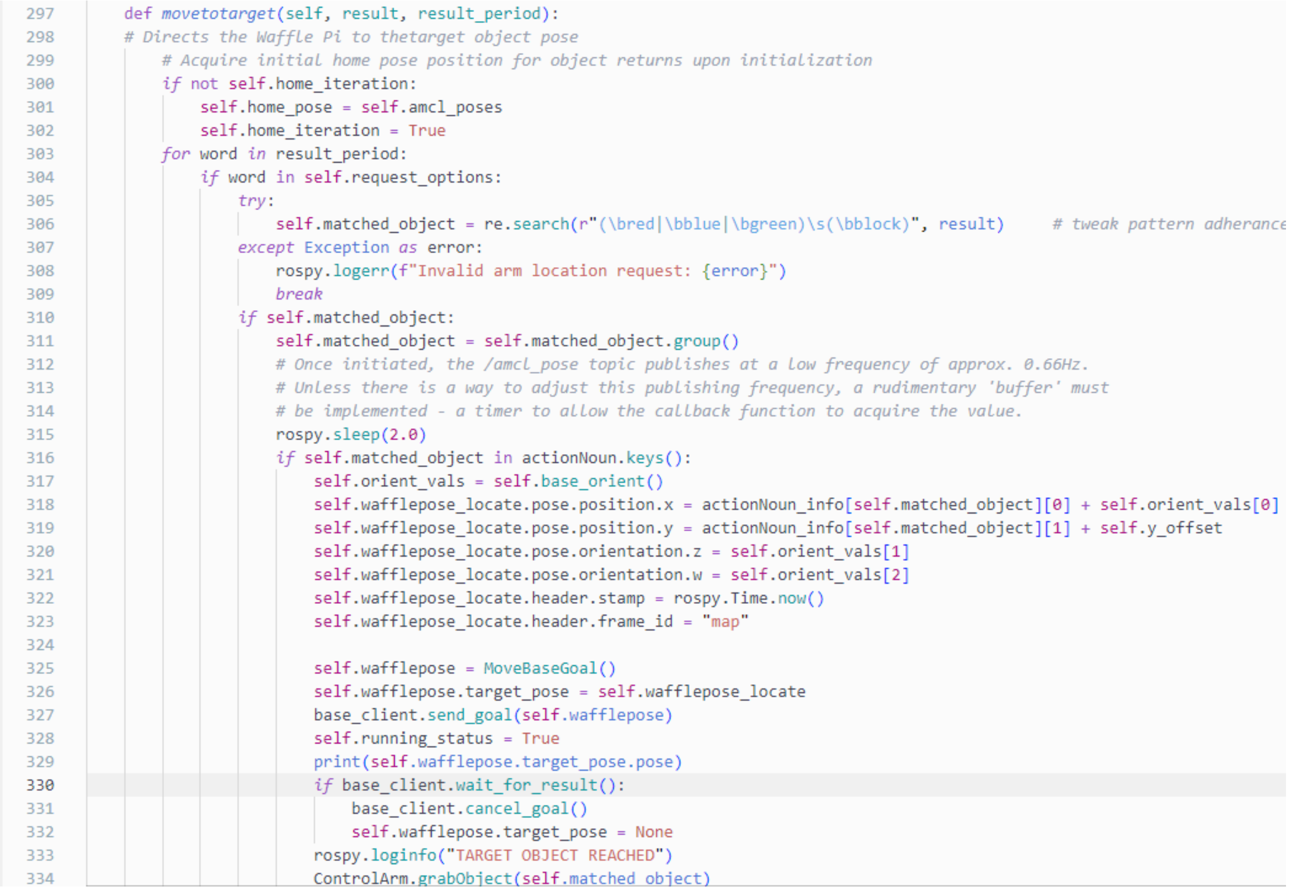
Overview of summarisation layer. A key improvement to this function would be to include a standard format for the resulting response from the LLM (e.g. JSON), to ensure robust, consistent response types.



```
def SummariseRequest(self, request):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "A user is asking a robot to pick up an object, "
            "the input text is a request from the user. Identify the main request that the "
            "user makes(for example, pick up the red block) and also rate how urgent the text "
            "is (1= not urgent, 10 = very urgent."},
            {"role": "user", "content": request}
        ]
    )
    summary = response["choices"][0]["message"]["content"]
    summary = summary.splitlines()
    split_summary = []
    for text in summary:
        if ": " in text:
            _,text = text.rsplit(": ", 1)
            split_summary.append(text.strip())

    split_summary[1] = int(split_summary[1])

    if split_summary[1] >= 8:
        self.reconfigure_params_fast()
    else:
        self.reconfigure_params_norm()

    self.final_request_str = split_summary[0]
    self.final_request_tupl = tuple(self.final_request_str.split())
    GoalAction.movetotarget(self.final_request_str, self.final_request_tupl)
```

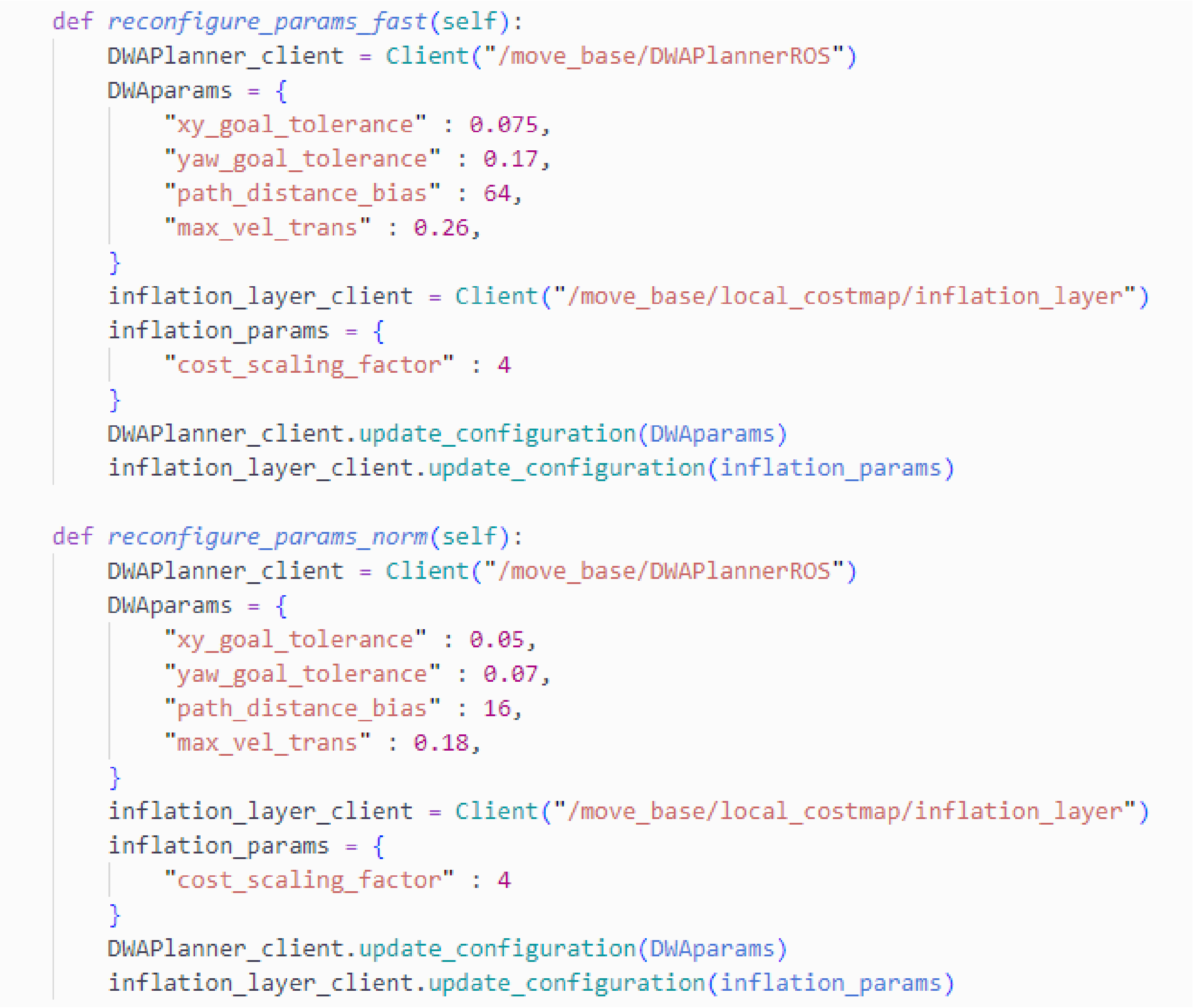
Appendix G: RegEx utilisation for target object identification

In showing the RegEx usage on line 306, the further algorithm for the identification of the target object pose and goal publishing to the MoveBaseGoal() class is supported.



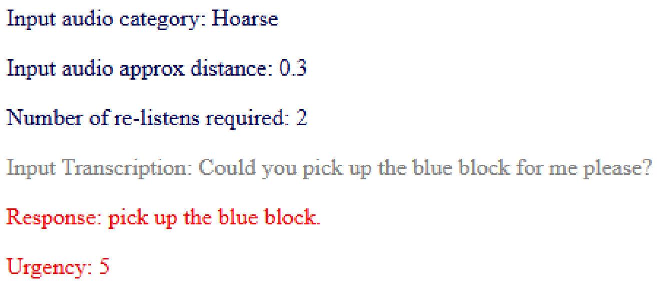
```
297     def movetotarget(self, result, result_period):
298     # Directs the Waffle Pi to thetarget object pose
299         # Acquire initial home pose position for object returns upon initialization
300         if not self.home_iteration:
301             self.home_pose = self.amcl_poses
302             self.home_iteration = True
303         for word in result_period:
304             if word in self.request_options:
305                 try:
306                     self.matched_object = re.search(r"(\bred|\bblue|\bgreen)\s(\bblock)", result)     # tweak pattern adherance
307                 except Exception as error:
308                     rospy.logerr(f"Invalid arm location request: {error}")
309                     break
310                 if self.matched_object:
311                     self.matched_object = self.matched_object.group()
312                     # Once initiated, the /amcl_pose topic publishes at a low frequency of approx. 0.66Hz.
313                     # Unless there is a way to adjust this publishing frequency, a rudimentary 'buffer' must
314                     # be implemented - a timer to allow the callback function to acquire the value.
315                     rospy.sleep(2.0)
316                     if self.matched_object in actionNoun.keys():
317                         self.orient_vals = self.base_orient()
318                         self.wafflepose_locate.pose.position.x = actionNoun_info[self.matched_object][0] + self.orient_vals[0]
319                         self.wafflepose_locate.pose.position.y = actionNoun_info[self.matched_object][1] + self.y_offset
320                         self.wafflepose_locate.pose.orientation.z = self.orient_vals[1]
321                         self.wafflepose_locate.pose.orientation.w = self.orient_vals[2]
322                         self.wafflepose_locate.header.stamp = rospy.Time.now()
323                         self.wafflepose_locate.header.frame_id = "map"
324 
325                         self.wafflepose = MoveBaseGoal()
326                         self.wafflepose.target_pose = self.wafflepose_locate
327                         base_client.send_goal(self.wafflepose)
328                         self.running_status = True
329                         print(self.wafflepose.target_pose.pose)
330                         if base_client.wait_for_result():
331                             base_client.cancel_goal()
332                             self.wafflepose.target_pose = None
333                         rospy.loginfo("TARGET OBJECT REACHED")
334                         ControlArm.grabObject(self.matched_object)
```

Appendix H. Dynamic Re-configure functions



```
def reconfigure_params_fast(self):
    DWAPlanner_client = Client("/move_base/DWAPlannerROS")
    DWAparams = {
        "xy_goal_tolerance" : 0.075,
        "yaw_goal_tolerance" : 0.17,
        "path_distance_bias" : 64,
        "max_vel_trans" : 0.26,
    }
    inflation_layer_client = Client("/move_base/local_costmap/inflation_layer")
    inflation_params = {
        "cost_scaling_factor" : 4
    }
    DWAPlanner_client.update_configuration(DWAparams)
    inflation_layer_client.update_configuration(inflation_params)

def reconfigure_params_norm(self):
    DWAPlanner_client = Client("/move_base/DWAPlannerROS")
    DWAparams = {
        "xy_goal_tolerance" : 0.05,
        "yaw_goal_tolerance" : 0.07,
        "path_distance_bias" : 16,
        "max_vel_trans" : 0.18,
    }
    inflation_layer_client = Client("/move_base/local_costmap/inflation_layer")
    inflation_params = {
        "cost_scaling_factor" : 4
    }
    DWAPlanner_client.update_configuration(DWAparams)
    inflation_layer_client.update_configuration(inflation_params)
```

Appendix I. Verbal Adherence – Live Test Results



Input audio category: Hoarse

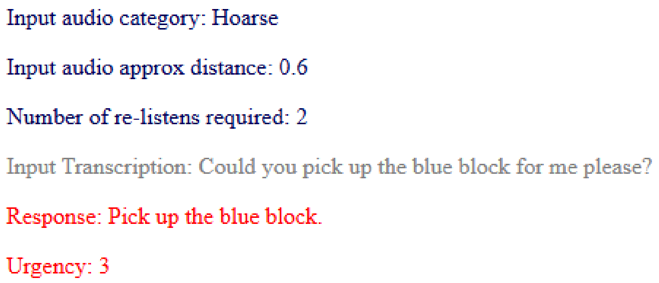
Input audio approx distance: 0.3

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: pick up the blue block.

Urgency: 5



Input audio category: Hoarse

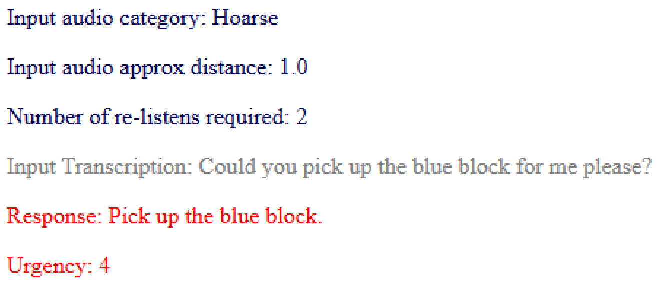
Input audio approx distance: 0.6

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block.

Urgency: 3



Input audio category: Hoarse

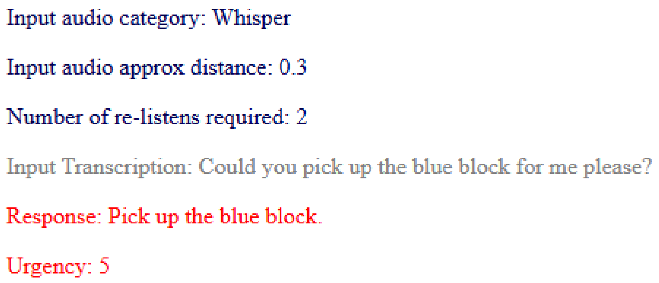
Input audio approx distance: 1.0

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block.

Urgency: 4



Input audio category: Whisper

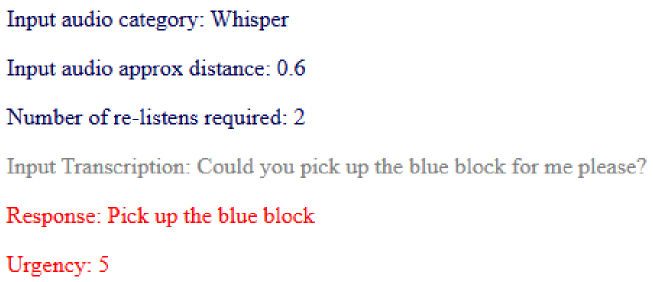
Input audio approx distance: 0.3

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block.

Urgency: 5



Input audio category: Whisper

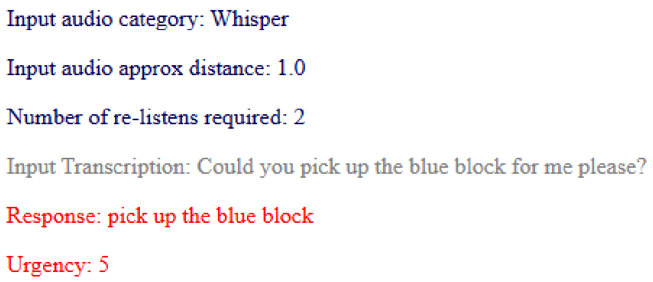
Input audio approx distance: 0.6

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block

Urgency: 5



Input audio category: Whisper

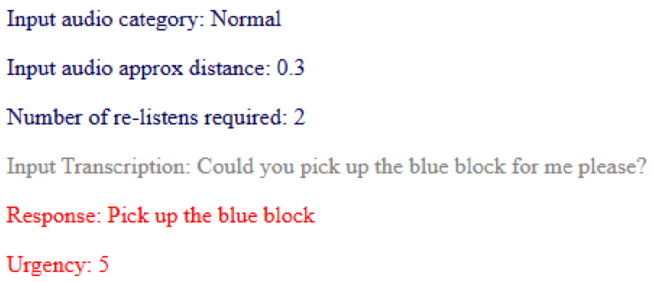
Input audio approx distance: 1.0

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: pick up the blue block

Urgency: 5



Input audio category: Normal

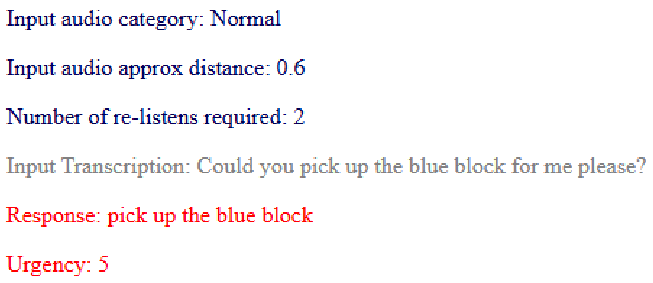
Input audio approx distance: 0.3

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block

Urgency: 5



Input audio category: Normal

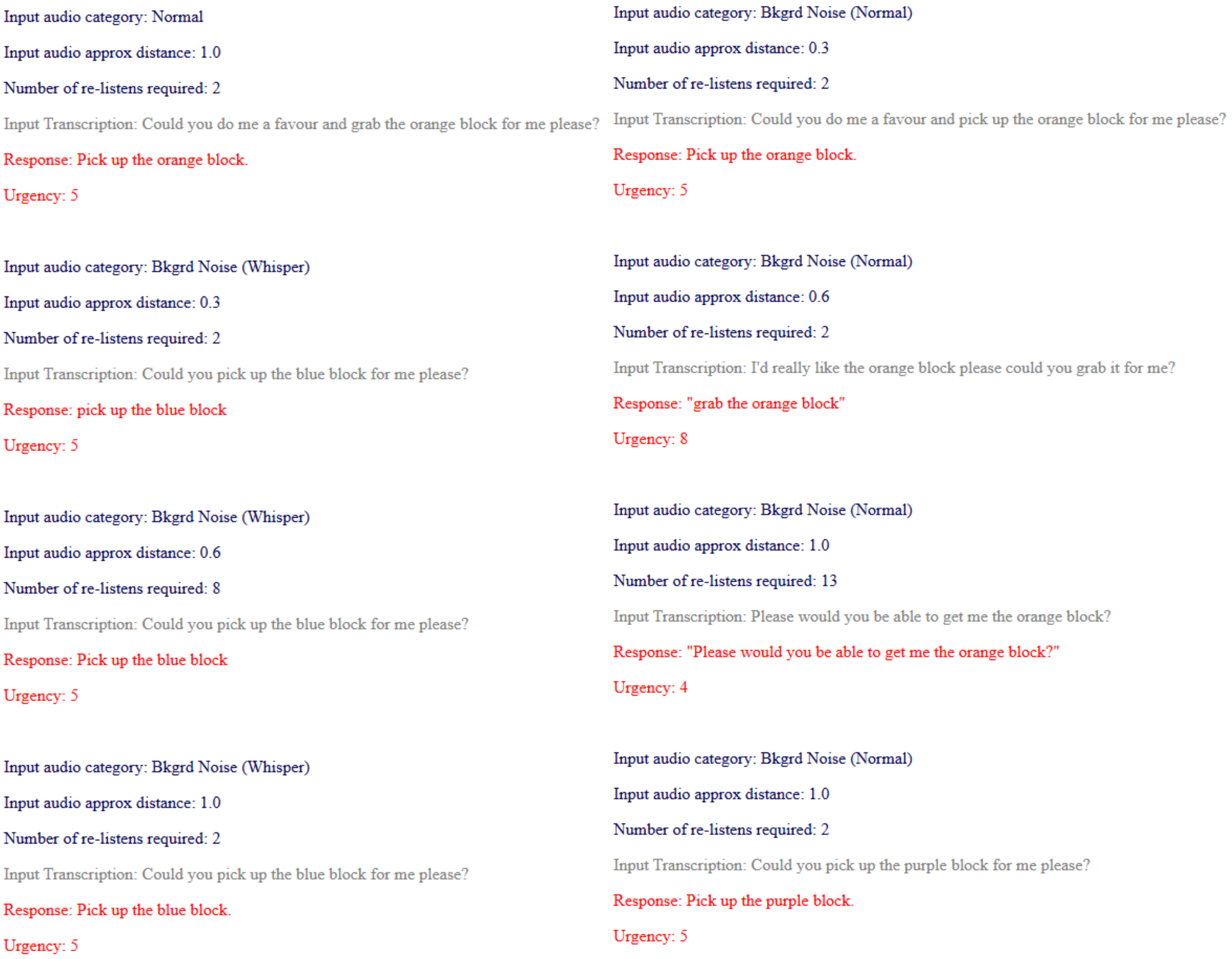
Input audio approx distance: 0.6

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: pick up the blue block

Urgency: 5



Input audio category: Normal

Input audio approx distance: 1.0

Number of re-listens required: 2

Input Transcription: Could you do me a favour and grab the orange block for me please?

Response: Pick up the orange block.

Urgency: 5

Input audio category: Bkgrd Noise (Whisper)

Input audio approx distance: 0.3

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: pick up the blue block

Urgency: 5

Input audio category: Bkgrd Noise (Whisper)

Input audio approx distance: 0.6

Number of re-listens required: 8

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block

Urgency: 5

Input audio category: Bkgrd Noise (Whisper)

Input audio approx distance: 1.0

Number of re-listens required: 2

Input Transcription: Could you pick up the blue block for me please?

Response: Pick up the blue block.

Urgency: 5

Input audio category: Bkgrd Noise (Normal)

Input audio approx distance: 0.3

Number of re-listens required: 2

Input Transcription: Could you do me a favour and pick up the orange block for me please?

Response: Pick up the orange block.

Urgency: 5

Input audio category: Bkgrd Noise (Normal)

Input audio approx distance: 0.6

Number of re-listens required: 2

Input Transcription: I'd really like the orange block please could you grab it for me?

Response: "grab the orange block"

Urgency: 8

Input audio category: Bkgrd Noise (Normal)

Input audio approx distance: 1.0

Number of re-listens required: 13

Input Transcription: Please would you be able to get me the orange block?

Response: "Please would you be able to get me the orange block?"

Urgency: 4

Input audio category: Bkgrd Noise (Normal)

Input audio approx distance: 1.0

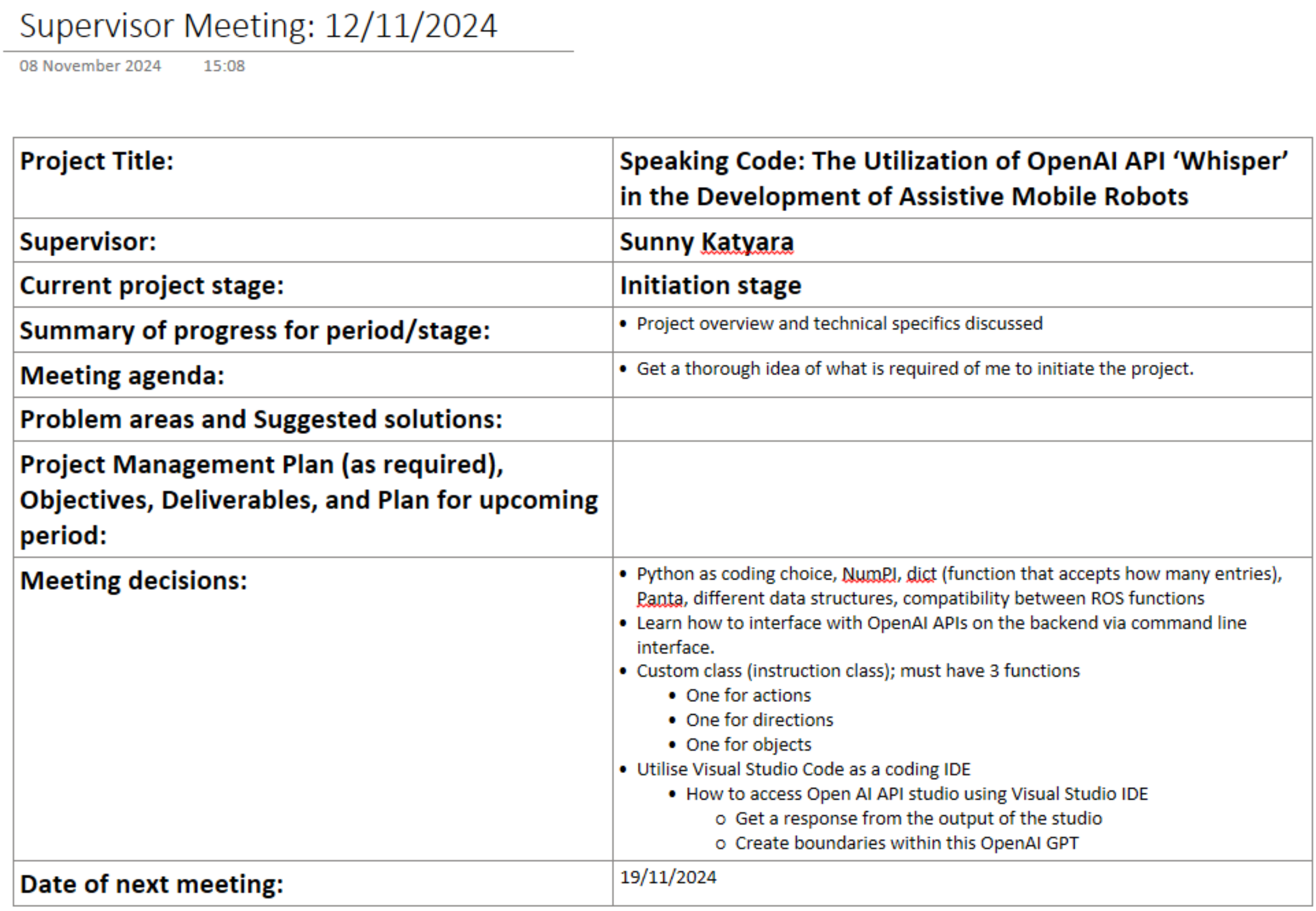
Number of re-listens required: 2

Input Transcription: Could you pick up the purple block for me please?

Response: Pick up the purple block.

Urgency: 5

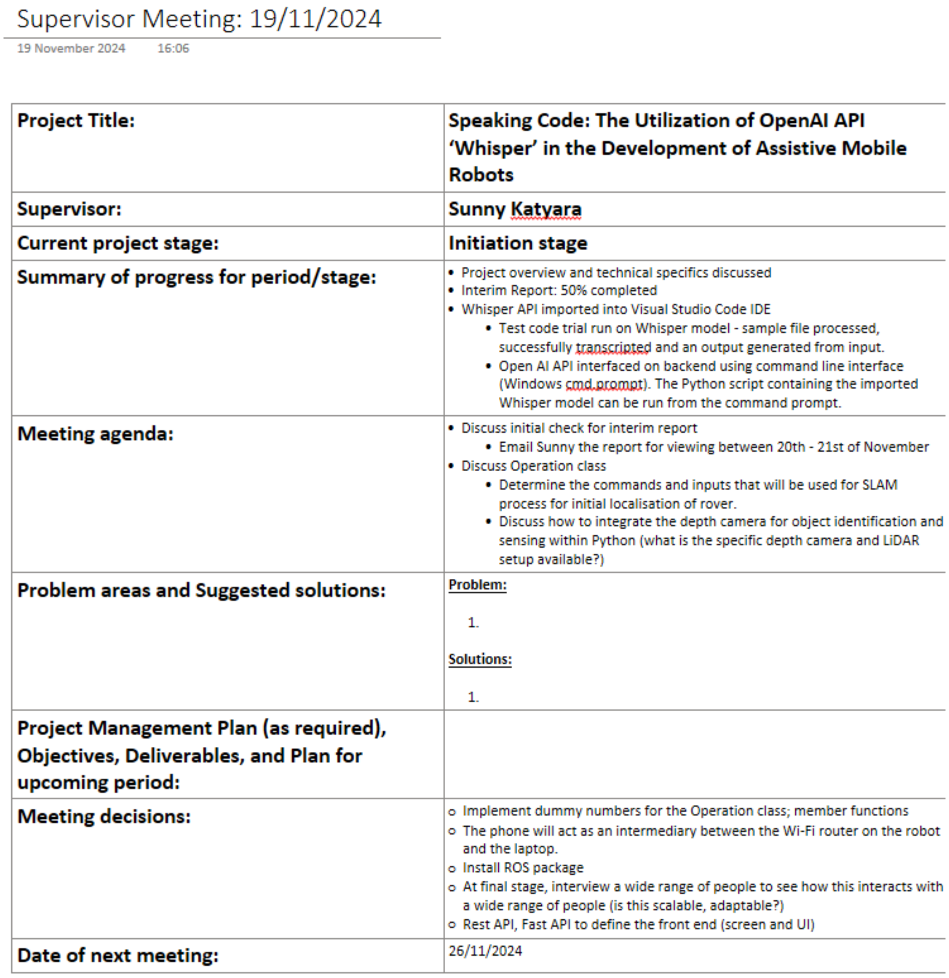
Appendix J. Initial Buy-Off Supervisor Meeting/s



Supervisor Meeting: 12/11/2024

08 November 2024 15:08

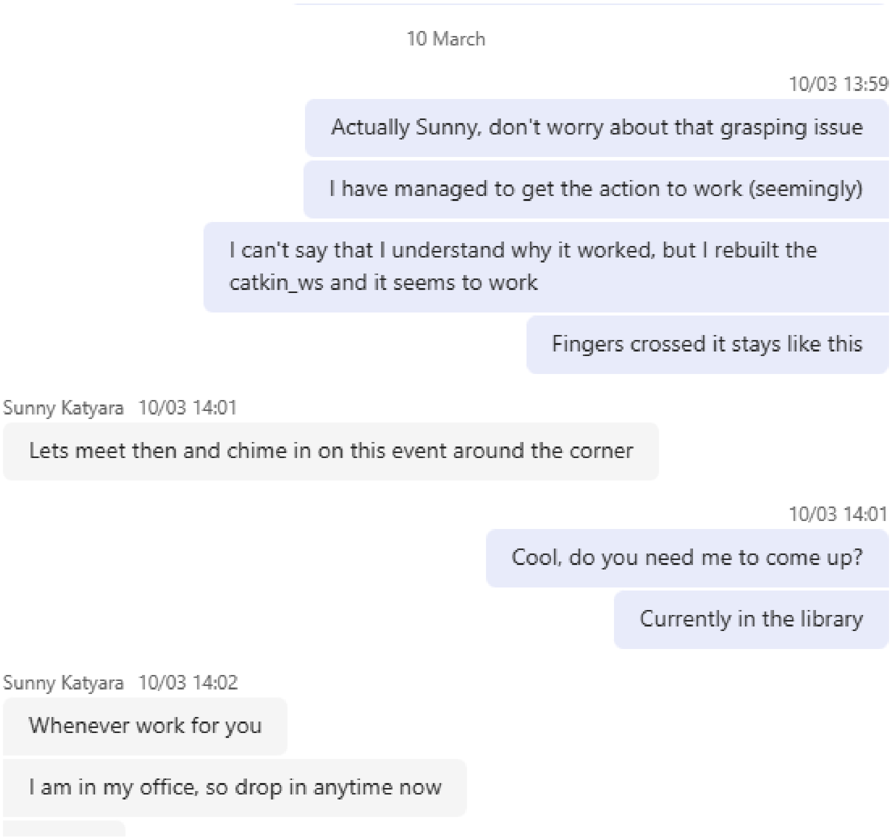
Project Title:	**Speaking Code: The Utilization of OpenAI API 'Whisper' in the Development of Assistive Mobile Robots**
Supervisor:	**Sunny Katyara**
Current project stage:	**Initiation stage**
Summary of progress for period/stage:	• Project overview and technical specifics discussed
Meeting agenda:	• Get a thorough idea of what is required of me to initiate the project.
Problem areas and Suggested solutions:	
Project Management Plan (as required), Objectives, Deliverables, and Plan for upcoming period:	
Meeting decisions:	• Python as coding choice, NumPI, dict (function that accepts how many entries), Panta, different data structures, compatibility between ROS functions • Learn how to interface with OpenAI APIs on the backend via command line interface. • Custom class (instruction class); must have 3 functions • One for actions • One for directions • One for objects • Utilise Visual Studio Code as a coding IDE • How to access Open AI API studio using Visual Studio IDE o Get a response from the output of the studio o Create boundaries within this OpenAI GPT
Date of next meeting:	19/11/2024



Supervisor Meeting: 19/11/2024

19 November 2024 16:06

Project Title:	**Speaking Code: The Utilization of OpenAI API 'Whisper' in the Development of Assistive Mobile Robots**
Supervisor:	**Sunny Katyara**
Current project stage:	**Initiation stage**
Summary of progress for period/stage:	• Project overview and technical specifics discussed • Interim Report: 50% completed • Whisper API imported into Visual Studio Code IDE • Test code trial run on Whisper model - sample file processed, successfully transcripted and an output generated from input. • Open AI API interfaced on backend using command line interface (Windows cmd.prompt). The Python script containing the imported Whisper model can be run from the command prompt.
Meeting agenda:	• Discuss initial check for interim report • Email Sunny the report for viewing between 20th - 21st of November • Discuss Operation class • Determine the commands and inputs that will be used for SLAM process for initial localisation of rover. • Discuss how to integrate the depth camera for object identification and sensing within Python (what is the specific depth camera and LiDAR setup available?)
Problem areas and Suggested solutions:	**Problem:** 1. **Solutions:** 1.
Project Management Plan (as required), Objectives, Deliverables, and Plan for upcoming period:	
Meeting decisions:	○ Implement dummy numbers for the Operation class; member functions ○ The phone will act as an intermediary between the Wi-Fi router on the robot and the laptop. ○ Install ROS package ○ At final stage, interview a wide range of people to see how this interacts with a wide range of people (is this scalable, adaptable?) ○ Rest API, Fast API to define the front end (screen and UI)
Date of next meeting:	26/11/2024



10 March

10/03 13:59

Actually Sunny, don't worry about that grasping issue

I have managed to get the action to work (seemingly)

I can't say that I understand why it worked, but I rebuilt the catkin_ws and it seems to work

Fingers crossed it stays like this

Sunny Katyara 10/03 14:01

Lets meet then and chime in on this event around the corner

10/03 14:01

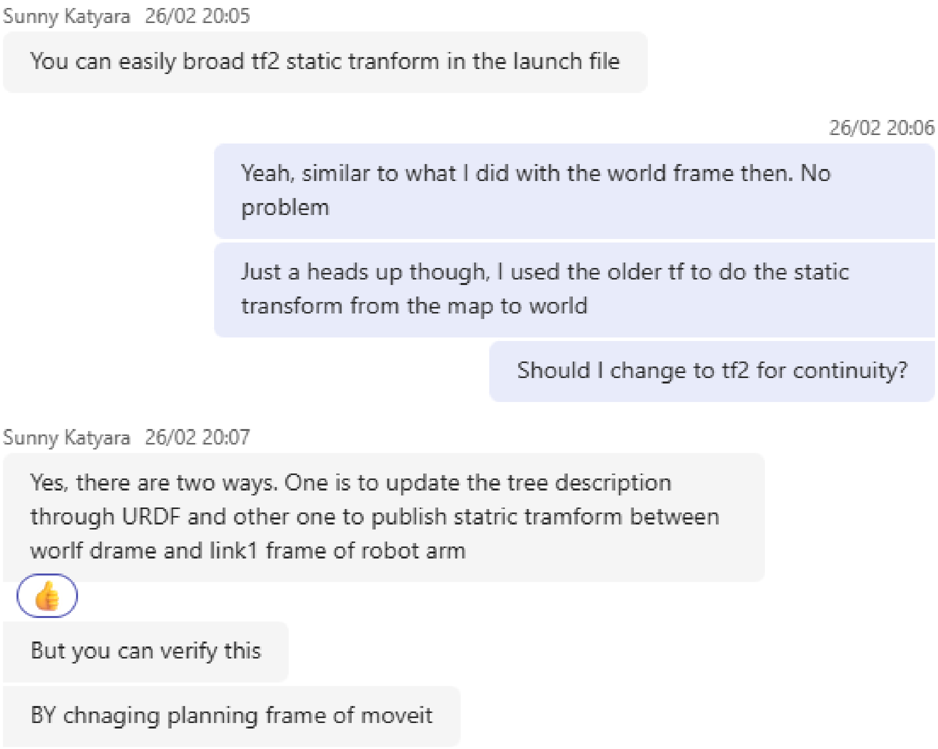
Cool, do you need me to come up?

Currently in the library

Sunny Katyara 10/03 14:02

Whenever work for you

I am in my office, so drop in anytime now



Sunny Katyara 26/02 20:05

You can easily broad tf2 static tranform in the launch file

26/02 20:06

Yeah, similar to what I did with the world frame then. No problem

Just a heads up though, I used the older tf to do the static transform from the map to world

Should I change to tf2 for continuity?

Sunny Katyara 26/02 20:07

Yes, there are two ways. One is to update the tree description through URDF and other one to publish statric tramform between worlf drame and link1 frame of robot arm

👍

But you can verify this

BY chnaging planning frame of moveit