ROBOT TRIAGE AND VARIABLE AUTONOMY FOR HUMAN-ROBOT TEAMING

By

ANIKETH RAMESH

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

Extreme Robotics Laboratory
School of Metallurgy and Materials
University of Birmingham
July 2024

Abstract

This thesis addresses the problem of systematically detecting, quantifying and mitigating runtime performance degradation in robots during task execution, through interaction with a human operator who can service robots through level of autonomy switching. Robots are increasingly popular today for executing a wide variety of tasks autonomously. However, robots often face runtime performance degradation (e.g. poor navigation, getting stuck, noisy sensor readings), which can cause them to fail task execution or perform sub-optimally. To continue task execution, it is crucial to detect situations where robots face issues so that operator assistance can be enabled through Variable Autonomy paradigms (i.e., switching between different levels of autonomy on demand). Although operators can constantly monitor a robot's task execution to detect situations where they need assistance, this is cognitively demanding. Furthermore, as the scale and complexity of robot missions increase, multiple robots will be required to execute tasks simultaneously and work together. This may result in multiple robots simultaneously requiring assistance with task execution, with varying degrees of relevance to the overall mission. In such situations, it is important to optimise the interaction paradigm so as not to overload the limited cognitive capacity of the human operator while not compromising the overall mission performance. AI agents can help with such situations, as they can effectively triage robots needing operator assistance and trigger autonomous recovery behaviours for robots if necessary. To this end, this thesis addresses the problem of human-multi-robot teaming by a) proposing a framework for detecting and quantifying performance degradation during task execution, b) proposing several methods to deal with the detected performance degradation via variable autonomy paradigms and human interactions, and c) systematically evaluating the proposed solutions in disaster response and remote inspection inspired scenarios via systematic HRI experiments with human participants.

Specifically, the first major contribution of this thesis is the'Robot Vitals and Robot Health' framework, which serves as a foundation for Human-Multi-Robot Teaming. Evidence is presented

that this framework can be used to estimate the robot's state and quantify its online performance degradation. The second contribution is using this framework to realise different variable autonomy architectures to enable a robot to overcome performance degradation by requesting the operators' help, self-regulating its autonomy level or blending varying degrees of operator input to improve task performance. Evidence is provided to demonstrate that conveying information about robot performance degradation to the operator through visual cues can significantly influence their approach to servicing the robot. This promotes a more risk-aware strategy, reducing the aggregate risk of robot failure during mission runtime. Lastly, insights from previous experiments are utilized to propose a novel interaction paradigm for Variable Autonomy Multi-Robot Systems, aimed at facilitating robot triage through multi-modal interface sensory cues. Experimental evidence shows that our design can significantly improve task performance and reduce aggregate risk of robot failure, which enhances the ease of use, transparency, and trust in the system without increasing the cognitive workload of the operator. An analysis of the experimental results is then presented, highlighting how human factors such as personality, trust, and user understanding of the system are crucial in improving the performance of Variable Autonomy Multi-Robot Systems. By carefully examining human interaction with robot teams and understanding key components responsible for improving task performance and minimising operator cognitive demand, this thesis serves as a solid methodological foundation for future work in the field of human-multi-robot teaming.

Publications and Author Contributions

Here we list all the publications arising from this thesis, and the scope of my contributions to each of them. Chapter 3 is based on the papers **P.1** and **P.2**. The work presented in chapter 4 is based on **P.3**, **P.4** and **P.6**. Lastly, section 5.1 is based on **P.5**. For all the papers listed below, my co-authors, Dr. Manolis Chiou and Prof. Rustam Stolkin, who are also my supervisors, contributed by assisting with ideation, providing suggestions on conducting experiments, and helping with writing and proofreading drafts.

P.1 Ramesh, A., Chiou, M. and Stolkin, R., 2021, March. Robot vitals and robot health: An intuitive approach to quantifying and communicating predicted robot performance degradation in human-robot teams. In Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction (pp. 303-307).

P.2 Ramesh, A., Stolkin, R. and Chiou, M., 2022. Robot Vitals and Robot Health: Towards Systematically Quantifying Runtime Performance Degradation in Robots Under Adverse Conditions. IEEE Robotics and Automation Letters, 7(4), pp.10729- 10736.

P.3 Braun, C.A., Ramesh, A., Rothfuss, S., Chiou, M., Stolkin, R. and Hohmann, S., 2023. Model Predictive Degree of Automation Regulation for Mobile Robots Using Robot Vitals and Robot Health. In-Press, 2023 World Congress of the International Federation of Automatic Control (IFAC).

P.4 Braun, Christian Alexander, et al. "Model Predictive Control of the Degree of Automation Optimizing Robot Health." 2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2023.

P.5 Ramesh, A., Englund, M., Theodorou, A., Stolkin, R. and Chiou, M., 2023. Robot Health Indicator: A Visual Cue to Improve Level of Autonomy Switching Systems. In VAT4HRI Workshop 2023 ACM/IEEE International Conference on Human-Robot Interaction

P.6 Ramesh, A., Braun, C.A., Ruan, T., Rothfuß, S., Hohmann, S., Stolkin, R. and Chiou, M., 2023, October. Experimental evaluation of model predictive mixed-initiative variable autonomy systems applied to human-robot teams. In 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 5291-5298). IEEE, doi:10.36227/techrxiv.24018825.v1.

I was the first author for papers **P.1**, **P.2**, **P.5** and **P.6**. In these papers I was responsible for modelling the problem, devising a hypothesis, designing the experiments and conducting them. Papers **P.3**, **P.4** and **P.6** were written in collaboration with Karlsruhe Institute of Technology (KIT). For papers **P.3** and **P.4**, I contributed equally with my main collaborator from KIT (we are listed as co-first authors). My collaborator carried out the experiments and provided the human behaviour models based on previous experiments conducted by their team. My contributions to these papers were mainly focused on ideation, formulation of the problem, structuring the scope of work, and writing the papers. Experiments for **P.6** were carried out at the University of Birmingham, and the co-authors from KIT mainly assisted during ideation, validating assumptions made to scope the problem and proofing drafts of the paper. Finally, **P.5** was carried out in collaboration with Umeå University in Sweden. I was the first author of this paper; I designed the systems and conducted the experiments. The second author was responsible for designing the questionnaires A.2.2 and A.2.3. My supervisors and Dr.Andreas Theodorou contributed to proofreading the paper and defining the scope of work carried out in this paper.

Dedicated to my family, friends, junk food, zoom meetings and caffeinated drinks.

ACKNOWLEDGMENTS

The Oxford dictionary defines ‘Acknowledgements’ as “a statement, especially at the beginning of a book, in which the writer expresses thanks to the people who have helpe” If we’re being honest, no one truly “helped” me in the most literal sense of the word. So, in theory, I should be acknowledging no one. Unfortunately, this thesis is about robotics, not about coining a new term for the experience of having people endure my rants, disengagement, meltdowns, confusion, and self-doubt, all while responding with unwavering positivity, kindness, and support. As much as I’d like to believe my intelligence got me here, I’ve been buoyed along by a sea of people brimming with love and compassion. So here’s me NOT acknowledging them.

First up, let’s talk about my supervisor, Dr. Manolis Chiou. If I had to sum up his role in one word, it would be ‘Sprezzatura’—the art of making everything look effortless while being incredibly effective. Manolis provided the perfect blend of support and challenge, maintaining just the right balance of distance while still being personally invested. Whether it was burning the midnight oil to help rewrite parts of my papers and thesis or being more invested in my growth than I was, Manolis has been an irreplaceable guide. The greatest asset of these four years has been trying to imbibe some of Manolis’s qualities and work ethic. Now, my other supervisor, Prof. Rustam Stolkin—I can’t imbibe his qualities because they are a superpower. He can talk to you for just an hour, and you’ll walk away with enough hypotheses and research ideas for four PhD students. Rustam, your ability to generate ideas has not only sustained my PhD but also provided me with enough party conversation starters for the next decade.

To the co-authors of my (our) papers—Madeleine Englund, Andreas Theodorou, Christian Braun, and Simon Rothfuß—it was as much fun hanging out with you as it was building on your

insights for my work. Your contributions were invaluable, and the camaraderie was a delightful bonus.

Given the word limit for this thesis, I'll have to lump a few special people into the category of 'family and friends'. While I may have conducted the experiments, written the thesis, and authored a few papers, you all are the ones who actually made this thesis possible. For all the slack you've cut me, for enduring my endless PhD excuses, for the countless pointless conversations at 3 A.M. where I used you as sounding boards, and for helping me bounce back when the stress was too much—thank you. Your unwavering support, endless patience, and ability to keep me grounded (and mostly sane) were the true foundations of this work. Thank you for being my cheerleaders and my reality checks.

Sometimes science is

a lot more art than science

A lot of people don't get that.

Rick, Rick and Morty

Contents

List of Figures

List of Tables

Acronyms

DoA Degree of Autonomy. vi, viii, 8, 37, 88, 98, 101, 120, 121, 181, 182

EKF Extended Kalman Filter. 54, 56, 57

HF High Friction. v, 64, 65, 67, 69–72

HI Human Initiative. 32–35, 40

HI-LoA Human Initiative Level of Autonomy. 8, 32, 33

HMRT Human Multi-Robot Teaming. 5, 13, 29, 30, 39, 77

HRI Human-Robot Interaction. 10, 18, 39, 43, 44

HRT Human-Robot Teaming. 5

LIDAR Light Detection and Ranging. 24, 47, 66

LoA Level of Autonomy. vi, viii, 8, 9, 11, 15, 19, 23, 24, 26, 28–38, 40–44, 78–80, 83–85, 87, 88, 93, 95–101, 103–108, 111, 112, 114–116, 118–121, 136, 139, 140, 176, 178, 181, 182

MI Mixed Initiative. 9, 15, 21, 33, 35, 36, 40, 42, 45, 47

MI-LoA Mixed Initiative Level of Autonomy. 9, 15, 35, 36

MRS Multi-Robot System. 6, 9, 11, 16, 29, 30, 38, 39, 41, 42, 44–46, 136, 149

NEWS 2 National Early Warning Score 2. v, 50, 51

OOTL Out Of The Loop. 28, 31, 34, 37

RI Robot Initiative. 33–35

RI-LoA Robot Initiative Level of Autonomy. 9

SLAM Simultaneous Localization and Mapping. 56–58, 62

UCD User-Centred Design. 10

UGV Unmanned Ground Vehicle. 20–22

UMUX Usability Metric for User Experience. 44

VA Variable Autonomy. 8, 9, 11, 13, 24, 26, 27, 29, 31, 32, 34, 38–48, 77, 124, 136, 173

VA HRT Variable Autonomy Human-Robot Team. 10

VA MRS Variable Autonomy Multi-Robot System. 10–13, 31, 38, 39, 41, 44, 48, 140, 176

Chapter One

Introduction

In 1966, the Stanford Research Institute developed the first fully autonomous intelligent robot (Hart et al., 2017) capable of making its own decisions about how to execute tasks. When given simple English instructions such as "move the block onto the table", the robot could generate a plan to accomplish the task and then execute it. The robot could detect the table, the block, navigate around obstacles, and then successfully move the block to the table. This landmark achievement marked the dawn of a new era of robotics and autonomous systems. Today robots are ubiquitous, finding applications in a wide range of domains like extinguishing fires at Notre Dame Cathedral (Cassel, 2021), spraying fertilisers (S.L, 2023), picking berries on British farms (*Automation in horticulture review* 2022), sorting groceries in warehouses (Morris, 2024), and decommissioning hazardous waste at nuclear sites (Chiou, Epsimos, et al., 2022). The wide range of applications showcases the remarkable strides in robotic technology and highlights the increasing awareness of robots' capabilities to address diverse challenges.

To identify which tasks are well-suited for robotisation, experts in the field commonly use the 3D's test. The 3D's refer to Dull, Dirty, Dangerous tasks (Marr, 2017). Dull tasks are repetitive and tedious, such as sorting items in a warehouse or assembling components on a production line. These tasks do not require human creativity or cognitive abilities, making them prime candidates

for robotic automation. By delegating these monotonous tasks to robots, industries can improve efficiency and productivity while freeing up human workers for more complex and intellectually stimulating endeavours. Dirty tasks, such as maintenance and monitoring in sewage systems, or sorting recyclables from garbage in domestic waste processing, are human labour-intensive tasks which can be deeply unpleasant despite being critically important for the basic functioning of society. Finally, dangerous tasks, such as search and rescue operations, inspection and maintenance of infrastructure (at height e.g. wind turbines, or underwater, e.g. offshore drilling), clean-up of legacy nuclear waste, space exploration, or bomb disposal, inherently expose human workers to hazards of various types and degrees. The deployment of robots in these scenarios minimises the exposure of human workers to life-threatening situations, making them a safer and more practical solution (Murphy, 2014). Furthermore, expert human workers can be potentially more productive by supervising such robots remotely from a safe zone, rather than entering the hazardous zones themselves to perform the work manually (R. Stolkin et al., 2022).

Regardless of a robot's advanced capabilities, it comes with inherent limitations (Delmerico et al., 2019). Therefore, robots are not always reliable and need human supervision during task execution. To date, deployments of robots in safety-critical and high-consequence applications have largely been directly teleoperated for two main reasons. Firstly, such applications (exploring a collapsed building or manipulating mixed nuclear waste) tend to involve significantly unstructured scenes and objects, which to date have required human intelligence to understand, in contrast to fully automated tasks which pervade manufacturing. Secondly, high-consequence industries tend to be highly conservative, with rigorous regulatory agencies and frameworks which govern the planning and execution of safety critical tasks (R. Stolkin et al., 2022). Even with the advent of sophisticated autonomous systems; these frameworks will continue to mandate a human-in-the-loop approach (Santoni de Sio and Van den Hoven, 2018; Middleton et al., 2022) as human oversight for autonomous systems, to ensure safety, accountability and ethical decision-making.

Therefore, most robotic applications in safety-critical and unstructured environments are

overseen by a skilled human operator, who ensures the task is accomplished successfully. This operator's job varies widely depending on the application and its context. This includes controlling a fire extinguishing robot manually and directing the water sprayed at the fire (Cassel, 2021), planning a path for a drone to spray fertilisers across a field, or inspecting the camera feed when a robot is conducting reconnaissance in a search and rescue task (Murphy, 2014). Robotics research problems focused on enabling humans and robots to work together to achieve shared goals, are called Human-Robot Teaming (HRT). In particular, scaling the number of robots that a single operator can manage together by increasing the autonomous capabilities of each robot in a multi-robot system, simplifying decision making for operators, and improving methods by which operation data can be easily visualised, are referred to as Human Multi-Robot Teaming (HMRT) research. However, in existing literature, Human-Robot Teaming and Human Multi-Robot Teaming are used interchangeably.

1.1 Robotics for Extreme Environments

Extreme or hazardous environments, characterised by harsh, unpredictable, and often dangerous conditions, pose significant challenges for human presence and activities. These environments show a paradox. On one hand, they are well-suited for robotisation, as the tasks carried out in such environments are largely dull, dirty, and dangerous, aligning well with the "3Ds" of robotisation (UKRI, 2020). On the other hand, due to the unstructured and dynamic nature of these environments, fully autonomous robots cannot be deployed reliably, making them well-suited for Human-Robot Teaming (J. Y. Chen and Barnes, 2014). Examples of extreme environments include nuclear facilities, deep-sea, disaster sites, space, or extraterrestrial terrains. Robots may need specialised hardware and software to function in such environments (UKRI, 2020). For example, in certain kinds of nuclear environments (e.g. high gamma radiation), robots and their sensors require shielding or radiation hardening design approaches to withstand radiation damage. Undersea

robots need to be built in a hydrodynamic manner to withstand hydrostatic pressure, minimise drag, improve efficiency and manoeuvrability. Hence, this thesis focuses on extreme environments due to their unique challenges and the critical need for effective human-robot collaboration. More specifically, the focus is on mobile robots in search and rescue and remote inspection scenarios, as they provide challenging testing fields for applications, and offer rich context for investigating and improving human-robot interaction, integrating Artificial Intelligence (AI) agents, interface design issues, and optimising operator workload (Chiou, Rustam Stolkin, et al., 2016).

While it is possible to deploy a single, multi-functional robot to execute tasks in extreme environments, this makes the system highly susceptible to single point failures. Therefore, it is more cost-effective to use a team of multiple robots with specialised, redundant functionalities, working concurrently towards completing mission goals. These are called Multi-Robot System (MRS). Multi-Robot System (MRS) adoption for tasks in extreme environments is receiving growing interest, because of their ability to efficiently distribute workload across the entire team, and cover more ground at lower costs (Rizk, Awad, and Tunstel, 2019; Murphy, 2014; Zhou and Tokekar, 2021). For example, in nuclear sites, where operators find it difficult to communicate with robots due to thick concrete shielding, multi-robot teams can pass messages to each other. Each robot in a multi-robot team may have their own tasks, or may be working together with the team to achieve a common goal. In search and rescue missions, a robot team will need to use mapping and self-organising techniques to spread out and identify targets. Then, the robots closest to the target may have to clear debris or grasp door handles to transport the stranded victims back to safety.

A crucial requirement for robots operating in extreme environments is the ability to make autonomous decisions. This is because robots operating in such environments are prone to lapses or delays in communication with the robot operator or mission control. Additionally, robots are also likely to experience runtime faults or unreliable sensor readings during runtime (Murphy, 2014; Morales et al., 2019). In such situations, robots should possess the intelligence or autonomous capabilities onboard to continue operating without interruptions. This includes navigating around

difficult obstacles, filtering noise from sensors, and modifying mission goals according to new information. Multi-robot systems may also have to redistribute mission goals among each robot or change formations depending on which robots get damaged or which robots are facing communication faults (Zhou and Tokekar, 2021). Current research provides a lot of interesting approaches and algorithms for perception, grasping, navigation, etc. to equip robots with autonomous capabilities to perform complex tasks with robustness against runtime adversities (C. Wong et al., 2017).

Robots in extreme environments are typically overseen by at least one mission supervisor. For example, robotic search missions require two operators per robot, namely the robot driver and the problem holder (Murphy, 2014). The robot driver is in charge of driving the robot through the disaster site, and the problem holder is in charge of observing the video feed, acquiring situational awareness, communicating with first responders and directing the driver according to mission goals. Similarly in aerial reconnaissance missions, each drone requires two operators (Minculete and Păstae, 2023). One operator drives the drone through a prescribed patrolling route, while the other spots features of interest in the camera footage and executes the required tasks during runtime. Operators in these missions require extensive training which is costly and takes time. As the complexity and scale of missions in extreme environments increase, with larger numbers of robots being deployed, this 2:1 ratio of operators to robots may become unfeasible. Therefore, there is a pressing need to reduce the operator to robot ratio.

1.2 Variable Autonomy Systems

A significant portion of robotics research posits that once a robot is equipped with advanced algorithms, it will be capable of executing the tasks smoothly with little to no assistance. While robots can potentially sustain long periods of autonomous operation, the role of a human in the loop re-

mains indispensable to mission execution (Rosenfeld et al., 2017). This is due to three reasons: 1) Robot autonomy is not always reliable; 2) Robots sometimes require assistance with mission execution and decision making; and 3) Robot operators are accountable for all mission-related decisions. Therefore, robot operators carry out a wide variety of tasks during mission runtime, such as monitoring robot camera feed to identify victims or points of interest, interpreting sensor data to acquire situational awareness, monitoring mission goals and relaying information to first responders. In case of MRS missions the operator is also in charge of monitoring the overall team's formation and delegating work to different robots (Kolling et al., 2015). In situations where robots require assistance with overcoming adversities, the robot operator may also have to take control of the robot to continue task execution.

Variable Autonomy (VA) robotic systems are robots that are capable of adjustable amounts of operator control during runtime. The adjustment spans across the continuum from Full Autonomy to Manual Control (Reinmund et al., 2024). This can be implemented as fixed Levels of Autonomy (LoAs), (Chiou, Bieksaite, et al., 2016) or variable Degrees of Autonomy DoAs (Christian A Braun, Flad, and Hohmann, 2019). A VA system where a human is in charge of switching between the Levels of Autonomy (LoAs), is called a Human Initiative Level of Autonomy (HI-LoA) switching system (Chiou, Rustam Stolkin, et al., 2016; Chiou, Bieksaite, et al., 2016). These are the most common implementations of VA systems. Typically HI-LoA systems have 2 LoAs: 1) Autonomy; and 2) Manual Control. HI-LoA implementations enable operators to provide assistance to robots when their autonomous capabilities are compromised. For example, in a mobile robot navigation task, an operator can give a robot a goal, and monitor the robot while it moves towards the goal autonomously. While the robot is moving autonomously, the operator can monitor the mission execution, acquire situational awareness and coordinate with first responders (Murphy, 2014). If the operator notices the robot is facing difficulties with navigation due to runtime adversities, or is unable to localise itself due to noisy sensor data the operator can switch the robot to manual control and assist it with task execution. When the adversities have been dealt with, the

operator can switch the robot back to Autonomy. This ability to switch between a robot's LoAs makes VA robotic systems effective at enabling operator assistance.

Mixed Initiative Level of Autonomy (MI-LoA) switching systems are Variable Autonomy implementations where both a human operator and an AI agent are capable of triggering Level of Autonomy (LoA) switches. AI agents are able to parse through large quantities of data, simulations, or even previous trials to learn how to address a variety of situations they encounter during mission runtime (Rosenfeld et al., 2017; Kaufmann et al., 2021). This makes their use key to alleviating the cognitive workload of VA robots and Multi-Robot Systems (MRSs). In contrast to Robot Initiative Level of Autonomy (RI-LoA) systems, i.e. VA implementations where only the AI or the robot's onboard controller is in charge of triggering LoA switches, MI-LoAimplementations allow AI agents to benefit from the judgement of a human operator, who is more mindful of varying mission requirements and ethical choices.

The importance of Mixed Initiative (MI) systems becomes evident when considering the challenges faced by human operators in multi-robot teams. The multitude of tasks that operators have to carry out during runtime, combined with the high-risk nature of the mission, imposes very high levels of cognitive workload on them. This cognitive workload increases with super-linear complexity for every additional robot added to the team (Kolling et al., 2015). This high cognitive demand will be increased further if the operator is tasked with regulating the LoA of each robot in the mission. For this, an operator has to expend mental resources to decide when a robot requires servicing, then switch the robot to manual control and help the robot overcome adversities. The operator then has to determine when to switch the robot back to autonomy. When operator cognitive abilities are overloaded, they tend to manage their attention poorly and service robots sub-optimally, thereby deteriorating overall mission performance. Furthermore, high cognitive workload can cause operators to feel overloaded, fatigued, and burnout quickly. Therefore, reducing the workload of operating robots in extreme environments is crucial to preventing operator burnout and improving the operator to robot ratio in robot missions.

1.3 Human-Robot Interaction for Variable Autonomy Multi-Robot Systems

Interaction between humans and robots is a fundamental part of Variable Autonomy Human-Robot Teams VA HRTs. The success of these teams relies heavily on effective communication and collaboration between human operators and robotic systems. Human-Robot Interaction design is the science of studying how people interact with robots, focusing on their behaviour and attitudes towards robots in relationship to the physical, technological and interaction features of the robot (Prati et al., 2021). This field focuses on understanding, designing and evaluating robotic systems for use by or with humans, to finally understand and shape the interaction between one or more humans and one or more robots (Michael A. Goodrich and Schultz, 2007). For this purpose, Human-Robot Interaction (HRI) is highly interdisciplinary, covering several research areas such as engineering, technology, psychology, design, anthropology, sociology and philosophy. Each of these disciplines addresses different aspects of the human-robot interaction. Therefore, they are all important for developing a successful interaction paradigm which respects both the human's strengths and limitations and the robots.

Effective interaction design for Variable Autonomy Multi-Robot System (VA MRS) will require HRI to move beyond the traditional engineering design approaches and adopt User-Centred Design (UCD) approaches (Pizzagalli, Kuts, and Otto, 2021). UCD focuses on optimising the interaction paradigm while being considerate of human factors like workload, vigilance and situational awareness. While improving the autonomous capabilities of robots in a VA MRS are beneficial, this introduces new problems like over-reliance of the operator on automation, out-of-the-loop performance problems and operator complacency. This is of particular concern during remote operation of robots in extreme environments (Murphy and Tadokoro, 2019). The interaction paradigm should promote a feeling of tele-presence for the operator. Tele-presence is "the

ideal of sensing sufficient information about the teleoperator and task environment and communicating this to the human operator in a sufficiently natural way, that the operator feels physically present at the remote site." as defined by Sheridan, Verplank, and T. Brooks (1978) (note that the term "teleoperator" in this quotation refers to the remote robot, and not the human supervisor). This includes the interface presenting all the information required by the operator to assess mission requirements and the robot's state in a manner that does not overload the operator's visuo-spatial abilities. Given the nature of tasks in extreme environments, issues like lapses in communication with robots and varying mission requirements make it difficult to determine which robots need assistance. Therefore, the interaction should be designed to provide operator situational awareness while compensating for runtime issues. Additionally, when designing interfaces for VA MRS systems, the interaction should be designed such that there is a careful balance between information required to control a single robot, and information about the other robots in the MRS. Information about which robot the operator is controlling as well as the LoAs of each robot in the team should be communicated to the operator in a simple and effective manner.

The variety of factors associated with optimal interaction design makes it a non-trivial task. As the autonomous capabilities of a robot increase, it abstracts increasing amounts of information from the human operator. However, this should not come at the cost of the operator's situational awareness or the transparency of robot actions. Therefore, sensor fusion techniques are essential for combining all information relevant to robot operation and mission objectives, and presenting it to the operator in a way that fosters trust and ensures transparency. Using best practices from existing literature, this information should be curated to prioritise the operator's attention and cognitive workload (Murphy and Tadokoro, 2019; Moniruzzaman et al., 2022). Thus, the thesis of this study is that carefully integrating information about a robot's performance degradation into various parts of the interaction paradigm can simplify autonomy regulation for VA robots.

1.4 Triage for Variable Autonomy Multi-Robot Systems

One of the main sources of mental demand in VA MRSs is what is referred to as 'Triage' in this thesis. This is the process of monitoring each robot's task execution, detecting when robots require help, deciding the nature of intervention, and servicing them in a timely manner. This thesis explores how different components of the interaction paradigm can contribute to reducing the cognitive demand of 'Triage'. Triage is an open problem in existing literature that requires a cross-disciplinary approach. Rather than building an AI agent for Triage, the inquiry is focused on understanding how to realise a Triage-Centric User Experience. This requires a systemic view of the problem, as a broader perspective would allow us to develop insights across the Human Multi-Robot Interaction paradigm.

Potential solutions include integrating AI agents into the control architecture to handle tasks like execution monitoring, planning, and target identification, allowing operators to focus more on triage (Kaufmann et al., 2021). Another option to offset the operator's mental demand is for AI agents to run state estimation algorithms that prioritise robots based on service needs (Rosenfeld et al., 2017) for easy servicing (Dahiya et al., 2022). Through Mixed Initiative implementations, AI agents could also trigger LoA switches or pre-programmed recovery behaviours to improve mission performance (Chiou, Hawes, and Rustam Stolkin, 2021; Rigter, Lacerda, and Hawes, 2020). AI agents could be integrated into the mission control interface in many different ways, e.g. through voice-based recommender systems, visual or haptic cues. Crucially, information from the robot's runtime and mission data must be presented effectively to enable quick decision-making without overloading the operator's visuo-spatial skills or cognitive capabilities. Most importantly, these elements should be carefully integrated to ensure scalability (Dahiya et al., 2022), promote trust in the AI agents and robots, ultimately providing a good overall user experience (Hussein, Ghignone, et al., 2018).

To build a triage-centric user experience, the first step is the creation of a shared knowledge representation for all team agents. In collaborative teams with diverse roles and competencies, a shared mental model of the environment serves as a common knowledge base for communication, coordination, and decision-making (Schelble et al., 2022). This shared understanding promotes trust and enhances transparency in reasoning about actions and decisions. In this thesis, developing a framework is explored to: 1) estimate each robot's state in a VA MRS system; 2)help operators easily determine when a robot needs assistance; and 3)serve as a shared knowledge representation for the entire Human Multi-Robot Team. To this end, the concept of the 'Robot Vitals and Robot Health Framework,' a simple and intuitive approach to prioritising robots requiring operator assistance and streamlining triage is introduced in this thesis.

The Robot Vitals and Robot Health Framework quantifies robot performance degradation. Intuitively, robots facing high levels of performance degradation are regarded as having a low 'health', therefore making it easy for robot operators to detect when to assist robots. In this study, the Robot Vitals and Robot Health Framework is applied to VA robotic systems. How information about a robot's performance degradation can facilitate HMRT is examined in two different ways: 1) how an operator can use information about a robot's performance degradation to assist with task performance and whether this can reduce the mental demand of HMRT; and 2) whether a robot can use information about its performance degradation to self-regulate its level of autonomy and request operator support when necessary. This thesis begins by exploring these questions in a single-operator, single-robot system to gain scalable insights. Subsequently, an experimental study is conducted to assess how robot performance degradation information can enhance VA MRS performance in a simulated multi-robot remote navigation task. This approach enables the systematic evaluation of the framework's effectiveness in improving Triage and overall system performance, from single-robot scenarios to more complex multi-robot environments.

1.5 Contributions of this thesis

Previous literature and investigations of how information about a robot's performance degradation can be used to facilitate human multi-robot teaming, is limited. Furthermore, Human Multi-Robot Teaming research that focuses on Variable Autonomy Multi-Robot System is even more limited. This thesis adds new contributions to these areas of research by:

- Proposing a framework for quantifying a robot's performance degradation during task execution. While a robot's performance degradation is a somewhat ill-defined phenomenon in general, there are specific aspects of a robot's behaviour during performance degradation that are symptomatic of imminent failure. Therefore, each of these symptoms serves as an indicator of robot performance degradation, which can be mathematically modelled and quantified. These different performance indicators are called 'robot vitals,' and a single scalar metric indicating the total performance degradation of the robot is called 'robot health.' This framework allows for real-time assessment and management of robot performance, providing a systematic approach to monitor and mitigate risks during mission execution.

- Proposing a set of 5 vitals - Rate of Change of Distance from navigational goal, Jerk along Axis of Motion, Robot Velocity, Rate of Change of Localisation Error and Laser Scanner Noise Variance; and a health metric that can generalise across several mobile robot morphologies, agnostic of the task or the navigation algorithms used.

- Presenting, to the best of my knowledge, the first systematic empirical experimental methodology to demonstrate that robot performance degradation can be quantified online during runtime, and show that the "Robot Vitals and Robot Health" framework adequately captures the effect of performance degradation on the robot. This experimental methodology can be used to evaluate the validity of new vitals, health metrics, or any subsequent frameworks to quantify robot performance degradation.

- Realising two adaptive automations for degree of autonomy switching, which choose the appropriate degree of autonomy by optimising the robot's health during the execution of tasks during runtime.

- Proposing an informed methodology and suggestions for designing AI agents that can trigger LoA switches to minimise the risk of robot failure. More specifically, this thesis proposes a Model Predictive Controller that can be integrated into a MI-LoA switching system. The assumptions made to design this controller, and the steps to improve it are clearly outlined. The main novelty of this controller is that the robot chooses the LoA that is estimated to have the highest robot health for the next tessellation in the robot's path plan.

- Designing a MI control system in which both the robot operator and the robots can trigger LoA switches, with specific improvements to minimise conflicts for control between the operator and the AI agent.

- Proposing the design and implementation of a 'robot health bar', a visual cue that informs the operator about a robot's runtime performance degradation to help them make LoA switching decisions that minimise the aggregate risk of robot failure.

- Providing evidence from a pilot study that shows how robot operators react and respond to a 'Robot Health Bar' while carrying out a remote robot navigation task using a Human Initiative Level of Autonomy switching system for a single robot. A comprehensive evaluation of the workload, operator trust, system transparency and the task performance is provided.

- Demonstrating that using the 'Robot Health Bar' can successfully change the operator's LoA switching style to a more safety-centric approach, which reduces the aggregate risk of robot failure during the task runtime, without increasing the operator cognitive workload. To the best of my knowledge, this study on the robot health bar is the first study to examine how adding a health bar to a user interface can affect the task performance and operator cognitive workload for a Variable Autonomy robot.

- Proposing a novel control interface forVariable Autonomy Multi-Robot System with multi-modal sensory cues for effective LoA switching, robot selection and data visualisation.
- Undertaking a principled, systematic study of how a 'robot health bar', and a 'low health sound alert', two sensory cues based on robot performance degradation, can impact the performance of an operator on a variable autonomy remote navigation task for 1, 2 and 3 robot teams.
- Providing a rigorous methodological foundation to carry out experiments with Variable Autonomy Multi-Robot System, using best practices from existing literature to evaluate the task performance, operator cognitive workload, trust, reliability, and usability of these systems.
- Providing an in depth analysis of the interaction of the human operator with an Human Initiative Level of Autonomy switching system with and without a robot health bar for single and MRS. This is the first such study to qualitatively and quantitatively report on metrics unique to the evaluation of Variable Autonomy Multi-Robot System systems.
- Demonstrating that using multi-modal sensory cues based on robot performance degradation as part of the interface to control Variable Autonomy Multi-Robot System can significantly improve operator response times, minimise aggregate risk of robot failure and promote higher usability and trust in the system.

1.6 Thesis Structure

The remainder of this thesis is organised as follows:

- Chapter 2 focuses on providing the context and background knowledge (e.g. the notion of levels of autonomy, robot performance degradation, fundamentals of interaction design and

human factors). It also reviews the relevant literature and examines the niche in the literature which the contributions of this thesis address.

- Chapter 3 reports on experiments with the Robot Vitals and Robot Health Framework. It demonstrates how the effect of runtime performance degradation on a robot during task execution can be quantified using the Robot Vitals and Robot Health Framework.

- Chapter 4 presents realisations of 3 different architectures of variable autonomy systems that use the Robot Vitals and Robot Health Framework to enable the robot to self-regulate its autonomy based on when it requires assistance from the robot.

- Chapter 5 presents the pilot study on using the robot health bar with a single robot and 2 robot Variable Autonomy systems. The experimental method, interaction design process and the results are presented elaborately. The final design informed by insights is presented for evaluating Variable Autonomy Multi-Robot System systems with one, two, and three robots, with and without multi-modal sensory cues based on the Robot Vitals and Robot Health Framework

- Chapter 6 summarises the thesis by discussing the broader impact of this research, future work and closing remarks. Lastly, important conclusions from this project are provided.

Chapter Two

Background

Triage for Variable Autonomy Multi-Robot Systems is a highly cross-disciplinary topic, necessitating insights, techniques, and tools from various fields. First, it requires an intuitive and robust state estimation framework to serve as the foundation for Triage. Subsequently, Variable Autonomy systems must be implemented using this framework. The interaction paradigm (i.e. the HRI) then needs to be designed both for the underlying human factors and for the robot AI capabilities to enable effective teaming. Finally, designing robust experiments (an open challenge) is crucial to evaluate the validity and effectiveness of these implementations. This chapter identifies key concepts from each relevant field, outlines gaps in the existing literature, and contextualises them.

2.1 State Estimation

Robust state forms the foundation of autonomous decision making in robots. As described in Barfoot (2024), the *state* of a robot is a set of quantities, such as position, orientation, and velocity, that, if known, fully describe that robot's motion over time. Barfoot (2024) further describes the problem of state estimation as "*Estimation is the problem of reconstructing the underlying state of a system given a sequence of measurements as well as a prior model of the system*". The

specific form of state estimation required can vary, depending on the application, and can be interoceptive (stimuli within the robot) or exteroceptive (stimuli from the environment). Interoceptive sensors, like accelerometers and gyroscopes, reflect the robot's actions in an environment, while exteroceptive sensors, like cameras and GPS, reflect the result of the robot's actions relative to the environment. Typically, a mix of both interoceptive and exteroceptive sensors are used to create robust state estimation methods. During state estimation, measurements from various sensors are first filtered to remove noise or artifacts, then combined to produce accurate metrics indicating the robot's state. Techniques such as Kalman filters or particle filters are often used to improve confidence in the state estimate by reducing fluctuations and noise. Alternatively, simpler methods like moving averages can also be employed to smooth out short-term variations and highlight underlying trends, ensuring reliable state estimation in dynamic environments.

Depending on the nature of the task or application, the criteria for state estimation vary, and therefore, the sensing and measuring techniques used vary accordingly. Low-level robot control focuses on estimating the robot's position and heading (Tzafestas, 2013). Navigation planning focuses on a relatively more exteroceptive estimation of the robot's position and representation of the environment for effective navigation (Tzafestas, 2018). In human-robot teams, where robots are capable of highly autonomous behaviour and LoA switching, the state estimation method must identify which robots are faulty, stuck, or require operator support (Dahiya et al., 2022; Rosenfeld et al., 2017). This is crucial for effective triage of multi-robot teams. Contrary to traditional fault detection methods, state estimation for Human-Robot Teaming (HRT) requires a higher level of abstraction, where not just sensor measurements but specific aspects of robot behaviours also have to be detected. The focus of this thesis is on creating a state estimate for the degree to which a robot requires assistance to continue executing without failing or malfunctioning.

2.1.1 Robot Performance Degradation

A variety of situations can cause impairments to a robot's ability to carry out its tasks. This includes failures, faults, malfunctions and errors. According to Honig and Oron-Gilad (2018), errors refer to system states which can lead to a failure; errors are caused by one or more faults at a system level. This paper adopts the definition of failure from Brooks, Begum, and Yanco (2016) as "a degraded state of ability which causes the behaviour or service being performed by the system to deviate from the ideal, normal, or correct functionality". For example, a robot may experience a navigation failure resulting from an error in obstacle recognition caused by poor illumination (fault). More specifically, for the mobile robot navigation context, Carlson and Murphy (2005) proposed a taxonomy for classifying Unmanned Ground Vehicle (UGV) physical failures based on their repairability and impact, which has been adapted in figure 2.1. According to Carlson and Murphy (2005), failures are categorised as field-repairable or non-field-repairable. Field-repairable failures can be fixed on-site, such as poor path planning, sensor adjustments or difficult obstacles, while non-field-repairable failures require extensive resources or specialised tools, like major hardware replacements. Additionally, failures are classified as terminal or non-terminal. Terminal failures can render a UGV non-operational, whereas non-terminal failures reduce efficiency without completely disabling the UGV. However in some cases, if the robot is left unattended, a robot with a

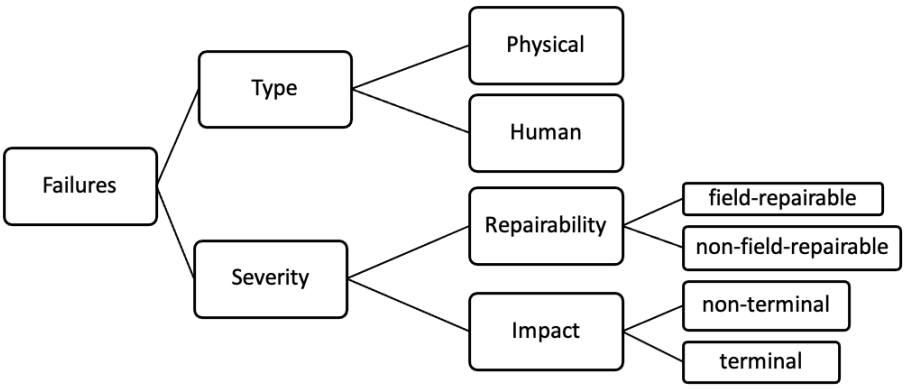


Figure 2.1: Taxonomy of mobile robot failures used in analysis adapted from Carlson and Murphy (2005) in Ramesh, Rustam Stolkin, and Chiou (2022)

non-terminal failure may turn into a terminal failure. In the context of this study, situations that cause any impairment in a robot's ability to carry out its tasks are referred to as performance degradation. In this thesis, performance degradation refers specifically to non-terminal, field-repairable failures of mobile UGVs.

2.1.2 Rule Base State Estimation

There are several approaches to detecting robot performance degradation using sensor measurements. One of the first studies on robotic fault detection was carried out by Visinsky, Cavallaro, and I. D. Walker (1994). This paper was a survey focused on detecting kinematic errors in pick-and-place robots by analysing the joint motor, sensors, power supply and control input signals for irregularities like spikes, surges, invalid commands or input biases. While this paper gives seminal insights into detecting performance degradation in robots concerning their behaviour, it does not directly provide useful metrics that can estimate the state of a mobile robot during remote navigation tasks. Another approach to detecting run-time faults in pick-and-place robots is to use a fuzzy logic rule base that observes output signals from the robot's sensors (Schneider and Frank, 1994). Fuzzy logic may be preferred over analytical models when the problem is ill-defined and cannot be broken down deterministically. An advantage of fuzzy logic over analytical models is the ability to encode expert knowledge into the system in semantic form rather than as a set of formulae. Unlike stochastic modelling, fuzzy logic based systems can be quickly deployed with minimal training and tuning required. This unique advantage of fuzzy inference systems enables it to be adapted even to state-of-the-art MI systems (Chiou, Hawes, and Rustam Stolkin, 2021). However, as the number of input variables increases, fuzzy systems are hard to scale as the number of rules required to define the state estimation rises exponentially. This is known as the 'curse of dimensionality'. Moreover, the over-reliance on human-defined rule bases has the risk of capturing the human's limitations and biases, potentially compromising the system's accuracy. My work aims to develop

a state estimation framework which can easily aggregate metrics based on both expert knowledge and analytical models of different aspects of the system without compromising the scalability.

2.1.3 Learning Methods for State Estimation

An alternative to using analytical approaches to state estimation is to use Machine Learning approaches to develop a mapping between robot operation data and the state estimate. For example, to measure the degradation of industrial machinery and components, machine learning can be used to predict Mean Time To Failures (MTTF) (Onur, Kaymakci, and Mercimek, 2020). A similar technique can be used to predict the MTTF for robots during task execution based on performance degradation. Triage can then be accomplished by prioritising robots based on their MTTF values. Soni, Prabakar, and Kim (2017) proposed to train individual classifiers to predict faults based on each sensor and actuator model. Using these models, a robot fault is predicted by individual voting by each classifier. In Yu et al. (2011), the authors use a support vector network to segregate erroneous behaviour by comparing the robot's trajectory to that calculated by its dead reckoning system. The prediction error is then put through a particle filter to remove noise, isolate instances of faulty behaviour and give fault proneness probabilities. To apply machine learning to robotics, it is important to have access to a database of labelled data. For example, the UCI machine learning repository (Dua and Graff, 2017) contains a dataset that maps robot grasping failures to the corresponding force and torque values collected at regular time intervals. Challenges in this approach include acquiring and processing training data, feature engineering (Domingos, 2012), labelling various types of faults and malfunctions, and training and deploying models on different robotic hardware (Soni, Prabakar, and Kim, 2017). Additionally, the main issue with applying this to robot triage in extreme environments is the lack of readily available data to detect all types of field-repairable, non-terminal failures for UGVs in remote settings.

One option is to generate data to enable AI agents to estimate robot states in a high-fidelity

digital twin through reinforcement learning. Hong et al. propose a reinforcement learning based approach to the design of a control switching policy (Hong et al., 2019). The operation of the multi-robot team is broken down into discrete states, such as navigating to unexplored regions, standing by, searching for victims, identifying victims, and performing manual control. Each state is further split depending on the number of unexplored regions, unvisited areas, obstacle collisions and victims identified. Using a task hierarchy and incentives for various state changes, a MAXQ (Dietterich, 2000) hierarchical reinforcement learning is used to create a policy that finds the optimal way to switch between tele-operation and autonomy. This reinforcement learning algorithm has access to data from each robot's navigation planner, thermal camera and infrared sensor. This algorithm was trained over 1800 (Doroodgar, Liu, and Nejat, 2014) unique scene layouts to achieve good generalisation for LoA switching in different adversities. While previous studies show that state estimation using reinforcement learning has proven effective, a significant drawback of reinforcement learning is its black-box nature, necessitating additional resources to make these models explainable. There is also a substantial overhead associated with data collection and training on new digital twins, especially when system modifications are required. It is crucial that these systems facilitate collaboration with human operators, and thus, they cannot be complete black boxes that exclude the operator from the loop. To encode expert knowledge, the reinforcement network must be modified with hierarchies, which complicates the deployment of these techniques. Unless zero-shot deployment techniques for real environments are developed with high transparency and explainability, these methods in isolation are unsuitable for robot triage. This is an avenue which is being actively pursued in recent literature (Mao, Dai, et al., 2024). Here, the aim is to develop a state estimation framework that can easily lend itself to fine-tuning through machine learning models while still retaining the intuitive explainability of rule-based systems and the ability to encode expert knowledge.

2.1.4 Competency Assessment

An alternate approach to state estimation is to create a vector of metrics representative of specific types of robot behaviour. This vector of metrics could encapsulate different aspects of the robot's behaviour, indicating its competence in executing the task. This could indicate a robot's behaviour at a higher level of abstraction or simply quantify the deviation of a robot from expected behaviour like dead-reckoning systems (Yu et al., 2011). One of the early papers to propose and implement a set of indicators to quantify performance degradation for VA mobile robots was carried out by (Valero et al., 2008). The authors proposed an agent-based LoA switching using four performance indicators - the robot's mean velocity after 4 and 15 seconds, displacement in the last 2 seconds and area explored. The trigger values of these indicators were tuned during experimentation. Chiou, Hawes, and Rustam Stolkin (2021) used the goal-directed motion error, i.e., the difference between the actual robot velocity and an expert planner's velocity, with a fuzzy rule base to detect performance degradation. Mendoza, Veloso, and Simmons (2012) estimated faulty behaviour in the 'CoBots' platform by calculating the robot's location independently using different sensors. To check for mismatches, the robot compares localisation information generated individually from wheel encoders, laser range finders and infrared scanners. The presence and magnitude of mismatches are used to detect faults. Additionally, the robot's knowledge base generates an estimated time to task completion for each task. When the task takes longer than the estimated time to complete, the robot flags a possible fault. In the absence of multiple sensors that can give location information, the robot's Light Detection and Ranging (LIDAR) scan data can be compared with the occupancy grid map to calculate a reliability estimate (Akail, Moralesl, and Murase, 2018). Ye (2007) uses the laser scanner data to generate a 'polar traversability index', quantifying the difficulty level associated with traversing any terrain the robot observes. The robot then incorporates this information into its navigation plan. This and other similar studies (Prágr and Faigl, 2019), show that traversability can serve as an indicator of robot navigation failure.

The main feature of these studies is that they use the knowledge about specific ways a robot can fail during task execution, model it, and use sensor data to detect such situations and quantify this effect. This enables the state estimation method to benefit from a combination of expert knowledge, heuristics from pilot studies, and learning models which can detect patterns in robot operation data. Common names for this include competency assessment (Conlon, Ahmed, and Szafir, 2024), or assumption alignment tracking (Cao et al., 2023). Furthermore, in the paper by Conlon, Ahmed, and Szafir (2024), they list references to other taxonomies referring to the same, such as introspection, algorithmic assurance, proficiency self-assessment and trust management systems. These papers provide a general mathematical framework to model and design such systems, which can be realised through numerical methods for aggregation and optimisation or learning.

Learning-based methods offer robustness, scalability, and the ability to process large datasets, while rule-based systems and analytical solutions provide intuitiveness and ease of implementation. Additionally, modelling expert knowledge and heuristics can serve as effective methods to estimate a robot's state for triage and servicing them during task execution. Therefore, it is important to find a common framework that has the merits of all these approaches and also allows for the creation of ensembles of different architectures (Dietterich et al., 2002), which individually indicate different aspects of a robot's state estimate. In this thesis work, the vector of performance indicators called *Robot Vitals* is proposed, each of which is symptomatic of different aspects of robot performance degradation. The vitals vector can be condensed into a single scalar metametric called robot health, providing an intuitive measure of robot performance degradation. This approach offers a transparent and easily understandable way to communicate robot status to operators, contrasting with the complexity of existing competency assessment architectures (Conlon, Ahmed, and Szafir, 2024).

2.2 Variable Autonomy

The various ways of implementing robotic systems with flexible levels of operator input are categorised under the term Variable Autonomy (VA) Robots (Chiou, Hawes, and Rustam Stolkin, 2021). VA Robots can also be defined as "an interaction strategy between human and robot agents in which the robot's Level of Autonomy varies during operation in response to changes in context" (Reinmund et al., 2024). This includes adaptive automations, shared control systems, and different implementations of LoA switching systems. It is important to distinguish between the notion of *Autonomy* and *Automation*. As described in Xu (2020), automation is the ability of a system to perform well-defined tasks and to produce deterministic results, relying on a fixed set of rules and algorithms without AI technologies. This term is very widely used to describe a variety of rule-based systems like industrial automation, control engineering systems, and adaptive automations. Xu (2020), further explains in references to the current era that the term *Autonomy* specifically refers to the ability of an Agent to perform specific tasks and take decisions independently. They can exhibit behaviours, learn or evolve to gain certain levels of human-like cognitive, self-executing, and adaptive abilities. They can also function successfully under some situations that are possibly not fully anticipated, and the results may not always be deterministic. The nature of autonomous systems and automations are mostly similar in the relationship with the requirement of human intervention, but differ in certain key technical aspects. However, contemporary literature on the subject largely uses the terms autonomy and automation interchangeably. In this section, various implementations of VA systems are analysed to identify areas with scope for improvement. This thesis then highlights how the Robot Vitals and Robot Health framework addresses these areas

2.2.1 Levels of Autonomy : Definition and Frameworks

A level of autonomy can be defined as the amount of control which a robot or any artificial agent has over its own decisions or actions. One of the earliest descriptions of the notion of Levels of Autonomy, and their stratification was developed by Sheridan, Verplank, and T. Brooks (1978), and further adapted Wandke* (2005) as listed below. These levels were organised based on - 1) Decision Making, 2) Selecting and Performing actions and 3) Information Sharing between the robot and the human.

1. The human does the whole job until it's handed over to the computer to implement.
2. The computer helps by determining options.
3. The computer suggests options, but the human doesn't have to follow them.
4. The computer selects an action, and the human may or may not perform it.
5. The computer selects an action and implements it if the human approves.
6. The computer selects an action, informs the human in time to stop it.
7. The computer does the whole job and necessarily tells the human what it did.
8. The computer does the whole job and tells the human what it did only if asked.
9. The computer does the whole job and tells the human what it did if it decides the human should be informed.
10. The computer does the whole job, decides if it should be done, and tells the human if it decides the human should be informed.

Subsequently, several other taxonomies have been developed building on Sheridan, Verplank, and T. Brooks (1978) to describe such VA systems for various contexts. Parasuraman,

Sheridan, and Wickens (2000) built on this to categorise Levels of Automation based on four stages: information acquisition, information analysis, decision making, and action implementation. Within each stage, there are 10 levels ranging from manual control to full automation. This framework allows for a nuanced understanding of how tasks can be distributed between humans and computers, with either entity capable of being in charge of task execution at different levels. Around the same time, In Kaber and Endsley (1997), Kaber and Endsley noted that over automated systems risk causing Out Of The Loop (OOTL) performance problems. This may include 1) over complacency or over-trusting the system, 2) Loss of Situational awareness, 3) Failure to observe system parameter changes as 4) Manual Control skill decay. They noted that (Sheridan, Verplank, and T. Brooks, 1978) was more applicable to teleoperation or control of a remote robot, and therefore built a more general framework that can be applied to a wider variety of human-computer control systems. To ensure that the operator is still engaged and involved in the operation of the system, Kaber and Endsley (1997) proposed a set of 10 LoAs, which can be optimally assigned to mitigate OOTL performance problems. The LoA presented in this paper were sorted based on four functions - Monitoring task execution, Generating a plan, Selecting and Implementation, which could be assigned to either agent in the human-computer control system. The authors also carried out extensive experiments to show that the use of LoAs were able to minimise the effects of OOTL performance problems, and improve the operator's situational awareness through the use of LoAs. The work of Beer, Fisk, and Rogers (2014) re-interprets these frameworks in the context of Human-Robot interaction, and provides a clear process to think about implementing these frameworks in practice. The authors of this paper propose a set of guidelines to determine the autonomy of the robot, the implications of the robot's autonomy on human variables, robot variables and interaction variables.

The SAE (Society of Automotive Engineers) defines six levels of driving automation for on-road vehicles (Committee, 2021), ranging from Level 0 (no automation) to Level 5 (full automation). Level 0 involves no automation, where the human driver is fully responsible for controlling

the vehicle. Level 1 provides driver assistance, such as adaptive cruise control. Level 2 includes partial automation, with systems that can control both steering and acceleration but still require human supervision. Level 3 offers conditional automation, where the vehicle can handle all driving tasks under certain conditions, but the driver must be ready to take over. Level 4 involves high automation, capable of performing all driving functions in specific scenarios without human intervention. Finally, Level 5 represents full automation, where the vehicle can drive autonomously under all conditions without any human input. This framework has been adopted extensively for self-driving applications and articulated for standardisation across countries, serving as a good foundation for safety and reliability testing. However, this framework is particularly focused on driving functions. It does not translate very well to mobile robots in extreme environments with terrain adversities, communication constraints, performance degrading factors, and unreliable sensor readings.

2.2.2 Variable Autonomy Frameworks for Multi-Robot Systems

The above-mentioned frameworks lack specific guidance on scaling VA to Multi-Robot System (MRS). This is a challenging problem, as any framework must address the entire spectrum of possibilities, from manually controlling each robot to achieving full autonomy for the entire MRS. Human Multi-Robot Teaming (HMRT) is fundamentally asymmetric, as the robot team always has more degrees of freedom than the human. Musić and Hirche (2017) delve into the problem of HMRT in detail and outlines several key considerations required to design the interaction. As described in Musić and Hirche (2017), when a single operator has to control a MRS, they can take either a supervisory or active role. The responsibilities of a human in a supervisory role are to select global and local behaviours and intervene only when necessary. Alternatively, humans in an active role provide control inputs to one or more robots continuously, as an active member of the team. LoAs are closely tied to this capability. To enable the operator to switch to an active

role from a supervisory role during mission runtime, the system needs to have autonomous control and (or) a task allocation system with redundancies to deal with such situations. As per Hussein, Ghignone, et al. (2018), the LoA of a swarm of robots can be measured using its level of human dependence. A swarm of robots are autonomous robots performing a collective action with local sensing and communication, lacking centralised control or access to global information. Mi and Yang (2013) extended the work of Parasuraman, Sheridan, and Wickens (2000) to swarms as a four layer structure (displayed in figure 1 of their paper). Rizk, Awad, and Tunstel (2019) proposes a very similar framework with 4 levels of automation to organise a task allocation to a heterogeneous MRS. In the first level, task execution by each robot is automated, but the operator is responsible for allocation. In the second level, task allocation or coalition (not both) is automated. In level 3, both coalition formation and task allocation are automated, and finally, level 4 automates the entire system.

P. Walker et al. (2013) describe their experiments on a swarm with two levels of autonomy - 1) Autonomous Dispersion and 2) Flocking to a user-defined go-to point. The authors studied how users varied the levels of autonomy for the swarm to explore a map. This study showed that operator input was needed to break the swarm into subgroups to disperse the swarm better and quickly find the targets. However, the swarm's performance declined when the operator took too much control. Applying swarming techniques to MRSs (Chamanbaz et al., 2017) may not always be possible. Each robot may sometimes have different tasks and require access to global information. Therefore, executing actions across the entire team may not always be possible. In such situations, the system requires an adaptive network to support effective HMRT. In Cummings (2004), the author proposes intra-vehicle LoAs. In systems with low network autonomy, each vehicle operates independently, following their original tasks without inter-vehicle communication and requiring human intervention for new tasks. As autonomy increases, vehicles begin to communicate with each other for basic functions like separation and threat avoidance, though they still rely on human input for new tasks. At higher LoAs, vehicles collaborate more fully, with the human operator

interacting only with a designated "lead" vehicle. Finally, in systems with full network autonomy, vehicles communicate and collaborate completely, adjusting their tasks based on predetermined algorithms without any human intervention. The lack of widespread consensus or standardisation in the stratifications for VA MRSs indicates that this is still an open problem.

2.2.3 Variable Autonomy Implementations

While the above mentioned frameworks offer valuable techniques and tools for selecting the appropriate Level of Autonomy (LoA) for various tasks, they do not specifically address the realisations and implementations of different Variable Autonomy (VA) systems. Additionally, they do not detail the specific types of information used by humans or computers to make decisions during task execution. Here, the existing literature is reviewed to identify common VA system implementations, their strengths and limitations.

Level of Autonomy Switching Systems

In early literature on human-robot systems, maintaining fixed responsibilities for the computer (i.e., the robot's control software) and operator throughout the task runtime was referred to as a fixed Level of Autonomy (LoA) or "static automation." In the context of automated aviation systems, Parasuraman, Bahri, et al. (1992) noted that static automation systems, which apply a constant level of system automation based on the workload demands of an average pilot, can penalise and underutilise the skills of a superior pilot. Kaber and Endsley (1997) also highlighted this issue, pointing out the potential for OOTL performance problems, a concern raised in several other studies. LoA switching is designed to address this particular issue. In LoA switching systems, humans collaborate with software/AI agents to control robots during task execution. These agents can be hosted on a computer local to the operator (Sheridan, Verplank, and T. Brooks, 1978), the cloud (Ermacora, Rosa, and Bona, 2015), or even onboard the robot (Chiou, Hawes, and Rustam Stolkin, 2021). De-

pending on the design of the overall human-robot system, the LoAs can be chosen to leverage the full spectrum from total manual control of the robot to Full Autonomy. This method of switching LoAs was originally referred to as *traded control* by Sheridan, Verplank, and T. Brooks (1978), but has taken a variety of terms like - adjustable autonomy (Mostafa, Ahmad, and Mustapha, 2019), adaptive autonomy (De Visser et al., 2008; Ermacora, Rosa, and Bona, 2015), dynamic autonomy (Bruemmer, Dudenhoeffer, and Marble, 2002) and sliding autonomy (Desai and H. A. Yanco, 2005) to name a few. LoA switching has also been studied in a variety of contexts other than robotics. Readers are referred to Reinmund et al. (2024) for a comprehensive review of such systems. As pointed out in Methnani et al. (2021), VA architectures differ mainly in terms of "which aspects of autonomy are adjusted, by whom (human, agent or both), how (continuous or discrete), why (pre-emptive or corrective), and when (design phase or operation)".

Initiative

Here the question of 'who' initiates changes in autonomy is addressed. For a single robot system, there are three types of initiatives possible - Human Initiative (HI), agent or Robot Initiative (RI), and Mixed Initiative (MI). Reinmund et al. (2024) define Human Initiative (HI) Systems as LoA switching implementations where the human operator has the sole capacity to change LoAs. This is the most common form of LoA switching systems. In these systems, a human operator interprets information generated by the robot during runtime through an operator control unit or GUI, and uses their judgement to decide the appropriate LoA for the robot. The design and implementation of HI-LoA systems have been subject to a few investigations in the literature. Researchers have explored the effect of several factors on the performance of HI-LoA systems, such as training, priming, interface cues, heavy cognitive workload (Gutman, Olatunji, and Edan, 2021), recommender systems, and sensor or communication faults. Their application has also been evaluated for different contexts like factories, warehouses, military operations, space, farming, and search and rescue. Common implementations of HI-LoA systems have at least two LoAs - Full Auton-

omy and Manual Control (Chiou, Rustam Stolkin, et al., 2016; Chiou, Hawes, and Rustam Stolkin, 2021). This approach allows humans to take control of the robots during complex, uncertain, or critical phases of the task and delegate control back to the autonomous system during routine or well-understood aspects of the task. This makes them particularly well-suited for Extreme Environments.

The primary merit of HI-LoA switching systems is that they are able to leverage human cognitive strengths, such as problem-solving, situational awareness, and decision-making for complex and unforeseen situations. The work of Chiou, Rustam Stolkin, et al. (2016) shows that on a remote robot navigation task, HI-LoA systems can outperform the robots with a single mode of operation, i.e. Autonomy or Manual Control alone. This is because operators can switch to autonomy when their performance is degraded and take control when the robot's performance is degraded. Studies on exploring operator preferences while using HI Systems (Gutman, Olatunji, and Edan, 2021) have also shown that users prefer to devolve control to autonomy when they face high workload but are willing to retain control when they are not facing high levels of workload. However, in Chiou, Hawes, and Rustam Stolkin (2021), evidence is presented to show that HI systems impose a higher workload on the operator in comparison to MI and Robot Initiative (RI) Systems, mainly because the extra cognitive overhead of observing the data from the robot's operation and determining the best LoA for it during runtime. Stressed operators are unable to operate the robot effectively for long periods and make sub-optimal decisions about servicing the robots, leading to a drop in overall task performance. Therefore, it is important to design the HI-LoA switching system in a manner that the cognitive workload of switching LoAs is within acceptable levels. This is a topic that is being studied extensively in existing literature, with different techniques available to reduce both the workload of both perceiving information and acting on it (Matthew S. Prewett et al., 2010b).In this thesis, the Robot Vitals and Robot Health Framework is used to evaluate if multi-modal cues on the operator's interface based on robot performance degradation is able to alleviate the operator's total cognitive workload, influence the operation style and improve task

performance.

In RI systems, the robot can change the LoA automatically without relying on the human operator (Chiou, Hawes, and Rustam Stolkin, 2021; Reinmund et al., 2024). Here, the robot uses some performance metrics informed by the state of the robot and the human-robot team to decide the appropriate LoA for any given situation. RI systems have been studied in existing literature, with most of the focus on proposing novel performance metrics or mitigating the effects of OOTL problems. The study by Chiou, Hawes, and Rustam Stolkin (2021) reported significantly quicker task completion and lesser operator cognitive workload with RI systems than HI systems. In the AI or Reinforcement Learning literature the LoA switching problem is treated as a problem of mapping a specific policy to the state of the environment. Therefore, these are also known as controller switching or policy switching systems (Comanici and Precup, 2010). In a paper by Wray, Pineda, and Zilberstein (2016), the problem of Robot Initiative Level of Autonomy switching is modelled as a transfer of control Markov decision process, specifically applicable to the semi-autonomous driving context. In their paper, the authors propose using Live states, i.e. a set of states that the system is safe to switch into at any given time. These states are calculated using an actor capability function, which maps the set of all states to the states the actor is allowed to transition into. To implement this, the probabilities of state transitions, executing LoA switching actions, and live states would individually have to be calculated for each situation, making this unsuitable for rapid deployment and triage-centric VA systems. While reinforcement learning techniques (Hong et al., 2019) could potentially learn these probabilities automatically, they present their own set of limitations (please refer to 2.1.3).

An alternative to the learning based approach is modelling policy switching as a myopic advice optimisation problem i.e., a human interacting with unreliable robot(s) needs to make actions that result in near-optimal performance of the human-robot team. These actions are a series of LoA switches that reduce the number of times robot(s) malfunction and impede task execution. In Rosenfeld et al. (2017), the authors provide a method to design an intelligent agent that will

advise the operator to take actions based on the reward for each action and the cost to execute it. This can be made into an RI system by allowing the agent to trigger LoA switches. The main advantage of advice optimisation over the MAXQ hierarchical reinforcement learning (Dietterich, 2000) and other learning based approaches is the ability to encode expert knowledge in the form of cost functions and heuristics. This removes the need to train to policies from scratch in a simulated environment. The disadvantage, however, is the added computational cost of solving an optimisation problem during run time. A potential solution to avoid real-time optimisation is using a lookup table. The robot could scan the environment for performance-degrading factors using sensors, identify similar situations in its lookup table, and switch to the corresponding LoA. Unlike the approach in Rosenfeld et al. (2017), this method is more tractable for real-time applications. This work explores whether the Robot Vitals and Robot Health Framework optimises the LoA in real-time and whether a robot-initiated LoA switching agent can be implemented through a lookup table based on past robot operation data

Jiang and Arkin (2015) define MI control as "A collaboration strategy for human-robot teams where humans and robots opportunistically seize (relinquish) initiative from (to) each other as a mission is being executed, where initiative is an element of the mission that can range from low-level motion control of the robot to high-level specification of mission goals, and the initiative is mixed only when each member is authorised to intervene and seize control of it.". In an MI-LoA switching system, both the robot or AI agent and the Human can trigger LoA switches. Results from (Chiou, Hawes, and Rustam Stolkin, 2021) show that these systems combine the strengths of both HI and RI systems. They result in improved performance in navigation tasks and lesser operator cognitive workload than HI tasks. However, they may lead to several *conflicts for control* between the operator and the agent. *Conflicts for control* refer to situations where the AI agent and the Human disagree about the appropriate LoA for the robot, resulting in rapid fluctuations in the robots LoA thereby resulting in sub-optimal task performance and a poor user experience. To reduce *conflicts for control*, operators need to know when to trust the AI agent's LoA regulation

choices and when to override them. One possible approach to minimising *conflicts for control* is to build a criticality hierarchy where robot initiative switches are made based on the operator's state and intention while using the robot (Panagopoulos et al., 2022). Alternatively, to reduce conflicts for control, the MI system can be appended with an additional module that negotiates the control transfer with the operator during runtime (Rothfuß et al., 2022). An alternative to designing hierarchical systems or additional modules is to design the MI system such that the AI system can only trigger autonomy to manually control LoA switches. This would potentially reduce the likelihood of *conflicts for control* and promote ease of use. In this study, an MI-LoA switching system using the Robot Vitals and Robot Health Framework is designed and its task performance and ease of use is compared to the state of the art MI system EMICS, described in Chiou, Hawes, and Rustam Stolkin (2021). This study is the first to implement an MI-LoA system that employs a model predictive controller to estimate the risk of robot failure and adjusts the robot's LoA accordingly to mitigate this risk.

Degree of Autonomy Regulation

(Backes, 1994) defines shared control as "the merging of teleoperation and autonomous control in real time during task execution". Shared control systems blend the input from an autonomous controller, and manual control inputs by an operator typically through joystick command. Active domains of research for remote robot navigation using shared control include finding the best approaches to realise a system under different constraints like communication issues (Storms, K. Chen, and Tilbury, 2017), autonomous obstacle avoidance (Pappas et al., 2020), managing asymmetrical access to information, haptic feedback and intent recognition (Abbink et al., 2018). The work by (Kucukyilmaz and Demiris, 2018), proposes a shared control setup where a regression model is trained to personalise the wheelchair controller to the operator, thereby reducing the effort required to control it and minimising jerkiness. The work by (Janabi-Sharifi and Hassanzadeh, 2010) identified key design considerations and their effects on shared control such as the relation-

ship between camera placement, time delay and haptic feedback on the performance of the robot in environments filled with obstacles. The major advantage of shared control is the ability of the operator to continuously stay in the loop and provide control inputs to the robot. This minimises the likelihood of OOTL problems and allows the robot to benefit from constant operator support, making it more responsive to different situations. However, due to continuously staying in the loop, shared control can impose cognitive workload on the operator, leading to a gradual decline in performance.

One way to address this is to augment shared control robots with the benefits of LoA switching. This can be achieved by allowing the ratio of input blending to be varied during runtime (Abbink et al., 2018). This is known as the Degree of Autonomy (DoA) Regulation. In DoA regulation, the ratio of operator input (denoted as α) can be varied to result in implementations of shared control across the continuum of Manual Control to Full Autonomy. This solves the "loss of information" (C. A. Miller, 2018) that arises from the discretisation of the continuum. Additionally, varying the values of α allows the incorporation of multiple criteria of optimisation simultaneously. That is, using a high-level model of the system, depending on different issues during runtime like communication delays, high workload, the value of α can be varied. In control systems literature, systems that can automatically regulate the DoA of a system are known as adaptive automation. Such systems can automatically adjust the operator support for the robot depending on measures of task difficulty (Cosenzo et al., 2010), task-load (Visser and Parasuraman, 2011), workload and skill-level (Nittala et al., 2018), competence (Basich et al., 2020) or performance (Calhoun et al., 2021). Therefore, the design of such systems is largely contingent on finding the appropriate system state estimation measure. While several studies have used realised adaptive automation-based metrics or formal methods to quantify performance degradation, to my knowledge, they do not provide a single, empirically validated metric to quantify the total performance degradation. Therefore, in this thesis 2 different adaptive automation systems based on the Robot Vitals and Robot Health framework are realised to estimate a robot's runtime performance

degradation over a finite time horizon.

Variable Autonomy Multi-Robot systems

Scaling VA to MRSs is a promising line of research with many open problems. Early research was carried out by Valero-Gomez, De La Puente, and Hernando (2011). In this paper, participants interact with VA MRS using a computer, mouse and a joystick. The authors compare two different approaches to realise VA for an MRS - 1) The operator selects a fixed LoA for each robot at the start, and tasks are executed strictly according to this preset level without the ability to change autonomy levels during the task execution, 2) The operator can dynamically adjust the autonomy levels and reconfigure tasks in real-time, allowing for intervention at any point to manage errors and optimise performance. Their study showed that a flexible approach that allowed real-time adjustments (i.e., option 2) was generally preferred because it improved performance and scalability. However, the authors also noted that if the autonomy adjustment is not designed carefully, it could overload the operator's cognitive abilities more than the static approach. In this study, operators make decisions about switching between LoAs based on their judgement by constantly monitoring the robot's operation. However, they are not provided with specific aids for enhancing their judgement or reducing the workload of managing the MRS. This paper also does not delve into methods for robot triage or assessing the likelihood of failure and criticality for overall mission success. The work by Dahiya et al. (2022) offers valuable insights into developing an AI agent capable of automatically scheduling triage. However, it presents a high-level mathematical model evaluated in a low fidelity simulation environment and does not elaborate on practical implementation details. Similarly, the approach proposed by Rosenfeld et al. (2017) suggests a scalable method for designing an AI agent to assist operators in triage operations. Nevertheless, this paper does not empirically evaluate the benefits of the agent compared to the state-of-the-art and requires extensive state space modeling to recreate (please see 2.2.3).

This study focuses on scenarios where robots in an MRS operate independently without forming spatial coalitions or groups. Therefore, from the work of Rizk, Awad, and Tunstel (2019) and Musić and Hirche (2017), it can be inferred that any VA MRS requires some autonomy for task allocation and execution while allowing the operator to both seamlessly intervene and alter the task allocation and assist any robot in the MRS that requires assistance with continuing task execution. For a distributed robot navigation task, for example, task allocation for each robot can be accomplished using an automatic waypoint planner. In that case, the operator would have to monitor task execution and assist any robot that faces problems with navigation by first selecting that robot and servicing it using any of the existing VA frameworks. This thesis, explores how cues based on robot performance degradation can be integrated into the overall HRI paradigm to improve task performance and alleviate operator cognitive workload.

2.3 Human Factors Evaluation and Interaction Interface Design

The preceding sections have explored the extensive research in VA and HMRT, focusing primarily on technical implementations. However, to fully understand and optimise these systems, the critical human factors involved in designing effective HRI paradigms needs to be considered. Human factors, as a discipline, is defined as the "scientific study concerned with understanding interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and other methods to design in order to optimise human well-being and overall system performance" (Czaja and Nair, 2012). In the context of human-robot interaction (Michael A. Goodrich and Schultz, 2007), this field focuses on identifying and optimising the interplay between humans and robots to enhance task performance, safety, user engagement, and satisfaction. While an exhaustive study of all human factors is beyond the scope of this thesis, this section briefly highlights

key considerations particularly relevant to triage for a Variable Autonomy Multi-Robot System (VA MRS). Here, the focus is on operator cognitive workload, trust, transparency, and situational awareness. These factors are crucial in shaping the overall effectiveness and user experience of VA MRS, complementing the technical aspects discussed earlier.

2.3.1 Workload

Parts of this section contain previously published text from the paper by Ramesh, Englund, et al. (2023). Operator cognitive workload refers to the mental effort required to perform a task or set of tasks. A well-designed Variable Autonomy (VA) system should strive to maintain operator cognitive workload within a 'sweet spot '— an optimal range where the workload is neither too high to cause stress and fatigue nor too low to lead to complacency or out-of-loop performance problems (Ashcraft, Michael A Goodrich, and Crandall, 2019). When workload is too high, it can degrade task performance due to stress, fatigue, and varying levels of trust in the system (Matthew S. Prewett et al., 2010b; Mizuno et al., 2011; Agrawal and H. Yanco, 2018). Conversely, low workload can result in complacency and over-trusting the system, as observed by Evans and Fendley (2017), and Hussein and Abbass (2018). Therefore, achieving this 'sweet spot 'is critical for effective human-robot teaming, ensuring operators remain engaged without being overburdened.

The workload sweet spot is influenced by how tasks are allocated and managed between humans and robots. For instance, operators' LoA switches should be informed by a clear understanding of the robot's capabilities and limitations rather than driven by trust issues. Evidence from the work carried out by Chiou, Hawes, and Rustam Stolkin (2021) shows that MI systems induce less operator cognitive workload than HI systems, reinforcing the importance of balanced human-robot interactions.

Drawing from Wickens (2002) multiple resource theory, cognitive workload can be divided

into three components: perception, cognition, and responding. While perception and cognition utilize the same mental resources, responding draws upon a distinct set of cognitive resources. This functional separation suggests that some secondary tasks (e.g., verbal responses) can be designed to have minimal interference with primary perceptual or cognitive processes. Extending this theory to Human-Robot Interaction, operators may be able to receive information about a robot's performance degradation during runtime without significant disruption to their decision-making about triage or LoA adjustments. Introducing a secondary task in VA experiments can help ensure that cognitive workload does not drop too low, preventing operators from becoming complacent or disengaged. However, the combination of operating a VA system while performing a secondary task may possible induce excessive cognitive workload and hinder effective human-robot teaming. Therefore, the interaction has to be designed carefully such that the balance of these factors optimise both operator engagement and system performance

Workload management becomes increasingly complex in MRS. As noted by Kolling et al. (2015), the cognitive complexity of MRS can rise super-linearly ($O(> n)$) when inter-robot dependencies create cascading demands. If this complexity exceeds an operator's cognitive capacity, it leads to high mental workload and stress. Strategies to mitigate this include enabling group-level actions, increasing neglect tolerance, and implementing hierarchical control paradigms (Hou et al., 2010; Musić and Hirche, 2017; C. Y. Wong et al., 2010). These approaches simplify control and help manage workload, preventing cognitive overload while maintaining operator engagement to avoid out-of-loop errors (Endsley, 2017).

For the experiments with a VA robot in this thesis, a secondary task is used to induce additional cognitive workload for the operator. This approach allows us to test whether incorporating real-time information about the robot's performance degradation into the operator control interface can 1) reduce cognitive workload and 2) enhance overall task performance for VA robots and VA MRS. Similar to the work of Chiou, Hawes, and Rustam Stolkin (2021), the NASA-TLX questionnaire is used for post-hoc workload evaluation after each trial.

2.3.2 Trust and Reliability

Trust is a fundamental aspect of VA systems, particularly in Human-Robot Teaming. Trust in robots is defined as 'the attitude that an agent will help achieve an individual's goals in a situation characterised by uncertainty and vulnerability' (Lee and See, 2004), and reliability is 'the quality of being trustworthy' (Lee and See, 2004). Trust and Reliability are often very closely linked and highly correlated. In VA systems, trust influences how operators interact with the system (Hussein and Abbass, 2018), including their willingness to rely on or override the decisions of the system, automated agents, or advisory functions (Jian, Bisantz, and Drury, 2000; J. Y. Chen and Barnes, 2014). The introduction of AI agents in VA systems adds complexity to the trust dynamic of the human-robot team. Robot operators tend to be more sensitive to errors made by AI than those made by humans (Jian, Bisantz, and Drury, 2000). This sensitivity can lead to rapid fluctuations in trust levels, especially in robot initiative and mixed-initiative systems where the AI can initiate LoA switches.

Interface design plays a crucial role in building and maintaining trust. An effective interface should provide clear, real-time information about the system's decision-making process, current LoA, and overall performance. It should also facilitate easy operator intervention when necessary. For example, implementing a management-by-consent approach, where the AI proposes actions for operator approval, has shown promise in enhancing trust and workload management (H. A. Ruff, Narayanan, and Draper, 2002). Systems where information about the robot is presented over the graphic/visual modalities are trusted more than the audio and textual modalities (Sanders et al., 2014). Additionally, standardised training paradigms can help establish baseline trust levels, and multiple interaction trials have been shown to increase both skill acquisition and trust in MI systems (Chiou, Talha, and Rustam Stolkin, 2019). In the context of multi-robot systems (MRS), trust becomes even more critical due to increased complexity. J. Y. Chen and Barnes (2014) from multiple sources that significant differences exist in people's perceived reliability of human aids

and automated aids. As the number of robots increases, proper workload management becomes essential to prevent operator stress and maintain trust (J. Y. Chen and Barnes, 2014).

Measuring trust in VA systems often involves observing patterns of system use. Two key indicators are misuse (over-reliance on the AI) and disuse (under-reliance on the AI) of AI-initiated LoA switches. Researchers can design experiments with occasionally unreliable AI to study how trust levels change over time. Alternatively, in situations where misuse and disuse cannot be clearly measured, another option is to use a trust scale developed by HRI experiments (Yagoda and Gillan, 2012). In conclusion, trust in VA systems, particularly those involving multiple robots, involves a complex interplay of system design, interface usability, and operator perception. Ongoing research continues to explore ways to optimise this balance, aiming to create systems that are both highly capable and trustworthy from the operator's perspective. In this study, the trust scale by (Yagoda and Gillan, 2012) is adapted to evaluate how operators perceive the trustworthiness and reliability of information about robot performance degradation displayed on the user interface.

2.3.3 Usability and Transparency

In simple terms, transparency in human-robot interaction helps operators understand what the system is doing and why. Studies have shown that when HRI is transparently designed, operators can accurately interpret the robot's capabilities, goals, and progress (J. Y. Chen and Barnes, 2014). This transparency leads to effective calibration of trust (Vitale et al., 2018), even with increasing levels of autonomy (J. Y. Chen and Barnes, 2014). Similarly, the aim of transparency as suggested by Theodorou, Wortham, and Bryson (2017) is "accurate interpretations of the agent's capabilities, goals, and current progress towards its goals". Transparent design has the potential to reduce the workload and promote the trust of the user in the system as well. Studies such as Dawson et al. (2015) and J. Y. Chen, Lakhmani, et al. (2018) show that transparency in algorithm functions can prevent mistrust and reduce workload by clarifying expectations. It is very closely tied

to the usability of a systems (Patel, Sonar, and Pinciroli, 2022). Usability and user experience are used interchangeably in existing literature. In Pohlt et al. (2018), the authors study the effect of gestures and touch input modalities in a smart manufacturing environment to assess user experience. User experience was evaluated through the Usability Metric for User Experience (UMUX) questionnaire (Finstad, 2010), measuring the effects of individual differences, performance, and previous knowledge. Different LoAs were implemented by varying the robots' autonomy in task execution, and it was found that these factors significantly impacted user experience, with intuitive interactions reducing workload and training periods.

Capelli, Secchi, and Sabattini (2019) demonstrated that using colour coding for robots on the interface in a Multi-Robot System (MRS) improved transparency. Operators found it easier to understand the robots' movements, which significantly influenced their navigation paths to the goal position. In another study, Etzi et al. (2019) evaluated HRI in collaborative manufacturing using a virtual reality headset. They found that lower system transparency led to longer response times from operators. For the pilot study on the impact of robot health-based visual cues on task performance, system transparency is measured using a questionnaire. This questionnaire evaluates if operators understand when and why to switch levels of autonomy based on robot performance degradation. Next, in the study with a VA MRS , response times are measured to assess transparency. Additionally, the UMUX questionnaire (Finstad, 2010) is adapted to evaluate User Experience.

2.3.4 Interaction Design

Effective interaction and interface design for robotic systems, particularly VA systems, requires careful consideration of several key factors. The complexity of these considerations increases as we move from single robot to MRS. For single robot interfaces, the primary focus is reducing operator stress through thoughtful design choices (Steinfeld, 2004). System responsiveness is crucial,

with users typically expecting command execution within 0.1 seconds (Murphy and Tadokoro, 2019). To optimise the interface, designers should minimise the number of clicks required for actions, group and customise tools without cluttering the screen, and avoid excessive visual overlays (Murphy and Tadokoro, 2019; Lee and See, 2004).

In VA robots, incorporating explainable AI guidelines becomes essential to enhance system transparency, facilitating operator understanding and trust in the system's decisions (Gunning, 2017). A dual-view approach can be beneficial in these systems: egocentric views are more appropriate for teleoperation scenarios, allowing for precise robot control, while allocentric views are well-suited for higher autonomy levels, providing a broader environmental overview (Valero-Gomez, De La Puente, and Hernando, 2011). The complexity further increases when designing for MRS. Interface design must account for managing larger robot teams efficiently, which can be achieved by implementing group-level action capabilities and designing for high neglect tolerance. MI control systems have shown promise in allowing agents to progress on tasks with reduced human input (Hardin and Michael A Goodrich, 2009). However, it is essential to carefully balance autonomy levels to prevent operator disengagement and out-of-loop errors.

Across all these systems, from single robot to MRS, the use of colour-coded visual cues and audio signals plays a crucial role in effective interface design. Colour-coding schemes can significantly enhance operator understanding of robot status and potential malfunctions (Steinfeld, 2004). For instance, using red to indicate critical issues, yellow for potential problems, and green for normal operations can quickly convey important information. These visual cues can be complemented by audio signals to alert operators to changes in robot status or environmental conditions. In MRS interfaces, colour-coded cues become even more critical as they allow operators to quickly assess the status of multiple robots simultaneously. They can indicate not only individual robot status but also the progress of group-level tasks or overall system state. Audio cues can serve as attention-grabbing alerts for critical events requiring immediate operator intervention. Designers must be mindful of the challenges associated with viewing complex 3D information on 2D screens, partic-

ularly in balancing contextual environmental information with robot-specific situational awareness (Hou et al., 2010). The RoboLeader project provides further insights into scalable interface design for mixed-initiative systems (J. Y. Chen, Barnes, et al., 2011).

Video games serve as inspiration for presenting real-time information transparently while maintaining optimal operator cognitive workload. Game players report receptiveness to visual cues when clear and consistent (Llanos and Jørgensen, 2011). These cues, often colour-coded user interface elements overlaid during game play (e.g., the 'Health Bar'), are used by remote players in online games to attract teammates' attention. This suggests that information about robot performance degradation can be efficiently integrated into the user interface through a 'Robot Health Bar', promoting usability and minimising additional perceptual effort. Such an approach aligns with quantifying performance degradation and could potentially enhance operator trust and system effectiveness in VA and MRS contexts. In this thesis, based on best practices from existing literature, and inspiration from video games, a novel triage-centric interface for Human Interaction with Variable Autonomy Multi-Robot System is contributed.

2.4 Experimental Evaluation and Metrics

Evaluating VA systems is an open, challenging problem due to the complexity created by the environment, the humans, and the robot acting as a team. Hence, for VA experiments on a single robot setup, the experimental procedure followed in experiments on other VA systems is used (Chiou, Rustam Stolkin, et al., 2016; Chiou, Hawes, and Rustam Stolkin, 2021; Panagopoulos et al., 2022; Rothfuß et al., 2022). In these studies, the operator is first provided training on basic robot navigation. Then, the operator is introduced to the Variable Autonomy system, explained how to use it, and given time to get comfortable with the use of these systems. Then, the operator carries out a simple remote mobile navigation task with a Clearpath husky robot. While the studies

by (Chiou, Rustam Stolkin, et al., 2016; Chiou, Hawes, and Rustam Stolkin, 2021; Panagopoulos et al., 2022; Rothfuß et al., 2022) use only LIDAR noise and obstacles to degrade the robot's performance, experiments in this thesis use LIDAR noise, uneven terrain, high friction terrain and obstacles. This is done to have a higher degree of similarity to real robot missions in extreme environments. Additionally, to recreate high levels of operator cognitive workload during task execution, state-of-the-art literature uses a 3D mental rotation task, which the operator has to carry out simultaneously at a specific time during the execution of the primary task.

In this thesis, a continuous secondary task is employed to elevate the operator cognitive workload throughout the entire task execution period. This methodological choice was made to investigate how operators interact with Variable Autonomy implementations under sustained high levels of workload rather than examining their responses to momentary workload surges. The aim here is to assess the robustness and effectiveness of the VA systems under challenging operational conditions that more closely resemble real-world scenarios by maintaining a consistently elevated cognitive load. Similar to the state-of-the-art literature, the key metrics used to evaluate the performance of the VA systems are primary task completion, workload assessments through the NASA-TLX questionnaires and performance on the secondary task. The aggregate risk of robot failure is also measured throughout the task to ensure safe robot operation during runtime. As noted in subsection 2.2.3, a higher amount of conflicts in MI and robo initiative systems can affect user experience negatively. Therefore, to evaluate if the Mixed Initiative realisation can provide a better user experience than state-of-the-art, the total conflicts of control from each implementation is also measured.

While there are a few studies that address the implementation of Variable Autonomy multi-robot systems (Hong et al., 2019; Valero-Gomez, De La Puente, and Hernando, 2011), they do not carry out a rigorous evaluation of the impact of providing information about the robot's performance degradation to the operator through cues integrated into the interface. Additionally, these studies do not carry out a principled study of VA systems with systematically introduced perfor-

mance degradation factors to degrade robot performance during runtime. As described in Hussein, Ghignone, et al. (2018), this study also evaluates the interaction effort, the operator's attention efficiency, and the task performance metrics. Additionally, the trust and usability of the VA MRS systems are evaluated using a questionnaire after each trial and qualitative feedback. To the best of my knowledge, this study represents the first of its kind, a comprehensive evaluation of the impact of multi-modal sensory cues based on a robot's performance degradation on VA MRS systems.

2.5 Conclusion

This section highlights several key research areas that collectively form the focus of this thesis. A review of existing literature reveals a significant gap: the absence of a simple, intuitive approach to quantify performance degradation that can serve as an effective state estimation tool for designing triage systems in Variable Autonomy (VA) robots and Variable Autonomy Multi-Robot System (VA MRS). To address this gap, the novel Robot Vitals and Robot Health Framework is proposed to quantify performance degradation, serving as a foundation for several VA implementations. The research directions in this thesis encompass three main areas: 1) investigating how the "Robot Vitals and Robot Health" framework can improve the usability of Human Initiative Level of Autonomy (HI-LoA) and Mixed Initiative Level of Autonomy (MI-LoA) systems, providing operators with more intuitive and actionable information about robot performance; 2) exploring how this framework can be utilised to realise adaptive automation systems that dynamically adjust the Degree of Autonomy based on predicted robot performance degradation; and 3) examining how multi-modal sensory cues, based on robot performance degradation, can facilitate triage in both single-robot and multi-robot VA systems while keeping the overall operator workload within the 'sweet spot ', promoting trust and system usability. Through these research directions, my work contributes to the development more effective, user-friendly, and scalable VA systems for single and multi-robot applications.

Chapter Three

Estimating performance degradation: the Robot Vitals and Robot Health Framework

In this chapter, the "Robot Vitals and Robot Health" framework is proposed to quantify runtime performance degradation in robots. The chapter is based on two papers - 1) Late Breaking Report at the HRI 2021 Conference (Ramesh, Chiou, and Rustam Stolkin, 2021), 2) the Robotics and Automation Letter presented at the IROS 2021 Conference (Ramesh, Rustam Stolkin, and Chiou, 2022). As highlighted in the previous chapters, there is a need to develop a framework that can assist operators with robot triage. Such a framework should be able to identify instances when a robot is facing performance degradation during autonomous task execution. Additionally, in scenarios where multiple robots are simultaneously experiencing performance degradation, a triage framework should help prioritise attending to the robots based on the severity of their performance degradation. Such a framework should also have some crucial properties:

1. It needs to facilitate the creation of a shared mental model for the Human Multi-Robot Teaming (HMRT).

2. It should be intuitive, easily explainable, and transparent.

3. It should work irrespective of the task given to the robot(s) and be robust to on-the-fly changes in task or mission goals.

4. It should be demonstrably correlated to one or many forms of performance degradation that the robot can experience

A good place to look for inspiration for such frameworks is patient triage systems in hospitals. Triage systems are used worldwide to allocate doctors and nurses, manage hospital resources and decide the order of treatment of patients based on the severity of their illnesses. Such systems use a standardised set of physiological parameters popularly known as a human's vital signs. Using a flowchart of different threshold values of each vital sign, a patient is assigned a severity score for their illness. One example of such a triaging system is the National Early Warning Score 2 (NEWS 2)(Smith et al., 2019) used across the United Kingdom. The flowchart given in figure 3.1 is used across the United Kingdom to detect and respond to clinical deterioration of patients. The simplicity, elegance, and potential for standardisation of this framework inspires the adoption of a similar approach for robot triage. The first step to building such a triage framework is demonstrating that runtime performance degradation can be detected, and its severity can be quantified in a similar fashion, agnostic of the robot's morphology and task.

As described in Chapter 2, 1) For the purpose of this thesis, the term "performance degradation" refers to any impairment in the capability of a robot to carry out its tasks, 2) The focus is specifically on field repairable non-terminal failures, i.e., failures that can be mitigated during task runtime through operator assistance. In addition to the proposed framework, experiments both in simulation and on a real mobile robot are conducted, demonstrating that the "Robot Vitals and Robot Health" framework successfully quantifies performance degradation. In these experiments, the robot carries out a navigation task, while being subject to several performance degradation. Through these experiments, it is shown that the "Robot Vitals and Robot Health" framework can be used to estimate robot performance degradation online and serve as a basis for triage. Addition-

Physiological parameter	Score 3	2	1	0	1	2	3
Respiration rate (per minute)	≤8		9–11	12–20		21–24	≥25
SpO_2 Scale 1 (%)	≤91	92–93	94–95	≥96			
SpO_2 Scale 2 (%)	≤83	84–85	86–87	88–92 ≥93 on air	93–94 on oxygen	95–96 on oxygen	≥97 on oxygen
Air or oxygen?		Oxygen		Air			
Systolic blood pressure (mmHg)	≤90	91–100	101–110	111–219			≥220
Pulse (per minute)	≤40		41–50	51–90	91–110	111–130	≥131
Consciousness				Alert			CVPU
Temperature (°C)	≤35.0		35.1–36.0	36.1–38.0	38.1–39.0	≥39.1	

Figure 3.1: NEWS 2 Severity score flowchart

ally, evidence that this framework can be used to develop Variable autonomy systems is provided. This experiment has highlighted the complex, fuzzy and highly variable nature of robot performance degradation and the difficulties of quantifying it in a systematic and principled manner. Lastly, a few lessons learnt while designing this experiment are listed.

3.1 Robot Vitals

For any task that uses robots, the task's performance metrics are post-hoc metrics that evaluate how well the task or the mission is being executed. However it is important to use metrics that describe how well a robot is performing on the task. Consider the example of a game of soccer between two teams as an analogy. A team's performance can be determined using metrics like

- the number of goals scored, ball possession, number of yellow and red cards issued, shots on target, and fouls. However, some players may feel unwell or unfit during the game, requiring medical aid or substitution. This can be monitored by measuring each player's pulse, respiration rate, body temperature and gait speed (Studenski et al., 2003) during the game. Analogously, robot performance degradation can be measured using specialised metrics that indicate the status of each robot easily and intuitively.

Robot vitals are online, real-time metrics that indicate performance degradation faced by a robot at any time during its mission. Each vital represents a specific aspect of robot behaviour. While a single vital may or may not give definitive information that a robot is failing, trends of a set of vitals can provide a robust indicator of adverse conditions. They may also help diagnose the nature of an adverse situation. Akin to a human's vital signs, trends in the robot vitals encode symptoms of robot behaviour during performance degradation. Ideally, robot vitals should account for all potential performance degrading factors. However, the robot vitals may practically be chosen to provide rich and robust information specific to the task or context of application.

Any robots' actions should reflect its intent (e.g. its task or mission objective). Any disruption to the robot's action, which causes the robot to deviate from the intended behaviour, can, therefore, be referred to as performance degradation. The robot vitals are a set of indicators that can be used to quantify how much a robot's actions are deviating from its intent. Beyond a certain threshold, this can require remediation behaviours or interventions. There are several types of robot morphologies, control algorithms, sensors, and tasks. Therefore, finding a systematic and generic approach to creating robot vitals for any situation is not easy. Here, a set of vitals that can be generalised in various situations are proposed.

As one example realisation, five vitals useful for experiments with a mobile robot in the Extreme Robotics laboratory's mock-up disaster scenario are presented, and the rationale for deriving them. Four vitals capture motion-related performance degradation. A fifth captures localisation-

related degradation. Probability distributions and transforms for each vital are determined empirically, based on preliminary experiments, observations of multiple robot platforms, and previous work on variable autonomy robots (Chiou, Hawes, Rustam Stolkin, et al., 2015; Chiou, Rustam Stolkin, et al., 2016; Chiou, Hawes, and Rustam Stolkin, 2021). This is not meant to be an exhaustive list. Depending on the robotic platform, task and the environment, a variety of metrics can be chosen as robot vitals by adhering to the above-mentioned principles.

A robot is defined as "suffering" if it is experiencing high performance degradation. Different aspects of performance degradation are related to their corresponding vitals, by defining a probability of robot "suffering" given each vital. Hence, as the performance degradation indicated by a vital increases, the probability of suffering given the vital should increase. Each vital is computed by processing real-time data sampled from the robot. Various techniques such as event detection, pattern matching and thresholding can be applied to emphasise significant features in the trend of each vital. The probability of suffering for each vital allows us to relate robot failure with specific aspects of robot behaviour, as given by the vital. This function enables us to incorporate a wide range of insights, such as information collected through expert knowledge, Monte Carlo methods, previous experiments, and machine learning models. For tasks where operators in the field have developed heuristics over years of experience with tasks and specific robot hardware or sensors, the probability of suffering gives them the opportunity to encode this expert knowledge as a probability distribution, rule-based system or even a fuzzy logic-based system. This value can then be converted into a probability between 0-1 for each vital, representing the probability of the robot suffering given that vital.

3.1.1 Rate of Change of Distance from navigational goal ($\dot{d}_g$)

This vital is used to indicate situations in which performance degrading factors cause a robot to not move towards its navigational goal. Such situations can be detected by observing the Rate of

Change (RoC) of distance from a robot's current position to its current navigational goal ($\dot{d}_g$). The odometry position estimate obtained after Extended Kalman Filter (EKF) sensor fusion is used as the robot's current position, and the goal is given by an operator or the navigation algorithm. The d_g is calculated using Euclidean distance to make minimum assumptions about the task, the algorithm used, and whether the map is known before the task. However, for more sophisticated applications d_g can be calculated using the distance remaining along a non-linear path. During little to no performance degradation (i.e., ideal behaviour), the robot moves towards the goal with uniform velocity. This results in a constant value $\dot{d}_g < 0$, barring few fluctuations. A $\dot{d}_g \approx 0$ has very little similarity to ideal behaviour and indicates that a robot is unable to move. Lastly, $\dot{d}_g > 0$ is dissimilar to ideal behaviour, and indicates that performance degradation has resulted in the robot taking a sub-optimal path or moving away from the goal.

To calculate the magnitude of similarity (d_{event}), $\dot{d}_g$ values over multiple time steps are observed, and compared to ideal behaviour using a convolutional matched filter (Turin, 1995). Preliminary experiments showed that $d_{event} > 0.3$ indicates a high degree of similarity i.e., the robot is facing little to no performance degradation. A $d_{event} < -0.3$ indicates dissimilarity and suggests that the robot is unable to move or is moving away from the goal. The probability of suffering is calculated as a function of d_{event} such that its value is high if $d_{event} < -0.3$ and low if $d_{event} > 0.3$. To increase the sensitivity of the probability distribution to $d_{event} \in [-1, 1]$, the sigmoid function given below is used with constants $a = -6$ and $b = -0.15$:

$$P(\textit{suffering}|\dot{d}_g, \dot{d}_{event}) = \frac{1}{1 + \exp\left((-a \cdot d_{event} + a \cdot b)\right)} \tag{3.1}$$

3.1.2 Jerk along Axis of Motion ($\dot{a}_z$)

This vital detects situations in which performance degrading factors like uneven terrain may result in sudden jerks or jittering along the axis of motion (z axis generally). Sudden dips in terrain elevation can rapidly increase the force on one side, thereby causing the robot to tilt or topple. A higher magnitude of jerk indicates that the robot is more likely to topple, making the probability of suffering higher. During preliminary experiments with a simulated Clearpath Husky robot, it was observed that sudden jerks of $\pm$ 30 degrees ($\approx \pm 0.5$ radians) or above along the z-axis may increase the likelihood of the robot toppling over. Therefore, the probability of suffering given jerk along the Z-axis should be high when $\mid \dot{a}_z \mid \approx \pm 0.5$ radians, and low if $\mid \dot{a}_z \mid \approx \pm 0$.

The jerk magnitude along the axis of motion is calculated using the rate of change of linear acceleration along the Z axis $\dot{a}_z$. a_z is usually measured using an Inertial measurement unit (IMU). Since IMU readings tend to be noisy; raw IMU output values are smoothened using a rolling window average and then downsampled to one reading per second before calculating $\dot{a}_z$. The function for $P(\textit{suffering}|\dot{a}_z)$ is calculated using an inverted bell curve as given below. The inverted bell curve was used to map low jerk along the axis of motion to a low probability of robot suffering, and highly jerky motion in either direction to a high probability of robot suffering.

$$P(\textit{suffering}|\dot{a}_z) = 1 \; - \; \frac{1}{(2\pi)^{\frac{1}{2}}\,\sigma_1} e^{\left(-\left(\frac{0.5}{\sigma_2^2}(\dot{a}_z)^2\right)\right)} \tag{3.2}$$

The values of $\sigma 1$ and $\sigma 2$ were calculated as 0.4 and -0.9 respectively to get the probability of failure close to 1 as $\dot{a}_z$ gets close to ± 0.5.

3.1.3 RoC of Localisation Error ($\dot{\delta}_{loc}$)

Robots sometimes encounter situations where their wheels are free to rotate, but the robot itself is stuck. Uneven terrain is an example of one such performance degrading factor. As the wheels continue spinning, the robots raw odometry estimate(x_1) continues to change. However, other position estimates from visual odometry (Nistér, Naroditsky, and Bergen, 2006) or EKF sensor fusion (x_2) remain relatively constant. Such situations result in localisation errors ($\delta_{loc} = x_1 - x_2$), i.e., the difference between redundant position estimates (Mendoza, Veloso, and Simmons, 2012) of a robot. Some Simultaneous Localization and Mapping (SLAM) algorithms are robust to different levels of localisation errors (δ_{loc}). However, the performance of a robot deteriorates after prolonged periods of high δ_{loc}. Hence, $\dot{\delta}_{loc}$ can be used as an indicator of when a robot's SLAM or localisation is compromised. While SLAM algorithms generally provide confidence measures, $\dot{\delta}_{loc}$ is used as a vital to reduce assumptions made about the robot's localisation algorithm.

During periods of low performance degradation, the localisation error is close to 0, barring small fluctuations. During periods of high performance degradation, the localisation error steadily increases. To detect such situations, the number of times steps $t_{event} = t$ that $\mid \dot{\delta}_{loc} \mid$ continuously takes a non-zero value are counted. In preliminary experiments, that robot failure was observed to be more likely when t_{event} was between 4-5 seconds. This is heuristically encoded as a function in which the probability of suffering linearly increases (with scaling constant $k = 0.2$) with the value of t_{event}:

$$P(\textit{suffering}|\dot{\delta}_{loc}, t_{event} = t) = \begin{cases} k \cdot t & \text{if } t \in [0,5], \\ 1 & \text{if } t \geq 5. \end{cases} \tag{3.3}$$

3.1.4 Robot Velocity ($\dot{x}$)

A robot's velocity is a salient indicator of performance degradation. During periods of low performance degradation, a robot velocity is constant unless acceleration or deceleration is required to turn, change directions, or around way points. This constant value is generally pre-set by the manufacturer or set by the operator before use. Navigation errors, SLAM algorithm limitations and hardware issues commonly cause a robot to halt during task execution, thereby causing a sharp drop in velocity. Alternatively, motor malfunctions and braking issues cause a robot to accelerate for long periods, thereby exceeding its standard operating velocity.

For this vital, the robot velocity is calculated by differentiating successive EKF fused position estimates of the robot. The probability of suffering is calculated as a function of the number of seconds ($t_{event} = t$) where a robots velocity is continuously trivial (i.e., close to 0), or exceeds the robot's max speed (1.0 m/s for a Clearpath Husky). That is, the number of seconds where $\dot{x} \leq \mid 0.01 \mid$ or $\dot{x} \geq \mid 1.0 \mid$ are counted. Preliminary experiments showed that the value of t_{event} is generally below 3-4 seconds when the robot is facing low performance degradation. Accordingly, the probability of suffering is encoded as the following sigmoid function with $a = 1.5$ and $b = 2.5$ such that the probability of suffering increases when t_{event} is higher than 3 seconds.

$$P(suffering|\dot{x}, t_{event} = t) = \frac{1}{1 + \exp\left((-a \cdot t_{event} + a \cdot b)\right)} \tag{3.4}$$

3.1.5 Laser Scanner Noise Variance (σ^2_{noise})

This vital detects situations where laser scanner noise impairs a robot's ability to perceive, map, or navigate its surroundings. Noisy readings create inaccurate representations of a robot's surround-

ings, thereby increasing the likelihood of collisions, sub-optimal path planning, and robot failure. The methods for evaluating noise estimation or the robustness of SLAM algorithms to different types and levels of laser noise are beyond the scope of this study, as the focus here is on understanding how a robot behaves in a given case where laser noise compromises SLAM. Therefore, the focus here is on the effect of additive white Gaussian noise on a Husky robot that uses the ROS navigation stack (Marder-Eppstein, Berger, et al., 2010b). The laser scanner measurement array is first rearranged as a square gray scale image. The noise variance (σ^2_{noise}) of this image is used as an estimate of the total laser scanner noise (Immerkaer, 1996). The σ^2_{noise} value is then calculated by convolving the image with a $3x3$ mask and applying summations on the resultant matrix. Preliminary experiments with a Husky robot showed that low noise ($\sigma^2_{noise} \approx 0.7$) had little to no effect on its navigation, and as the noise increased to $\sigma^2_{noise} \approx 1.4$, the robot's likelihood of halting or failing increased. This effect of noise variance values between 0.5 to 1.5 on the robot is captured by designing $P(suffering|\sigma^2_{noise})$ as the sigmoid function in 3.5 defined over σ^2_{noise}, with constants $a = 5$ and $b = 1$.

$$P(suffering|\sigma^2_{noise}) = \frac{1}{1+\exp\left(\left(-a\cdot\sigma^2_{noise}+a\cdot b\right)\right)} \tag{3.5}$$

3.2 Robot Health

The robot health is an overall scalar estimate of a robot's ability to carry out its tasks autonomously without its capabilities being impaired by any performance degrading factors. While carrying out a task, performance degradation will cause a robot's health to dip, making it more likely that the robot may fail or perform sub-optimally. Therefore the health of a robot can be monitored to detect situations where it is necessary to provide assistance to the robot to improve its performance,

prevent imminent failure or mitigate the effect of a performance degrading factor. The robot health therefore, gives a simple, intuitive metric to quickly determine the severity of a robot's performance degradation, simplifying the process of operator intervention or the triggering of pre-programmed recovery behaviours.

Algorithm 1 Suffering and Health Estimation

Initialization: ▷ Occurs once at the start
Create publishers for `/probSuffering_total` and `/robotHealth`.
Create subscribers for:
  P_{magROC}, $P_{velocity}$, P_{posErr}, P_{linAcc}, P_{snr}.
Set a timer to compute metrics every 0.5 seconds.
Initialize `probabilityOfsufferingVector` as $[0,0,0,0,0]$.
Initialize empty `robotHealthVector`.

while true **do** ▷ Continuous health publishing loop (ROS spinning)
  On receiving updates: ▷ Triggered asynchronously by subscribers
  Update the corresponding element in `probabilityOfsufferingVector`.

  Every 0.5 seconds: ▷ Periodic computation triggered by timer
  Compute $P_{total} \leftarrow \frac{\sum_{i=1}^{5} P_i}{5}$.
  Publish P_{total} to `/probSuffering_total`.
  if `robotHealthVector` length < 10 **then**
    if $P_{total} > 0$ **then**
      Append $P_{total} \cdot \log(P_{total})$ to `robotHealthVector`.
    end if
    Set $H \leftarrow 0$.
  else
    Compute $H \leftarrow \sum(\texttt{robotHealthVector}) + (1 - P_{total})$.
    if $P_{total} > 0$ **then**
      Remove the oldest element from `robotHealthVector`.
      Append $P_{total} \cdot \log(P_{total})$ to `robotHealthVector`.
    end if
  end if
  Publish H to `/robotHealth`.
end while

Robot health combines the effect of several performance degrading factors into a single meta-metric. This is somewhat analogous to expert ensembles (Dietterich et al., 2002), or strategic decision-making that draws on opinions from diverse subject experts in crisis management. There are different ways of creating a "combination of experts" AI. However, this chapter focuses mainly

on determining the feasibility of such a system and realising it, not finding the most optimal method of creating such a system. Here, for proof of principle, the robot health is computed in terms of probability of robot "suffering" given the robot vitals (see section 3.1). The entire pipeline of computing the robot vitals and probability of suffering throughout the robot's runtime is given in algorithm 1. The total probability of robot suffering at time t is:

$$P(suffering)\big|_t = \eta \sum_{v \in V^t} P(suffering|v)P(v)|_t \tag{3.6}$$

where,

$$\text{where } V^t = \{\dot{d}_g, \dot{a}_z, \dot{\delta}_{loc}, \dot{x}, \sigma^2_{noise}\} \tag{3.7}$$

Therefore, equation 3.6 results in

$$\begin{aligned} P(suffering)\big|_t = (P(suffering|\dot{d}_g, \dot{d}_{event})\big|_t + P(suffering|\dot{a}_z)\big|_t \\ + P(suffering|\dot{\delta}_{loc}, t_{event} = t) + P(suffering|\dot{\delta}_{loc}, t_{event} = t) + \\ P(suffering|\dot{x}, t_{event} = t) + P(suffering|\sigma^2_{noise}))\big|_t \end{aligned}$$

Where v is any robot vital from $V = \{\dot{d}_g, \dot{a}_z, \dot{\delta}_{loc}, \dot{x}, \sigma^2_{noise}\}$ at time t, and η is a normalisation constant. For the sake of simplicity, the probability of observing each vital $P(v)$ is assumed to be 1, i.e., a perfect observation model. In tasks where the robot's vital has partial observability due to communication issues or component malfunctions, the value of $P(v)$ could be replaced with more sophisticated models to incorporate the probability of observing the vital at any point in time.

Information entropy is a standardised metric used to quantify the amount of information uncertainty or 'surprise' in a random variable's possible outcomes. Low entropy is observed when

a robot is operating under little to no performance degradation (i.e. 'normal' operating conditions). Sudden severe performance degradation, or a gradual rise in degradation, will cause entropy to increase. If the problems are mitigated and the robot returns to normal operating conditions, then entropy will fall again. To associate high and low health with low and high performance degradation respectively, the additive inverse of information entropy is used as the robot health. The robot health between two time intervals t_1 and t_2 is calculated using information entropy as follows:

$$H^{t_1:t_2} = \sum_{t=t_1}^{t=t_2} P(suffering)\big|_t \cdot log(P(suffering)\big|_t) \tag{3.8}$$

This approach of using entropy as a measure of robot health is based on the insight that uncertainty in the robot's behaviour is induced by performance degradation. Therefore using the robot health, the operator can very quickly intervene when the robot is not behaving as expected. During a steady state of robot operation there is low uncertainty, but transient behaviour would increase the information entropy. The information entropy is used to detect transient behaviour in experiments as it is especially effective when there is a clear distinction between areas of normal operations and those with performance degradation. For subsequent experiments with LoA switching systems that use Robot Health, where the performance degrading factors were more spread out, it was more important to communicate the severity of performance degradation irrespective of the duration, a simpler formulation was used as given below. This equation calculates the robot health as the probability that the robot is "not suffering", and averages it over a rolling window to smoothen the curve and remove the effects of minor fluctuations due to noisy data.

$$H^{t_1:t_2} = \frac{\sum_{t=t_1}^{t=t_2}(1 - P(suffering)\big|_t)}{t2 - t1} \tag{3.9}$$

3.3 Applying the Robot Vitals and Robot Health Framework to Different Contexts

The above-presented vitals should work well for most types of mobile robot navigation scenarios, provided that the effects of the environment on performance degradation are captured by one or more vitals. However, some parameters will need tuning for different platforms in different settings. Such tuning can be done empirically or learned from simulations or real robot experiments, and/or inferred from specifications of the robot's sensor systems. Consider the case of adapting Robot Vitals from the rugged Clearpath Husky robot to a small lower-spec Turtlebot. After preprocessing robot data to create the vitals, the thresholds for event detection need to be adjusted. For example, the jerk ($\dot{a}_z$) required to topple a Turtlebot may be less than for the Husky. Alternatively, consider the case of a high-spec robot, with very advanced SLAM system designed for operating in noisy, uncertain environments. In this case, the minimum threshold of (σ^2_{noise}) to indicate performance degradation may need to be set higher, and can be determined through simulations with varying noise levels. Finally, once the event thresholds are fixed, the $P(\textit{suffering})$ function's shape can be modified by adjusting its constants.

Overall, this framework should in principle, apply more widely to other types of tasks and robots, e.g., manipulation scenarios, unmanned aerial vehicles, or underwater robots. For example, when applying this framework to drones, factors like the wind gust speed, rotor speed, camera occlusion etc are the main sources of performance degradation and therefore, vitals should be chosen to indicate the effect of these factors on the drone's performance. Selecting and tuning new vitals for new applications hinges on the questions: "What are the key ways in which the robot can fail?". "What are the sensor readings when performance starts to degrade, and what processing, functions (and function parameters) of the sensor readings robustly detect this?". Ultimately, a robot vital translates to a probabilistic model of a robot suffering given vital, $P(\textit{suffering}|v)$.

Equation 3.9 for robot health assigns the same weightage to the probability of suffering for each vital. This may not always be the best approach, because each performance degrading factor may not always equally impact the probability of robot's failure. In such situations, the health metric could use different weights for each vital based on the principal components in the trends related to robot failure. When combining multiple vital probabilities into a single robot health metric, the choice of combination approach (e.g., sum, product, etc.) has important implications for the behaviour and sensitivity of the resulting metric. Using a sum, as shown in Equation 3.6, effectively averages the vital probabilities, providing a smoothed representation of overall robot suffering. This approach may be preferable when robustness to occasional noisy or outlier vital readings is desired, as the effect of a single high-probability vital would be dampened by the other low-probability vitals. However, this "smoothing" effect could also potentially mask severe performance degradation if one or more vitals indicate a high probability of suffering while others remain low. Alternatively, taking a product of the vital probabilities would make the robot health metric more sensitive to individual vitals signalling high suffering probabilities. In this case, even a single vital indicating severe degradation would result in a low overall robot health value, effectively sounding an alarm. However, this increased sensitivity could also make the metric more susceptible to false alarms from transient noise or errors in individual vitals. The choice between these approaches to design the robot health metric therefore depends on the desired trade-off between robustness and sensitivity, as well as the specific application requirements and constraints.

3.4 Experimental Validation

The goal of the experiments carried out in this study was to determine if it is possible to quantify a robot's runtime performance degradation, and whether the Robot Vitals and Robot Health Framework was able to capture this. In these experiments a robot is tasked with navigating an ob-

Table 3.1: Different conditions tested in the experiments

Experiment I, 15 Trials in each condition				
	Intensity of Performance degradation			
	Baseline	Level 1	Level 2	Level 3
Noise Scale	0	0	0	0
% Uneven Terrain	0	10	20	40
		Level 1	Level 2	Level 3
Noise Scale		0.4	0.5	0.6
% Uneven Terrain		0	0	0
		Level 1	Level 2	Level 3
Noise Scale		0.4	0.5	0.6
% Uneven Terrain		10	20	40
Experiment II, 14 Trials in each condition				
		Intensity of Performance degradation		
		Level 1	Level 2	Level 3
Noise Scale		0	0	0.4
High Friction (HF) Terrain Used		No	Yes	Yes

stacle course autonomously, while being subjected to different types and severities of performance degradation. During this task, the robot's health and overall task performance were measured. Two experiments were carried out using a Clearpath Husky Robot - first in a high fidelity simulation environment (referred to as Exp. I) and the second using the real robot (Exp II). In each of these experiments, the robot was tasked with autonomously navigating from point A to B in the arenas as shown in figures 3.2 and 3.4 . A repository containing the ROS code for Robot Vitals and Robot Health, and all code necessary to replicate these experiments, is provided under MIT license[2].

In both experiments the robot uses the ROS navigation stack (Marder-Eppstein, Berger, et al., 2010a) with the dynamic window local planner and a global planner that uses Dijkstra's algorithm (Dijkstra, 1959). Within the navigation stack, if the robot remains inactive beyond a certain threshold time, the planner clears the cost map and rotates the robot to create a fresh scan of the environment. To minimize confounding factors introduced by this rotational behavior, the rotate recovery feature in the navigation stack was disabled. The robot is not given any prior in-

[2] https://github.com/anikethramesh/robotVitals

formation about the map, performance degrading factors, and the nature of the terrain or boundary conditions. In case the robot aborts its navigational goal during runtime, the goal gets reset so that it can find a new path. This is done to avoid the robot failing the experiment. However, if the robot is stuck, unable to find a path to the goal or aborts navigation for 30 seconds despite resetting the goal, the experimental trial is terminated.

Three common field repairable performance degrading factors observed in adverse environments were used in the experiments: High Friction (HF) terrain, uneven terrain, and laser noise. These performance degrading factors can greatly impair an autonomous robot's navigation ability, but do not cause similar issues when the robot is manually controlled. Small dust particles in air or smoke can deflect laser beams thereby inducing noise in a laser scanner, but do not degrade the robot's camera feed. Laser scanner noise degrades the robot's ability to perceive its environment, introducing localisation and navigation errors. These errors affect the velocity of the robot, the speed at which it moves towards the goal, and the laser noise ($\dot{d}_g$, $\dot{x}$, and σ^2_{noise}) . Uneven terrain is a performance degrading factor commonly encountered in extreme environments. For example, earthquakes can cause fissures in the ground or buildings to break leading to pieces of debris in the robot's path. While a robot's laser scanner can detect obstacles that are large enough, obstacles/debris that are lower than the laser scanner's altitude are not detected, thereby causing problems in autonomous navigation. Traversing uneven terrain causes instability. Changes in elevation or inclination may result in the laser scanner detecting the ground as an obstacle, thereby degrading navigation. This can cause changes in the jerk, reduce the velocity of a robot and reduce the rate at which it moves towards the goal (values of $\dot{d}_g$, $\dot{x}$, $\dot{a}_z$, and $\dot{\delta}_{loc}$). Finally, robots face difficulty turning and moving smoothly on HF terrain. Various environmental factors such as sand dunes, muddy areas and urban environments may present areas with high friction that limit a robots ability to turn. Robots may slip, skid, or even halt on such surfaces. The values of $\dot{d}_g$, $\dot{x}$, and $\dot{\delta}_{loc}$ are affected in such cases. That is, due to constant slipping and sliding, the errors may be induced to the robot's ability to localise itself. Different intensities of these performance degrading factors

were combined to create multiple experimental conditions (see Table. 3.1).

3.4.1 Experiment I

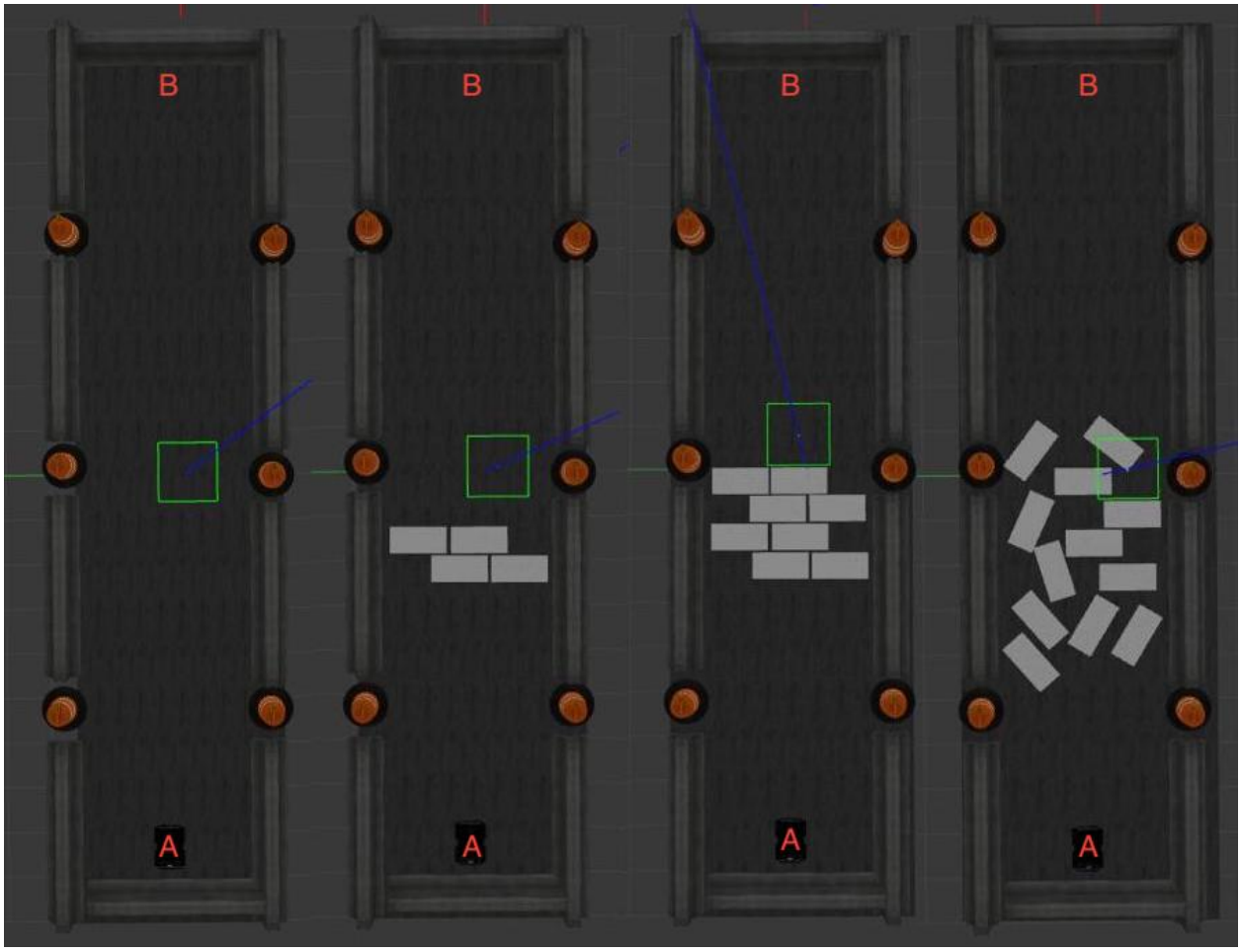


Figure 3.2: The simulated arenas: From left to right, uneven terrain covers 0%, 10%, 20%, and 40% of the area respectively.

Gazebo, a high-fidelity robotics simulation with a realistic physics engine, was used for Exp I. The robot was equipped with wheel encoders for odometry, an LMS-111 LIDAR scanner and a UM6 IMU sensor. Seven seconds after the start of each trial, varying degrees of random additive Gaussian white noise were introduced into the laser scanner to degrade the robot's localisation using the Box-Muller transformation (Box, 1958). Laser noise was turned off after seven more seconds. These timings were chosen such that the robot's performance was degraded for some

time, but the goal was not aborted by the robot's navigation stack. This way, sufficient performance degradation was introduced without causing complete robot failure. To control noise magnitude, the standard deviation of the Gaussian kernel is multiplied by several noise scale values, Table 3.1. For Exp I, four different arenas with uneven terrain were designed as shown in Figure 3.2. These arenas had 0%, 10%, 20% and 40% of their total area covered with the "FRC 2016 Rough Terrain" Gazebo block. HF terrain was not used for Experiment I for two reasons - 1) The simulation environment did not replicate the behaviour of robots in HF terrain properly, and 2) HF terrain does not cause robot performance degradation when a straight path without turns is used.

Results of Experiment I

The task completion time T_{comp}, and average health values for each trial, sorted by the different levels of performance degradation, are plotted in Figure 3.3. With no performance degrading factors, the average value of T_{comp} is 40 seconds and average robot health during runtime is -0.7. Performance degradation increases T_{comp} and reduces average health. However, the ranges of T_{comp} and average health for different levels of performance degradation vary. The combination of laser noise and uneven terrain results in more performance degradation than either of these factors alone. This evidence suggests that the effect of multiple performance degrading factors on robots may not always be additive in nature, and that quantising the effects of performance degrading factors into different levels is complex and often non-linear. However, their intensities can be measured in relative terms. Among different performance degrading factors, laser noise induces the lowest level of degradation (minimum increase in T_{comp} from baseline performance). The presence of noise and uneven terrain results in the lowest range of average health values (-1.22 to -1.484).

To assess the relationship between (T_{comp}) and the average robot health metric, a non-parametric correlation analysis was deemed appropriate due to the ordinal nature of the performance degradation levels and the non-normality of the data distributions. Spearman's rank cor-

relation test was chosen as it is a robust and widely used method for measuring the strength and direction of monotonic relationships between two variables, without making assumptions about the underlying data distributions or linearity of the relationship. This test is particularly suitable when working with ordinal data or when the data violates the assumptions of parametric tests, such as normality or same variances of the distributions. The Spearman's rank correlation test showed a strong and significant negative correlation ($p < 0.001$, $\rho = -0.93$) between T_{comp} and the average robot health.

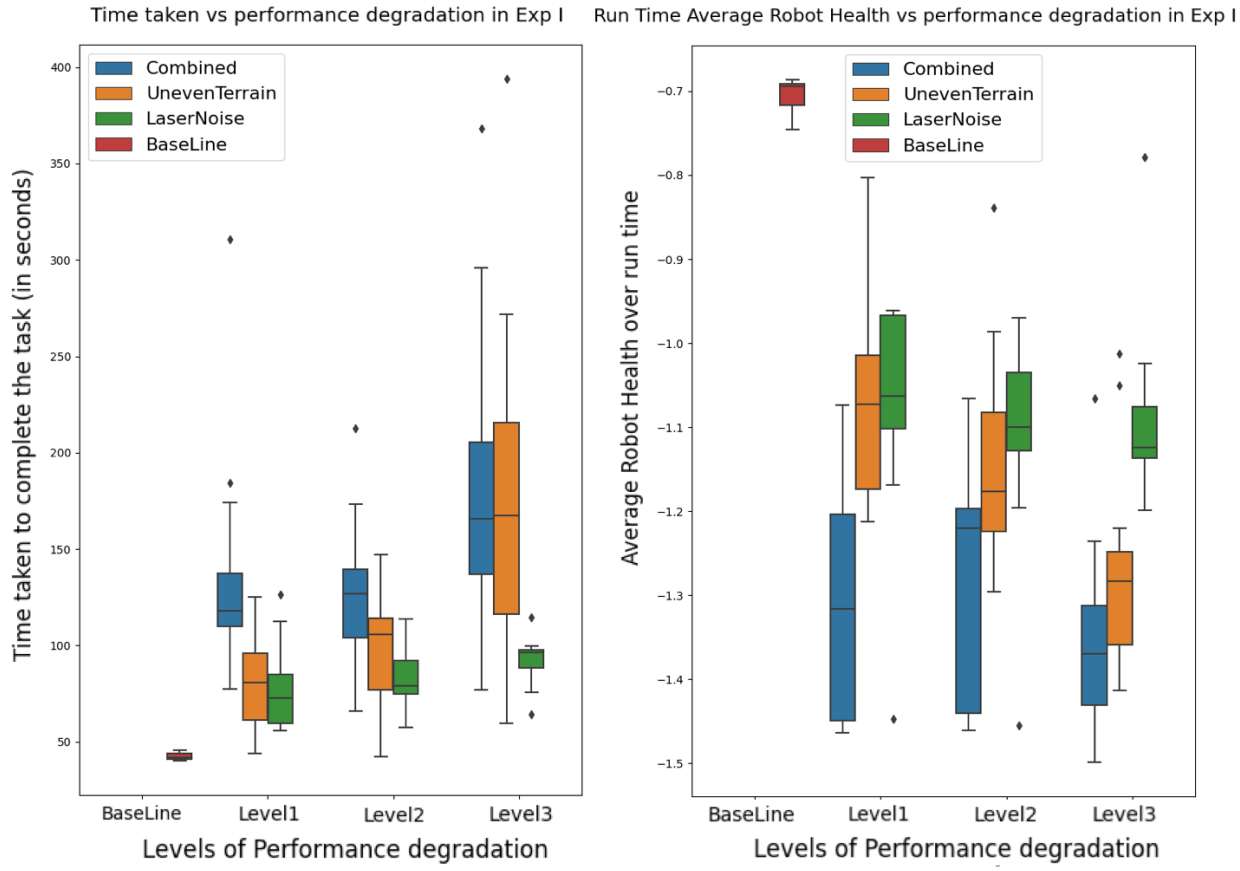


Figure 3.3: Exp I - **Left:** Task completion time T_{comp} (lower is better); **Right:** Average robot health (higher is better); The diamonds represent outliers.

3.4.2 Experiment II

Exp II was carried out using a real Husky robot in the experimental setup depicted in Figure 3.4. This setup was inspired by real tasks in extreme environments like search and rescue operations or disaster site inspections, where robots must navigate through complex and unpredictable environments. Since the robot used for this experiment did not possess an IMU, $\dot{a}_z$ values were not calculated. Therefore, robot health was calculated using four vitals $V = \{ \dot{d}_g,\ \dot{\delta}_{loc},\ \dot{x}$ and $\sigma^2_{noise}\}$ in Equation 3.6. While carrying out Experiment I, it was observed that uneven terrain generates significant jerks during robot motion, which may lead to the robot tipping over or colliding with walls. To avoid risking hardware damage due to this, uneven terrain was not used for the real robot experiments. Instead, HF terrain, obstacles and laser noise were introduced to degrade the ability of the robot to turn and move smoothly while minimising the risk of physical damage. In real-world scenarios, robots often encounter situations where the ground surface may shift or become unstable due to factors such as loose debris, uneven terrain, or environmental disturbances. These changes in the ground surface can significantly impact a robot's odometry and localisation accuracy, as the robot's wheel encoder position estimations rely heavily on assumptions about the ground being stationary and predictable. The reason behind using a movable tile for the HF terrain was to simulate real-world scenarios where the ground surface may unexpectedly change or become unstable, leading to odometry and localisation errors. Therefore, the square tile of HF terrain was not fixed on the floor. As the tile slipped and moved during the robot's turns, it would introduce inconsistencies and uncertainties in the robot's motion estimation, mimicking the challenges faced in environments where the ground surface is not static or well-defined. A square tile of side length 1.2 meters wrapped with a high-friction rubber mat was used as HF terrain for this experiment. The arena used for Exp II, and the position of the HF terrain is shown in Figure 3.4. An area in the robot's path was marked with blue tape as a fixed area where laser noise will be introduced. The locations where performance degradation was introduced were marked and kept

constant. That is, after each trial, in case the robot collided with obstacles and moved them, or caused the HF terrain to slide due to turns their positions were reset before the next trial. In each trial, the experimenter triggered laser noise using the experimenter joystick as soon as any of the robot's wheels entered this area. The laser noise was active for a period of 5 seconds. Similar to Experiment I, the duration was chosen based heuristically, to minimise the chance of the robot aborting the navigational goal.

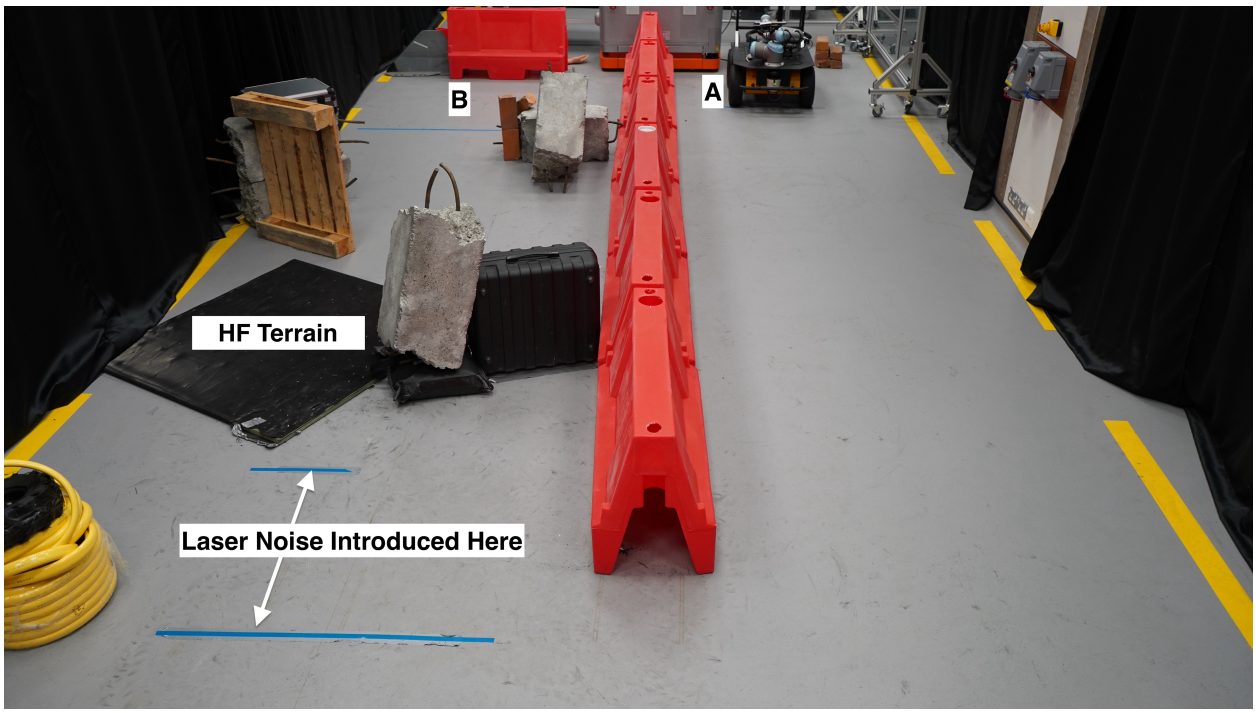


Figure 3.4: The arena in Exp II: Points A and B, marked with blue tape, denote the start and goal positions. The different performance degrading factors used are annotated.

Results of Experiment II

Average health and T_{comp} values for different levels of performance degradation were calculated and plotted in Figure 3.5. The range of T_{comp} values increases with levels of performance degradation. The middle quartile of average health observed in level 3 is lower than that of level 2, however the range of values are similar. The robot was unable to complete the task in 2 trials of level 2, and 3 trials of level 3. In these trials the robot momentarily experienced a localisation error

on the HF terrain, and then could not find a collision-free path, even after resetting the goal. Similar to experiment 1, the Spearman's Rank Correlation test was applied to determine if there is a correlation between T_{comp} and the Average Health. The Spearman's Rank Correlation test showed a significant ($p < 0.001$), with strong negative correlation ($\rho = -0.77$) between the average robot health and T_{comp} values.

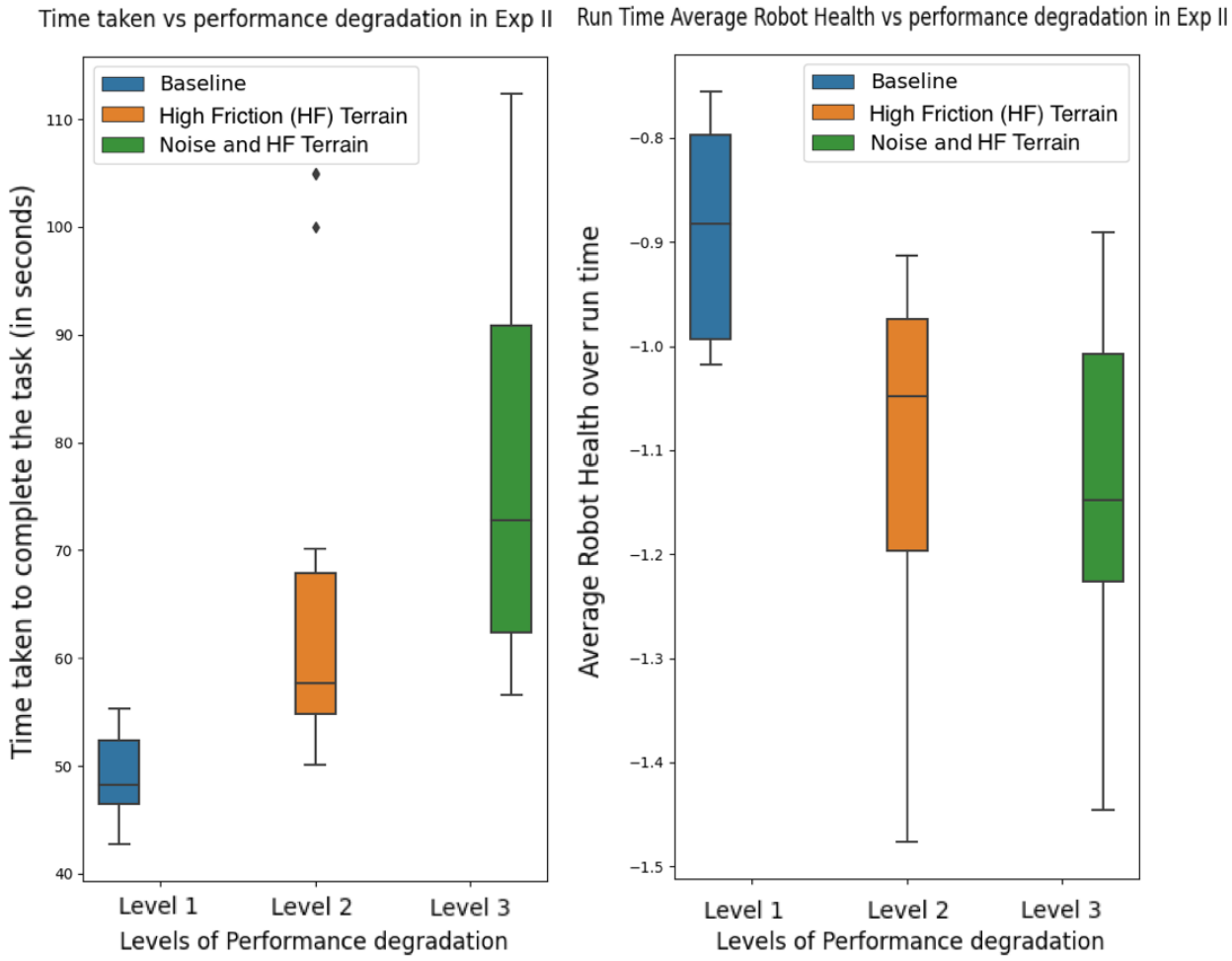


Figure 3.5: T_{comp} (left) and average robot health (right) boxplots for Exp II. The diamonds represent outliers.

The variation of instantaneous robot health values over time, for various different combinations of degradation factors, is plotted in Figure 3.6. Given the same experimental conditions, robotic hardware and navigation algorithms, the performance degradation induced in a robot varied in each trial. This variation is due to combination of a stochastic path-planner, with stochastic

noise, and physical terrains that may cause different outcomes for small variations in e.g. approach angles. The high variance in robot performance between different trials indicates that along with performance degrading factors, stochasticity in the environment of operation plays an important role in affecting the performance of the robot in any task.

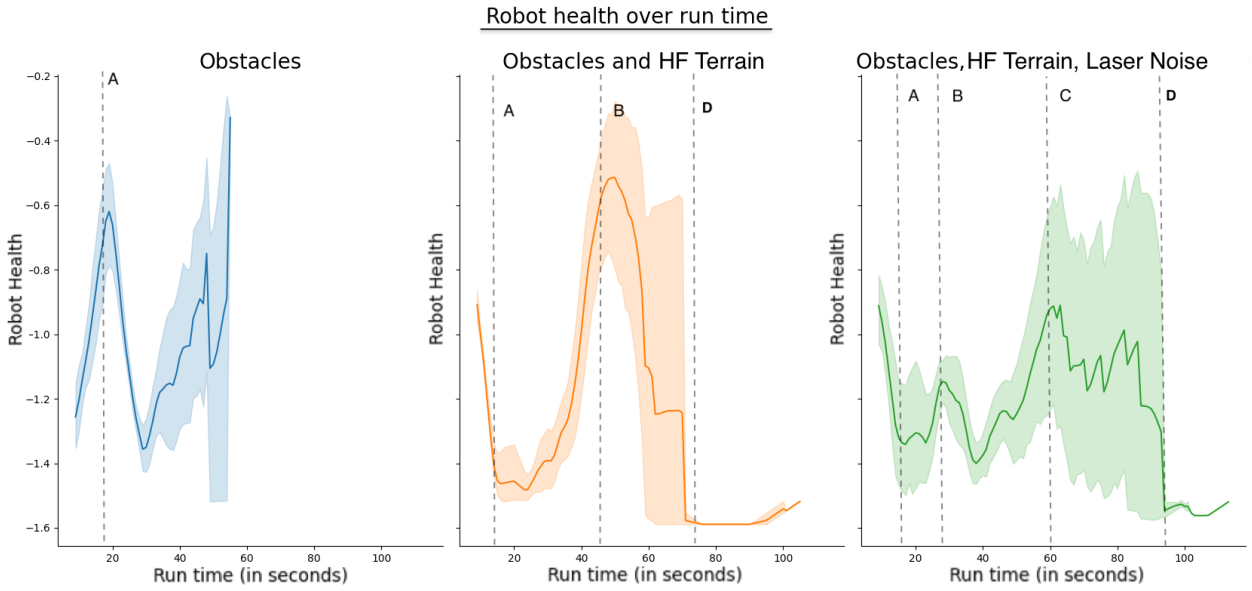


Figure 3.6: The health trend over runtime for all conditions in Exp II. (L to R) Level 1, Level 2, Level 3. Error bands around the lines represent the 95% confidence interval of values. Dotted lines indicate when the robot entered the area with obstacles (A); HF terrain (B); laser noise (C); (D) indicates the approximate timestamp where robot failures were observed.

As seen in Figure 3.6, the health trend for level 1 dips during the 20 to 30 seconds time period. In this period, the robot encountered obstacles, and slowed down to create a new navigation plan. The health then continued to rise until task completion, indicating little further performance degradation. In level 2, the health sharply drops around the 50 second mark. This is consistent with the robot encountering HF terrain and facing navigation errors. In level 3, the robot health trend was characterised by high fluctuations due to the combined effect of laser noise, obstacles and HF terrain. The introduction of multiple performance degradation factors in quick succession causes the health to stay below -0.8 throughout the runtime.

3.5 Discussion

Our experimental results illustrate how, as the severities and types of performance degrading factors increase, T_{comp} increases (indicating posthoc, an overall decline in mission performance). Meanwhile the average robot health also decreases (see Figures 3.3 and 3.5). This suggests that the proposed health metric, which combines inputs from the proposed vitals metrics, provides a useful indication of the degree to which a robot suffers difficulties. As shown in Figure 3.6, the robot health metric, at each time step, successfully tracks robot performance degradation in runtime. Furthermore, there is a strong negative correlation between T_{comp} and average robot health. This shows that using the "Robot Vitals and Robot Health" framework, a robot can estimate its own performance degradation online during missions, and that it has similar accuracy as commonly used offline, post-hoc performance metrics, i.e., T_{comp}. Hence, the proposed system of "robot vitals" and "robot health" can successfully provide an autonomous robot with awareness about its own performance degradation, at any instant, while executing missions.

The same approach can also be applied to semi-autonomous (shared control or variable autonomy) systems in which a human and autonomous agents collaborate to control a remote robot. This kind of self-monitoring is an important step towards avoiding catastrophic mission failures, since it can detect deteriorating performance, and trigger remediation measures such as remote human intervention and supervision, or autonomous recovery behaviours. A simple example of this is- if a robot has low health, and this is due to jerky motion and the robot is incorrectly classifying the ground as an obstacle, this would be indicated by very high values of $P(\textit{suffering}|\dot{a}_z)$ and $P(\textit{suffering}|\dot{x}, t_{event} = t)$. In such situations, the robot can alert the operator to clear the cost map or even trigger this recovery behaviour autonomously.

Exp II provides insights on the utility of such a system. In some trials, the robot got stuck momentarily, or failed when it was unable to find a collision free path through the arena. During

these instances the robot's health dropped below -1.4. Thus, a threshold-based control switcher (Chiou, Hawes, and Rustam Stolkin, 2021) could be used to initiate recovery behaviours. Alternatively, an operator could take control of the robot to provide it a collision free path, or teleoperate it. Furthermore, in "one human, many robots" multi-robot paradigms, robot health of each robot can be used to prioritise those robots most in need of operator attention.

In a variety of applications, but especially in high-consequence or extreme environment applications, there is increasingly a demand for "explainable AI". The vitals-based health metric provides intuitive explainability, with the different vitals providing rich diagnostics information to help humans understand any particular problematic circumstance. While comparatively less explainable and intuitive, a set of vitals can also be found using methods like principal component analysis or machine learning. Such approaches can mine large amounts of data from robot operation to find parameters that best represent its performance degradation. As opposed to black box AIs this framework increases the transparency of the system, with the added ability of allowing experts to encode their knowledge. Recommender systems or natural language interfaces can provide explanations of the robot's state very easily using the Robot Vitals and Robot Health Framework.

While the health metric in Exp I used five vitals, Exp II used only four, as the robot lacked an IMU. This illustrates the generalisability and scalability of the framework, which is intended to handle different numbers and types of vitals according to different robots and tasks. Additional vitals can be readily added to the health metric using a $P(\textit{suffering}|v)$ function that relates changes in the vital with the probability of the robot suffering. This function can be derived from Monte Carlo studies, reliability studies, or expert knowledge. One limitation of this study, as mentioned in section 3.2, is the assumption of a perfect observation model for the observability of each vital (i.e, $P(v) = 1$). However, in field applications where the sensor data observability is error-prone, this can be replaced with a probability model $Pv|m_{1:N}$, where the value of the vital v has to be inferred from a series of N sensor measurements $m_1, m_2, m_3 \ldots m_N$.

Lastly, one explanation for the variance in some of the results is the stochasticity in the robot's autonomous planner and experimental setup. In each trial, minor differences in the laser scanner measurements and robot odometry, different map representations and update frequencies are some of the factors that caused the robot to take different paths to the navigational goal. Some of the paths through the obstacles resulted in higher likelihoods of collision and failure. Due to this stochasticity, an increase in the level of performance degradation in the experiments did not always induce a commensurate increase in T_{comp}. The stochasticity and path variability can be reduced greatly if the robot knows the map before the start of each experiment. However, this option was not exercised, to examine how performance degradation affects the robot's ability to create a map. Consequently, this work identifies that designing experiments where all performance degradation and variability are controlled is a key challenge for subsequent studies in this field.

3.6 Future Scope

There are several promising avenues for future work on the Robot Vitals and Robot Health Framework. A key area for development is streamlining the process of designing and tuning the robot vitals and the robot health metric. For situations where extensive data is available from previous experiments, machine learning pipelines can be created to adaptively tune the robot vitals based on a combination of real-time robot operation data and data from previous experiments (Carlucho et al., 2017). This adaptive tuning is particularly useful for long-term missions where robots may experience gradual changes, deterioration, or wear and tear due to environmental factors, a challenge well-documented in long-term autonomy research (Howe et al., 2020). In such cases, machine learning pipelines that re-tune thresholds for different vitals can be highly beneficial, saving the time taken to retrieve the robots for periodic updates and recalibrations. This process would involve collecting performance data during testing to analyse deviations from the initial baselines in the trends of each vital, and using the machine learning pipelines to continuously adjust the

thresholds for the probability of suffering for each vital. This approach would potentially enhance the robot's preparedness for operation in real-time environments and minimise the probability of catastrophic mission failure.

For applications in extreme environments, standardising an experimental arena with commonly observed performance-degrading factors could be useful, similar to the mock-up arenas developed by the National Institute of Standards and Technology (NIST) (Wang, Lewis, and Gennari, 2003). This approach builds on established practices in standardised robotics testing (Perille et al., 2020). Before deployment in a real environment, a robot could be tested on a representative task in the mock-up course, under the observation of expert operators. This controlled testing would facilitate the calculation of robot vitals and health metrics, and the establishment of appropriate thresholds specific to each robot. Tuning thresholds based on performance in these mock-up courses would enable operators to better understand each robot's behaviour and response to different levels of performance degradation. This tailored approach ensures that the framework is optimised for specific robots and anticipated challenges, facilitating better human-robot teaming, a key aspect of effective robot deployment in extreme environments (J. Y. Chen and Barnes, 2014).

3.7 Conclusion

This chapter has proposed a novel framework of "robot vitals" and "robot health," enabling robots to detect and quantify their own performance degradation online, during task execution. A systematic approach is outlined for designing vitals tailored to specific robots, tasks, and environments. Additionally, a scalable framework is proposed for combining information from multiple vitals into a single, overall health metric.

In this chapter, systematic experiments were presented where both simulated and real mobile robots encounter performance degradation of various types and severities. Results show that

the "Robot Vitals and Robot Health" framework can detect the presence and severity of performance degradation caused by a wide variety of circumstances. Instantaneous online detection of performance deterioration has been demonstrated, which correlates strongly with time stamps at which different adversities are encountered. The Robot Vitals and Robot Health Framework can seamlessly accommodate different numbers and types of vitals, allowing it to be tailored to the specific requirements of diverse robots and tasks. The main benefit of this framework is its versatility. Therefore users can devise their own metrics for robot vitals and robot health using various methods like machine learning, fuzzy inference, or expert knowledge etc. The robot health metric is a simple, intuitive metric that can enable the creation a simple shared knowledge representation of the entire HMRT.

By quantifying the runtime performance degradation of each robot in real-time, this framework lays the foundation for the development of triage systems for human multi-robot teaming applications. Using this framework as a robust foundation for a robot's state estimation can enable the creation of AI systems that assist operators with triage and interaction paradigms that simplify triage decision-making. Additionally, this framework allows robots to self-assess their performance and alert the operator when they require assistance with task execution. In the next chapter, three different realizations of VA systems are presented. These systems leverage the Robot Vitals and Robot Health Framework to self-regulate their autonomy and request operator assistance based on the level of performance degradation experienced during task runtime.

Chapter Four

Mixed-Initiative Variable Autonomy based on Robot Health

This chapter addresses the challenge of enabling robots to self-regulate their autonomy and request operator support during the task to mitigate performance degradation. As described in Chapter 2, existing approaches to accomplish this include LoA switching, Shared Control and Adaptive Automations for Degree of Autonomy Regulation. While there are several implementations of these systems in the existing literature, to the best of my knowledge, there is a gap for implementations based on a singular, empirically validated metric that can quantify the total performance degradation experienced by a robot during runtime. To address this gap, the Robot Vitals and Robot Health Framework introduced in Chapter 3 is proposed as a foundation for realising VA systems that can self-regulate their autonomy. The key contributions in this chapter are the realisation and rigorous experimental evaluation of the following three novel VA systems that use the Robot Vitals and Robot Health Framework as a robust basis for state estimation.

1. Model Predictive Degree of Autonomy (DoA) and LoA regulation using parameter optimisation
2. Model Predictive DoA and LoA regulation using optimal control

3. Model Predictive Controller for Mixed Initiative LoA Switching

All these realisations follow the simple intuition that the need for operator assistance increases with the magnitude of performance degradation experienced by the robot during runtime. item 1 and item 2, both adaptive automations for DoA regulation are evaluated on a low fidelity numerical simulator (MATLAB). Informed by insights from experiments on these systems item 3, a novel MI system for LoA switching is proposed. This is evaluated rigorously through participant studies in a high-fidelity robotics simulation environment (Gazebo). Additionally, the performance of item 3 is compared to the state of the art MI LoA switching system by Chiou, Hawes, and Rustam Stolkin (2021), and the findings are presented along with some comments on the future work. The rest of the chapter is divided into two different sections. Section 4.1 describes the problem formulation for the 2 adaptive automation systems which optimise robot health. Section 4.3 describes the Model Predictive Controller for Mixed-Initiative LoA switching in detail.

4.1 Adaptive Automation Systems

Shared control, one of the approaches to realising VA systems allows for continuous blending of both the robot's and the operator's inputs. This approach is useful when the robot requires operator assistance through the task but has complementary capabilities to the operator. For example, during remote robot navigating through rough terrain, the operator might provide higher level directional inputs to the robot, while the robot's onboard controller handles obstacle avoidance (Pappas et al., 2020) or focuses on minimising jerky motion (Kucukyilmaz and Demiris, 2018). While this approach keeps the operator in the loop throughout the mission's runtime, it inherently lacks flexibility. If the operator is occupied with other tasks, the robot's inability to assume more control can overburden them. It may also prevent sufficient operator input when human judgment is crucial, potentially compromising task efficiency.

Adaptive automation represents a dynamic approach to Variable Autonomy. In contrast to shared control systems, these systems adjust the ratio of control inputs between the robot's onboard intelligence or robot's controller, and the operator's control input. This ratio is denoted by $0 \leq \alpha \leq 1$. When the value of α is 0 the robot is in manual control, $\alpha = 1$ represents full autonomy. This flexibility enables the system to optimise the balance between robotic autonomy and human control based on real-time assessments of the situation, robot performance, and operator state. For instance, the system can increase autonomy when the robot is performing well in familiar environments, reducing operator workload. Conversely, it can shift more control to the operator when the robot encounters difficulties or when human expertise is particularly valuable. The versatility of the parameter α enables the realisation of different VA architectures within the same framework. By allowing α to vary continuously across its range, the system can implement DoA regulation, providing fine-grained control over the continuum between manual control and full autonomy. Alternatively, by restricting α to a set of preset discrete values, the system can offer LoA switching.

The most crucial aspect of realising adaptive automation systems is the considered decision variable, i.e, the criteria for optimising the value of α. Chapter 3 demonstrates that the total performance degradation of a robot can be quantified using a single scalar value. Therefore, the robot health metric is used as the state estimator for designing the adaptive automation system. Rewriting the robot health metric equation 3.9 in the continuous form, we have

$$H(t) = \int_{t-T_\mathrm{i}}^{t} (1 - P_{\mathrm{suf}}(\tau))\, \mathrm{d}\tau \tag{4.1}$$

Where T_i is the time span considered for computing $H(t)$. As this definition features an integral, the health of the robot at time t cannot be affected by taking action at the current point of time but rather needs to be precisely steered over the considered time span instead. This is an

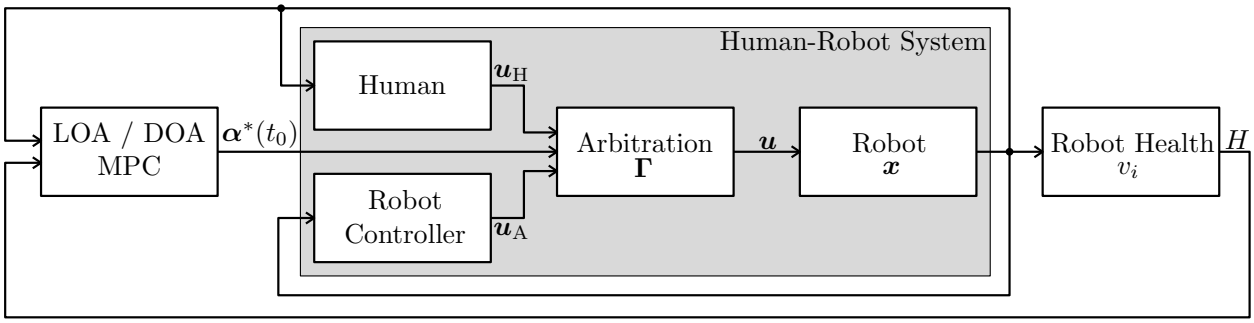


Figure 4.1: Structure of the considered adaptive automation systems

important implication for the design of systems striving to optimise robot health. They cannot only react to the current health status in a feedback control fashion but also need to predict which actions will lead to which health in the future and choose actions based on this prediction. This is also validated by intuition, as a robot's performance degradation at any point in time is also affected by the sequence of actions it has taken. For example, robots may or may not get stuck in areas with uneven terrain based on the specific path they take through it. In situations where a robot doesn't account for the presence of uneven terrain during path planning, it is more likely to get stuck. Therefore, to include a predictive component in the design of the adaptive automation, the problem in this section is formulated as a Model Predictive Control system.

Figure 4.1 depicts the HRT considered in this work: The inputs of the human operator u_H and the robot control algorithm u_A are arbitrated based on the current LoA or DoA to form a common input $u = \Gamma(u_\mathrm{H}, u_\mathrm{A}, \alpha)$ before being applied to the robot. The robot states x are observed by the operator, the robot's controller (i.e., the robot's onboard intelligence) and the MPC; while the operator and the robot's controller use this information to generate their inputs, the MPC uses the state information alongside model information about the system and robot health H to compute the value of α that optimises the health over a finite time horizon with the length T_H.

The total probability of suffering of a robot $P(\textit{suffering})$ is calculated using equation 3.6, given the vitals v_i. The health can then be computed using the equation 4.1. Tthe value of α is then calculated to optimise robot health $H : \mathbb{R}^{N_\mathrm{V}} \to \mathbb{R}_{<0}$. This is accomplished by adjusting the value

of α such that each robot vital $v_t \in V$ computes to a higher value of robot health.

$$\alpha^* = \underset{\alpha}{\operatorname{argmax}}\, H\left(P_{\text{suf}}\left(v_1\left(x\left(\alpha\right),\alpha\right),\ldots\right)\right) \tag{4.2}$$

Please note that throughout this chapter the time-dependencies of each of the variables is omitted for brevity and readability. For the sake of this study, the robot with unicycle dynamics is modelled following (Broggi et al., 2015). This assumption is made for the simulation experiments in this section, which are intended to demonstrate the feasibility of a simple model predict control system on a numerical simulator. In the next section, the need for this assumption is eliminated.

$$\dot{x} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\left(\theta\right) & 0 \\ \sin\left(\theta\right) & 0 \\ 0 & 1 \end{pmatrix} u \tag{4.3}$$

$$u = \Gamma\left(u_{\text{A}}, u_{\text{H}}, \alpha\right) \tag{4.4}$$

$$\Gamma_{\text{pb}}\left(u_{\text{A}}, u_{\text{H}}, \alpha\right) = \begin{pmatrix} \alpha_1 u_{\text{A},1} + \left(1-\alpha_1\right) u_{\text{H},1} \\ \alpha_2 u_{\text{A},2} + \left(1-\alpha_2\right) u_{\text{H},2} \end{pmatrix} \tag{4.5}$$

Here, individual DoAs α_1 and α_2 are considered for the two control inputs, velocity (β) and angular velocity ω, allowing for control authority shifts ranging from MC ($\alpha_i = 0$), over to shared control ($\alpha_i \in (0,1)$) to a FA robot operation ($\alpha_i = 1$) in each of the two dimensions. LoAs with N_{L} levels can be implemented by setting α to a certain value from a countable set $\Lambda = \{\alpha_1, \ldots, \alpha_{N_{\text{L}}}\}$.

Finding the optimal value α^* can be treated both as a parameter optimisation problem and an optimal control problem. In parameter optimisation, an optimal fixed value of α is calculated for a given time horizon to to maximise the value of robot health in that time horizon. This is a special case of optimal control, where α is calculated as $\alpha(t)$, a function of time, for potentially better performance. The choice between these two are practical considerations for each use case (explained in detail in later sections). For now, the realisation of both implementations are explained below. That is - 1) The implementation of a DoA regulation system where the value of α varies over the entire continuum between manual control and full autonomy, and 2) A discretised LoA switching system. The MATLAB (Simulink) block diagrams for the implementations in sections 4.1.1 and 4.1.2 are given in the appendix. The main block diagram of the entire system is given in Figure A.3. The block diagram given in Figure A.2 is the human machine system model, and lastly the block for computing the robot vitals and robot health is given in Figure A.1. Details of the implementation follow.

4.1.1 Parameter optimisation

Based on the above mentioned definition of the system, the following two problems are considered for Parameter Optimisation

Problem 1: *Adaptive Automation for LoA*. Design a LoA regulation module optimising (4.2) s.t. (4.3), (4.5) and models of $u_{\mathrm{H}}(x)$ and $u_{\mathrm{A}}(x)$ considering $\alpha \in \Lambda$.

Problem 2: *Adaptive Automation for DoA*. Design a DoA regulation module optimising (4.2) s.t. (4.3), (4.5) and models of $u_{\mathrm{H}}(x)$ and $u_{\mathrm{A}}(x)$ considering $\alpha \in [0,1]^2$.

Both the above modules predict the evolution of the robot health over a time horizon of time T_p to compute α^*. The optimal time horizon is used for a time horizon $T_s < T_p$.

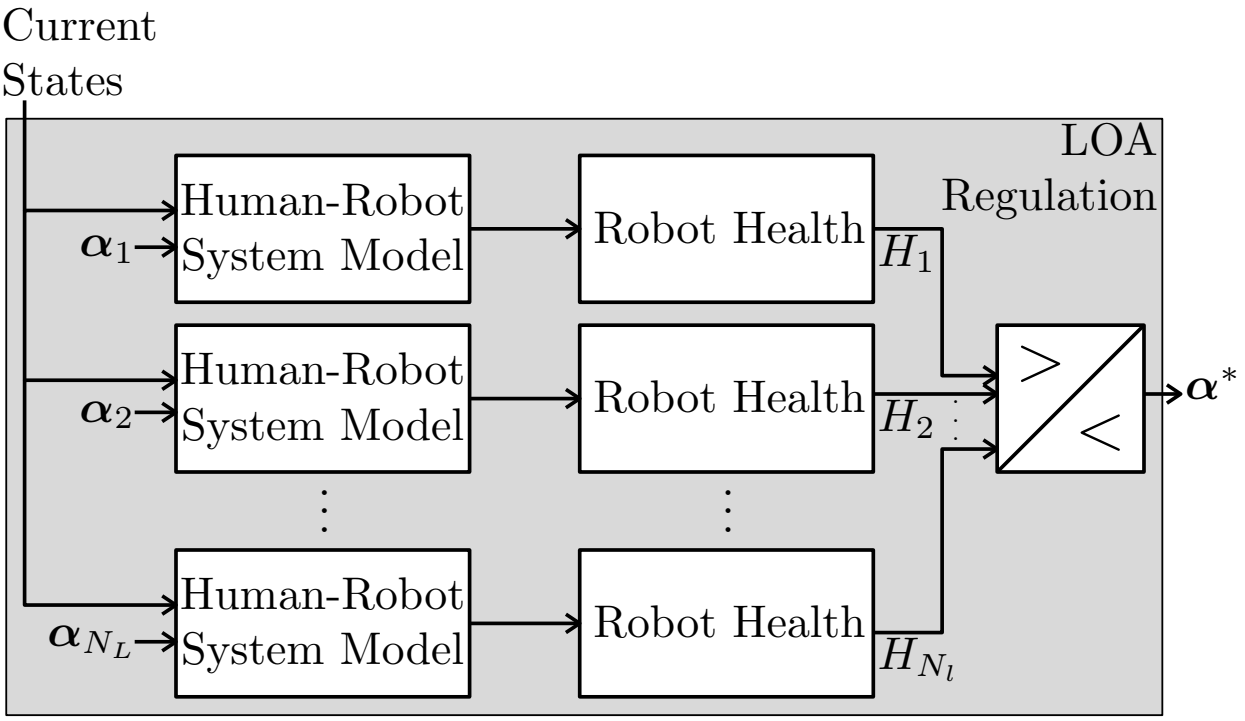


Figure 4.2: LoA Regulation module.

4.1.1.1 LoA Regulation

In LoA regulation, the total number of LoAs is fixed and discrete, typically limited to less than ten levels as suggested by Vagia, Aksel A. Transeth, and Sigurd A. Fjerdingen (2016b), to make the system comprehensible to the human operator. This is because any practical implementation of LoA switching will have a predetermined number of LoAs set by the system design engineer, making the set Λ countable. Therefore, problem 1 reduces to calculating robot health (H_i) N_L times for all $\alpha_i \in \Lambda$ and then comparing the resulting H_i. The α_i leading to the highest H_i is subsequently applied to the human-robot system. Figure 4.2 shows the structure of the resulting Model Predictive Adaptive Automation for the LoA case. Omitting potential model errors (i.e., assuming perfect models), this system is guaranteed to find the globally optimal solution which would yield the maximum value of robot health among the N_L possibilities. As the simulations of each of the human-robot system models are independent of each other, they can easily be executed in parallel on modern desktop CPUs.

4.1.1.2 DoA Regulation

In the DoA case, the optimisation domain $[0,1]^2$ is uncountable. Since an infinite amount of H evaluations would be required, the adaptive automation for LoA switching cannot be extended to this case. As (4.2) is a nonlinear constrained optimisation problem, the Karush-Khun-Tucker conditions (Gordon and Tibshirani, 2012) (KKTC) are necessary conditions for optimality. The derivative of the robot health is considered using the chain rule, as the gradient of the objective function forms the core of the KKTC. As H results from the probability of suffering P_{suf} of the robot, the derivative of P_{suf} with respect to α plays a major role:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} H\left(P_{\text{suf}}\left(v_1\left(x(\alpha),\alpha\right),\ldots,v_{N_{\text{v}}}\left(x(\alpha),\alpha\right)\right)\right) \tag{4.6}$$

$$= \frac{\mathrm{d}}{\mathrm{d}\alpha} \int_{t_0}^{t_0+T_H} 1 - P_{\text{suf}} \mathrm{d}t \tag{4.7}$$

$$= \int_{t_0}^{t_0+T_H} -\frac{\mathrm{d}P_{\text{suf}}}{\mathrm{d}\alpha} \mathrm{d}t \tag{4.8}$$

Due to the sum structure of P_{suf}, the overall derivative is a sum of the derivatives of the individual probabilities of suffering for each vital $P(s|v_i)$:

$$\frac{\mathrm{d}P_{\text{suf}}}{\mathrm{d}\alpha} = \frac{\mathrm{d}}{\mathrm{d}\alpha} \eta \sum_{i=0}^{N_{\text{v}}} P(s|v_i) = \eta \sum_{i=0}^{N_{\text{v}}} \frac{\mathrm{d}P(s|v_i)}{\mathrm{d}\alpha} \tag{4.9}$$

$P(s|v_i)$ are usually nonlinear, hence we have:

$$\frac{\mathrm{d}P(s|v_i)}{\mathrm{d}\alpha} = \frac{\mathrm{d}P(s|v_i)}{\mathrm{d}v_i}\frac{\mathrm{d}v_i}{\mathrm{d}\alpha} \tag{4.10}$$

In the vitals presented in the Chapter 3, α influences the vitals through the states:

$$\frac{\mathrm{d}v_i}{\mathrm{d}\alpha} = \frac{\mathrm{d}v_i}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}\alpha} \tag{4.11}$$

The influence on x stems from the influence on the inputs:

$$\frac{\mathrm{d}x}{\mathrm{d}\alpha} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\left(\int_{t_0}^{t_0+T_1} \dot{x}\mathrm{d}t + x_0\right) = \int_{t_0}^{t_0+T_1} \frac{\mathrm{d}\dot{x}}{\mathrm{d}u}\frac{\mathrm{d}u}{\mathrm{d}\alpha}\mathrm{d}t \tag{4.12}$$

Finally, the inputs result from the arbitration which is parameterised by α:

$$\frac{\mathrm{d}u}{\mathrm{d}\alpha} = \frac{\mathrm{d}\Gamma(u_\mathrm{A}, u_\mathrm{H}, \alpha)}{\mathrm{d}\alpha} \tag{4.13}$$

Thus, (4.6) results from the combination of (4.8) to (4.13). The concrete form of the gradient is application-specific and depends on the used arbitration, robot dynamics and the choice of robot vitals. Four different solutions are possible for Problem 2 depending on the features of the gradient:

4.1.1.2.1 Reduction of the Problem: Considering linear input blending, the following derivatives result with (4.5):

$$\frac{\mathrm{d}\Gamma(u_{\mathrm{A}}, u_{\mathrm{H}}, \alpha)}{\mathrm{d}\alpha} = \begin{pmatrix} u_{\mathrm{A},1} - u_{\mathrm{H},1} & 0 \\ 0 & u_{\mathrm{A},2} - u_{\mathrm{H},2} \end{pmatrix} \tag{4.14}$$

The result in (4.14) is interesting for two reasons. In situations where the robot and the human behave identically ($u_{\mathrm{A,i}} = u_{\mathrm{H,i}}$), the value of $\frac{\mathrm{d}\Gamma(u_{\mathrm{A}},u_{\mathrm{H}},\alpha)}{\mathrm{d}\alpha}$ becomes 0. This essentially indicates that when the navigation commands from the robot and humans are identical, the choice of α for DoA regulation is arbitrary, and does not influence the outcome. Second, since the value of $\frac{\mathrm{d}\Gamma(u_{\mathrm{A}},u_{\mathrm{H}},\alpha)}{\mathrm{d}\alpha}$ does not depend on the value of the blending coefficients α, the best solution will always be at the extreme ends of α, i.e., 0 or 1. In simpler words, when human and robot inputs are blended in a straightforward manner, the robot health does not depend on complicated calculations of both these inputs, and the best control strategy will always be to set the LoA to either full autonomy $\alpha = 1$ or full manual control $\alpha = 0$.

4.1.1.2.2 Analytical solution: If the overall gradient does depend on α and the KKTC can be solved for α, candidate points for the optimal solution can be achieved this way. If more than one candidate point results, the optimum can be determined using a function evaluation similar to the procedure in Subsection 4.1.1.1. If feasible, this is a very computationally efficient solution.

4.1.1.2.3 Iterative, gradient-based solution If the gradient can be computed but solving the KKTC for α is hard or infeasible, gradient-based iterative solvers can be used. This way, locally optimal solutions may result.

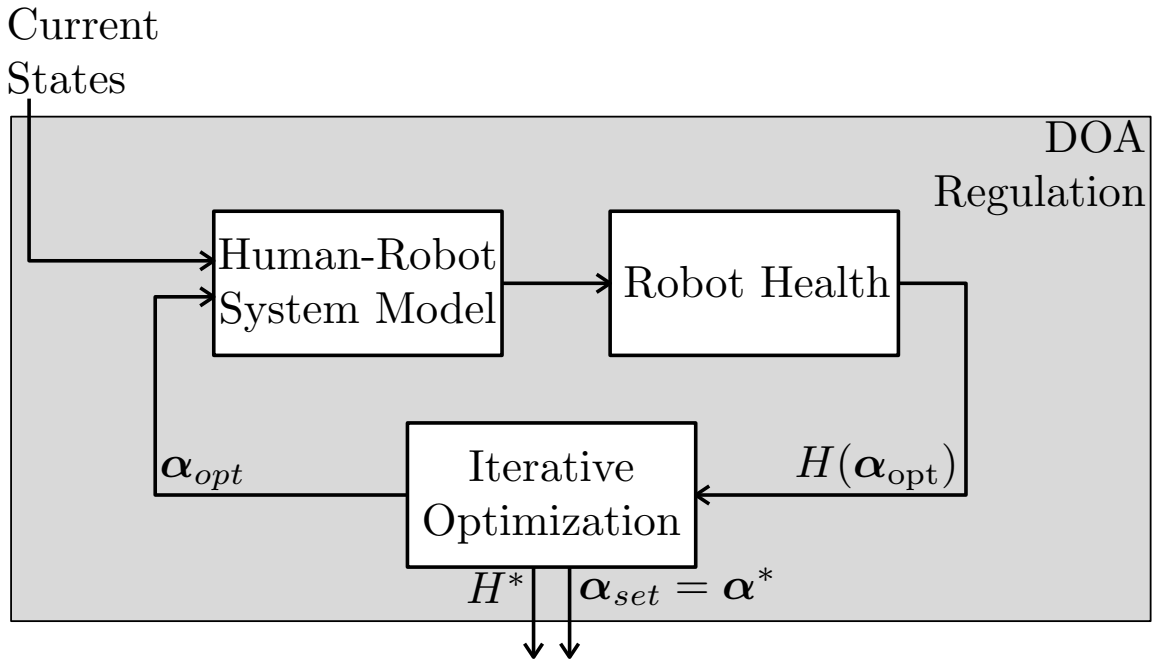


Figure 4.3: DoA regulation module

4.1.1.2.4 Iterative, gradient-free solution If the gradient cannot be computed e.g. in the case of piece wise defined functions, gradient-free iterative solvers can be applied. While they may also generate locally optimal solutions and are often less computationally efficient compared to the gradient-based approaches, they are the most generally applicable solution to Problem 2 as they do not rely on a specific form of the optimisation problem.

Therefore for parameter optimisation, the gradient free solution is examined. The resulting Model Predictive Adaptive Automation for the DoA case is shown in Fig 4.3

4.1.1.3 Experimental Evaluation

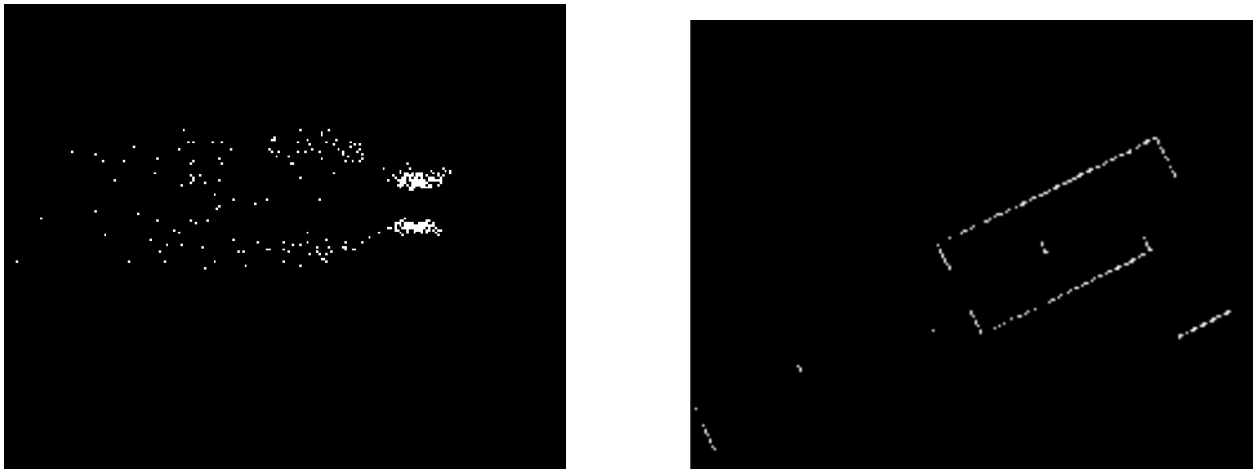
A simulation study was conducted to evaluate the proposed Model Predictive Adaptive Automation systems for DoA and LoA regulation, and their ability to effectively manage the robot's level of autonomy under varying circumstances. The key aspect examined was how these systems could dynamically adjust the autonomy ratio α to compensate for situations where either the remote op-

erator's control inputs or the robot's autonomous navigation capabilities experience performance degradation. Specifically, it is hypothesized that the adaptive automation system can effectively detect situations where the operator's control inputs are restricted and adjust α to rely more on the robot's onboard navigation control inputs. Conversely, it is also hypothesized that in situations where the robot's onboard navigation controller inputs are restricted, the adaptive automation system can vary α to rely more on the operator's control inputs.

The intuition behind examining the system's response when operator control inputs are restricted is based on multiple factors. As explained in the previous chapter, the robot uses its laser scanner to map the surrounding area and calculate velocity commands. The presence of noise in the laser scan (as shown in Figure 4.4) distorts the robot's cost map. The robot's navigation planner uses the cost map to detect obstacles, identify free spaces, and plan a path. Laser noise causes the robot to incorrectly register empty spaces as obstacles, as illustrated in Figure 4.5. Consequently, to mitigate the perceived risk of collision, the navigation planner reduces the amplitude of autonomous navigation commands. Similar situations may occur in the presence of obstacles or due to conservative motion constraints in the navigation stack. However, manual navigation commands provided by the operator can compensate for this. If the operator is confident about navigating without risk, they can provide velocity commands with higher amplitude.

Conversely, there are situations where the operator's velocity commands may have low amplitude or be entirely absent. Common causes include latency or input signal distortion during remote teleoperation, the operator being distracted by external supervisors, or leaving the robot unattended for short periods of time. Therefore, scenarios in which the velocity commands provided by both the operator and the navigation planner are restricted in different parts of the map are examined. The ability of the adaptive automation system to dynamically vary α across these situations to ensure effective control and operation is examined.

For the experiments carried out in this study, a highly abstracted version of the experimental

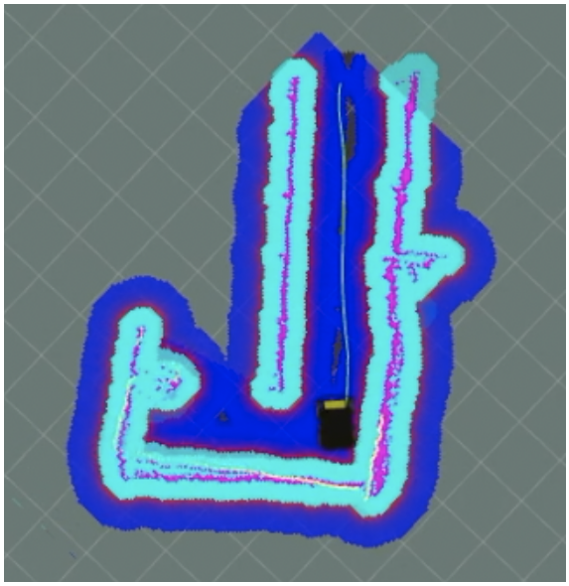


(a) Laser scan of an area with noise

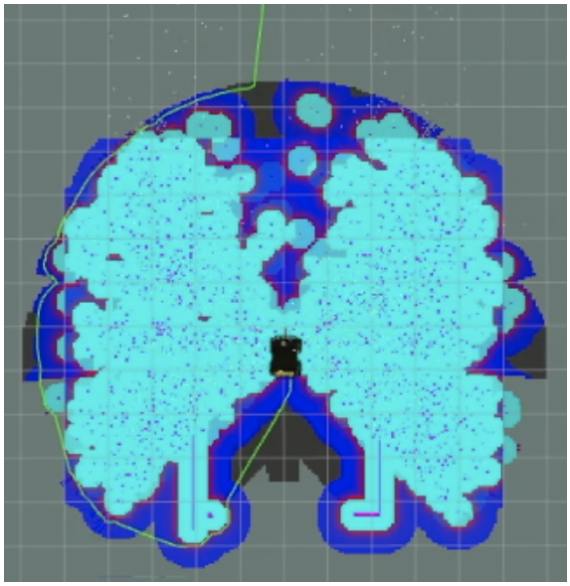
(b) Laser scan of an area without noise

Figure 4.4: Laser scanner visualised with and without laser noise

setup in the previous chapter is used. The experiment was a mobile robot navigation task in the scenario depicted in Fig 4.6. In the figure, the initial robot position is marked with a triangle to indicate its heading. The goal point is marked with a large cross and the intermediate way-points are labelled 1 to 6. During the task runtime, the robot traverses four different areas with adverse condition, denoted by different colours in the figure. Each of the adverse conditions limits the range and magnitude of control velocities provided by either the robot's navigation stack or the robot operator, necessitating the adaptive automation to change the value of α to accommodate for these conditions. 1) In the grey area, noise is introduced into the laser scanner data. The noise causes the robot's navigation stack to reduce the range of command velocities provided to 20% of the magnitude during normal operation. 2) In the pink area, the laser noise is reduced and the navigation stack's control inputs increase to 60% of the normal operation. Here, the operator getting distracted and limiting their range of command velocities to 20% of normal operation is simulated. 3) In the orange area, the robot faces obstacles and uneven terrain. While the robot's navigation controller is unaware of the performance-degrading factors, the operator can recognise these factors and thus adds a conservative intermediate waypoint between waypoints 4 and 5 to avoid the orange area, as indicated by the dashed line. Lastly, while traversing the blue area, an internal fault in the robot's onboard navigation stack which causes the velocity input commands



(a) Navigation Cost map before Laser Noise



(b) Navigation Cost map during Laser Noise

Figure 4.5: Effect of performance degradation on robot cost map

to be reduced to 20% is simulated. All the other areas coloured in white were not affected by performance degradation.

For simplicity and to minimise the number of confounding variables, it is assumed that the adaptive automation system has complete information about the environment. This assumption is made to examine the feasibility of the system without imposing practical constraints and confounding variables. However, the next section of this chapter relaxes this assumption and motivates it with realistic considerations. Lastly, a critical design choice is made that as the robot's health decreases, greater preference should be given to manual control by the human operator. When the robot's health is high, its onboard navigation is capable of functioning autonomously, and therefore, the robot should be autonomous.

Evidence from the previous chapter showed that the robot health metric is robust to adding and removing any vitals from calculating the probability of suffering. For this experiment, only use three of the robot vitals presented in the previous chapter, which are most important for this study are used. The study to uses only three vitals to avoid the requirement of complex physics

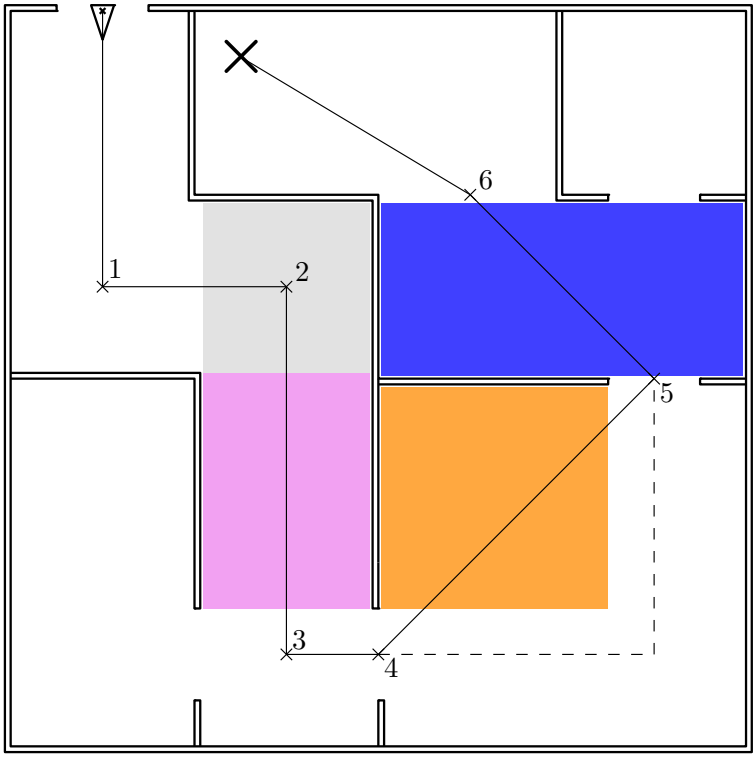


Figure 4.6: Map of the arena for the navigation task

modelling, which is not the focus of this thesis. Therefore, the vector of vitals used was $V = \{\dot{d}_g, \dot{a}_z, \sigma^2_{noise}\}$. $\dot{d}_g$, the rate of change of distance from the navigational goal is computed as:

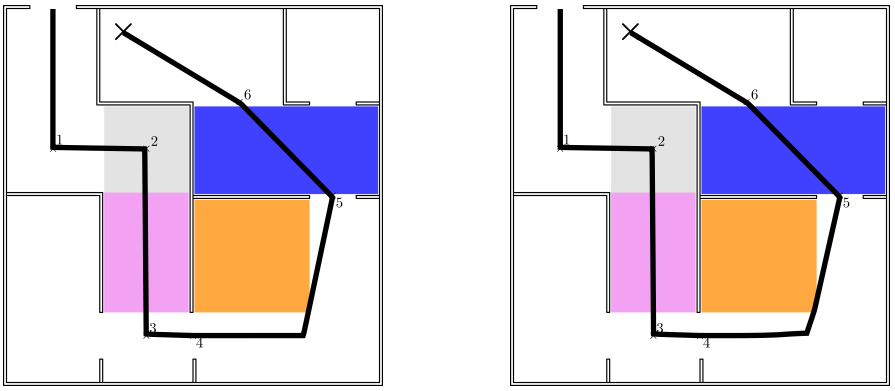
$$\dot{d}_{\mathrm{g}} = \frac{1}{\left\|d_{\mathrm{g}}\right\|_2} d_{\mathrm{g}}^{\top} \begin{pmatrix} \beta\cos(\theta) \\ \beta\sin(\theta) \end{pmatrix} \text{ with } d_{\mathrm{g}} = \begin{pmatrix} x_{\mathrm{g,i}} - x \\ y_{\mathrm{g,i}} - y \end{pmatrix}. \tag{4.15}$$

Here, $\left(x_{\mathrm{g,i}}, y_{\mathrm{g,i}}\right)$ denotes the coordinates of the currently active way-point. The probabilities of suffering and parameter mapping were fine-tuned heuristically to the scenario considered in this study. The entire system was implemented in Simulink as shown in Figure A.3. The robot dynamics were modelled as in (4.3), human behaviour models for both the simulation model, as well as the prediction model, were based on extensive tele-operation experiments conducted as part of the ARCHES project ((Wedler et al., 2021)): When told to pursue a goal via way-points as described above, the operators usually first turned the robot until the correct heading

had been reached and then commanded velocities subsequently. This behaviour was implemented by switching between a proportional controller for θ and a proportional controller for d_g using the inputs ω and β. Apart from the different types of performance degradation, the model of the automation was identical to the one of the human as this structure was also used successfully in (Wedler et al., 2021). Based on these models, $\alpha = \alpha_1 = \alpha_2$ is set in (4.5) as neither the operator nor the automation influenced both inputs simultaneously. All results were achieved with $T_p = 5\,\mathrm{s}$ and $T_s = 0.25\,\mathrm{s}$.

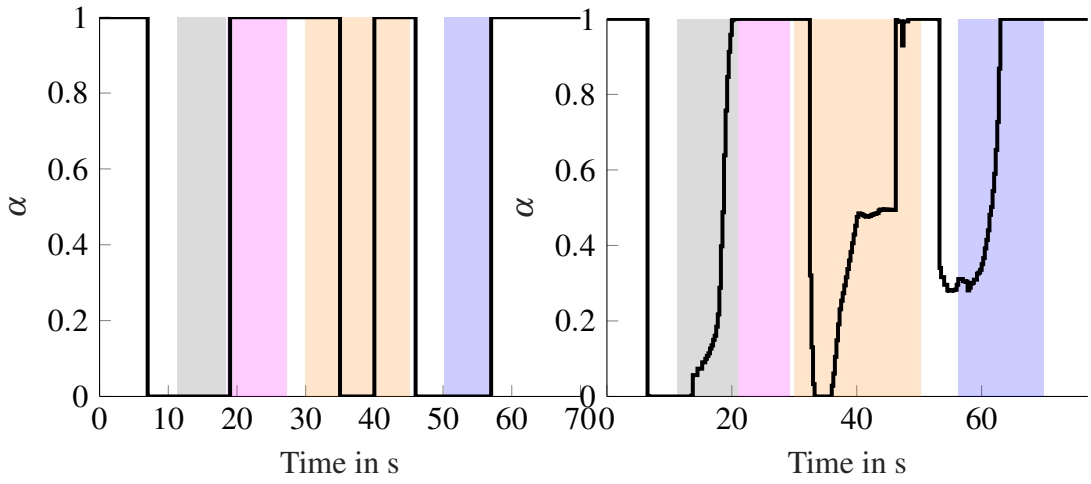
4.1.1.3.1 Results - Adaptive Automation for LoA switching The proposed Adaptive Automation system was tested on the above described experimental setup and the value of alpha was then plotted for the runtime. Fixed values for the autonomy ratio α resulted in an LoA switching system with two LoAs - full manual control ($\alpha = 0$) or full autonomy ($\alpha = 1$). This LoA switching approach is denoted as $\Lambda = 0,1$. The resulting robot trajectory for this discrete LoA case is shown in Figure 4.7a, with the corresponding evolution of α over time shown in Figure 4.8a. As seen in Figure 4.7a, the combined efforts of the human operator, the AAS, and the robot's navigation stack successfully guided the robot through the white, grey, pink, and blue regions to ultimately reach the goal position, while also avoiding the problematic orange area. In regions where the performance predictions for the human and autonomous control were comparable, the AAS devolved full control authority to the robot's navigation system ($\alpha = 1$). This can be observed in the white areas at the start and end of the trajectory, as well as immediately after traversing the pink region. However, when facing adverse environmental conditions in the grey, pink, and blue zones, the AAS quickly assigned control authority to whichever entity (human or autonomy) was least impaired. This decision was dynamically updated as the robot moved through the different regions. Particularly, between way points 4 and 5, the AAS performed a control authority transfer to leverage the human operator's skills in avoiding the uneven terrain of the orange area, while also taking advantage of the autonomous system's ability to chart an optimal path directly towards way point 5 once the uneven terrain was successfully avoided. Overall, the LoA switching approach

demonstrated the AAS's capability to seamlessly transition control authority between the human and autonomous navigation, enhancing the robustness and performance of the human-robot team under various adverse operating conditions.



(a) Trajectory for LoA case. (b) Trajectory for DoA case.

Figure 4.7: Trajectory of the robot's navigation during the experiment with Parameter optimisation



(a) Resulting LoAs. (b) Resulting DoAs.

Figure 4.8: Simulation results for LoAs and DoAs with Parameter Optimisation. Background colours indicate the traversal of areas the correspondingly coloured in the maps. The orange area is avoided by the robot, but shown here for context

4.1.1.3.2 Results - Adaptive Automation for DoA switching The simulation study next examined the case where the adaptive automation system could regulate the autonomy ratio α as a continuous value between 0 and 1. DoA regulation was implemented by using a gradient free

pattern search algorithm (MathWorks, 2024) in the Model Predictive Adaptive Automation. The path taken by the robot and the value of α were plotted for the runtime in Figures 4.7b and 4.8b respectively. Similar to the LoA switching system, the AAS is able to use DoA regulation to reach the goal successfully while avoiding the orange area (i.e, uneven terrain). Similar to the discrete LoA switching case, the DoA regulation approach successfully coordinated the overall Human-Robot Team to reach the goal position while minimising the effects of performance degradation. In periods where the predicted performance of the operator and the robot was comparable, such as the start, end and immediately after the pink area, the DoA system correctly assigned full control authority to the robot's onboard navigation by increasing the α to 1. While facing performance degradation where the robot health on manual control was predicted to be higher than autonomy, the robot gradually adjusted α to leverage the relative strength of manual control. In the grey and blue regions, where autonomous performance was more severely degraded, the DoA system slightly favoured human control by reducing α. The main difference observed in Figure 4.8b compared to the LoA switching system is that instead of binary switches, the DoA system facilitated smooth transitions in autonomy when moving between areas with different types of performance degradation. During the traverse from way-point 4 to 5, a sharing of control spanning the whole range of DoAs occurs to trade off between avoiding the orange area and pursuing way-point 5. Input blending is executed smoothly across the entire range $\Lambda = [0, 1]$, thereby combining the strengths of both autonomous navigation and manual control.

4.1.2 Optimal Control

Parameter optimisation finds the best value for α in a finite time horizon. Alternatively, in optimal control finds the optimal trajectory sequence of the values of α over a prediction horizon, improving the performance of a dynamic system. While this is computationally more expensive, it can lead to better performance and stability of the system, and can also handle multi-variable systems

and complex constraints more effectively. In this section, a realisation of an AAS for DoA and LoA regulation with Optimal Control is presented. The problem formulation, the calculations involved, the process of implementing the system and the results from experiments are all described. The experimental methodology from the previous section is largely followed here as well.

Figure 4.1 depicts the system considered for this work. The AAS uses the system's state, model information and robot health H to compute the LoA or DoA sequence $\alpha^*(t)$ that optimises H over a finite time horizon with the length T_H. Due to the receding horizon principle of model predictive control, only the first LoA or DoA of the sequence $\alpha(t_0)$ computed for the current time step t_0 is applied to the arbitration Γ. The remaining part of the sequence (for $t \in [t_0 + T_\mathrm{c}, T_\mathrm{H}]$) is updated in the following cycles with a cycle time of T_c before being applied to the system. In line with best practices in control systems literature, the predicted input to the system α is considered to be constant during a cycle $t \in [t_0, t_0 + T_\mathrm{c}]$. Throughout this section the time-dependencies are mostly omitted for better readability; however, they are explicitly showed if they are significant for the expression.

The goal of the adjustment of LoA or DoA is to optimise robot health over T_H. The objective function can be formulated as follows:

$$\alpha^*(t) = \underset{\alpha(t)}{\operatorname{argmax}}\, H(t_0 + T_\mathrm{H}), \quad t \in [t_0, t_0 + T_\mathrm{H}] \tag{4.16}$$

As α remains constant during each prediction cycle, (4.16) is the problem of finding an optimal sequence of $N_H = T_\mathrm{H} T_\mathrm{c}^{-1}$ LoAs or DoAs. In contrast to the previous section, (4.16) thus constitutes an optimal control problem as $\alpha(t)$ is treated as a sequence rather than a system parameter fixed for T_H. Therefore, parameter optimisation is a special case of this optimal control problem if $T_\mathrm{C} = T_\mathrm{H}$ is set.

The two following problems are considered in order to design the AAS for LoA or DoA regulation. While both aim to optimise the objective function (4.16), they differ in the arguments

$\alpha(t)$ available for doing so:

Problem 1: LoA Switching Design an AAS for LoA switching optimising (4.16) such that the model information about the HRT described above considering $\alpha(t) \in \mathscr{L}$ with $\mathscr{L}$ denoting the countable set of sequences (N_{H}-tuples) that can be created from the set of considered LoAs Λ.

Problem 2: DoA regulation Design a DoA-MPC module optimising (4.16) s.t. the model information about the HRT described above considering $\alpha(t) \in \mathscr{D}$ with $\mathscr{D}$ denoting the uncountable set of all sequences (N_{H}-tuples) that can be created from the set of considered DoAs $[0,1]$.

As $\Lambda \subset [0,1]$ always holds, Problem 1 is always a special case of Problem 2. However, due to its structure, Problem 1 allows for solution concepts that are infeasible for Problem 2. Thus, Problem 1 is considered on its own, and as a special case of Problem 2.

4.1.2.1 LoA Regulation

As the set of possible LoAs Λ is countable, the number of possible sequences of length T_{H}, or N_{H}-tuples, is $N_{\mathrm{S}} = |\Lambda|^{N_{\mathrm{H}}}$. Here, $|\Lambda|$ denotes the cardinality of Λ. Since N_{S} is finite, Problem 1 allows for the evaluation of the performance of all possible solutions and the subsequent selection of the best one. Figure 4.2 visualises this approach where N_{S} model instances of the HRT are initialised with the currently observed states and evaluated for $t \in [t_0, t_0 + T_{\mathrm{H}}]$ using a fixed sequence $\alpha_i(t),\ i \in \{1,...,N_{\mathrm{S}}\}$. Subsequently, the resulting robot health H_i can be computed and the optimal solution $\alpha_j(t)$ can be determined by comparing the resulting robot health values to select H_j, the one with the highest value. This approach is of interest as it is guaranteed to find the globally optimal solution and can easily be implemented in a parallel fashion since the time-consuming evaluations of the HRT models are mutually independent.

4.1.2.2 DoA Regulation

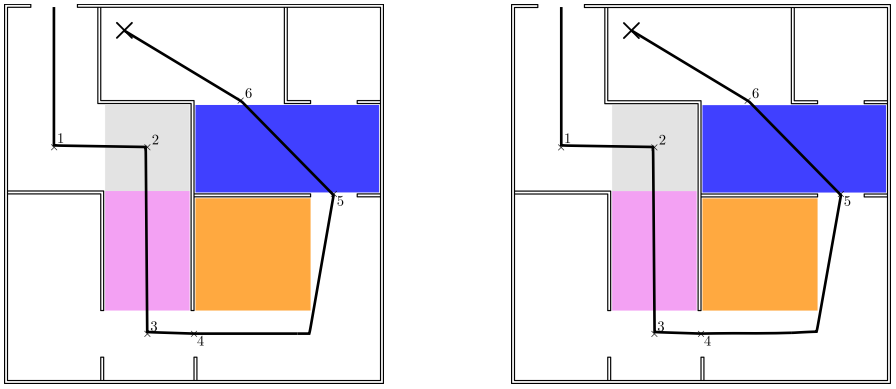
Considering Problem 2, the set of possible DoAs is uncountable, hence the number of possible sequences is infinite. Thus, the system proposed in Subsection 4.1.2.1 is no longer applicable as an infinite amount of model evaluations would be required.

Following most of the control system literature, this issue is addressed using an iterative optimisation algorithm. The resulting structure is depicted in figure 4.3. The human robot system model is again initialised with the currently observed system states in each prediction cycle, but instead of evaluating all possible sequences, only the sequences $\alpha_{\text{opt}}(t)$ selected by the optimisation algorithm are evaluated. The resulting robot health values are used to iteratively update the currently best solution until convergence is reached. As the solution of the previous prediction cycle is, apart from a time shift of T_{c}, likely similar to the optimal solution for the current cycle, the optimisation is always initialised with the solution of the previous iteration shifted by T_{c} to improve computational efficiency.

Due to the usage of an iterative optimisation, no general statements about optimality or even convergence can be made; however, in most practical cases at least a locally optimal solution can be achieved. As $\Lambda \subset [0,1]$ always holds, all possible solutions for a LoA switching AAS are solution candidates for a DoA regulation AAS, too. Hence, the potential solution performance of the latter is always at least equal to the one of the former.

4.1.2.3 Experimental Evaluation

In order to assess the adaptive automation presented above, a proof of concept study was developed in simulation. It was studied whether the AAS was able to effectively manage the robot's LoA/DoA. It was studied whether the AAS was able to effectively manage the robot's LoA/DoA. As this study focuses on evaluating the general functionality of the system, perfect model and map



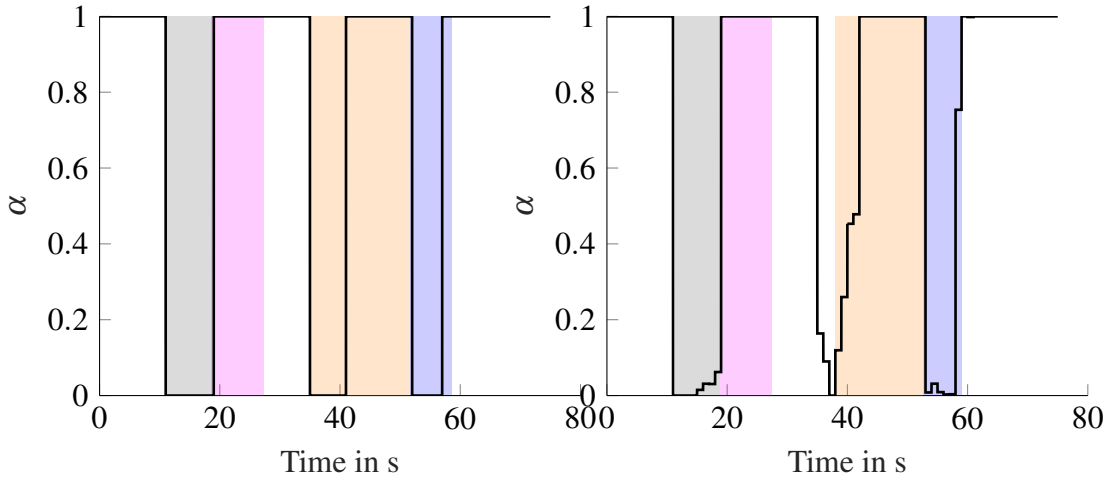
(a) Trajectory for LoA Case using optimal control

(b) Trajectory for DoA case using optimal control

Figure 4.9: Trajectory of the robot's navigation during the experiment using optimal control

knowledge were assumed. The whole system is considered in continuous time. This experimental setup was very similar to one described in section 4.1.1.3, with a few minor differences in the severity of the performance degrading factors. These were made to slightly modify the response of the AAS. In the grey area, laser noise limits the range of commanded velocity to 10% of the maximum velocity. In the pink area the operator getting distracted is simulated and the range of velocities is limited to 30% of the maximum velocity. The laser scanners are still impaired by noise in this area, but the navigation stack able to use 60% of the nominal velocity input range. There is no change in the orange area, the area has uneven terrain and the operator has to recognise the uneven terrain and give the robot an alternate way point. Lastly, in the blue range the robot's onboard navigation malfunctions, resulting in an input range of 30% of the maximum velocity. The robot vitals used, the robot and operator model are the same for this experiment and are shown in figures A.3, A.2 and A.1.

4.1.2.3.1 Results - Adaptive Automation for LoA switching For the LoA switching case, the set of admissible LoAs were $\Lambda = \{0, 1\}$ i.e., either the human or the operator is in charge of commanding the robot. Figure 4.9a shows that the resulting path of the robot. This system is able to successfully arbitrate control between the operator and the robot's navigation, and pass all the way-



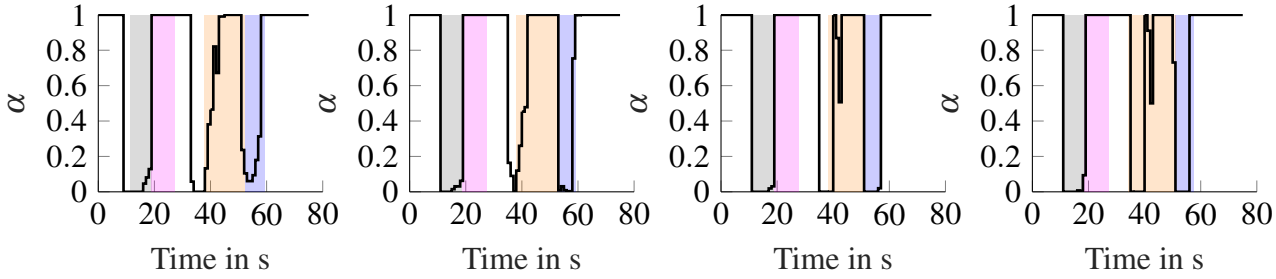
(a) Resulting LoAs. (b) Resulting DoAs.

Figure 4.10: Simulation results for LoAs and DoAs for the optimal control case. Background colours indicate the traverse of the correspondingly coloured areas in the respective maps shown in figure 4.9. The orange area is avoided by the robot, but shown here for context

points before reaching the goal state. The LoA evolution illustrated in Figure 4.10a demonstrates the effectiveness of the proposed AAS. Initially, control is successfully transferred to the human while traversing the grey area, where the robot's navigation is impaired. Once the pink area is reached, where the operator is distracted and the RC encounters less noise, control is handed back to the robot's onboard navigation stack. To avoid the orange area, the operator receives control authority at around 35s. However, instead of completely following the overly conservative dashed path of the human, control is shifted to the navigation stack to barely avoid the bumpy orange area while maximising the $\dot{d}_g$ vital. In the blue area, control is correctly handed to the human during the malfunction of the robot's navigation as it is recognised by the $\dot{d}_g$ vital and then handed back to the robot's onboard navigation stack to reach the goal. The AAS for LoA switching provides a reasonable arbitration of control authority, giving it either to the operator or to the robot based on who is predicted to have the best capability of handling the task thus maximising robot health and consequentially maximising the overall performance of the human robot system.

4.1.2.3.2 Results - Adaptive Automation for DoA switching In this scenario, the models are provided analytically, allowing for the use of gradient-based optimization algorithms. However, flexibility in terms of the model type is prioritized, as real-world implementations planned for future applications may involve gray-box or even black-box models without gradient information. Consequently, the gradient-free pattern search algorithm (Audet and Dennis Jr, 2002) is used for the implementation of DoA regulation.

The robot trajectory achieved with this system is depicted in Figure 4.9b. Again, all way-points are precisely passed and the goal state is reached. Figure 4.10b shows the evolution of the DoA chosen by the AAS. Similar to LoA switching case, control authority is given to the operator in the grey and blue areas as well as to avoid the orange area, while the robot controller is used in the pink area. In contrast to the results of LoA switching, the AAS for DoA regulation uses collaborative shared control at approximately 38s to smoothly shift control back to the automation. This allows for an even more efficient health trajectory around the orange area as the robot controller is able to start steering the robot towards way-point 5 even before the orange area is fully passed. A corresponding trend is observed Figure 4.10b, where the health starts increasing close to the 40 second mark.



(a) DoA-MPC using $T_H = 4$s and $T_c = 4$s.

(b) DoA-MPC using $T_H = 4$s and $T_c = 1$s.

(c) DoA-MPC using $T_H = 15$s and $T_c = 1$s.

(d) DoA-MPC using $T_H = 75$s and $T_c = 1$s.

Figure 4.11: Comparison of DoA regulation results for multiple T_H. Background colours indicate the traversal of the corresponding coloured areas.

Apart from the collaborative time span, slight artefacts can be observed in the grey and blue area. The reason for this is the rather coarse prediction cycle-time T_c of 1s: The transition,

e.g. from the grey to the pink area, would have happened exactly within one cycle, thus leading to a DoA-only optimal for half of the cycle time. To avoid this, the AAS gives a slight amount of control to the RC before the transition, leading to a slightly slower traverse (and thus slightly lower robot health), synchronising the transition of the grey to the pink area with the beginning of a new cycle time. This leads to an optimal DoA for the whole cycle T_c by sacrificing only a small amount of performance in the previous cycles. These artefacts can be removed by decreasing T_c, however they showcase the capability of the AAS to use even small effects to optimise robot health and thus performance.

In addition to T_c, the considered time horizon T_H plays an important role, too, as it scales the amount of information available for planning the optimal DoA. To analyse its impact, simulations with varying T_H are conducted and the results are depicted in Figure 4.11. Figure 4.11a shows the special case of a parameter optimisation similar to the previous section, Figure 4.11b and Figure 4.11c result from setting $T_H = 4$s and $T_H = 15$s, respectively, and Figure 4.11d shows the result if T_H covers the full scenario. Despite their significantly different T_H, Figure 4.11c and Figure 4.11d achieve very similar results. Figure 4.11a shows a more elaborate use of shared control, which stems from an averaging effect: As only one DoA can be optimised for the time horizon, an intermediate DoA is chosen as a compromise if the environment changes within the time horizon.

Table 4.1: Comparison of the average robot health. A higher value means less performance degradation.

Configuration	**Average health per second**
Fully automated	0.7583
Fully manual	0.9380
$T_H = 4$s, $T_c = 4s$	0.9860
$T_H = 4$s, $T_c = 1s$	0.9878
$T_H = 15$s, $T_c = 1s$	0.9892
$T_H = 75$s, $T_c = 1s$	0.9895

Considering the overall robot health averaged over the duration of the task, the configura-

tions of Figure 4.11 achieve the values given in Table 4.1: All of the DoA-MPC configurations significantly outperform a fully manual ($\alpha(t) = 0\ \forall t$) or fully automated ($\alpha(t) = 1\ \forall t$) operation of the robot. Increasing T_{H} and thus increasing the information available for computing the DoA improves the achieved robot health monotonously with the configuration considering the whole duration performing best. Nevertheless, the comparable DoA evolution of $T_{\mathrm{H}} = 15\mathrm{s}$ also leads to comparable robot health. Despite the significantly different DoA sequence shown in Figure 4.11a and Figure 4.11b, their achieved robot health is still far better than in the fully manual and fully automated case, thus indicating that DoA regulation Adaptive Automation Systems even of relatively low computational complexity might provide a significant benefit for applications.

4.1.3 Key Takeaways and Insights

Section 4.1 integrated insights from two papers (Christian Alexander Braun, Ramesh, Rothfuß, et al., 2023; Christian Alexander Braun, Ramesh, Rothfuss, et al., 2023) on Model Predictive Control (MPC) for optimising robot health, comparing parameter optimisation with discrete Levels of Automation (LoA) and optimal control with a continuous Degree of Automation (DoA). The differences between LoA switching and DoA switching are summarised in Table 4.2. Parameter optimisation with LoA-MPC is simpler to implement and computationally less demanding, showing effectiveness in stable environments by achieving high average robot health. However, it lacks flexibility in dynamic conditions, leading to sub-optimal performance. Conversely, DoA-MPC offers granular control, enabling precise adjustments to automation levels, and demonstrates superior performance in dynamic environments with higher average robot health. The trade-offs involve balancing computational complexity and flexibility; parameter optimisation suits predictable scenarios, while optimal control excels in unpredictable settings but demands sophisticated algorithms and more computational resources. Simulation results confirm these findings, with DoA-MPC consistently outperforming LoA-MPC in dynamic environments.

The control sharing problem studied in equations 4.5 and 4.13 has been implemented with linear input blending for the sake of simplicity. While linear systems are simpler to implement, they allow only proportional transitions which are not always preferable. Alternatively non linear blending of inputs can be considered for different control responses in equation 4.5. This would result in $\Gamma_{\text{pb}}(u_{\text{A}}, u_{\text{H}}, \alpha)$ implemented as $f(\alpha)u_{\text{A}} + (1 - f(\alpha)u_{\text{H}})$, where f can be a non linear function. The main advantage of this is that it allows context aware control, where one input needs to dominate in certain scenarios depending on the environment, operator preferences or the system dynamics. For example, if f is implemented as a non-linear sigmoidal function of both α and signal strength s during remote operation, it can leverage the smooth transition properties of non-linear blending to prioritize autonomous control (high u_{A}) when the signal strength s is low or when α favours automation. The non-linearity of f ensures that as the signal strength improves or α shifts to favour human control, the blending transitions smoothly from autonomous control to human input (high u_{H}). This avoids abrupt changes in control allocation, thereby maintaining system stability even in highly dynamic environments.

Table 4.2: Comparison of LoA-MPC and DoA-MPC

Aspect	**LoA-MPC**	**DoA-MPC**
Implementation Complexity	Simpler to implement	Requires model of system dynamics to implement
Computational Complexity	Computationally less expensive	Computationally demanding
Flexibility	Low, only fixed options for LoAs	High, offers granular control adjustments
Use Cases	Ideal for environments with low frequency of performance degradation	Better for complex or rapidly changing scenarios

4.2 Limitations of Control System Based Approaches

Section 4.1 proposed realisations of VA systems that regulate LoA and DoA at the control level. Assuming unicycle dynamics, these methods provide a foundational approach for developing

model predictive control systems that optimise the robot's automation based on its health. These sections offer valuable insights into designing control systems using a robot health metric. However, implementing this Adaptive Automation System on commercially available off-the-shelf robots can be very time-consuming. This is because commercial robot morphologies vary widely and cannot always be modelled using unicycle dynamics. For example, a robot with bicycle dynamics not only considers position and heading angle but also includes an additional Degree of Freedom for the angle of the front wheel. In such cases, calculations would need to account for four degrees of freedom, increasing the time and complexity of adapting the system. Other morphologies, like snake robots, have fundamentally different control dynamics, necessitating extensive calculations and hyper-parameter tuning to integrate robot health into the control layer. Therefore, while these adaptive automation approaches offer significant benefits, it's important to consider more versatile implementation strategies that can accommodate a wide range of robot types without requiring fundamental redesigns of the control system.

4.3 Model Predictive Mixed Initiative Level of Autonomy Switching

While the adaptive automation systems discussed in section 4.1 offer DoA and LoA regulation based on robot health, their implementation across diverse robot morphologies poses challenges. This realisation led to the adaptation of the Model Predictive Controller (MPC) approach for easier integration with off-the-shelf robotic platforms. Inspired by the state of the art VA systems proposed by Chiou, Hawes, and Rustam Stolkin (2021), in this section, a novel MPC is presented for LoA switching that predicts the optimal LoA for each tessellation of the robot's path plan. The implementation of a Mixed Initiative Variable Autonomy (MI VA) system, which uses this Model Predictive Controller to trigger LoA switches, is described, with key assumptions and considera-

tions outlined to facilitate future research. Furthermore, this system is compared to the state-of-the-art Mixed-Initiative LoA switching systems, and its ability to reduce cognitive workload while enhancing user experience and task performance is evaluated.

While most early VA systems solely relied on adaptive automation systems that automatically adjust their LoA (e.g. (Parasuraman, Bahri, et al., 1992; Kaber and Endsley, 1997)), Mixed-Initiative (MI) systems provide the operator with assistance in setting the LoA through adaptive automation while still allowing for the flexibility to deviate from the chosen LoA if desired (Jiang and Arkin, 2015). Multiple studies found benefits in terms of user preference and performance when using MI systems as opposed to pure adaptive automations (Li et al., 2013; H. Ruff et al., 2018; Calhoun, 2021; Chiou, Hawes, and Rustam Stolkin, 2021). Nevertheless, the cooperation of the adaptive automation and the operator in setting the appropriate LoA leads to new challenges like the *conflict for control* (Chiou, Hawes, and Rustam Stolkin, 2021), which occurs if the operator and the adaptive automation disagree on which LoA to choose. Additionally, it is still an open research question if the performance of current adaptive automations and MI systems can be improved further.

To this end, the next section describes the implementation of the MPC for the given use case. Subsequently, the experimental methodology used to compare the system to the state of the art is described, and the results and insights gathered are presented

4.3.1 Design of the Model Predictive Controller for LoA switching

This section outlines the problem formulation, describes the required assumptions, and motivates them with relevant literature. The main contribution here is the description of a data-driven method to estimate the expected performance degradation due to adversities in the environment by combining techniques from automated path planning and computer vision.The Robot Vitals and Robot

Health Framework (Ramesh, Rustam Stolkin, and Chiou, 2022) is used as a performance criterion to map the state of the environment to the optimal LoA during runtime. The implementation of this system is described, along with a discussion on how the approach can be generalized.

4.3.1.1 Problem Formulation

Consider a variable autonomy robot with LoAs $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$, carrying out a navigation task in an extreme environment. Here each value α is one possible LoA realisation from $\mathscr{A}$, the continuum between Manual Control and Full Autonomy. This environment is filled with adversities or performance-degrading factors that may degrade the robot's performance during its runtime. This work is scoped to field repairable and non-terminal (Carlson and Murphy, 2005) performance degradation that the robot faces during runtime, i.e., any performance degradation faced by the robot can be fixed by switching the robot to Manual control by a remote operator or by triggering pre-programmed recovery behaviours.

Once the robot is given a navigation goal, the robot's automated planner creates a feasible path plan P. This path plan can be decomposed into a series of n tessellations $P = \{s_0, s_1, \ldots, s_n\}$, where each tessellation s_i is an area in the map through which the robot has to locomote to reach the goal. The set of all possible tessellations is denoted by S.It is assumed that the tessellation sizes are chosen so that the effect of performance degradation that the robot experiences in each tessellation is independent of the other. Implementing tessellation decomposition online its outside the scope of this thesis. However, readers are referred to studies by Patle et al. (2019) and Sánchez-Ibáñez, Pérez-del-Pulgar, and García-Cerezo (2021) on the topic for a thorough review of path plan decomposition techniques.

During task execution, a robot can encounter a variety of performance-degrading factors. Each factor denoted by f is an element belonging to the set of all possible performance degrading

factors $\mathscr{F}$. In each tessellation, a robot can encounter a set of m performance degrading factors $F = \{f_1, f_2, \ldots, f_m\}$, $m \geq 0$. Alternatively, $F = \emptyset$ indicates that there is no performance degradation present in the given tessellation. Let $I : S \mapsto \mathscr{P}(\mathscr{F})$ represent a function that can scan tessellation i and return a set of all performance degrading factors present in it, such that $I(s_i) = F_i,\ F_i \subseteq \mathscr{F}$. Techniques to detect information about the environment are being researched extensively in the existing literature. One approach is to use image semantics and computer vision techniques (Ruan et al., 2022; Zhu et al., 2017) to detect performance degrading factors. Alternatively, air-ground collaborative teams can be used where a drone looks ahead on the map, detects performance degrading factors, and communicates it to the ground robot (I. D. Miller et al., 2022). Therefore, it is assumed that in any tessellation s_i, the robot is able to look ahead and calculate $I(s_{i+1}) = F_{i+1}$ using the onboard camera and other sensors.

Let function $H_{exp} : \mathscr{F} \times \mathscr{A} \mapsto R$ denote a function that calculates the expected Robot Health (as described in Chapter 3) for a given tessellation given the performance degrading factors and the robot's level of autonomy. This gives $H_{exp}(F_i, \alpha_i) = H_i$. Therefore, the level of autonomy most appropriate for a tessellation is one that maximises the robot's health:

$$\alpha_i^* = \arg\max_{\alpha \in A} H_{\exp}(F_i, \alpha_i) \tag{4.17}$$

The MI VA system designed using the proposed MPC is represented in figure 4.12. Every time a robot enters a new tessellation, the state information $I(s_{i+1}) = F_{i+1}$ is calculated and provided to MPC system. The MPC AI then chooses the appropriate LoA for the system and sends that command to the Control Mixer. The Control mixer then sends velocity commands to the robot accordingly from either the human (through joystick input), or from the robot controller responsible for autonomous navigation. In the unlikely scenario of two different LoAs (i.e, α) having the same expected health, the LoA remains unchanged. Additionally, the operator at any given time has the capability to trigger switches, and changing the LoA set by the MPC system.

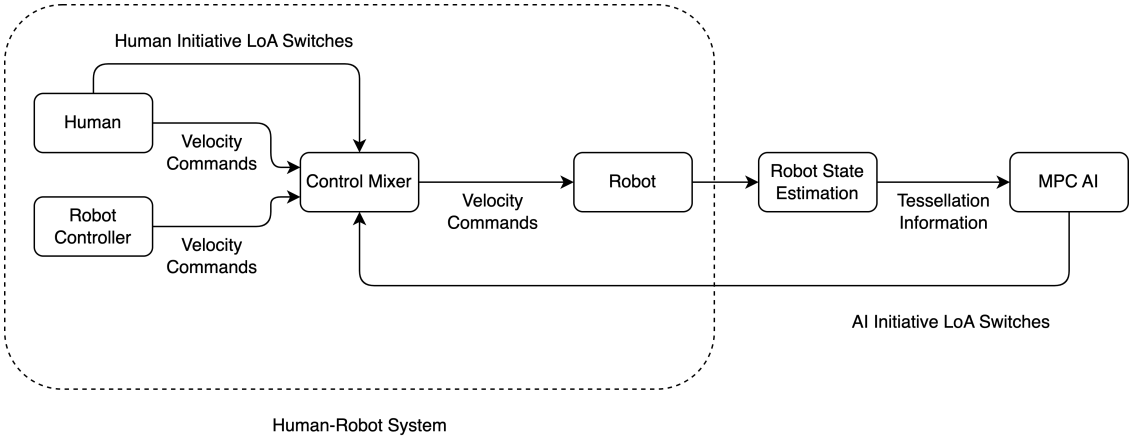


Figure 4.12: Block Diagram of the Mixed Initiative Variable Autonomy System that uses the Model Predictive Controller AI

4.3.1.2 Implementation

For clear demonstration, an example of an indoor scenario is introduced to adapt the model. Based on the environment, the tessellations are segmented as shown in Figure 4.13. Using the shortest feasible path P from the start to the goal (shown as a dotted white line) the arena was split up into tessellations (marked A, B, C, D). The shape of each tessellation was decided based on the kind of performance degrading factor present in that area. The letter A denotes areas where no performance degradation is present (i.e., $F = \emptyset$). Laser noise was present in B, and uneven terrain was present in C. In D, both laser noise and uneven terrain affected task performance.

We use data from chapter 3 to calculate the expected robot health for each type of performance degrading factor. That is, robot health from previous simulations of a mobile robot navigation through an area with high laser scanner noise, is used to estimate the H_{exp} in the tessellation marked B. Similarly, H_{exp} for tessellations C and D were calculated based on the robot health values for uneven terrain without and with laser noise (respectively). These values were calculated for each level of autonomy and stored in a lookup table.

Therefore, H_{exp} for each ordered set (F, α) was substituted as the average robot health

Figure 4.13: The tessellations (A to D) and the feasible robot path plan P (dotted line) for the navigation task

when navigating through the performance degrading factor F using LoA α. The algorithm then uses (4.17) in real time to decide the optimal LoA for the tessellation. Similarity measurement (Qi et al., 2016), i.e., matching each new tessellation with the appropriate value of H_{exp}, is hard coded for this implementation. However, this can be addressed in future work. More generalisable approaches in the future could adapt scene recognition and traversability estimation techniques (Chavez-Garcia et al., 2018; Sevastopoulos, Oikonomou, and Konstantopoulos, 2019; Matsuzaki, Masuzawa, and Miura, 2022) to calculate the expected health for any tessellation.

4.3.2 Experimental Methodology

Figure 4.14: Navigation Task Arena (L to R): 1) Empty, 2) With obstacles and uneven terrain, 3) 2D map with start and finish points marked, and locations where performance degradation is introduced.

The experiment investigates the effect of two MI VA system designs on task performance, overall user experience, and cognitive workload. This system (i.e., the MPC system) is compared to the state of the art Expert Guided Mixed Initative Control System Chiou, Hawes, and Rustam Stolkin (2021) (EMICS). Each participant in this study used both EMICS and MPC to carry out a mobile robot navigation task in a simulated arena filled with performance degrading factors. Simultaneously, they carried out a secondary mental rotation task during the experiment. The

secondary task was used to induce additional cognitive workload on the operator, and simulate the high stress nature of robotic missions in extreme environments.

The robot used an MI VA system with 2 LoAs - 1) Waypoint-based autonomous navigation and 2) Manual Control by an operator using a joystick. For waypoint navigation, the Husky robot used the standard ROS Navigation stack. The operator could dynamically switch between the LoAs using buttons on their joystick. Two experimental conditions were tested - EMICS and MPC MI VA System. It was hypothesized that, in comparison with EMICS, the MPC MI VA system would yield a better user experience, improved performance in the navigation task, greater cognitive availability for a secondary task, and lower operator cognitive workload.

4.3.3 Experiment Design

The primary task was a mobile robot navigation task based on an arena used in previous experiments (Ramesh, Rustam Stolkin, and Chiou, 2022). The design of this arena was based on environment typically encountered in Urban Search and Rescue tasks. A 2D scan of the arena was first created for robot navigation planning. After the scan was created, performance degrading factors like unforeseen obstacles, uneven terrain, and laser noise were added to the arena to degrade autonomous navigation performance (see Figure 4.14). To simulate the uncertainty and dynamic situations encountered during navigation tasks in extreme environments, the presence and location of these performance degrading factors were neither incorporated into the robot's planner, nor were they communicated to the operator before the task.

A 3D object mental rotation task was used for the secondary task (Ganis and Kievit, 2016). This is a visuospatial task added to increase the overall cognitive demand of the task and recreate high workload environments for robot operators (e.g., the need to multitask in the remote inspection or robot-assisted disaster response). Here participants were successively presented with two 3D

objects and asked if they are the same or different. An example of two sets of 3D objects are shown in Figure 4.15 Objects that were the same but rotated were classified as the same, but mirrored objects were classified as different. Like the experiments carried out in (Ramesh, Englund, et al., 2023), the secondary task was carried out by the operator simultaneously alongside the primary task. While carrying out the primary task, the participant simultaneously looks at the secondary task and says yes or no (i.e. yes-same, no-different) and the experimenter presses the corresponding keys to log the data. The control unit used by each participant is shown in Figure 4.16.

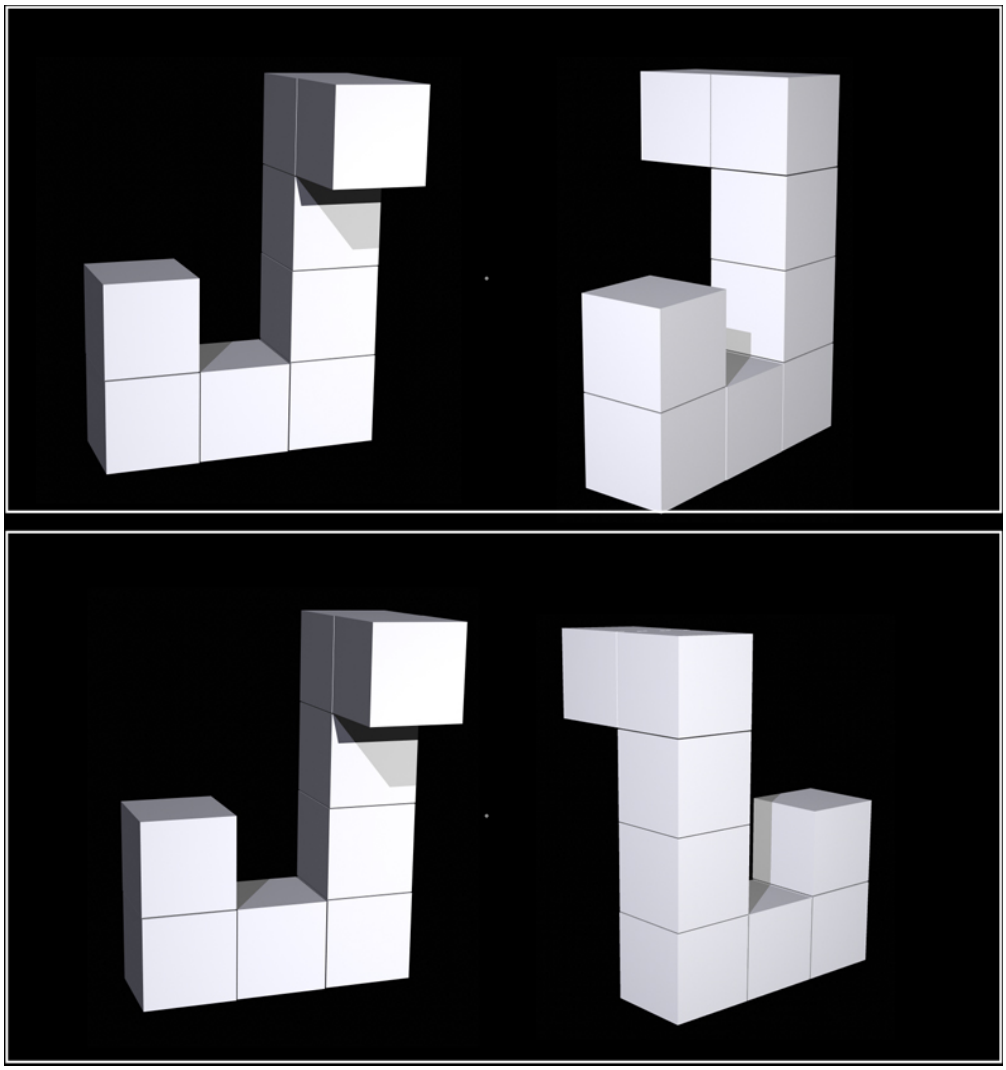


Figure 4.15: Examples of two sets of 3D Objects presented to participants in the secondary task. The objects on the top are different from each other and the bottom are the same.

4.3.4 Experimental Procedure

A total of 15 test subjects participated in the experiments, with usable data from 14 subjects. Data from one participant had to be discarded because of a technical error during the experiment. The experiment had a within-subjects study, where all participants carried out both conditions. To minimise the learning and fatigue effects, the order of the conditions was counterbalanced for half the participants. On arrival, participants first filled out a background information questionnaire to indicate if they had prior experience playing games, operating heavy machinery, or using AI tools for work. Then, participants were introduced to basic robot navigation and LoA switching on a specialised training arena (Ramesh, Englund, et al., 2023; Chiou, Talha, and Rustam Stolkin, 2019). They used this arena to practice manually controlling the robot and were asked to demonstrate minimum proficiency before moving on to the task. This was done to ensure confounding factors due to varying skill levels were minimised. Next, a different training arena was loaded with additional obstacles and sensor noise. Here participants were introduced to a couple of scenarios autonomous navigation is degraded (e.g., due to unforeseen obstacles and laser noise), how they impact the navigation, and how LoA may improve navigation in such situations. It was explained to the participants that the scenarios demonstrated were not exhaustive and that there may be other factors in the experiment which can degrade robot performance. To prevent priming participants with knowledge of how either of the MI VA systems worked, they were not provided training to use the EMICS and MPC systems.

Participants were then introduced to the secondary task. After explaining the task to them, they were given time to practice the task, following which their baseline performance on the secondary task was recorded. Before starting the first experimental condition, the 2D map of the arena was shown to the participants. The start and end points of the primary (navigation) task were shown to the participants. They were informed that the waypoints for navigation were predetermined and set by the experimenter automatically for both experimental conditions. Each participant was told

explicitly to focus on the primary task, prioritise robot safety and minimise the risk of robot failure. They were then asked to focus on the secondary task only when they had the time.

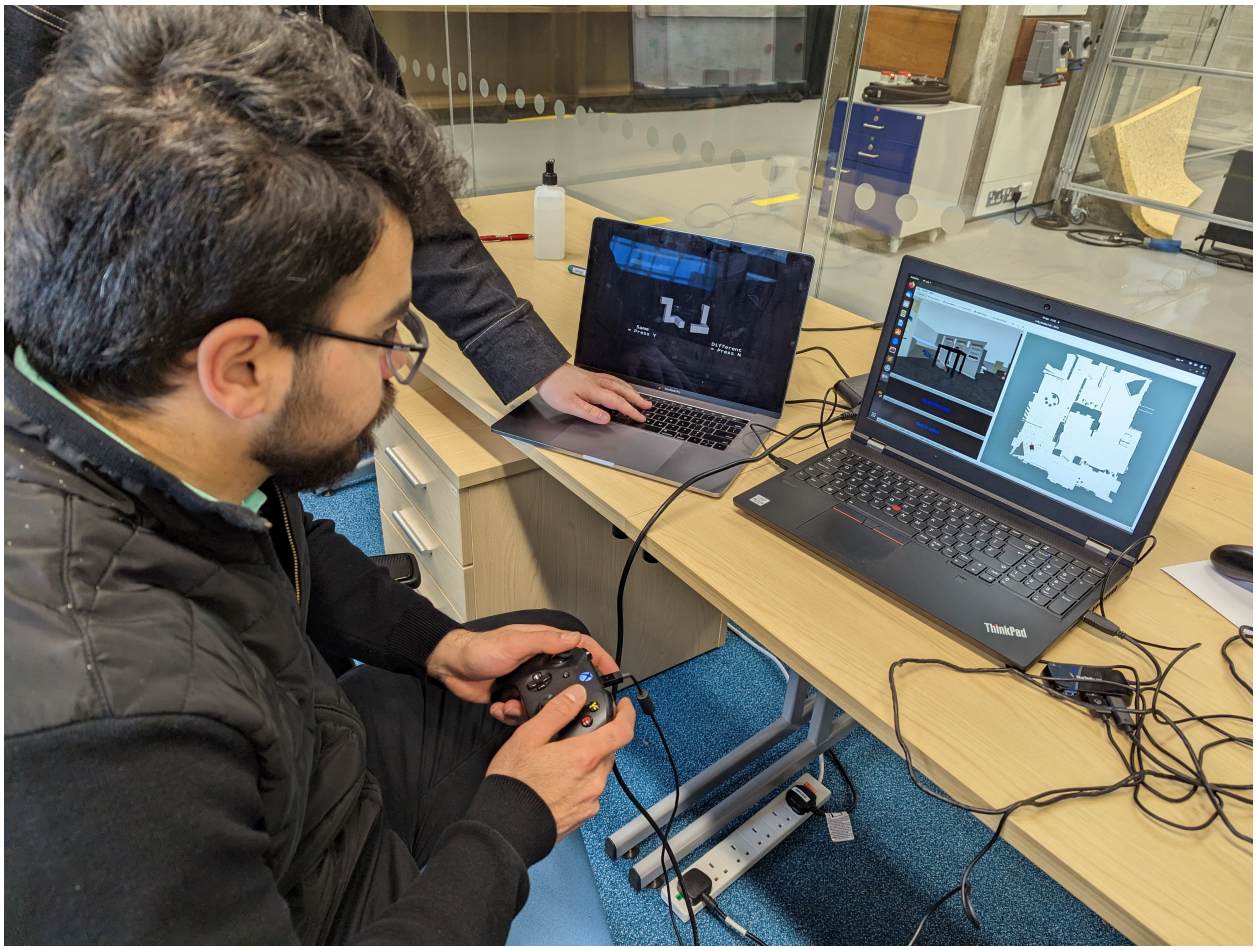


Figure 4.16: The control unit used for the experiment. The participant controls the robot using a joystick and views it on the screen. The secondary task is shown on another laptop, and is monitored by the experimenter

Lastly, before starting the first experimental conditions, each participant was told that an AI would be assisting them with LoA switching. It was made clear that both the participant and the AI had equal authority and the option of switching LoAs, i.e., that none of the agents was the ultimate authority or ultimately in charge of LoA switching. Participants were also told that the AI might sometimes trigger LoA switches the operator disagreed with, i.e., the operator may have to override the AI's decision. In such situations, participants were asked to verbally state 'Conflict' so that the experimenter can note it down. Independently, the experimenter also observed and

counted the number of conflicts, i.e., instance of operators overriding AI initiated LoA changes with 2 seconds. This was done to compensate for situations where the operator may forget to verbally acknowledge conflicts. After finishing each experimental condition, the participants filled out a raw NASA-TLX rating questionnaire to indicate the overall cognitive workload experienced during each experimental condition. Lastly, before ending the experiment, each participant was asked, "If you were to carry out this navigation task in a new environment using the same robot, which AI would you prefer for assistance with LoA switching?".

4.3.5 Results

Results from experiments conducted on 14 participants were computed and summarised in Table 4.3. The participants were within the age group of 23-35, with the approximately 65% male and 35% female participants. Around 50% of the participants reported no prior experience with operating remote heavy machinery, or with using AI and robots for work on a regular basis. 46% of the participants indicated they frequently play video games involving driving, flight simulation and third person shooters, RPGs (roleplaying games), and sports. The paired Wilcoxon's Test was used to check for statistical differences between both conditions in each metric, as they were non-parametric matched pairs of data and had a sample size of N=14. The null Hypothesis H_o was that no statistical difference exists between the two experimental conditions.

In the primary task (i.e., robot navigation), no statistically significant differences were observed in the total runtime. The EMICS system and MPC system varied significantly ($p < 0.05$) in how they utilized the LoAs. While 64% of the task was spent in autonomy using the EMICS system, the MPC used autonomy for only 55% of the total runtime. Consequently, the mean robot health for the MPC system during runtime was 3.591, and during the use of autonomy was 3.791; which was higher than the EMICS system which had an average health of 3.506 and 3.329 during autonomy. Conversely, the average response time in the secondary task was significantly

Table 4.3: Summary of Statistical Analysis

Primary Task	**EMICS**	**MPC**	**Paired Wilcoxon Test, Two Tailed**
Total Runtime (Seconds)	Mean = 170.907, SD = 19.590	Mean = 177.788, SD = 18.759	Z- Value = -1.1614, W Value = 34, p = 0.246
Average Health	Mean = 3.506, SD = 0.492	Mean = 3.591, SD = 0.416	Z- Value = -0.596, W Value = 43, p = 0.549
Number of LoA Switches	Mean = 13.714, SD = 3.604	Mean = 11.786, SD = 4.979	Z- Value = -0.816, W Value = 39.5, p = 0.410
Average Health during Manual Control	Mean = 3.329, SD = 0.516	Mean = 3.283, SD = 0.455	Z- Value = -0.534, W Value = 44, p = 0.596
Average Health during Autonomy	Mean = 3.579, SD = 0.562	Mean = 3.791, SD = 0.398	Z- Value = -1.099, W Value = 35, p = 0.271
Percentage of Runtime the Robot was Manually Controlled	Mean = 36.307, SD = 12.868	Mean = 44.572, SD = 8.361	Z- Value = -2.417, W Value = 14, p = 0.016*
Percentage of Runtime the Robot was Autonomous	Mean = 63.693, SD = 12.874	Mean = 55.430, SD = 8.416	Z- Value = -2.47, W Value = 14, p = 0.015*
Conflicts for Control Reported by Experimenter	Mean = 6.429, SD = 1.910	Mean = 2.214, SD = 1.424	Z- Value = -3.296, W Value = 0, p = 0.001*
Conflicts for Control Reported by Operator	Mean = 3.857, SD = 2.381	Mean = 1.714, SD = 1.489	Z- Value = -2.620, W Value = 8, p = 0.009*
Secondary Task	**EMICS**	**MPC**	**Paired Wilcoxon Test, Two Tailed**
Average Response Time (Seconds)	Mean = 7.923, SD = 3.348218	Mean = 10.443, SD = 6.099	Z- Value = -2.215, W Value = 21, p = 0.026*
Total Answered	Mean = 30.067, SD = 14.405	Mean = 27.2, SD = 14.189	Z- Value = -1.3497, W Value = 31, p = 0.177
Accuracy %	Mean = 88.158, SD = 10.088	Mean = 89.169, SD = 7.121	Z- Value = -0.313, W Value = 35, p = 0.756
NASA-TLX	**EMICS**	**MPC**	**Paired Wilcoxon Test, Two Tailed**
Total Workload	Mean = 50, SD = 14.821	Mean = 46.667, SD = 12.605	Z- Value = -1.224, W Value = 33, p = 0.222

*$p<0.05$

($p < 0.05$) better when EMICS was used (EMICS = 7.923 Seconds, MPC = 10.443 Seconds). While the mean number of objects detected in EMICS (30.067) was higher than MPC (27.2), the overall accuracy in both experimental conditions was relatively the same (88.158% for EMICS and 89.169 for MPC). Conflicts for control reported by participants were 55% less when they used the MPC system than the EMICS system. This was confirmed by the experimenter's observations, where in comparison to EMICS, 65% fewer conflicts were observed for the MPC system.

Although the perceived cognitive workload of the MPC system was lower on average, no significant differences were observed in the NASA-TLX scores of both conditions. Out of 14 participants, 9 said they would prefer using the MPC system if they were to carry out a navigation task in a new environment. Among the others, 4 felt no difference, and 1 participant preferred EMICS over MPC.

Most participants reported a better overall user experience while using the MPC system and felt the conflicts of control were the main reason for the difference between the systems. Upon further enquiry, participants reported that they trusted the LoA switches of the MPC system more, contributing to its overall ease-of-use. For example, one of the participants said - "It felt the MPC system made LoA switches that I would have". Another participant said they "found MPC more reliable, as it intimated me about the problems before I even saw them, whereas EMICS was useless to the point that I'd rather switch manually".

4.3.6 Discussion

The study carried out in this section demonstrates that information about a robot's runtime performance degradation in the form of Robot Vitals and Robot Health described in 3 can be used to realise a predictive LoA switching AI agent for Mixed-Initiative Variable Autonomy systems. State-of-the-art approaches (Chiou, Hawes, and Rustam Stolkin, 2021) to designing such systems

highlight they are prone to conflicts for control between the AI and Operator. While negotiation (Rothfuß et al., 2022) and operator state estimation (Panagopoulos et al., 2022) provide generalisable methods to counter conflicts for control, a predictive approach is proposed here to tackle some of the fundamental reasons conflicts happen. This predictive approach improves upon the performance criteria used by the EMICS system, by using robot health as a performance criterion. The robot health uses a set of multiple vitals, which combine different aspects of robot performance degradation to give more depth to the metric. Hence, the MPC system is able to understand the environment better and suggest LoA switches that the operator is more inclined to agree with.

Experimental evidence suggests that the proposed predictive approach to switching is comparable to state-of-the-art in total runtime, causes fewer conflicts for control, and improves the overall user experience. The experiment also shows that the MPC system uses autonomy more prudently. The MPC system uses teleoperation in situations where the aggregate risk of robot failure is higher and uses autonomy in situations where the effect of performance degradation on the robot is low. Additionally users reported fewer conflicts for control and made fewer switches during the runtime. This suggests that participants accepted AI initiated LoA switches more while using the MPC system. This indicates that the MPC system enables effective servicing of robots and improves overall robot safety during its runtime.

The aim of using a secondary task was to optimise towards the 'sweet spot' of cognitive workload by preventing low workload situations which can lead to operator complacency. Operators took longer to respond to their secondary tasks when using the MPC system. This is likely due to the additional time spent in manual control for the MPC system. Since operators were explicitly told to prioritise robot safety, extended periods of manual control added more cognitive demand to the operator thereby impacting the time taken to respond to the mental rotation task. Subsequent studies can help alleviate this demand by allowing operators to trigger pre-programmed recovery behaviours (Rigter, Lacerda, and Hawes, 2020) when they want to focus on a secondary task.

The MPC system is a predictive system and can minimise the conflicts for control and risk of robot failure by pre-emptively making LoA switches. On the contrary, reactive systems like EMICS, which monitor robot performance online, help mitigate unforeseen problems. Reactive approaches to LoA switching can also easily encode expert knowledge and ground truth information during missions. Therefore, in the future, AI agents that combine the strengths of both predictive and reactive LoA switching capabilities for Mixed-Initiative Variable Autonomy systems can be developed. One possible limitation of this study is that smaller tessellation sizes may result in multiple LoA changes in quick succession. This may cause more conflicts for control and higher workloads. However, the use of DoA as shown in the previous section can address this, by enabling input blending and smoother transitions between different levels of operator control over the robot's actions.

Anecdotally, one of the participants who preferred EMICS over MPC stated - 'While MPC is Objectively better at LoA switching, EMICS makes trivial mistakes which are easier to identify and correct. However, MPC may make smarter mistakes that are harder to detect. Hence, I would rather use EMICS'. As the MPC MI VA was more complex (i.e., predictive using an aggregate metric) than the EMICS (straight forward goal-directed error metric and reactive approach), actions of MPC MI VA were harder to anticipate. This remark indicates that transparency and explainability should be key design concerns for MI robotic systems.

4.4 Conclusion

This chapter explored how the Robot Vitals and Robot Health Framework can be used to design variable autonomy systems. Mainly, the effectiveness of the robot health metric as a state estimation framework for variable autonomy robots was explored. Three different proposed realisations were presented that implement variable autonomy using the robot health framework: 1) Model

predictive degree of automation (DoA) and level of autonomy (LoA) regulation through parameter optimisation, 2) Model predictive DoA and LoA regulation using optimal control, and 3) A model predictive controller for mixed-initiative LoA switching.

For 1 and 2, an analytical formulation to adjust the DoA based on system state was derived. These were treated as both an optimal control and a parameter optimisation problem for a highly abstracted navigation task in a numerical simulator. Experiments show the proposed realisations are able to regulate the robot's LoA and DoA effectively, so operator support is blended to enable a robot to overcome adversities during navigation. This control system design can serve as an effective foundation for designing adaptive automation systems or shared control systems that can vary the degree of operator input blending required to overcome adversities. The main advantage of such systems is they benefit from the properties of the Robot Vitals and Robot Health Framework, like the inherent robustness, scalability and the ability to encode expert knowledge. However, adapting this framework to off-the-shelf robots may require extra considerations and careful tuning of robot parameters.

Section 4.3 presented a novel approach to realise a Model Predictive Controller AI for Level of Autonomy switching and implemented a Mixed Initiative Variable Autonomy system using this controller. For each tessellation on the robot's path plan the AI determined the optimal Level of Autonomy based on the expected robot health (Ramesh, Rustam Stolkin, and Chiou, 2022), a metric to estimate performance degradation due to environmental adversities. On a mobile robot navigation task, the proposed approach yielded performance comparable to state-of-the-art Mixed Initiative Variable Autonomy Systems, with fewer conflicts for control between the AI and the operator and a better user experience. Experiments on this system also showed that the AI switched to autonomous navigation in situations with a low risk of robot failure. When there was a high probability of robot failure, i.e., its autonomous capabilities being compromised due to performance-degrading factors, the AI switched to manual control by a human operator. Therefore, the proposed approach enables autonomous robots to seek assistance from human operators to

prevent or mitigate failure, thereby improving robot safety during task execution. Future work on this system can focus on integrating computer vision and machine learning algorithms that estimate the expected robot health in real time.

As the total number of robots in an MRS increases, it becomes crucial to manage the operator's cognitive demand effectively Kolling et al. (2015). This chapter provides evidence that each robot can predict situations where it is likely to face issues with autonomous operation and then request operator support to help it with task execution, thereby preventing cognitive overload and promoting a good user experience. This self-supervision capability can reduce the effort required to manage and operate each robot in a multi-robot system, laying the groundwork for a more efficient triage process. The next step is to understand how an operator can easily visualise information from an MRS where each robot can communicate when it needs assistance (Dahiya et al., 2022). Therefore, the next chapter focuses on streamlining human interaction with VA MRS to enable quick and effective robot triage and servicing. The impact of communicating information about the robot's performance degradation to the operator through multi-modal sensory cues added to the interface is studied. Instead of using the MPC-MI for each robot in the MRS, Human Initiative Level of Autonomy switching is used to minimise confounding factors and the evidence gathered is presented.

Chapter Five

Human-Initiative Variable Autonomy based on Robot Health for multiple robot control

This chapter addresses the problem of facilitating quick and effective triage decision-making for servicing robots during task execution. As discussed in Chapter 2, there is a notable gap in the literature regarding game-inspired variable autonomy robot control interfaces. Specifically, interfaces that incorporate multi-modal sensory cues based on robot performance degradation are relatively under-explored. These cues can potentially ease the operator's cognitive workload and enhance task performance. This is important, particularly for the Human Initiative Level of Autonomy Switching systems (see Chapter 2), which can impose a high workload on the operator. Using the Robot Vitals and Robot Health framework for robust state estimation, this chapter explores how information about robot performance degradation can be integrated into the robot control interface through sensory cues and interface design. Based on best practices from existing literature, the aim in this chapter is to build a triage-centric control interface for Variable Autonomy Multi-Robot Systems, which facilitates quick robot Triage and servicing while promoting trust and transparency. We plan to submit the work presented in Section 5.2, to the ACM THRI journal.

Designing interfaces for Human-Robot Teaming (HRT) and Human-Robot Interaction (HRI)

presents unique challenges compared to traditional Human-Computer Interaction (Murphy and Tadokoro, 2019). These interfaces must bridge the gap between complex robotic systems and human operators, enabling effective control and monitoring in dynamic physical environments. To address these challenges, the guidelines proposed by Murphy and Tadokoro (2019), which emphasise the importance of presenting information in a way that aligns with natural human intuition and the task environment, are adapted. Additionally, Chapter 2 highlighted the benefits of integrating colour-coded visual cues and audio cues into the overall interaction paradigm. While Visual cues can quickly convey important information, Audio cues effectively capture the operator's attention, making them particularly useful for critical events that require intervention. For example, traffic lights use three colours to convey complex traffic management decisions, reducing cognitive load, while alert sounds quickly indicate emergencies (e.g., ambulances and alarms).

In the context of this work on VA robot navigation tasks, sensory cues can communicate critical information such as the robot's current Level of Autonomy (LoA), the severity of performance degradation, and the likelihood of robot failure without operator intervention. This would allow us to convey complex system states (i.e. at a high level of abstraction) relevant to Triage intuitively, potentially reducing cognitive load on operators while improving their ability to make timely decisions about robot servicing. Designing these systems in a transparent manner is crucial for calibrating operator trust, enabling them to understand a robot's capabilities and limitations, and thereby facilitating triage decision-making. Inspiration is also drawn from popular video games, where real-time information is presented transparently while maintaining optimal cognitive workload. In games like DOTA 2, colour-coded UI elements like health bars, as shown in Figure 5.1, effectively convey complex information to players, who use it for their gameplay strategy. In the popular online game Overwatch, remotely situated players use these cues to grab their teammates' attention. Similarly, robot operators can use colour-coded visual cues to achieve quick situational awareness with minimal perceptual effort. Based on this intuition, it is hypothesised that providing operators with information about the robot's performance degradation through multi-modal sen-

Figure 5.1: Example of Health Bar used in Games

sory cues can improve the servicing of Variable Autonomy (VA) robots during runtime, promote transparency, while reducing the operator's cognitive workload.

To evaluate this hypothesis, a systematic two-step approach is adopted. First, a pilot study is conducted to understand how robot operators interact with a 'Robot Health Bar' visual cue, and examine its effect on task performance, operator cognitive workload, and the perceived trust and transparency of the interface. Based on the insights gained from this pilot study, a refined control interface is designed for single robot and multi-robot systems. This new Triage-Centric Variable Autonomy interface integrates multi-modal sensory cues based on robot performance degradation. It is assessed for its impact on Human-Robot teams with one, two, and three robots. This study mainly focuses on the effectiveness of robot health based multi-modal cues in improving task

Figure 5.2: Condition A arena (L to R): 1) Empty, 2) With obstacles and uneven terrain, 3) 2D map with way points marked and locations where laser noise was introduced.

performance while minimising added workload and promoting trust, transparency, and a good user experience.

5.1 Robot Health Bar: A Visual Cue for Variable Autonomy Robots

This section describes the pilot study in which the 'Robot Health Bar ', a visual cue added to the variable autonomy robot operation interface, is designed and evaluated. The experiment carried out in the pilot study was a mobile robot navigation task using the Clearpath Husky Robot, simulated on a high fidelity robotics simulator (Gazebo). The overall experimental procedure resembles that of the experiments carried out to evaluate the Model Predictive Mixed Initiative Level of Autonomy Switching system described in section 4.3. The experimental arenas used for the experiments resembled those shown in Figure 4.14. However, the areas where the performance degrading factors were introduced were changed for this set of experiments. These arenas are displayed in Figure 5.2.

As shown in Figure 5.2, these arenas were designed to mimic urban search and rescue scenarios, and were populated with performance degrading factors commonly found in them such as obstacles, uneven terrain and laser noise. A 2D laser scan of an empty map (Figure 4.14 - Right) was generated before the performance degrading factors were added. The difference between the map used for robot navigation and the actual arena affects robot navigation planning, thereby adding another performance degrading factor. Two different arenas were created to compare operator performance with (condition A) and without the Health Bar (condition B).

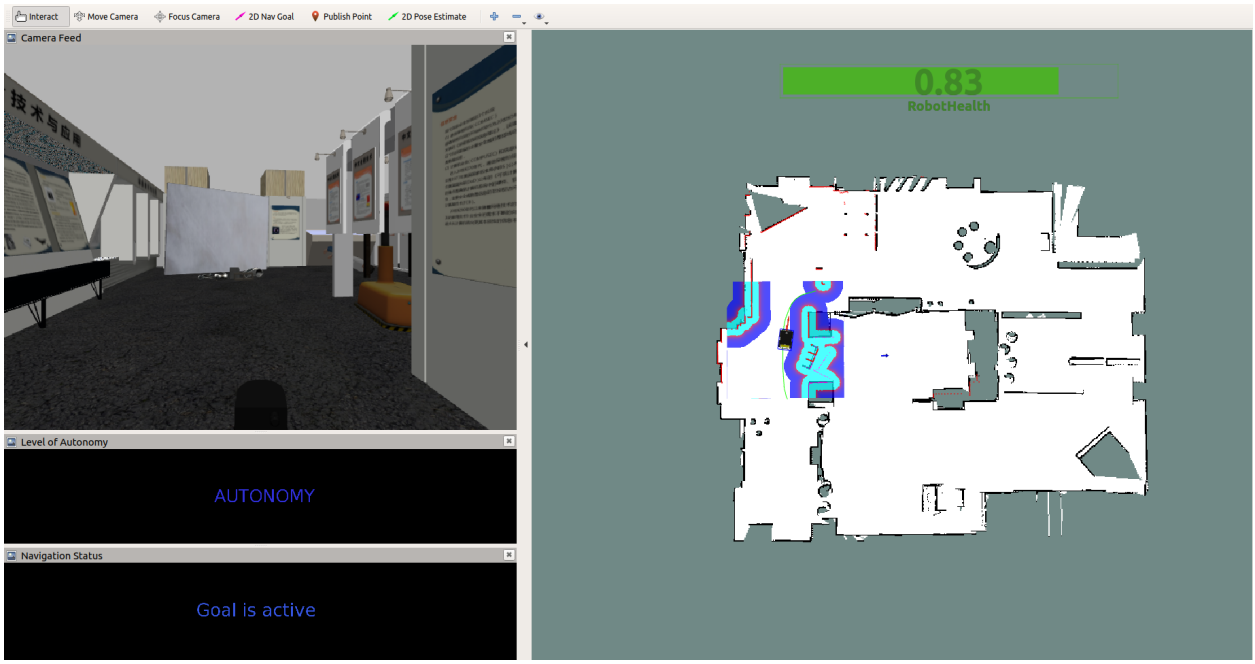


Figure 5.3: Condition A - Interface with Robot Health Bar. The green colour indicates the robot is 'healthy'

Figure 5.4: Condition B - Interface without Robot Health Bar

RViz was used to visualise the robot map, sensor data, and give commands to the robot. The standard RViz interface in Figure 5.4 was used for Condition B, and the interface with a 'Robot Health Bar ' shown in Figure 5.3 was used for Condition A. The Husky robot used two LoAs - 1) Waypoint-based navigation and 2) Manual Control by an operator using a Joystick. The Robot Health at each instant was calculated using the Robot Vitals and Robot Health framework (Ramesh, Rustam Stolkin, and Chiou, 2022), and the Health Bar was created and displayed using the JSK Visualisations[1] ROS Package. Here, Robot Health is defined as 'an overall scalar estimate of a robot's ability to carry out its tasks without its capabilities being impaired by any performance degrading factors '. Therefore, an operator can monitor the robot's health to detect situations where a robot is likely to fail, and trigger LoA switches to assist the robot. To improve readability, the Robot Health was standardised to the range $[0, 1]$. The Red-Amber-Green colour coding convention was used for the Health Bar as shown in Figure 5.3. Based on preliminary experiments, health above 0.7 was decided as 'Healthy ', and the colour of the Health Bar was green. Values between

[1] http://wiki.ros.org/jsk_visualization/

$0.5 - 0.7$ were coloured in amber, and values below 0.5 were coloured red. Instead of sharp colour changes, the Health Bar gradually went from green to amber to red. This was done to minimise rapid colour changes due to small fluctuations in health near the threshold values. A secondary task was used to induce additional cognitive workload in this experiment, which was a 3D object rotation task, identical to the secondary task in the experiments carried out in section 4.3. Participants carried out this task throughout the task's runtime. Both the primary task and the secondary task were displayed on adjacent screens as shown in figure 4.16. Before starting each trial, the participants were explicitly instructed 'to prioritise the robot navigation task and minimise the likelihood of robot failure '. They were also told to simultaneously do the secondary task to the best of their ability, and that they were not being evaluated on their performance in it.

5.1.1 Experiment Methodology

Eight test subjects participated in the experiment and performed both conditions (i.e. within subjects design). The order of the conditions was counterbalanced to minimise learning and fatigue effects. First, participants had to fill out background information (see appendix A.2.1). Then they were introduced to basic robot navigation in a training arena similar to the one used in Chiou, Hawes, and Rustam Stolkin (2021). All participants trained on this arena till they were able to demonstrate a minimum proficiency in robot navigation. This ensured that confounding factors due to variations in skill levels were minimised. Next, they were shown the 2 different LoAs, how to switch between them and were given some time to practice LoA switching on the training arena.

Participants were then introduced to the interface with the Health Bar and asked open-ended questions without prior explanation (see appendix A.2.2). After answering the questions, they were given the following instructions regarding the Health Bar: "The Robot Health Bar indicates how much a robot's performance is degraded by environmental factors. These environmental factors can be anything ranging from bad terrain to laser noise. During low health, the Health Bar will

Table 5.1: Summary of Statistical Analysis

Primary Task	**Condition A - With Health Bar**	**Condition B -Without Health Bar**	**Pairwise T Test, Two Tailed**
Completion Time	M 174.38, SD = 30.91	M 175.25, SD = 32.35	T-Score = 2.36, P Value = 0.96
Percentage of the Task Robot was Manually Controlled	M 49.13, SD = 19.97	M 39.77, SD = 17.77	T-Score = 2.36, P Value = 0.08
Percentage of the Task Robot was Unhealthy	M 61.91, SD = 21.80	M 78.03, SD = 21.36	T-Score = 2.36, P Value = 0.17
Percentage of the Task Robot was Manually Controlled when Unhealthy	M 64.59, SD = 21.16	M 45.17, SD = 18.67	T-Score = 2.36, P Value = 0.02*
Secondary Task	**Condition A - With Health Bar**	**Condition B -Without Health Bar**	**Pairwise T Test, Two Tailed**
Average Response Time	M 10,616.18, SD = 3,673.76	M 9,053.49, SD = 3,679.02	T-Score = 2.36, P Value = 0.22
Total Answered	M 20.63, SD = 5.80	M 23.63, SD = 8.35	T-Score = 2.36, P Value = 0.19
Accuracy %	M 90.13, SD = 7.00	M 94.59, SD = 4.98	T-Score = 2.36, P Value = 0.02*
NASA-TLX	**Condition A - With Health Bar**	**Condition B -Without Health Bar**	**Pairwise T Test, Two Tailed**
Mental Demand	M 80.63, SD = 16.78	M 80.00, SD = 17.73	T-Score = 2.36, P Value = 0.89
Physical Demand	M 28.75, SD = 13.30	M 36.88, SD = 25.35	T-Score = 2.36, P Value = 0.19
Temporal Demand	M 66.88, SD = 26.98	M 70.00, SD = 17.11	T-Score = 2.36, P Value = 0.60
Performance	M 43.75, SD = 29.73	M 43.13, SD = 24.63	T-Score = 2.36, P Value = 0.94
Effort	M 58.75, SD = 30.21	M 59.38, SD = 27.83	T-Score = 2.36, P Value = 0.91
Frustration	M 49.38, SD = 28.21	M 53.13, SD = 27.51	T-Score = 2.36, P Value = 0.36

*p<0.05

become redder. During high health, the Health Bar will become greener. The lower the robot's health, the more likely it will fail. You may use the Health Bar to help you determine when the robot requires a LoA switching". The effect of performance degrading factors on the Health Bar was demonstrated by introducing obstacles and laser noise in the training arena. Participants were also given time to familiarise themselves with the interface before starting the navigation task. Next participants were familiarised only with the secondary task, and allowed multiple practice runs till they felt comfortable with it. Their baseline on the secondary task alone was measured before the experiment.

Before each condition, participants were shown the start and finish points in the navigation task and told that the next way point would be assigned automatically on reaching the current one. Then the participants carried out each experimental condition followed by a NASA-TLX form to evaluate the perceived cognitive workload during the task. Lastly, participants completed open-ended questions and a transparency/trust questionnaire after the experimental trials and the NASA-TLX forms (see subsection A.2.2 and subsection A.2.3).

5.1.2 Results

Robot health values under 0.7 were classified as ‘unhealthy ’. This criterion was heuristically determined based on previous studies on the robot vitals and robot health framework (Ramesh, Rustam Stolkin, and Chiou, 2022). Results from the experiments conducted on 8 participants were computed and tested for statistical differences using two-tailed pairwise T-Tests. These results are summarised in table 5.1.

The total percentage of runtime that the robot was autonomous, manually controlled and ‘unhealthy ’ was calculated for each experimental condition. The percentage of run time that the robot health was ‘unhealthy ’ showed no statistical differences between the two conditions. Similarly, no significant differences were observed in the percentage of run time the robot was manually controlled. However, the percentage of run time that the robot was manually controlled when it was unhealthy, showed a significant ($p < 0.05$, $t = -2.36$) difference between both conditions. That is, operators manually controlled unhealthy robots for longer when the health of the robot was displayed (i.e., Condition A). As a result, the robot was ‘unhealthy ’ for an average of 61.91% of the runtime in condition A, but 78.03% of the runtime in condition B.

The change in perceptual effort required to carry out LoA switching tasks was measured using the operator’s accuracy on the secondary task. When the health bar was not displayed (Condition B), operators showed significantly higher levels of accuracy ($p < 0.05$, $t = 2.36$) on the secondary task. This indicates the perceptual effort required to use the interface with the robot health bar, is higher. However, the NASA-TLX scores showed no significant differences in the overall cognitive workload imposed by both conditions.

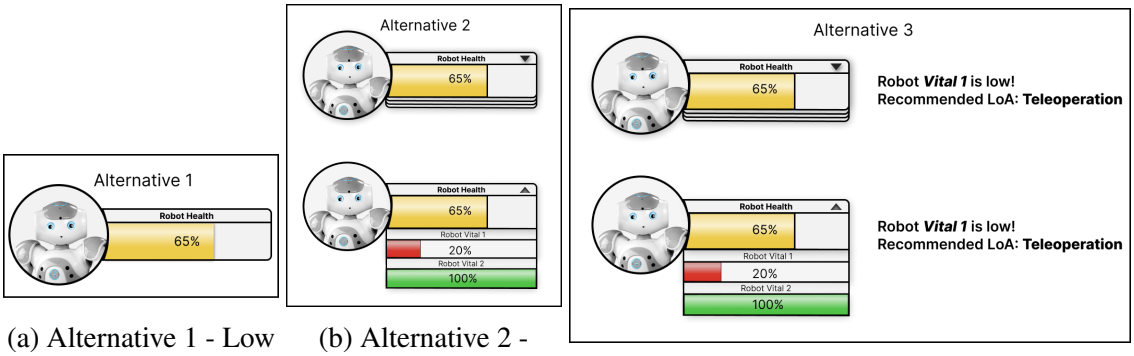
Open-ended questions asked before the experiment showed that some participants were slightly confused about what the Robot Health Bar was. One participant said - "The health bar looks like a timer, because it has a number. I thought it meant seconds". Others successfully

grasped the idea behind the UI element. Another participant thought "When the number goes down the robot dies", indicating a game-like perception of the Health Bar's function. However, all participants understood that it was intended to assist the operator or alert them to the robot's performance degradation

Most participants preferred the interface of condition A (with the 'Robot Health bar '), finding it intuitive for regulating LoA - "When the health was low, it was better to manually control the robot". One participant said -"Without the health bar I had to actively use my brain to detect when the robot needed help". However, three participants felt the colours were sometimes inaccurate and did not match the robot's state. One of them pointed out that while red and green clearly helped indicate which LoA was better, amber was confusing because they did not know what to do. Although not significant, operators with experience operating robots (based on background information) seemed to prefer more transparency in the UI, while the rest preferred less transparency. Novice operators (as reported by the background information), instead, stated that they responded to the colours of the UI to a greater extent than the participants with more experience in operating robots.

Pearson's correlation showed a significant negative trend between how often the participants operated remote-controlled vehicles and how easy it was to know when to change LoA for alternative 1 (shown in Figure 5.5a) in the questionnaire ($p < 0.01$). Significant positive trends were also observed for how often they used AI for work and how easy it was to know when to change the LoA for alternative 3 in the questionnaire (shown in Figure 5.5c) ($p < 0.05$), and how easy it was to understand why to change LoA for alternative 2 (see Figure 5.5b) ($p < 0.01$) and alternative 3 in the questionnaire(Figure 5.5c) ($p < 0.01$) (see Table A.3.1).

In terms of determining when to change LoA for the alternatives in the questionnaire, four participants rated either the highest or both the lowest and highest levels of transparency equally highly on the Likert scale (Figure A.2a). In contrast, the remaining four participants rated the



(a) Alternative 1 - Low transparency

(b) Alternative 2 - Medium transparency

(c) Alternative 3 - High transparency

Figure 5.5: Alternatives ranging from low to high transparency in the questionnaires for the Robot Health Bar Experiment

lowest level of transparency highest in comparison to the other two levels. Overall, the medium level of transparency was rated the lowest. As for why to change LoA, six participants either increased or maintained their rating for each increase in the level of transparency, resulting in the highest level of transparency being rated the highest (Figure A.2b). However, two participants decreased their rating on the Likert scale for each increase in the level of transparency.

5.1.3 Discussion and Insights

Our experimental results illustrate how informing an operator about a robot's performance degradation through visual cues can affect the operator's driving and LoA regulation style. When shown the health bar, operators triggered LoA switches to mitigate situations where the robot was 'unhealthy '. This style of HI-LoA regulation reduces the aggregate risk of robot failure. The interface with the health bar did not impose a significant additional cognitive workload on the operator during the experiment, and did not significantly change the task completion time. Contrary to the hypothesis, LoA regulation using the Health Bar required significantly ($p < 0.05$) higher perceptual effort. In condition A, Participants carried out fewer rotations and gave less accurate answers. One likely explanation for this is that adding a Health Bar increased the number of points on the UI the operator had to focus on, which increased the perceptual effort for the primary task.

Table A.3.1 shows that participants with little to no experience operating robots prefer significantly less transparency than experienced robot operators. Participant feedback revealed that while novice operators responded more to the colours of the health bar, experienced operators preferred having more information displayed to inform their LoA switching decisions. This indicates that participants value transparency only when they understand the necessity of transparency. People with different levels of experience have varied mental models about the Robot, leading them to perceive the system and the Health Bar differently. This makes it difficult to standardise explanations. Depending on their experience, each participant may want to clarify different aspects of the Human-Robot System, leading to differences in the way they are primed for the experiment. Therefore, rigorous training coupled with detailed explanations about the different components of the system are required to minimise the differences in perception. Finally, a limitation of this study is the small sample size, as it is a pilot. Therefore, all results and insights presented in this study require validation with a larger set of participants. To this end, a rigorous experimental validation was conducted in section 5.2, involving more participants and a redesigned interface that incorporated feedback from this study. The major themes in participant feedback are listed below.

5.1.3.1 Participant Recommendations to improve the Robot Health Bar and User Interface Design:

- Make the Health Bar more Salient, so that it attract attention when necessary
- Do not include a drop down-menu in the Health Bar as shown in the Questionnaire
- Use a percentage to the health value instead of a value between $[0, 1]$
- To attract attention to 'low health' use multi-modal sensory cues like sound alerts or UI elements e.g. By Shaking or Blinking the health bar after a threshold value.
- The robot health should be shown on top of the robot as well instead of just the screen, so

that operators don't have to look at multiple points in the screen to get adequate information about the robot's performance.

5.1.4 Conclusion

This pilot study explored whether visual cues about a robot's performance degradation can reduce the perceptual effort required to make LoA switching decisions for remote mobile robot navigation tasks. Inspired by video games, a 'Robot Health Bar' was designed. This health bar displayed the total runtime performance degradation the robot is facing. A total of 8 participants carried out a mobile robot navigation task with and without the health bar UI element under a high cognitive workload, and their performance was measured. Adding a Robot Health Bar to the robot control UI significantly changed how the operator makes Level of Autonomy Switching Decisions. When the Health Bar was displayed, operators were more attentive and took control of the robot more to minimise the risk of the robots failing. Visual cues that indicate a robot's health can serve as an effective way of ensuring safer control of robots, especially in extreme environments where robot missions have high levels of risk and environmental adversities. User feedback indicated that 1) the health bar needs to be more salient to attract more attention, and 2) the use of multi-modal sensory cues can better direct operator attention to low health situations. Finally, operators need to be provided with more rigorous training to understand how to use the health bar to aid the tasks.

5.2 Triage Centric Control Interface for Variable Autonomy Multi-Robot Systems

Our pilot study showed that visual cues based on the robot's performance degradation can significantly influence operation style and reduce the aggregate risk of robot failure. However, operator

feedback indicated modifications were necessary to the robot health bar and interface design for users to leverage its benefits fully. These design considerations are crucial to reducing the cognitive demand of operating VA robots and scaling the interface to MRS. Thus, there is a need to fundamentally redesign the overall interaction paradigm to facilitate quick and effective robot triage. This redesign incorporates a new health bar, multi-modal sensory cues, and distribution of information in a manner that reduces the perceptual effort required to detect robot performance degradation. In this section, the process for designing the Triage-Centric Control Interface for a Variable Autonomy Multi-Robot System (VA MRS) is described. This process combined insights from my previous work with best practices from existing literature, primarily the ecological design guidelines proposed by Murphy and Tadokoro (2019).

5.2.1 System Design

We scope this interface design to a remote multi-robot navigation task using Clearpath Husky Robots, with two LoAs: Waypoint-based Autonomous Navigation and Manual Control via Joystick. For autonomous navigation, the standard ROS navigation stack (Marder-Eppstein, Berger, et al., 2010a) is implemented, utilising the move base package (Marder-Eppstein, 2024) with a dynamic window local planner and a Dijkstra Algorithm based global planner. The planner used a 2D laser scan of the experiment arena (without performance degrading factors) as input. An automated waypoint planner provided robot goals. For manual control, joystick movements were mapped to robot velocity commands. Different buttons on the joystick were mapped to the LoAs to allow the human operator to easily switch between them, and the control mode of the robot was set accordingly. This interaction paradigm implements a total of six cues: two for LoA switches, two for robot selection, and two for runtime performance degradation. The interface design shown in figure 5.6 is described below.

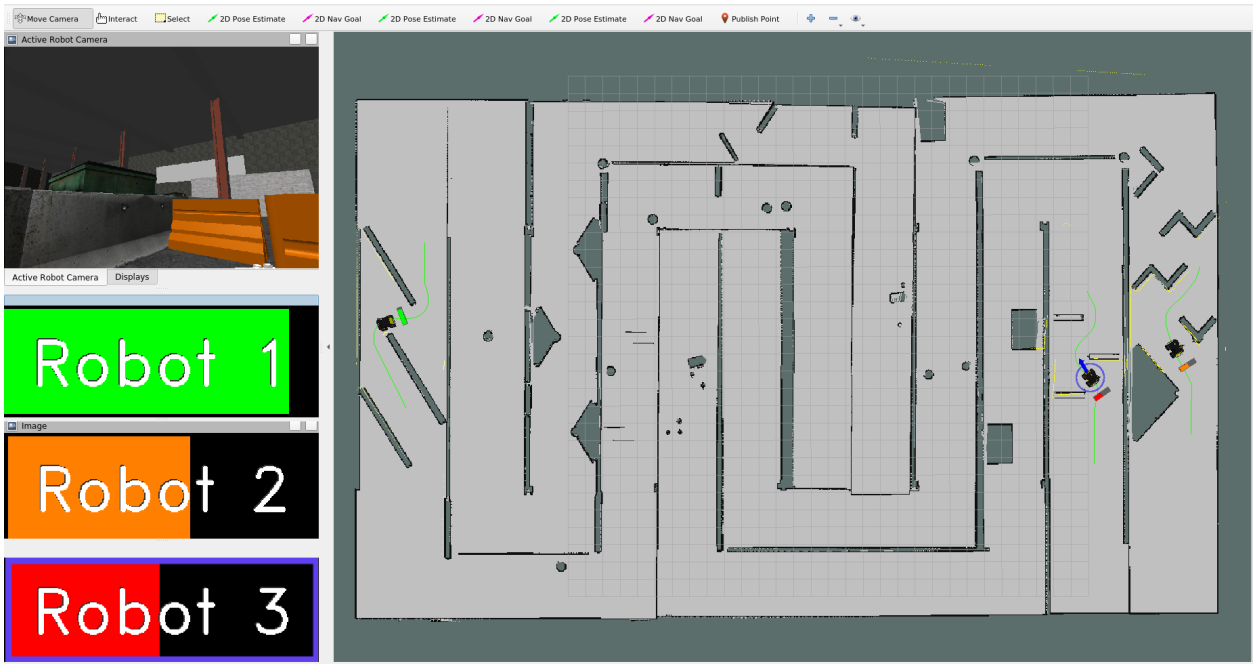


Figure 5.6: Triage Centric Control Interface, with 3 Robots during Task execution

5.2.1.1 Health Cues: Robot Health Bar and Low Health Sound Alert

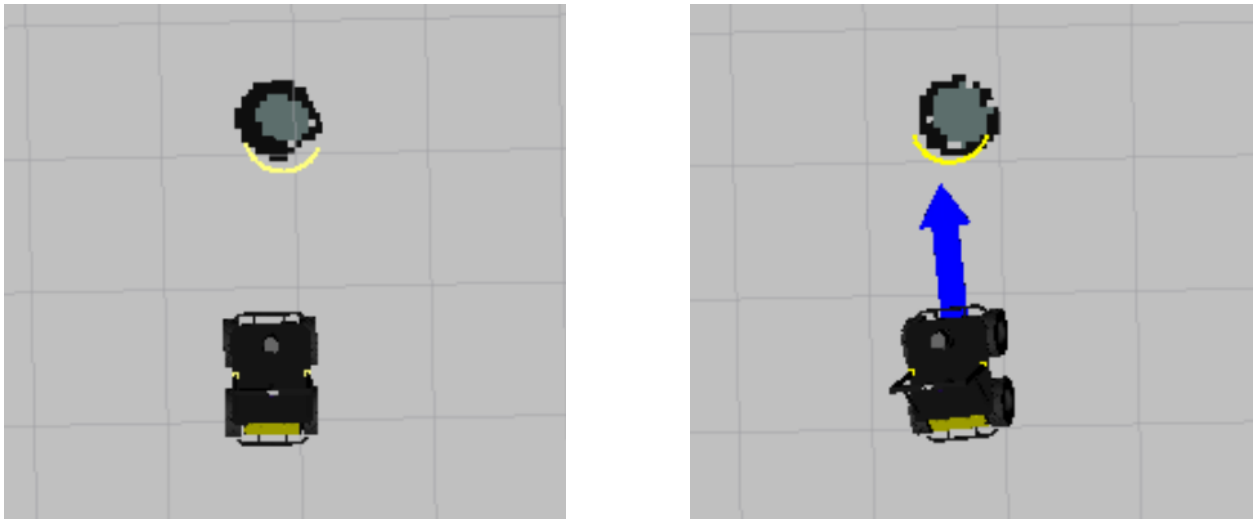
For each robot in the MRS, runtime performance degradation was calculated using the Robot Vitals and Robot Health framework described in 3. Inspired by video games, a robot health bar was added as part of the control interface. This health bar was attached to the robot's frame at a fixed distance and angle, ensuring it remained in a constant position and orientation relative to the robot. Based on feedback from the pilot study, the robot health values were standardised to the range of $[0, 100]$ for simplicity, with 100 indicating no performance degradation and 0 indicating robot failure. The health bar was displayed by mapping health values to the length of a rectangle, which adjusted its size proportionally to the robot's health. A shadow was added under the robot health bar to give it depth and distinguish it from other map elements.

Similar to the pilot study described in section 5.1, the red-amber-green colour scheme was used to indicate different intervals of health values. Green was used when the robot's health was high and operator intervention was not required. Red was used when the robot faced severe per-

formance degradation and was likely to fail unless manual control was initiated. Amber was used for intermediate states. In the pilot study, operators noted that gradually changing the colour from red to green was sometimes not perceived. Therefore, the colour changes were made sharp and at fixed thresholds to attract operator attention quickly. Operators were trained to interpret the colours (explained in a later section). Based on preliminary trials, the colour change thresholds from green to amber and from amber to red were set at 65% and 40%, respectively. These values were chosen to enable operator intervention well before catastrophic robot failure.

The robot health bar visual cue was used for each robot in the team, enabling the operator to intervene whenever any robot faced performance degradation. However, detecting low health for any robot required the operator to focus on it specifically and constantly monitor it. Operator feedback from the pilot study showed that sometimes the operator may not see the health bar and be distracted. Therefore, additional cues are required to grab their attention and make them focus on the health bar. A sound alert cue was also used to indicate low health for each robot. The low health alert was designed to interrupt previous sound alerts, quickly capturing the operator's attention. To minimise conflicts between successive sounds, a timeout was added for low health sound alerts. For two-robot experiments, only one low health sound alert was played per robot every 8 seconds. This threshold was set to 12 seconds for three-robot experiments. These timeout intervals were chosen heuristically based on preliminary trials, ensuring enough time between successive sound alerts to avoid overloading the operator while still effectively alerting them to performance degradation.

In addition to the health bars attached to each robot's frame, larger health bars were displayed on the left side of the screen under the camera feed as shown in 5.6, providing a clear and constant visual reference for the operator. Each of these health bars corresponded to a specific robot denoted by the robot number, labelled with the robot's number and using the same red-amber-green colour scheme to indicate health status. This setup ensured that even if the operator was focused on the camera feed, they could still quickly assess the health of all robots with a quick



(a) Robot in Autonomous Navigation Mode (b) Robot in Manual Control Mode

Figure 5.7: Heading Arrow Visual Cue used to denote when robot is in Manual Control

glance without too much perceptual effort or needing to focus on each individual robot's health bar on the map.

5.2.1.2 Level of Autonomy Switches

Throughout the task, operators should have access to information about each robot's LoA. Feedback from the pilot study (section 5.1) indicated that when information about a robot's state was distributed across the screen, operators required extra perceptual effort to focus their attention on different parts of the control interface to gain state information about each robot in the MRS. In the presented interface, the objective was to provide operators with maximum information about the robot at a single point of focus. Controlling the robot manually requires the operator to look at the 2D map to understand the robot's location in the context of its surroundings, and then to look at the camera to notice any performance-degrading factors missing from the 2D map. Therefore, a visual cue in the form of an arrow was introduced beneath the robot model on the map to indicate the robot's heading direction and Level of Autonomy (LoA).

As shown in figure 5.7, when a robot's LoA was set to Manual Control, a direction arrow

appeared under it to denote which direction the robot and its camera were facing. Similar to the health bar, this was linked to the robot's frame so that it moved with the robot. Instead of using separate cues for each Level of Autonomy, the arrow appears during Manual Control and was removed when the robot was in Autonomy. This was done both to simplify the interface, and to reduce the number of cues each operator had to remember to understand the robot's state. As suggested by Murphy and Tadokoro (2019) an arrow is used to indicate heading so that the design of this interface is consistent with other look and feel of heading indicators in video games. Along with the visual cue, a sound cue was implemented to alert the operator about the robot's LoA to accommodate for situations where the operator may be looking at the camera and neglect the map. Every time a LoA switch was triggered, this sound cue stated the robot number and its LoA. In cases where multiple LoA switches were triggered across multiple robots occur in rapid succession, the most recent sound cue interrupts the previous one. This approach is taken instead of a sequential one to minimise operator confusion due to delay.

5.2.1.3 Switching between Robots

While monitoring task execution by the VA MRS, the operator could select and control only one robot at a time. That is, the operator was only able to see the camera feed of the selected robot and could initiate LoA switches for or manually control only that robot. Each robot was assigned a specific number, with robot 1 selected by default at the start of the trial. To switch between robots, the shoulder buttons on the joystick had to be used: the right shoulder button switched to the next robot $2,3,4,\ldots N$, and the left shoulder button switched to the previous robot $N,N-1,\ldots 1$. For example, to switch from robot 2 to robot 5, the right shoulder button on the joystick had to be pressed three times. While other methods are available for directly switching to a specific robot (e.g., using number keys on a keyboard, voice activation), these methods could increase the mental demand on the operator, introduce confounding factors, or cause the operator to look away from the screen. Instead, the operator was provided haptic feedback through joystick vibration every

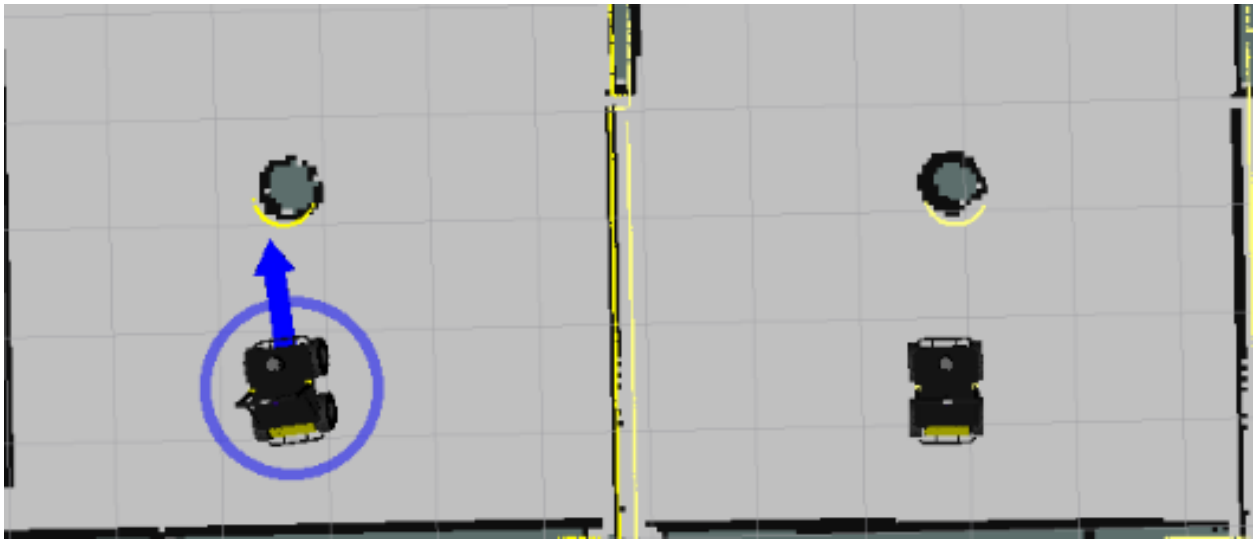


Figure 5.8: Active Robot Marker Visual Cue: The robot on the left is currently active, and is in manual control

time a robot switch was made. The main advantage of the haptic cue is that its hard to ignore and is perceived more quickly by the brain thereby promoting a better user experience. Additionally, an active robot marker visual cue was used to indicate the active robot through visual feedback. This was a blue circle drawn around the active robot as shown in figure 5.8. Each time the robot was switched the circular marker switched to the newly selected robot, thereby constantly indicating the active robot during runtime.

5.2.1.4 Interface Layout and Visual Design

The first three guidelines proposed by (Murphy and Tadokoro, 2019) $G1.1 - G1.3$ outline how the user interface must target the user's role and eliminate extraneous features. The interface proposed here was identified as an 'End User' interface used by robot operators during the task runtime. Therefore, all information displayed on the screen was limited to whatever was relevant to monitoring task execution, Triage, and servicing the robots through LoA switching. In accordance with $G1.2$, no picture-in-picture displays or selectable windows were used during mission runtime.

Guidelines $G2.1 - G2.7$ deal with optimising how the information is distributed on the user's display setup and outline principles to optimise the visual design in accordance with viewing behaviours and cognitive capabilities. Accordingly, the user interface was created to be compatible with any typical laptop or desktop monitor. As shown in Figure 5.6, the panel on the right displays all the robots, the 2D map of the arena, the navigational goals of each robot, and the multi-modal cues described above. Therefore, it is considered the primary window and the UI's largest panel. No pop-ups or overlays are used except the small overlays introduced due to the 'Health Bar' visual cues. The camera feed is displayed to the left of the map panel, consistent with general user viewing behaviour. The interface was designed to be observed from left to right, with no additional information presented in the top or bottom rows. Lastly, no text is overlaid on top of the images to minimise the strain of reading small text. As suggested in guideline $G2.6$, a combination of sound and haptic cues were used as an alternative to this. Since the experiment focuses only on a navigation task $G2.7$ is not applicable.

With respect to content, every visual cue and panel used in the UI serves only one function each, with no overlays. To avoid overloading the operators cognitive capabilities, multiple simultaneous camera feeds are not displayed together. Instead, at any point during runtime only the camera feed of the robot selected by the operators is displayed. While the map provides an allocentric, i.e. bird's eye view of the whole experimental arena, the camera feed of the robot gives the operator an ego-centric view for any robot when selected. This enables operators to have two levels of situational awareness for the task. It was also ensured that there was no unintended 'visual capture' because of any of the panels or the interaction cues. All the interaction cues used serve a specific function and are removed when not required. Lastly, $G3.1.4$ states that 'any interface designs should be consistent with the look-and-feel with existing apps'. Since computer games are closer to robot interface designs than traditional apps, they are used as a reference for designing the interaction cues in this study. Thus, the UI design incorporates aspects of computer games like an active sprite marker, health bar, direction arrow, and low health alert sounds.

The interface consists of the 4 major components - 1) Robot models, 2) Map, 3) Interaction Cues and 4) Camera feed. No graphs, grids or complex shapes are added to it. The health bar uses the colours red and green for low and high robot health respectively, to facilitate easy comparison and evaluation of robot status. This colour code also helps operators make quick decisions about 'normal' and 'off-normal' robot behaviour. Additionally, the size of the health bar changes proportional to the robot's health. A shadow is added under the robot health bar to give it depth, and denote that it is different from the map. The semantically related information is also grouped together. The robot status information for each robot in the MRS is displayed on the robot. This includes information about which robot is active, its level of autonomy, the heading direction, and how much performance degradation the robot is facing. The colour blue is used to indicate the active robot and heading direction during manual control, as it contrasts the colour of the map and obstacles in RViz. Red, amber and green were reserved specifically for robot health-related information to avoid additional perceptual effort caused by confusing colour-coding schemes. This colour coding scheme also aligns with *G*3.1.4, by adapting to best practices from existing computer game interfaces. Lastly, to help the operator quickly identify the health of the robot currently displayed in the camera feed, the active robot was highlighted with a purple outline around its health bar. This ensured that if the operator was focused on the camera feed instead of looking at the map, they could still quickly assess the health of all robots with a quick glance, without too much perceptual effort or needing to focus on each individual robot's health bar on the map.

5.2.1.5 User Experience and Interaction Design Consideration

While the guidelines by (Murphy and Tadokoro, 2019) forms the foundation of the interface design, other principles were also used to enhance user experience and interaction. The power of 10 rule by (Nielsen, 1994), states that operators expect the computer to respond within 0.1 seconds, get impatient at 1 second, and assume there's a problem if there's a 10 second delay. To align with these expectations, it was ensured that all UI elements, including the health bar, active robot marker, and

heading arrow, refresh every 0.1 seconds throughout the experiment runtime. Additionally, to align with the principles of ecological interface design (Burns and Hajdukiewicz, 2017) and to avoid overloading the operator's cognitive capabilities, the total number of interface cues was limited to six. This ensures that the operator can comfortably hold and process all relevant information within their working memory, thus enhancing situational awareness and decision-making efficiency.

Our interaction design minimises the number of clicks required for each functionality, typically to a single joystick button press. However, the complex nature of multi-robot systems introduces some exceptions. For instance, due to bandwidth limitations in this experimental setup, switching between camera feeds of different robots may take 0.1-0.4 seconds, especially as the number of robots increases or when operators rapidly switch between robots. Also, operators need to switch robots one by one instead of directly switching to the desired robot, which may take more than 0.1 seconds depending on how quickly they press the shoulder buttons. The focus of this experiment is to study how robots respond to performance degradation. Because this experiment is carried out in simulation, there is no risk of physical damage to the robot. Performance degradation can cause robots to collide or topple on uneven terrain, which would be impractical and costly to test in real-world environments. Additionally, while practical considerations such as communication delays between real robots deployed in extreme environments are important, they are beyond the scope of the current work. Thus, simulation allows us to safely and effectively investigate these critical aspects.

Considering the scope of this study, the design did not account for transitional users (as per guideline *G*4.4 by (Murphy and Tadokoro, 2019)) who progress from novice to expert proficiency through extended practice. This experiment involves only two trials per participant, with counterbalancing to mitigate learning effects. Based on the pilot study, significant changes in operator proficiency over these two trials were not anticipated. To maintain experimental consistency, requests for interface customization, such as resizing or rearranging panels, adjusting the brightness of the camera feed, reducing the volume of sound alerts, or remapping joystick controls, were not

entertained.

5.2.2 Experimental Evaluation

The aim of this study, is to evaluate the impact of multi-modal cues based on robot performance degradation on the operator's ability to triage and service variable autonomy (VA) robots effectively. The experiment involves a multi-robot navigation task where each robot navigates to a point of interest. A single remote operator supervises the MRS task execution in a mock-up disaster environment using a high-fidelity robotics simulation software (Gazebo). The operator can monitor task execution and assist the robots through LoA switching when they encounter performance degradation. It is hypothesised that providing information about robot performance degradation to operators will improve task performance without increasing their workload, while enhancing usability, trust, transparency, and system reliability.

To evaluate the hypothesis, two versions of the Triage-Centric Control Interface are compared : 1) With Health Cues (i.e. with robot health bars and low health sound alert), and 2) Without Health Cues (no robot health bars and low health sound alert). A combination of metrics were used, including data from robot operation during the trials, post-hoc questionnaires to assess perceived workload, usability, reliability, and trust, as well as qualitative feedback from participants. Details of the background information questionnaire and the usability and reliability evaluation questionnaire are provided in the appendix under subsection A.4.1 and subsection A.4.2. Lastly, after each participant completed both trials, they were asked for qualitative feedback on their experience. The arena's design, the operator training procedure, the description of the experimental conditions, and the evaluation metrics are described below.

Table 5.2: Overview of Experiment Design - 10 Trials Each

	Group 1 - 1 Robot		Group 2 - 2 Robots		Group 3 - 3 Robots	
	Condition A	Condition B	Condition A	Condition B	Condition A	Condition B
Robot Health Bar	Yes	No	Yes	No	Yes	No
Low Health Sound Alert	Yes	No	Yes	No	Yes	No
Haptic Feedback for Switching Robot	No	No	Yes	Yes	Yes	Yes
Sound alert for LoA switch	Yes	Yes	Yes	Yes	Yes	Yes
Active Robot Marker	No	No	Yes	Yes	Yes	Yes
Robot Manual Control Heading Arrow	Yes	Yes	Yes	Yes	Yes	Yes

5.2.2.1 Experiment Design

The study used a mixed experimental design with both within-subjects and between-subjects factors. Three groups, each consisting of 10 different participants, carried out the experiment in a between-subjects design. These groups, referred to as Group I, Group II, and Group III, used 1 robot, 2 robots, and 3 robots, respectively, as the robot team size. Each group had two within-subject conditions (A and B). In Condition A, participants used the control interface with the Health Cues. In Condition B, participants carried out the multi-robot navigation task without the Health Cues. Table 5.2 describes the groups and conditions.

The experiment design with the factors evaluated is detailed in Table 5.2. Out of the six cues used as part of the overall interaction paradigm, two were kept constant across all the trials. Because the haptic feedback was specifically for indicating that the operator has switched between the robots, this was not used for Group 1 experiments. Similarly, the active robot marker was not used for Group 1, because the robot was always active. The factors evaluated within subjects for each group were the Health Cues - Robot Health Bar and the Low health Sound Alert. The total task performance across all conditions was evaluated within subjects across the groups. Therefore, a total of 30 participants were required to carry out a total of 60 trials as part of this experiment.

5.2.2.2 Experiment Procedure

The format of the experiments in this study was consistent with the previous experiments. Each participant was first briefed about the experiment and asked if they had any pre-existing medical conditions such as sensitivity to flashing lights, sounds, or other stimuli. They were informed that their participation in the study and the use of their data were voluntary, with approval from the University of Birmingham Ethics Committee, and in accordance with the Data Protection Act of 2018. After addressing any questions, participants signed a consent form before starting the experiment. They then filled out a questionnaire with background information, including their experience with playing games, operating robots, and using AI tools (see subsection A.4.1). On completion of the experiment, each participant received a £10 Amazon gift voucher for their participation. The following paragraphs describe the procedure for training participants, familiarising them with each aspect of the interface, and carrying out the trials.

Navigation and LoA Switching Training: At the beginning of each trial, participants received standardized, systematic robot navigation training according to the best practices outlined by (Chiou, Talha, and Rustam Stolkin, 2019). This training ensured that all participants had a minimum acceptable level of proficiency in manual control before using the HI-LoA system. Participants were shown both the RViz and Gazebo environments and instructed on how to supply navigation commands to the robot through the joystick. They were shown how the robot moves in RViz and the corresponding effects in Gazebo. They were instructed to use both the map layout and the camera to navigate the robot and made aware of the benefits of this approach compared to only using the map or the camera. Participants were then given time to familiarise themselves with navigation using the RViz interface without looking at Gazebo. They were told to inform the researchers once they felt comfortable with manual control before proceeding with the experiment. Once comfortable, each participant was asked to carry out a baseline navigation test, completing

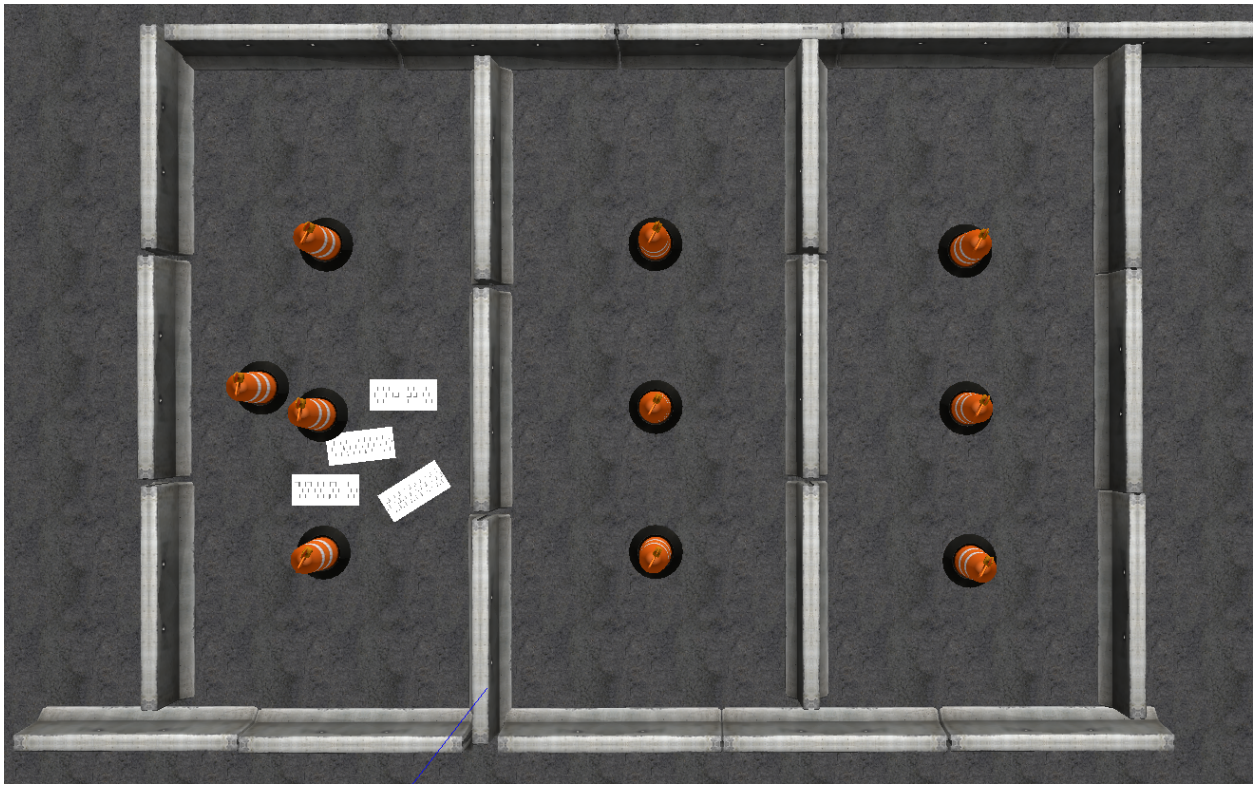


Figure 5.9: Multi-Robot Training Arena

one full lap of the training arena in under a minute, to ensure a minimum level of proficiency and minimise confounding factors during the experiment. After demonstrating proficiency in manual control, participants were introduced to Autonomous way point navigation and Level of Autonomy (LoA) Switching. They were shown how to switch between different levels of autonomy during task execution and given a few minutes to familiarise themselves with LoA switching during autonomous navigation.

Performance Degradation and Multi-Robot Training: After this, a new arena was loaded to introduce the participants to performance-degrading factors, as shown in figure 5.9. In this arena (figure 5.10), participants were shown how factors like uneven terrain and laser noise can compromise autonomous navigation. The map included two robots for Group 2 and Group 3 participants and was also used to familiarise them with multi-robot variable autonomy navigation. First, par-

ticipants were shown how different performance-degrading factors can affect task performance during autonomous operation. Robot 1 was given an autonomous way point where laser noise was activated, and participants observed the resulting performance degradation. They were then asked to switch to manual control to assist the robot through the laser noise area and switch back to autonomy afterwards. Next, each participant was shown the Gazebo model of the arena for Robot 2, which contained uneven terrain not represented in the RViz map layout but visible through the camera. Participants saw how the robot got stuck on uneven terrain and were instructed to use manual control to help the robot overcome this obstacle, then switch back to autonomy for the remainder of the task.

Next, each participant was shown how to switch between different robots using the joystick shoulder buttons and asked to confirm that they felt the haptic feedback each time the active robot was switched. They also observed that the active robot marker changed to the selected robot each time and that the heading arrow appeared when a robot was in manual control and disappeared when in autonomy. After ensuring the participant understood how to manage the variable autonomy MRS using these cues and addressing any questions, all robots in the map were set to autonomy and given target points simultaneously. The participant was asked to monitor task execution by the MRS, demonstrating their ability to switch between levels of autonomy and between robots to complete the training navigation task successfully. For Group 1, operators were informed about the sound alerts for LoA switching and the heading arrow. Robot 1 was then given a way point to reach, demonstrating their ability to switch between levels of autonomy.

Health Bar Training: Next, participants were introduced to the health bar and low health sound alert. Once shown the robot health bar, every participant was primed with the following explanation:

"*This is a robot health bar. This health bar shows you how capable the robot is of performing the task autonomously. So irrespective of which level of autonomy the robot is in, the health*

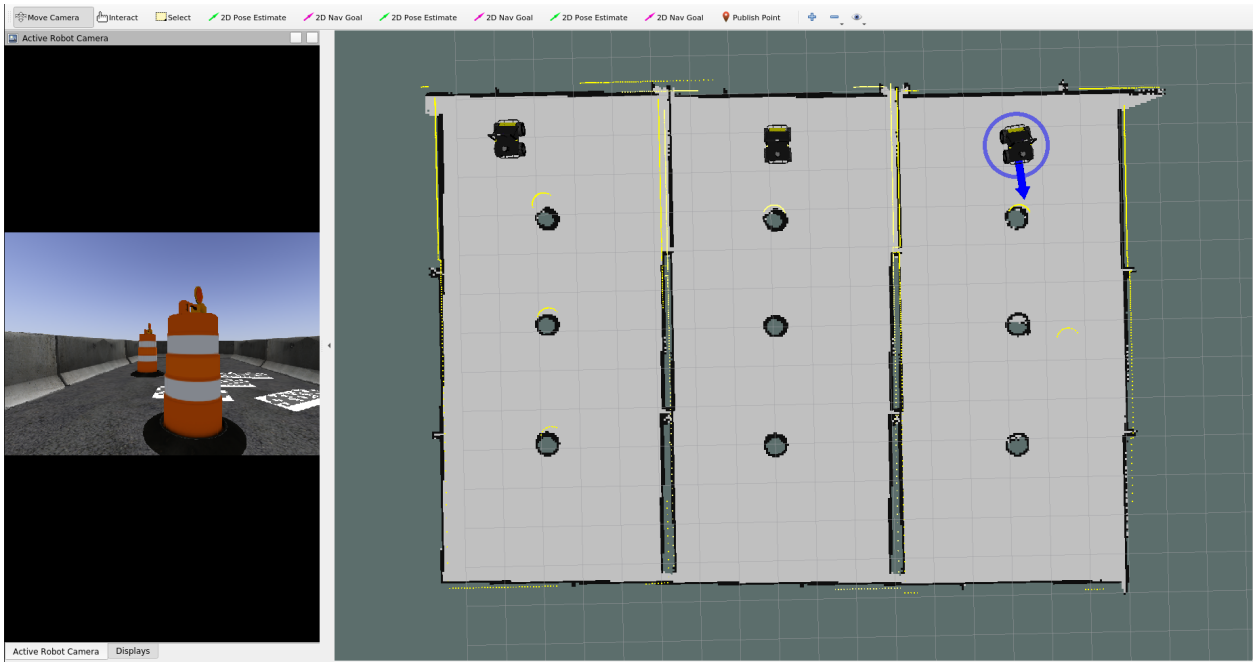


Figure 5.10: Training Arena Control Interface with No Health Cues for Group 3

bar shows you that if the robot were autonomous, how capable it is of performing the task. And the colours of this health bar follow the traffic light system. So when the health bar is green, it means the robot is quite capable of performing the task on its own. If the health bar is orange, it is still capable of performing the task on its own, but could benefit from operator assistance. If the robot's health goes into red, it is not capable of performing the task autonomously and needs operator assistance for task completion. However, this health bar is not 100% accurate all the time, so sometimes it may not accurately capture the robot's performance on the task. How you choose to use this health bar to help you with task execution is up to you."

For each participant utmost care was taken to ensure the explanation given was kept the same word for word, with a few exceptions made when participants interjected or wanted additional clarification. This was done to ensure the mental model of the health bar formed by each participant was as similar as possible, and not biased differently by the experimenter's explanation. Next the participants were shown how the robot health drops when the robot entered an area with laser noise. When the robot health dropped below 40% (rationale given in subsubsection 5.2.1.1) the

low health sound alert was triggered. Participants were told that this alert will play for each robot when the health drops below 40%. Finally, each participant was informed that the robot health for the active robot which is displayed under the camera will have a purple outline to denote which robot is selected, corresponding to the active robot. After answering any questions they had, and providing clarifications, when participants conveyed they are comfortable with the interface, they were considered trained.

Experimental Trials The experimental trials were conducted in the arena shown in figure 5.11, designed to simulate remote inspection tasks in extreme environments. This arena accommodates up to four robots performing tasks simultaneously. The main goal was to ensure the arena was large enough for each robot to have its own independent path while remaining compact enough to fit on a computer screen. This design decision was informed by feedback from pilot studies. Previous experiments with similar variable autonomy robots and the pilot study on this system showed that this field of study is particularly prone to significant differences in operating styles between trials due to learning effects. Therefore, most of the design considerations for this study were made to minimise the consequences of learning effects on task performance in each trial. In a pilot study for two robots, separate and unconnected arenas were used. However, to fit both arenas on the screen, the map had to be zoomed out too much, making it difficult for operators to track each robot's location and detect when they were stuck. Additionally, separate arenas led to operators subconsciously neglecting one robot and focusing on the other for extended periods. To mitigate these issues, it was crucial to design the arena so that operators perceived all robots as part of the same mission, simultaneously inspecting different areas of the map. Therefore, this unified arena design was used. Instead of using parallel paths, the robot paths were designed to motivate operators to use the active robot marker visual to detect the robot being controlled, rather than locating them based on their spatial arrangement on the screen.

Each robot's path included three different types of performance-degrading factors: uneven

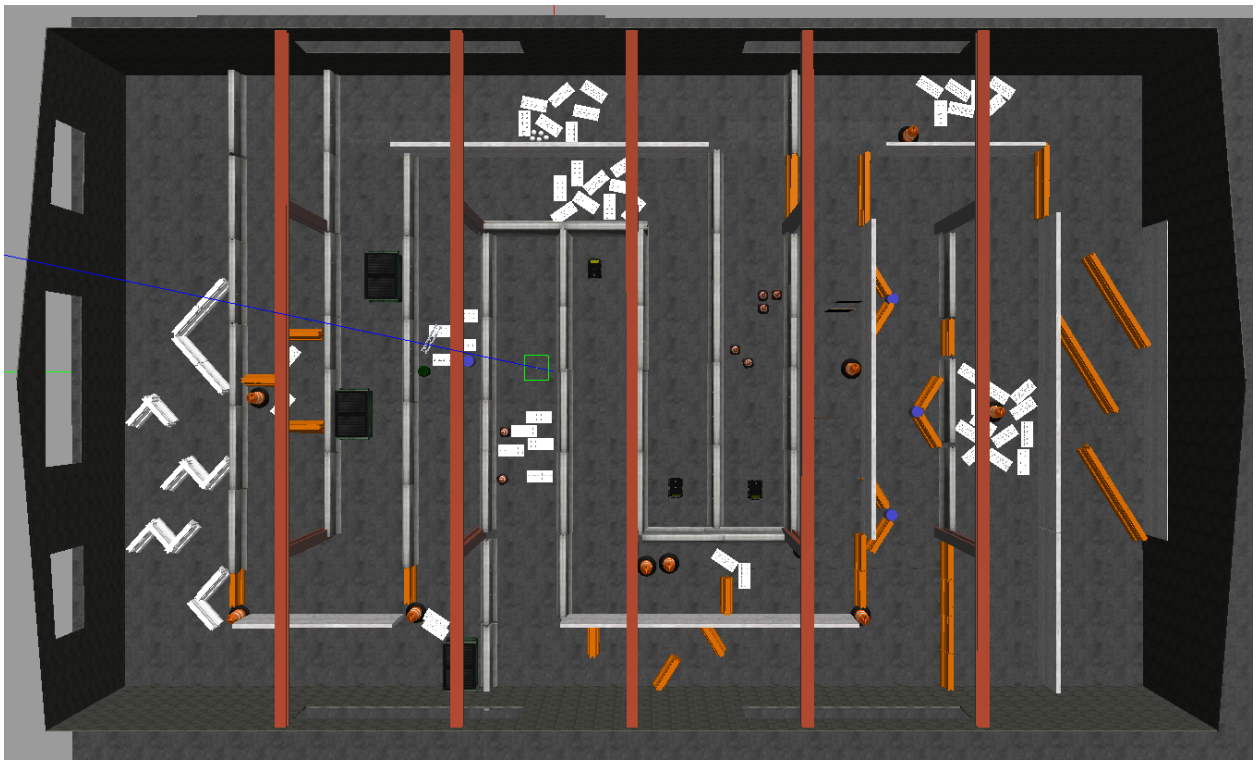


Figure 5.11: Experiment Arena for Multi-Robot Experiments, with three robots in their starting positions for Condition B

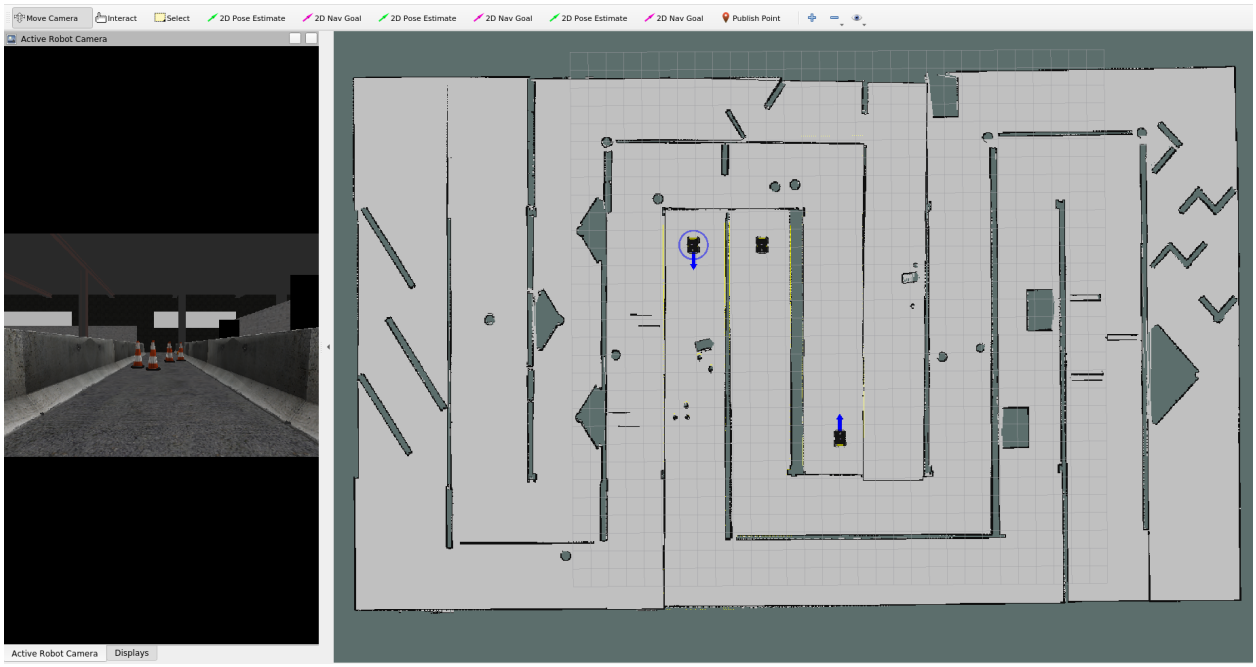


Figure 5.12: User Interface without Health Cues

or slippery terrain, challenging obstacles, and laser noise. These factors were arranged so that no two robots encountered the same intensity of performance degradation simultaneously. This setup ensured that each robot experienced a unique form of performance degradation at any given time, resulting in varying robot health values during the degradation. However, this was not entirely guaranteed, as the operation style and speed of task execution could delay one robot more than another, resulting in robots simultaneously facing low health situations. The approach in this study aimed to understand the basis for MRS LoA switching triage decisions by the operator, both with and without the health bar. In condition A, the robots traversed the path from start to end as shown in the figure 5.6. In condition B, the robots traversed the path in the opposite direction, using the user interface without health cues as shown in figure 5.12. This mimicked the nature of patrolling tasks in extreme environments, where the robot has to repeatedly patrol the same area by going back and forth multiple times. Additionally, making the robots traverse the same path in the opposite direction ensured comparable results between trials while preventing the operator from remembering too much about the map's layout, which could have induced learning effects.

Before starting each trial, participants were briefed that the experimenter would set the way points for the robots. Once the experiment commenced, the robots would move autonomously towards their designated goals. Participants were instructed that they were responsible for monitoring the task execution and providing assistance to the robots whenever necessary to prevent failure and ensure the successful completion of the task. They were advised that there was no time pressure, but they should avoid deliberately wasting time. Additionally, participants were reminded to focus equally on all robots and ensure that the tasks were executed simultaneously for each robot. After getting briefed, each participant carried out two trials of the experiment. After each trial the participants filled the questionnaire shown in subsection A.4.2 and the NASA-TLX. Lastly, after the participants finished both trials and questionnaires, as part of their qualitative feedback, they were asked a few questions to better understand their responses to the questionnaires.

5.2.3 Results

A total of 33 participants took part, with usable data from 30 participants. Data from three participants was excluded due to issues with the laptop, which caused it to fail to record the data twice and unexpectedly shut down once. Therefore, data from 30 participants were analysed and reported.

For each trial, data from the navigation task was recorded during runtime as Rosbags. The main parameters of the evaluation were understanding how operator behaviour and task performance change due to the presence of cues based on robot performance degradation. Plots based on different metrics are plotted below, with rigorous statistical analysis to evaluate if there are significant pairwise and group-wise differences. When statistical tests were conducted to measure group-wise differences, the data from all the robots were combined by concatenating them into a one-dimensional vector. However, the plots in this section also show data for each individual robot in the team, as this approach allows us to identify differences in performance between individual robots, which could be obscured when only considering the average or median values of all robots

together.

5.2.3.1 Risk Analysis

During the navigation task, the robot health of each robot was calculated across all trials for both conditions. The robot health was used as a proxy for the aggregate risk of robot failure during the task. That is, the lower the robot's health the higher the risk of the robot failing or performing the task sub-optimally. If the robot's health was below 45% it was considered low health. This threshold is based on previous experiments and extensive data linking robot behaviour with the robot health values. The value of low health threshold is slightly different from that used in the previous section ($threshold = 0.7$), as the calculation for the robot health metric was changed to reduce non-linearities in the health metric.

Figure 5.13: Percentage of the Runtime the robots had Low Health (Lower is better)

To calculate the impact of each condition on the aggregate risk of the robot failure, the percentage of the runtime the robot had low health was calculated and plotted in Figure 5.13. The trend showed that the percentage of runtime the robot had low health was generally lower for the Health Bar condition than the No Health Bar condition. Pairwise T-tests and Wilcoxon signed-rank

tests were used to study the impact of the conditions (i.e., health bar vs. no health bar) on each group, depending on the distribution of the data. Specifically, T-tests were employed for normally distributed data, while Wilcoxon's tests were used for non-normal distributions. Group 1 (One tailed, Less, $T-score=-5.816, p<0.0005$), Group 2 (One tailed, Less, $W=-5.817, p=0.047$) had significant differences in the percentage time the robot had low health. Specifically, significant results in the one sided results showed that the time the robot spend in low health for the health bar condition was lesser than the no health bar condition. However, group 3 showed no statistically significant difference ($W=0.6433, p=0.2633$) in the time spent in low health.

To further analyse the impact of both conditions on the aggregate risk of robot failure, the average health of each robot during runtime was calculated and plotted it in figure 5.14. The average health of the robot was also calculated for each LoA across conditions. The trend in figure 5.14 suggests that, overall, the median health of robots is higher with the Health Bar condition compared to the No Health Bar condition, although this is not the case for every individual robot. Pairwise comparisons for group 1 ($T-Score=11.32302, p<0.0001$), group 2 ($W=47.00, p=0.03$) showed strong evidence that the average robot health was higher when the operator used the Health Bar and Low Health Sound Alert, than when the operator didn't use those cues. No statistical significance was observed for pairwise comparisons in Group 3.

Next, the impact of the conditions on the time taken by the operator to notice when the robot is at risk of failure and intervene was evaluated. For this, the time difference is calculated between when a robot enters low health, to when the operator switches the robot from autonomy to manual control. Situations where 1) the robot was already in manual control were not included in the calculations, 2) the operator switched away from manual control to autonomy when the robot was in low health, and 3) situations where the operator switched to manual control but the robot was not in low health anymore; were not included. Once the robot entered a low health state, if the operator made multiple autonomy switches before the robot's health improved, the time taken until the final intervention was recorded. For each trial, the median of servicing time was calculated instead

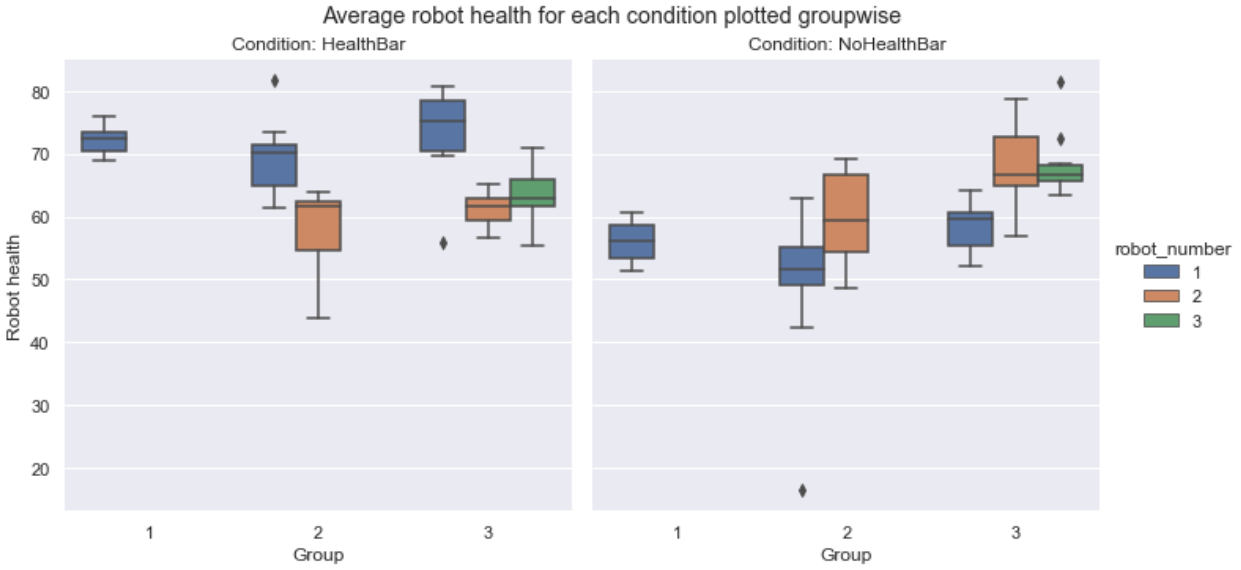


Figure 5.14: Average health of each robot during runtime (Higher is better)

of the mean. This was done to minimise the effects of outliers like - 1) operator halting a robot deliberately while servicing another robot and therefore neglecting the robot's health dipping; or 2) switching to manual control and getting close to the end point but not realising the robot is yet to reach the goal point. The resultant figure was plotted in figure 5.15.

The general trend of the graph showed that operators responded to service the robot quicker in the health bar condition than in the no health bar condition, by around 6-10 seconds. Group 3 had higher variance in the data when there was no health bar. Wilcoxon's signed rank test was used for group 1 and group 3 as they were not normal distributions, and a paired T-Test was used for group 2 as it was normal. One-tailed pairwise tests for group 1 ($W = 1.000, p = 0.005$), group 2 ($T - score = -3.191, p = 0.002$) and group 3 ($W = 148.00, p = 0.041$) show that the differences are significant across all conditions, and that the health bar condition has a lower median than the no health bar condition. This indicates that the health bar condition is able to significantly ($p < 0.05$) improve the operator's response time to low health situations, by saving around 6-10 seconds per triage decision.

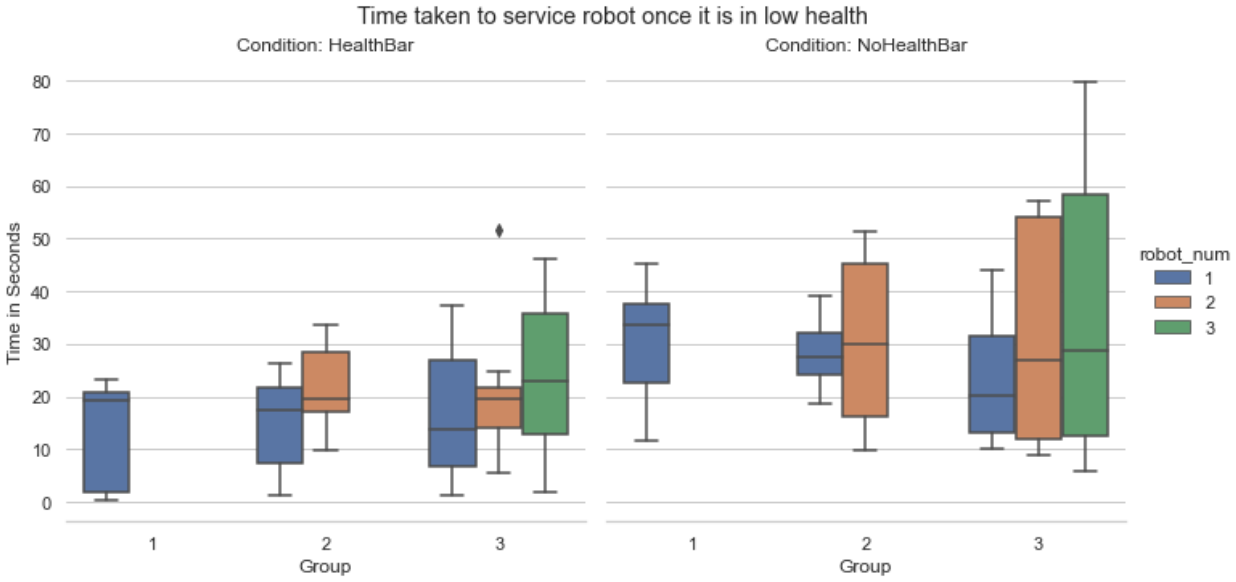


Figure 5.15: Time Taken to service the robot when it enters low health per interaction

5.2.3.2 Task Runtime Analysis

Pairwise comparisons between each group showed that there is no significant difference in the time taken to complete the task for each individual robot in the MRS ($p < 0.05$). The time taken to complete the task for each participant (i.e. for all robots) was calculated as the time taken by the last robot in the team to complete the task and plotted in figure 5.16. For group 1 and group 2, 6 out of 10 participants completed the task quicker with the Health Bar condition. However, only 2 out of 10 participants completed the 10 quicker for the Health Bar condition in group 3.

The median of the total time taken to complete the task in group 1, was lower ($Median =$ 179.43) for the Health Bar condition than the No Health Bar condition ($Median = 190.64$). Similarly, the median time-to-completion was lower for the health bar condition ($Median = 203.76$) than the No Health Bar condition ($Median = 211.53$) in group two. For group 3, the Median time taken for the Health bar condition was 235.58s, which was 11% higher than the no health bar condition $Median = 211.543$. Since the data was not normally distributed for all conditions, the Wilcoxon Signed Rank Test was used for checking for Statistical Differences. For group 1

($W = 10.0, p = 0.084$), and group 2 ($W = 23.0, p = 0.695$), there were no significant difference between the 2 sets of observations. The participant data for Group 3 was normally distributed and therefore, the paired T-Test was used. The difference between the time taken for group 3 was statistically significant ($t = 3.074, p = 0.013$), indicating there the 11% increase in the time taken for the Health Bar condition is statistically significant. This indicates that using the Health Bar did not significantly change the time taken.

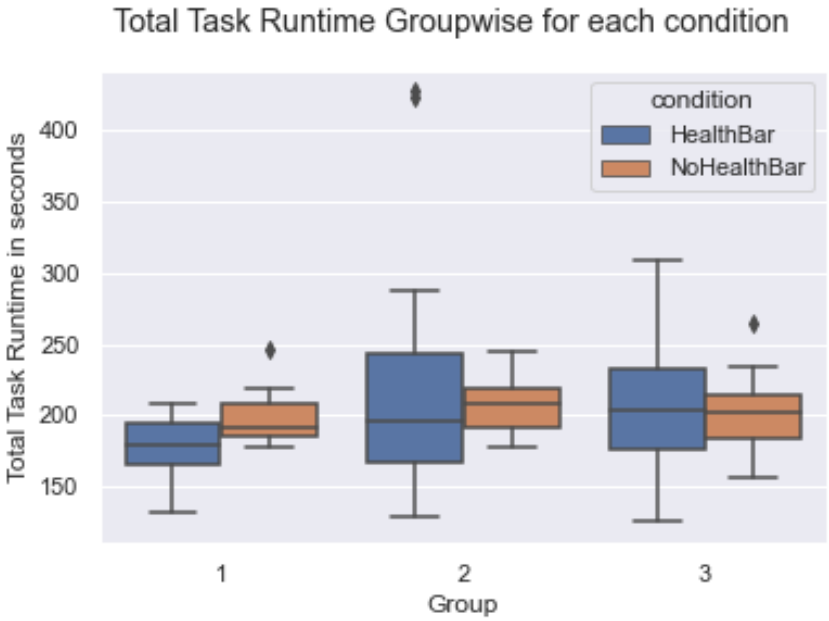


Figure 5.16: Task Runtime sorted group-wise for each condition

To assess the effect that the robot health bar had on the coordination, synchronisation and the overall strategy of the operator towards managing the MRS, the time difference between the first robot to complete the task and the last robot to complete the task was also calculated. The results are plotted in figure 5.17. The graph showed that the time difference in the runtime of the first and last robot was generally higher for the health bar condition, than the no health bar condition and the data had higher variance. Shapiro-Wilk tests on each of these showed they are normal distributions, hence a paired Two Tailed T-Test was used to check for a statistical difference. For group 2, $t = 2.795, p = 0.021$ and for group 3, $t = 3.553, p = 0.006$. $p < 0.05$ for both groups indicates that differences are statistically significant. That is, the median time difference in group

3 for the health bar condition ($Median = 72.996$) was 2.6 times higher than the no health bar condition ($Median = 28.024$); the median time difference in group 2 for the health bar condition ($Median = 35.437$) was 3.12 times higher than the no health bar condition ($Median = 11.335$).

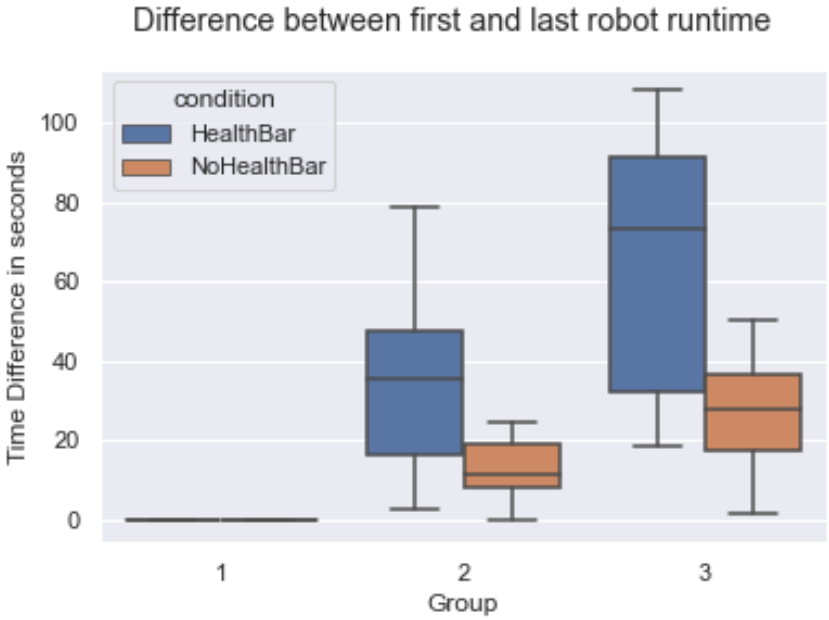


Figure 5.17: Difference in Runtime between first and last robot sorted group wise for each condition

To study the impact of different conditions on how the operator allocates attention to the robot, the frequency of the operator switching between the robots during the task runtime was calculated. This was only carried out for the multi-robot groups (group 2 and group 3). The cycling frequency was calculated as the total number of robot switches during runtime divided by the total time taken to complete the task. To standardise this graph and improve readability, the total number of robot switches per minute were plotted in figure 5.18.

The graph shows that per minute, the operator switches between the robots fewer times for the "Health Bar" condition than for the "No Health Bar" condition. Additionally, the number of switches per minute was higher overall for group 3 than for group 2. The operator made a median of approximately 5 and 9 LoA switches when using the health bar for groups 2 and 3, respectively. On the other hand, for the no health bar condition, the operator made a median of 7 and 10 LoA

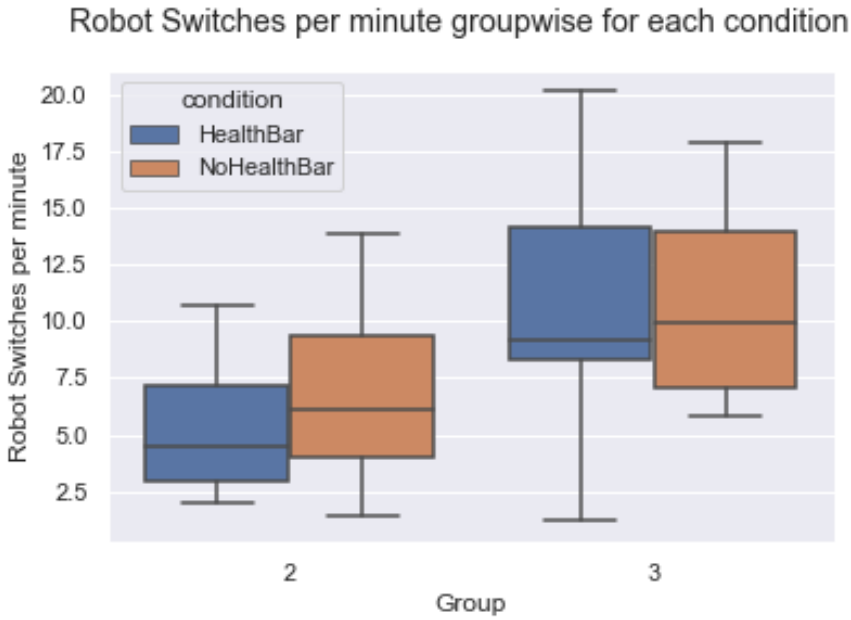


Figure 5.18: Number of robot switches per minute group wise for each condition

switches for groups 2 and 3, respectively. The paired two-tailed t-test for group 2 ($Tscore = -2.034, p = 0.07$) and group 3 ($Tscore = -0.311, p = 0.76$) showed no conclusive evidence that they were different distributions. However, given the non-normal distribution of the data, the Mann-Whitney test was used to further analyse the differences between the groups. The Mann-Whitney test is a non-parametric test that does not assume a normal distribution and is suitable for comparing medians between two independent groups. The test showed that differences between the groups for the health bar condition ($U = 78.000, p = 0.038$) and the no health bar condition ($U = 78.000, p = 0.038$) were statistically significant. The statistical analysis there shows that while the t-tests did not indicate significant differences in the number of LoA switches, the Mann-Whitney test revealed that the health bar condition significantly reduced the number of switches compared to the no health bar condition.

To calculate the impact of the conditions on the servicing of the robots, the duration each robot was in Manual Control was calculated across conditions and groups and plotted it in figure 5.19. The general trend showed that with an increase in the number of robots, the percentage of runtime each robot spent in manual control generally decreased. Pairwise comparisons for

group 1 ($Tscore = 1.737, p = 0.116$), and group 2 ($Tscore = 0.68843, p = 0.49950$) and group 3 ($Tscore = 0.64484, p = 0.524$) showed no significant differences.

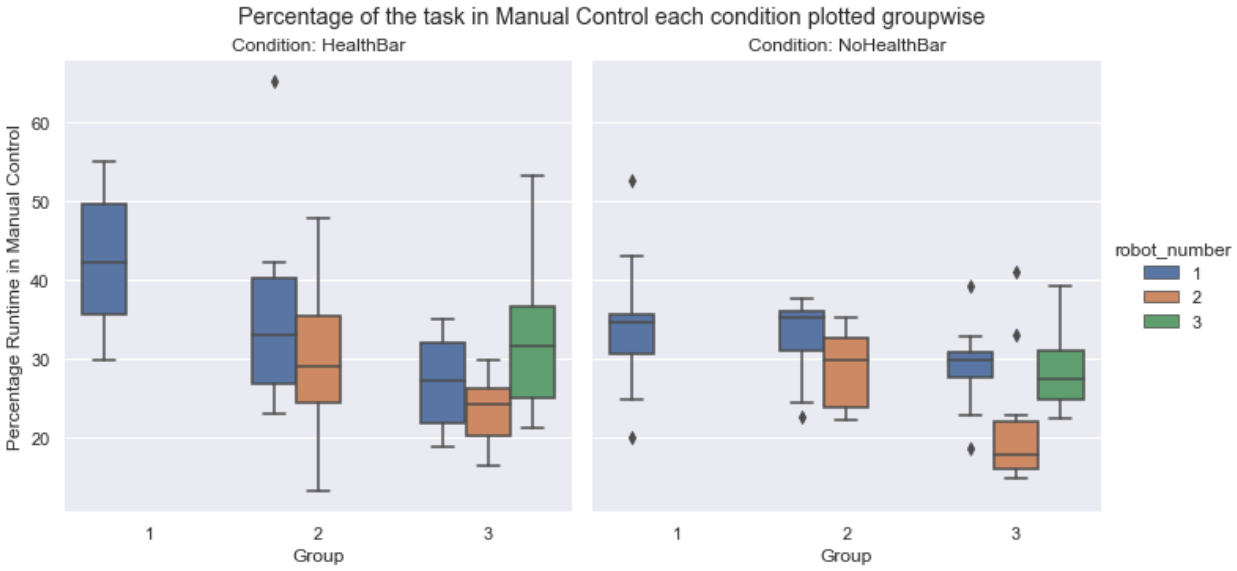


Figure 5.19: Time spent in manual control for each robot sorted group wise

5.2.3.3 Level of Autonomy Switches

The total number of Level Of Autonomy switches was calculated for each robot in each group and plotted in figure 5.20. The general trend showed little noticeable change in the number of LoA switches observed between the conditions. For group 1, Shapiro Wilk Test showed that the distributions were normal for both conditions, however the paired T-Test showed provided no evidence ($T - score = 0.00, p = 1.00$) to conclude the distributions were different. For group 2 since both distributions were not normal, the Wilcoxon test was used to make pairwise comparisons. The test provided insufficient evidence ($W = 53.50, p = 0.162$) to conclude the distributions were different. For group 3, a paired T-Test ($T - score = 3.22038, p = 0.00315$) returned significant differences between the distributions. This indicates that for group 3, the mean number of LoA switches carried out by the operator is significantly higher for the health bar condition than the no health bar condition, highlighting a notable impact of the health bar condition on operator behaviour for this

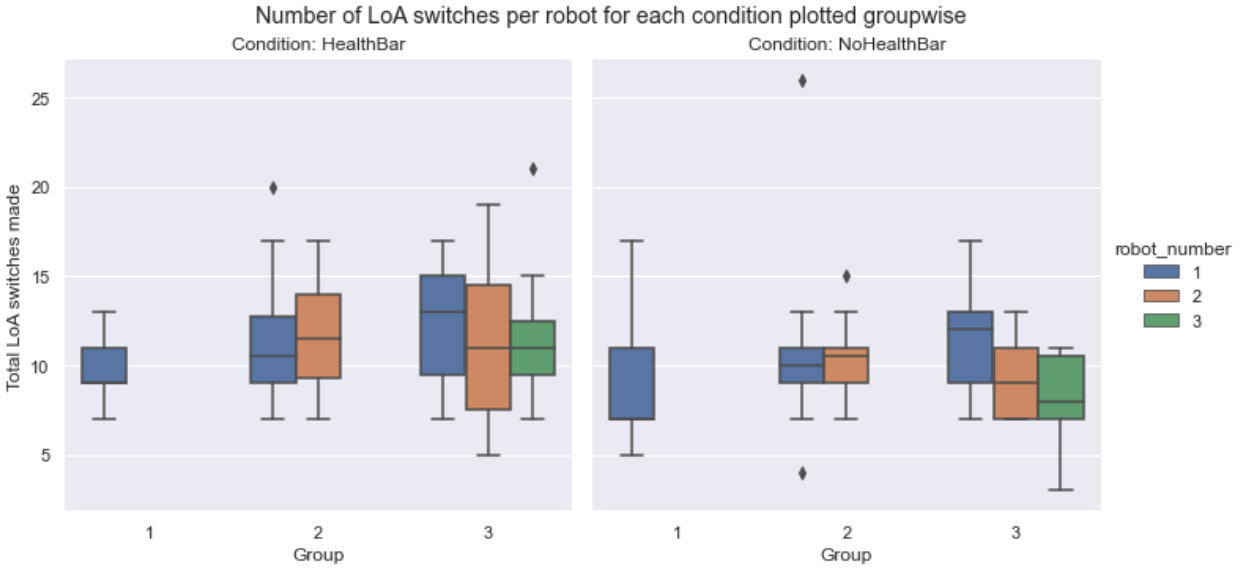


Figure 5.20: LoA switches made by the operators for each plotted group wise per condition group.

5.2.3.4 Cognitive Workload

Participants in this study filled out a NASA-TLX questionnaire after each trial to indicate the perceived workload from the remote robot navigation task. The median was calculated for both conditions groupwise, instead of the mean, as it is less sensitive to outliers than the mean. The results showed that the median mental and physical demand scores were lower for the 'Health Bar' condition compared to the 'No Health Bar' condition across all groups. The median temporal demand was lower for Groups 1 and 3 when operators used the Health Cues but higher for Group 2. For Groups 1 and 2, operators reported better performance and less effort when using the health bar but experienced lower effort and frustration during the 'No Health Bar' condition. Table 5.3 presents the median values of the Total Workload aggregated across all groups for each condition, showing that the total workload was lower for the 'Health Bar' condition compared to the "No Health Bar" condition for all groups. Comparisons between conditions were done for each group, and no statistical significance was found. The Wilcoxon Signed Rank test was used for these

comparisons, as it is a non-parametric test suitable for small sample sizes. Next, the Mann-Whitney U test was used on combined data from Groups 2 and 3 to compare across conditions, and no significance was found either. Only data from Groups 2 and 3 were combined instead of Groups 1, 2, and 3, because Group 1 does not have all the cues present in Groups 2 and 3, making it technically different. Therefore, although the median total workload for the 'Health Bar' condition was lower than that of the 'No Health Bar' condition for all groups, no statistically significant difference was observed in the data collected.

Table 5.3: Aggregated NASA TLX Scores for all groups between conditions

Median	Total Workload	
	Health Bar	No Health Bar
1 Robot	17.08	29.58
2 Robots	38.75	41.67
3 Robots	42.50	44.17

5.2.3.5 Usability, Trust and Reliability Evaluation

The usability questionnaire given in subsection A.4.2 was filled out by each participant after each trial. The first four questions were adapted from the UMUX questionnaire for a usability metric (Finstad, 2010). Q1, Q2 and Q4 were positively worded, and Q3 was negatively worded. The score for the answers therefore was calculated as: $((Q1-1)+(Q2-1)+(7-Q3)+(Q4-1))*4.1667$. The scores for each group were calculated and plotted in plotted in figure 5.21.

The trend showed that usability was generally higher for the Health Bar condition for group 1 and 2. For group 3, the median usability was same for both condition ($M = 79.17$). The average usability for the health bar condition was 80.97 across all groups, which was higher than the that of the no-health bar condition (73.47). Paired Wilcoxon Test for Group 1 ($W = 5.5, p = 0.043$) showed that the health bar condition had significantly higher usability than the No Health bar condition. However, paired Wilcoxon test for Group 2 and Group 3 together showed no statistically

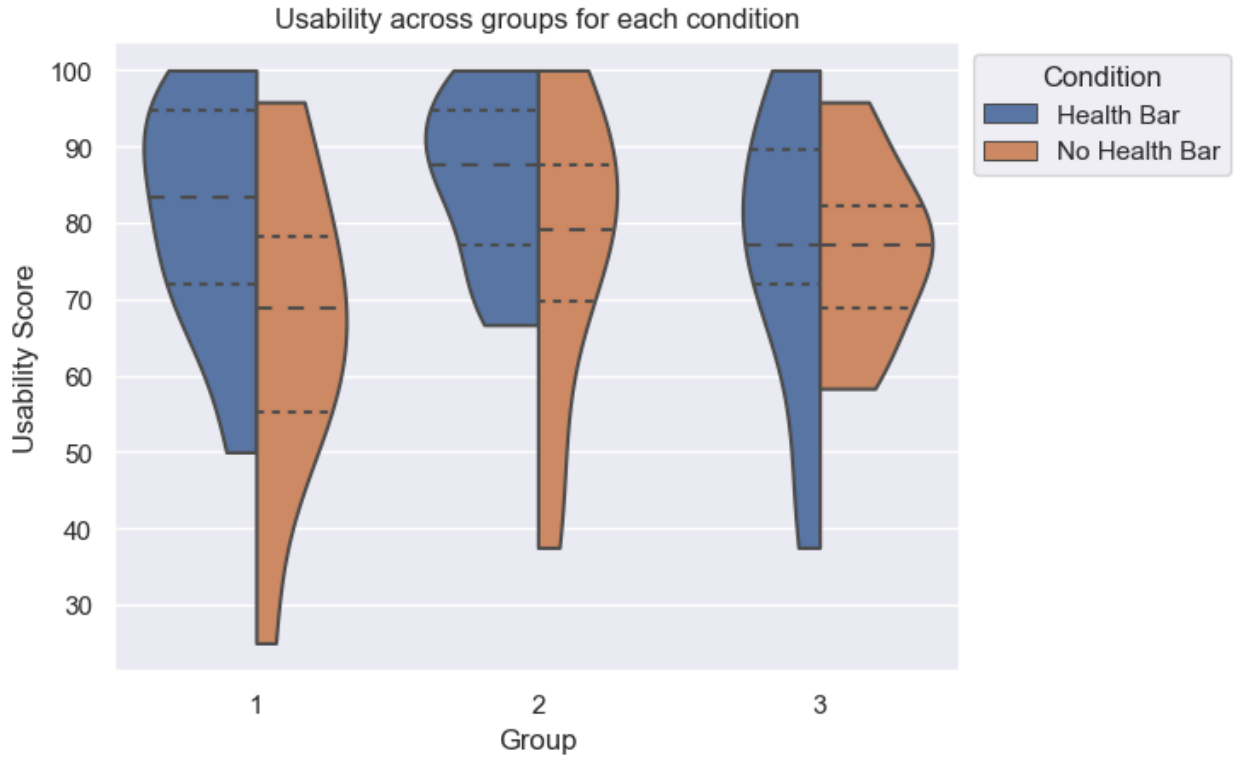
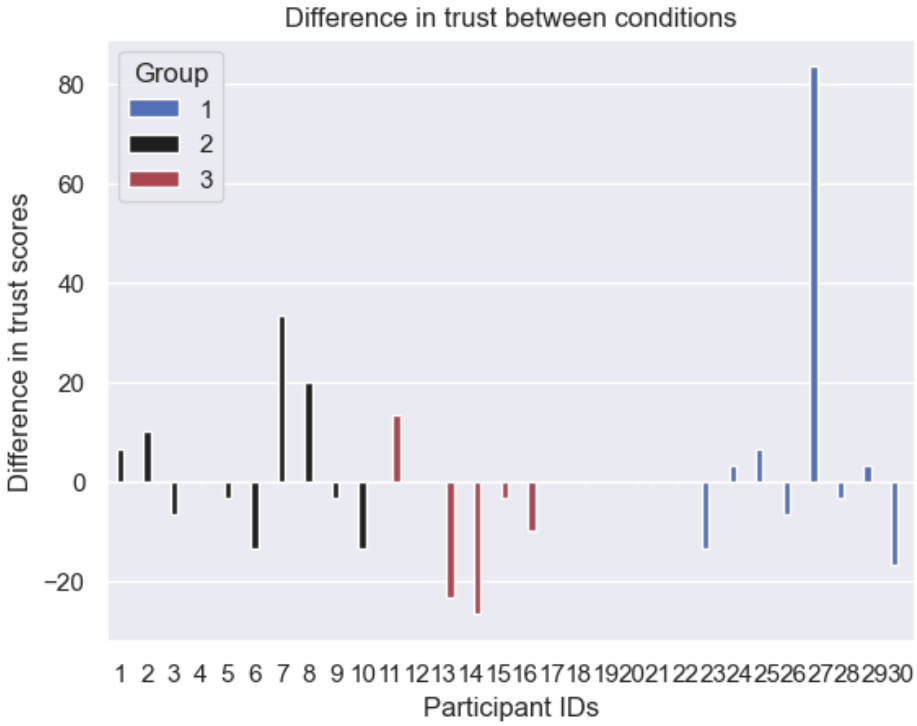


Figure 5.21: Usability Metric for User Experience Groupwise for Each condition (Higher is better)

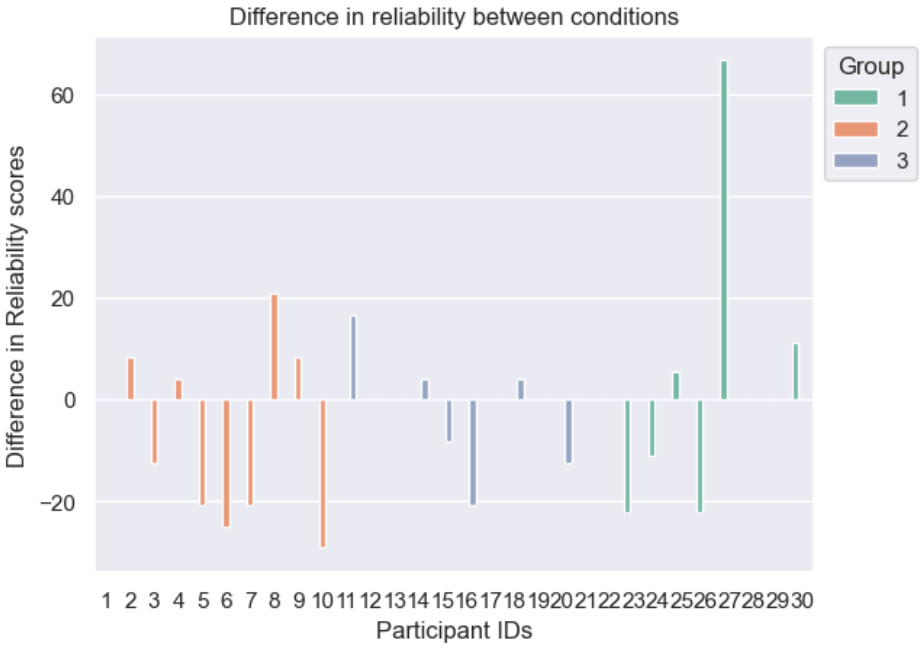
significant difference in usability between conditions.

In subsection A.4.2 Q5,Q6,Q7,Q9 focused on the reliability of the system and Q8, Q9, Q11, Q12 and Q13 focused on Trust in the system. Questions 5-9 were positively worded and Q11-Q13 were negatively worded. The difference between the perceived trust for the health bar condition and the no health bar condition was calculated and plotted in figure 5.22a. No evidence was found to indicate a significant difference in the perceived trust between the interface used for both conditions across groups. The total number of participants who trusted the interface without the Health Bar more were 4 in group 1, 5 in group 2, and 4 in group 3.

The reliability score for group 1 did not include the calculation for Q7, as haptic cues were not used for single robot experiments. The difference in reliability plot was calculated and plotted in figure 5.22b No significant differences were found across groups. In terms of perceived reliability, 3 participants in group 1, 5 participants in group 2, and 3 participants in group 3 felt



(a) Difference in Trust between conditions (Positive means higher trust in Health Bar condition.)



(b) Difference in Reliability between conditions (Positive means higher reliability of Health Bar condition.)

Figure 5.22: Differences in Trust and Reliability between conditions.

that the interface without the Health Bar was more reliable.

Table 5.4: Preferred Interface for each participant

Theme	1 Robot	2 Robot	3 Robot
Need Health Bar to Scale the System	10	8	8
Do not need Health Bar to Scale the System	0	2	2
Prefer using the Health Bar Interface	10	8	8
Prefer to use interface without Health Bar	0	2	2

5.2.3.6 Participant Qualitative Feedback

At the end of the experimental trials, participants were presented with three open-ended questions listed below:

Q1 If you had to do the same task again in a different environment, which interface design would you prefer and why?

Q2 If you had to operate a multi-robot system with more than 6 robots, which interface would you prefer and why?

Q3 Do you have any recommendations for improving the interface design?

Specifically, participants were asked about operating a system with more than six robots because previous research indicates that operator performance starts to saturate after this point (Olsen Jr and Wood, 2004). This was intended to understand their strategies and preferences for managing a larger multi-robot system, which could inform future interface design and operational strategies. They were encouraged to provide detailed responses and were given the opportunity to elaborate on their answers if they wished to do so. All participants using 1 robot felt that if they were doing the task in a different environment, they would prefer the interface with the Health bar and Low health sound cue. They unanimously also noted, that they would prefer having the

health cues if the task had to be scaled to operate more than 6 robots together. For group 2 and 3, 8 participants reported that they would prefer using the interface with the Health Bar and Low health Sound Cue if they were operating the robot in a new environment, or if the MRS had to be scaled to operate more than 6 robots. However, 2 participants in both these groups felt they preferred the interface without any cues based on the Robot Health added. These results are collated in the table 5.4. The answers to the open ended questions were collated, and analysed for common themes. These are tabulated in table A.4.1 given in the appendix. The general themes of the feedback indicate that generally users preferred the interface with the health bar, but got frustrated when they felt the health bar didn't represent the robot's actual condition. Participants felt that the sound alerts were useful to get their attention and keep them in the loop, but felt frustrated when it played when the operator has already noticed the robot is facing issues. Some participants noted they would like to be able to toggle the visibility of the health bar. Lastly, a few participants suggested that they would like if instead of switching between robots manually, there was a key to automatically switch to the robot with low health with manual control activated.

5.2.4 Discussion

Our experiments with Variable Autonomy Multi-Robot Systems (VA MRS) reveal two significant findings about the Triage-centric control interface design to facilitate Human Multi-Robot Teaming (HMRT). The first important finding is that a triage-centric control interface, which communicates information about robot performance degradation, enhances an operator's ability to manage multiple robots efficiently. This finding underscores the importance of performance degradation based sensory cues in multi-robot supervision. By integrating multi-modal cues such as a health bar and low health sound alerts, this interface design improved operators' median response times to robots in critical condition across all experimental groups by 6-10 seconds for each intervention. By providing a clear representation of the risk of robot failure, it served as an effective cognitive

tool, enabling operators to quickly assess and respond to the varying needs of multiple robots. This improvement in response time suggests that such multi-modal sensory cues can help mitigate the challenges of divided attention in multi-robot systems, leading to more efficient supervision and maintenance of robot fleets.

The second crucial finding is that the Triage-centric approach improves operational efficiency without increasing operator workload. Analysis of robot cycling frequency from figure 5.18 showed that operators using the interface with the health cues reduced their need to manually poll each robot's status. This streamlined process of monitoring task execution demonstrates how a well-designed, triage-focused interface can mitigate the challenges of divided attention in multi-robot systems, leading to more efficient supervision of robot fleets. Although not significant, the cognitive workload when operators used the health bar was generally lower for the health bar condition than the no health bar condition. This indicates that the change in the operation style was achieved without any change in the total operator workload. Therefore, these findings collectively suggest that the health bar improved operational efficiency and risk management without increasing operator workload, demonstrating its potential as an effective supervisory tool for triage in multi-robot systems.

Results from the navigation task show that when operating 1 or 2 robots, operators were able to use the robot health bar and sound alert to significantly ($p < 0.05$) reduce the time a robot spends in low health, thereby reducing the time the robot spends in situations where it is likely to fail or malfunction. Thus, when operating 1 or 2 robots, operators were able to significantly reduce the aggregate risk of robot failure throughout the runtime. For 3 robot experiments however, the results showed that operators were unable to significantly reduce the risk of robot failure for all the robots in the team together. Analysis of Figures 5.13 and 5.14 revealed that in Group 3, two out of three robots displayed relatively clustered box plots, suggesting that operators generally struggled to distribute their attention equally among all three robots. This observation is further supported by two data points. Figure 5.17, which demonstrates a statistically significant difference

($p<0.05$) in task completion times between the first and last robot for both group 2 and group 3 when the health bar and low health sound cue was utilised. Secondly, while the trends in figure 5.16 show indications that operators are able to finish the task quicker using the health bar for Group 1 and Group 2, they took significantly longer to finish the task in group 3. These findings indicate that operators did not consistently apply the health bar information uniformly across all robots. For 3 robots, operators were more likely to achieve a relatively staggered task completion. Despite no significant differences in the duration of Manual Control between conditions across groups, the Health Bar condition showed reduced aggregate robot failure risk and time spent in high-risk situations. This indicates that robot health based sensory cues promoted a more risk-aware operational strategy among participants. The health bar effectively enhanced risk perception and decision-making without increasing Manual Control time, suggesting improved efficiency in robot supervision.

Haptic cues were implemented exclusively for Groups 2 and 3, necessitating separate analyses: Group 1 was evaluated independently, while data from Groups 2 and 3 were combined. Questionnaire results showed that the Health Bar and Low Health sound alert significantly improved usability for single-robot operations, but no significant differences in trust, reliability, or usability were observed in multi-robot scenarios. This may be attributed to the perceived unreliability of sensory cues, particularly during complex tasks. Qualitative feedback indicated that while participants preferred the health bar interface, some experienced frustration due to its perceived unreliability, especially in Group 3 experiments. This discrepancy stemmed from misalignment between operators' expectations and the actual parameters governing the health bar's behaviour. Mental models formed during training and practice sometimes diverged from the system's actual functionality. For example, the robot health dropped only after 2-3 seconds of being stuck, a design choice to avoid temporary fluctuations due to path recalculations (please see 3). However, some operators, anticipating issues, provided assistance before this delay, leading them to perceive the health bar as less responsive than expected. This may explain the staggered operator style ob-

served in Group 3. Despite these challenges, almost all participants expressed a preference for the health bar interface in future tasks, with different environments or more robots. This suggests that the benefits of improved situational awareness and quicker response times outweighed the occasional frustrations, highlighting the potential of the health bar as an effective tool in multi-robot supervision systems.

5.2.5 Important Conclusions

This study explored if the robot vitals and robot health framework can help the operator with triage for VA MRS. This was accomplished by evaluating the impact of multi-modal sensory cues (Health Bar and Low Health Sound Alert) indicating robot performance degradation on the Variable Autonomy Multi-Robot systems operation. On a remote variable autonomy multi-robot navigation, the proposed cues were able to improve the time taken by the operator to respond to robots at risk of failure or malfunctioning for 1,2 or 3 robot teams. For single robot or dual robot teams, operators were able to use the health bar and sound alerts to reduce the aggregate risk of robot failure, and minimise the time robots spent in high risk scenarios, without increasing their overall cognitive workload. User feedback revealed a preference for the health bar interface despite occasional perceived unreliability, particularly in complex tasks. These findings show the potential benefit of sensory cues based on robot performance degradation multi-robot triage for remote variable autonomy multi-robot tasks in high-complexity scenarios, while also highlighting the need for further optimisation and careful design for teams with 3 or more robots. Future work needs to focus on improving the reliability of these cues, and developing non-invasive ways of improving the operators ability to uniformly distribute their attention across multiple robots, thereby facilitating Human Multi-Robot teaming.

Chapter Six

Conclusion

As robots become increasingly autonomous, the role of the operator will involve transitioning from continuously monitoring the task execution of each robot, to assisting them when they face performance degradation during runtime and servicing them when required. In this thesis, the problem of quantifying and communicating information about robot performance degradation to facilitate effective robot triage for Human Multi-Robot Teaming (HMRT) is addressed. The proposed Robot Vitals and Robot Health framework offers a systematic and intuitive way to combine the effects of different performance degrading factors and estimate and quantify the total performance degradation that a robot faces during runtime. Experiments with this framework highlighted the inherent randomness and unpredictability associated with robot behaviour when it encounters dynamic performance-degrading factors, as well as the difficulty of expressing total performance degradation simply and intuitively. Nevertheless, this work empirically demonstrates that runtime performance degradation can be estimated and quantified, with the Robot Vitals and Robot Health framework effectively capturing its effects. Analogous to human health, robot health intuitively indicates which robots are likely to fail and, therefore, require assistance to prevent failure. Using this framework as a robust foundation for state estimation in a higher level of abstraction, something under-explored in previous literature, streamlining robot triage and servicing in the context of Human-Robot Teaming (HRT) is explored. This was accomplished through Variable Autonomy

(VA) and systematically integrating performance degradation information into different aspects of the overall interaction paradigm with a human teammate.

Variable Autonomy enables robots to function under different levels or degrees of autonomy with different levels of human input. This is particularly useful when servicing robots undergoing severe performance degradation. While the literature contains a few examples of VA systems, there is a lack of a simple and intuitive approach to reason about the basis for switching between different levels or degrees of autonomy. This lack makes promoting trust in the system, ease of use, and design transparency difficult. A new Mixed Initiative Variable Autonomy system based on the Robot Vitals and Robot Health framework was developed, and its ability to enhance operator experience without sacrificing task performance or increasing workload was demonstrated. Experiments showed that robots can observe the environment, calculate the expected performance degradation, and switch to the Level of Autonomy (LoA) best suited to mitigate it. Additionally, two adaptive automations for Degree of Autonomy (DoA) regulation were proposed in this thesis. This can be a starting point for researchers to develop model predictive control systems based on robot performance degradation. Experiments in this thesis provide a solid methodological foundation for studies with human participants, especially the principled approach to training and testing while accounting for various confounding factors. These experiments demonstrated that instead of operators constantly monitoring a robot's task execution to detect when to service robots, the robots could report to operators when they need assistance, streamlining the triage process towards Multi-Robot System (MRS).

As a natural extension of enabling robots to autonomously request operator assistance based on their performance degradation, the focus was next on optimising how this information is communicated to the operator, especially as human-robot teaming is scaled to multi-robot systems. To accomplish this, the overall Human-Robot Interaction (HRI) paradigm was systematically redesigned to create a Triage-Centric User Experience. This was done in two steps. Inspired by video games, a robot health bar was first added to the operator's user interface to indicate a robot's

performance degradation. The impact of this health bar was evaluated on user experience, operation style, task performance, and the operator's cognitive workload. In a remote robot navigation task with Human Initiative Level of Autonomy Switching, experiments from my research showed that supplying information about robot performance degradation through a visual cue significantly changed the operation style, promoting a more risk-aware strategy to service the robot without additional cognitive workload. Thus, this study demonstrated that operators are able to use visual cues based on robot performance degradation to effectively triage and service robots during task runtime.

Based on insights from my experiments and previous studies, a novel Triage-Centric control interface was developed to enable a single operator to monitor task execution by a Variable Autonomy Multi-Robot System. Drawing on best practices from existing multidisciplinary literature, the overall interaction was specifically tuned to facilitate human multi-robot teaming without overloading the operator's cognitive capabilities while promoting system usability. Additionally, two multi-modal sensory cues based on robot performance degradation were added: a robot health bar and a low health sound alert. Experiments on a remote multi-robot navigation task using this interface confirmed that, while operating Variable Autonomy Multi-Robot Systems, operators were able to use the Health Bar and Low Health Sound Alert to lower response time to robots in need of assistance, while significantly promoting a sense of usability in the system without additional cognitive workload. These experiments highlighted the many difficulties and intricacies associated with designing experiments for human teaming with Variable Autonomy Multi-Robot Systems while minimising the effect of confounding factors such as operator style, personality traits, previous experience, and individual differences in the perceived reliability of autonomous systems. To the best of my knowledge, the experiments with Variable Autonomy Multi-Robot Systems in this thesis represent the first principled study of this kind, thus serving as a useful methodological foundation for subsequent inquiries in this field.

6.1 Contributions

The research conducted for this thesis has provided the following contributions to the field of Variable Autonomy, Human Multi-Robot Teaming, and HRI:

- **Estimating and Quantifying Runtime Performance Degradation:** This thesis introduces the 'Robot Vitals and Robot Health' framework, a simple and intuitive approach to quantifying online performance degradation in robots during task execution. A set of five vitals and a health metric were proposed, designed to generalise across various mobile robot morphologies, independent of the task or navigation algorithms used. The systematic empirical experimental methodology proposed in this thesis demonstrated that robot performance degradation can be quantified, and the 'Robot Vitals and Robot Health' framework adequately captures its effects. This methodology serves as a benchmark for evaluating the validity of new vitals, health metrics, or subsequent frameworks in quantifying robot performance degradation. The main advantage of this framework is that it serves as a robust state estimation framework for online competency assessment in robots.

- **Variable Autonomy Architectures:** Building on the Robot Vitals and Robot Health framework, advanced Variable Autonomy architectures were developed. Two adaptive automations for degree of autonomy switching were realised, optimising the robot's health during runtime. Additionally, an informed methodology was proposed for designing AI agents that can trigger Level of Autonomy (LoA) switches to minimise the risk of robot failure. This includes a Model Predictive Controller integrated into a Mixed Initiative Variable Autonomy system, which chooses the LoA estimated to yield the highest robot health for the next tessellation in the robot's path plan. A Mixed-Initiative Control system allowing both the robot operator and the robots to trigger LoA switches was also designed, with specific improvements to minimise control conflicts. Furthermore, a novel Human Initiative LoA switching

system for Multi-Robot Teams was developed, incorporating multi-modal sensory cues to facilitate quick triage and servicing. This system enhances the operator's ability to manage multiple robots efficiently, leveraging visual and auditory cues based on robot performance degradation to streamline decision-making in complex multi-robot environments.

- **Triage-Centric Multi-Robot Control Interface:** A novel control interface designed to enhance Human-Robot Interaction in Variable Autonomy systems was introduced in this thesis. This interface incorporates multi-modal cues, such as the 'robot health bar' and a 'low health sound alert', which informs operators about a robot's runtime performance degradation and facilitates quick robot servicing through LoA switches to minimise aggregate failure risk. This study provides a comprehensive evaluation of workload, operator trust, system transparency, and task performance with this feature. Additionally, this Multi-Robot Control Interface facilitates effective triage, robot servicing, and data visualisation through multi-modal sensory cues.

- **Experimental Studies on Variable Autonomy Multi-Robot Systems:** This thesis presents systematic experimental studies on a Variable Autonomy Multi-Robot System (VA MRS). A principled, systematic study on the impact of the 'robot health bar' and a 'low health sound alert' on operator performance in remote navigation tasks was conducted with 1, 2, and 3 robot teams. A rigorous methodological foundation for experiments with VA MRS was proposed to evaluate task performance, operator cognitive workload, trust, reliability, and usability. An in-depth analysis of human interaction with Human Initiative systems, both with and without a robot health bar, for single and multi-robot systems was provided. This research offered the first qualitative and quantitative report on metrics unique to VA MRS systems evaluation and provides empirical evidence and insights previously absent in the Human Multi-Robot Teaming literature.

6.2 Future Work

Here, a few promising directions for future research that emerged from this thesis are presented. The main focus of future research should be streamlining robot triage and scaling ideas in this thesis to multi-robot systems without overloading the operator's capabilities.

- **Minimising the need for operator intervention**: An interesting research direction involving the robot vitals framework would be developing AI controllers that mimic operator intervention for robots in different scenarios. This is important to ensure long-term autonomy for robots, e.g. rovers for undersea or extra-terrestrial exploration. The nature of such environments prevents direct human oversight for long periods, necessitating that robots self-correct issues to maintain continuous operation. Several reinforcement learning (Rigter, Lacerda, and Hawes, 2020) approaches can potentially be adapted for this by using the 'Robot Vitals and Robot Health' framework for robot state estimation. Whenever a robot detects an issue, the robot can either seek the operator's help at a cost or fix the issue autonomously and gain a reward. Using this technique, a robot can be taught to self-supervise its mission execution and reduce the operator effort required to control it. Lastly, the robot's vitals and health metrics may have to be updated in long-term missions to accommodate drift or wear and tear in components.

- **Mixed Initiative Level of Autonomy Switching to Multi-Robot Systems:** Chapter 4 provided insights into how Mixed Initiative Level of Autonomy Switching can be implemented for a robot using the Robot Vitals and Robot Health framework. Future research could explore the integration of a higher-level AI agent within Multi-Robot Systems (MRS) where each robot can self-regulate its autonomy to request operator support. This AI agent could serve as an intelligent mediator, prioritising which robot requires operator attention the most at any given moment, and automatically facilitating that interaction. Such a system could po-

tentially optimise the allocation of the operator's cognitive resources across the robot team, enhancing overall mission effectiveness. Investigating this type of AI-driven triage system's design, implementation, and impact represents a promising direction for advancing human-multi-robot teaming in complex operational environments. The most significant challenge in building such an AI agent is enabling true Mixed Initiative support, i.e. allowing both the operator and the AI agent to decide which robot most critically requires operator attention simultaneously. Resolving potential conflicts between these decision-makers and integrating their inputs seamlessly will be crucial for the system's effectiveness.

- **Multi-Modal Sensory Cues for a Triage Centric User Experience:** A crucial area for future research is optimizing how information is presented to operators in complex Multi-Robot Systems. This presentation must promote transparency and explainability to foster trustworthy robotic systems and enable meaningful human control. As these systems scale, effectively communicating the necessary information for quick triage and servicing decisions becomes increasingly challenging. The design must promote trust and transparency while managing the operator's cognitive load. Deciding which information to convey through visual, auditory, or haptic modalities, each with its strengths and limitations, is an open problem. Future studies could explore adaptive interfaces that adjust information presentation based on the situation, operator state or LoAs, using algorithms to prioritise real-time information and potentially integrating augmented or virtual reality interfaces (Jang et al., 2019). Researchers should also consider individual differences among operators to develop customisable interfaces that enhance human multi-robot teaming. The goal is to create an information ecosystem that maintains situational awareness across multiple robots without overwhelming the operator, facilitating timely and accurate decision-making in complex environments.

- **Applicability to Robotic Arms:** A promising frontier for the Robot Vitals and Robot Health framework is its application to robotic manipulator arms. For instance, during assembly

tasks or precision operations, the framework can identify subtle performance degradations caused by factors such as mechanical wear, excessive load, or environmental changes like temperature fluctuations. In industrial settings, the Robot Health metric can enhance collaborative robot setups involving multiple robotic arms and human operators. This scenario can be modelled as a game-theoretic problem where multiple agents (humans and robotic arms) collaborate (Mukherjee et al., 2022). Here, the robotic arms can take actions to improve their health based on external criteria or task-specific goals, optimizing their performance in a dynamic, multi-agent environment.

6.3 Closing Thoughts

Overall, I believe the advancements in Variable Autonomy presented in this thesis pave the way for significant improvements in real-world applications, particularly in collaborative multi-robot systems. As the autonomous capabilities of robots continue to evolve, their integration into various sectors such as disaster response, industrial automation, and remote inspection becomes increasingly viable. The implementation of the Robot Vitals and Robot Health framework can enhance the resilience and efficiency of these robotic systems, allowing for real-time adjustments in autonomy based on performance metrics. The main feature of this framework is it moves robot servicing to a higher level of abstraction, making it readily usable for both human operators and AI agents for competency assessment and quick servicing of robots. This will not only improve performance but also improve the ease-of-use for human operators, enabling them to manage multiple robots effectively even under high-stress conditions.

Looking towards the future, robots represent the next frontier in technological progress, poised to take on roles that are dull, dirty, dangerous, or those which humans cannot perform en masse. The vision includes fleets of robots exploring other planets, diving into the depths of our

oceans, and revolutionising mining operations with their ability to self-supervise and self-correct. The technology developed in this thesis can provide a useful contribution in these scenarios, helping robots to maintain optimal performance and adapt to unexpected challenges autonomously. As robots are increasingly deployed across a wide range of applications, ensuring seamless human-robot teaming becomes ever more important. This shift will not only enhance operational efficiency but can create new job opportunities, such as robot doctors who specialise in diagnosing and solving robotics 'health' issues. By leveraging advanced technologies like augmented and virtual reality and integrating sophisticated AI and machine learning algorithms, we can create a more seamless and efficient framework for human-robot interaction, driving innovation across various fields and ushering in a new era of robotic exploration and labour.

Appendix One

Appendix

A.1 Mixed-Initiative Variable Autonomy based on Robot Health

A.1.1 Simulink Block Diagrams

The DoA-MPC and LoA-MPC for both parameter optimisation and optimal control were implemented through Simulink in Matlab.

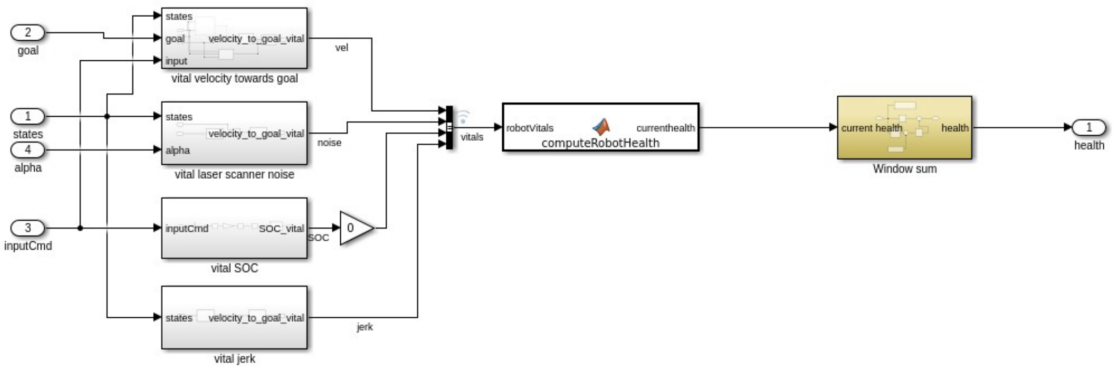


Figure A.1: Block to Compute The Robot Vitals and Robot Health

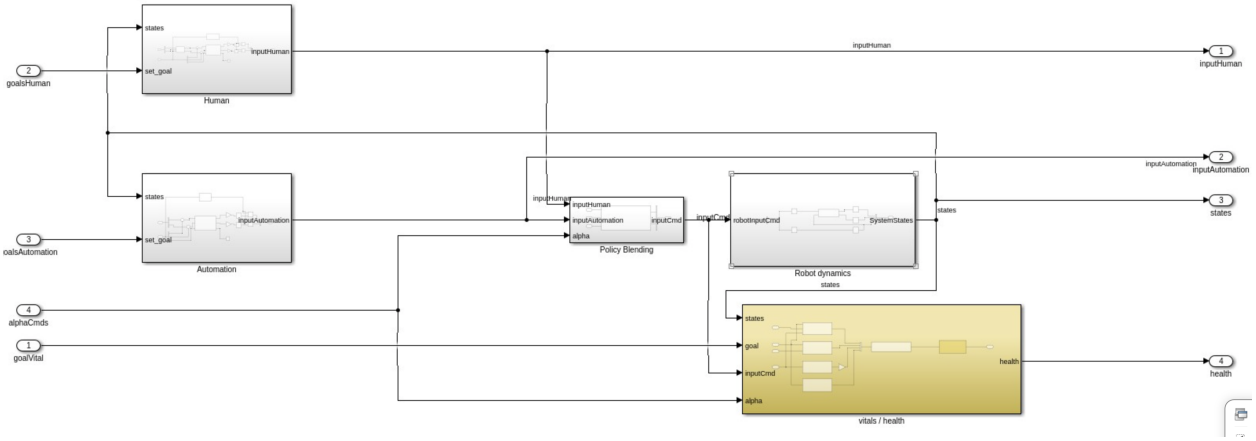


Figure A.2: The Human Machine System

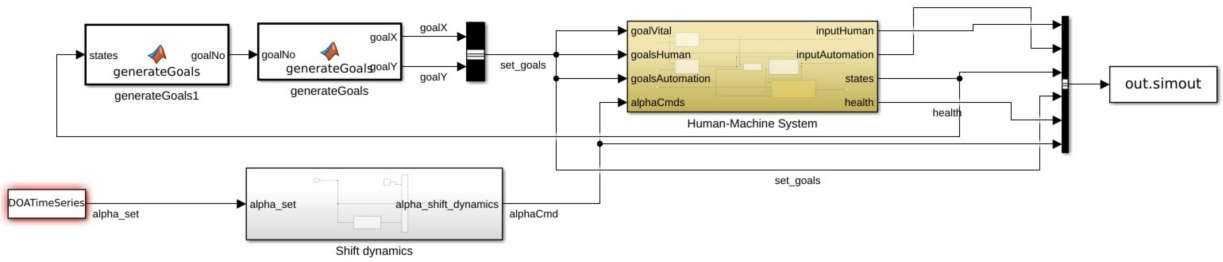


Figure A.3: Block Diagram of the MPC for DoA and LoA regulation

A.2 Visual cues for Human Initiative Level of Autonomy Switching

A.2.1 Background information questionnaire

The background questionnaire consisted of the following questions:

Q1: How often do you operate or use to operate remote controlled vehicles (e.g. Robots, Drones, Heavy machinery)

Q2: How often do you play or used to play video games involving driving, flight simulation and

third person shooters, RPG and sports?

Q3: Do you use AI (e.g. Autonomous Robots, Machine Learning Algorithms, AI Tools) for work?

Q4: Do you use AI (e.g. Personal Assistant) in your personal life?

The possible answers ranged between 1-5; i.e., least to most often.

A.2.2 Open-ended questions

Questions asked before participants had performed the experiment under both conditions are listed below:

Q1: What are you thinking as you look at this?

Q2: What is your first impression of this UI element?

Q3: What do you think this UI element does or will do?

All questions asked after participants had performed the experiment under both conditions are listed below:

Q1: Was anything surprising or did not perform as expected in either of the interfaces?

Q2: Was the interface without the 'Robot Health Bar ' easy to understand?

Q3: Was the interface with the health bar easy to understand?

Q4: What did you think about the colours used in the health bar when the health changed?

Q5: Was the LoA switching behaviour of the interface without the 'Robot Health Bar ' transparent?

Q6: Was the LoA switching behaviour of the interface with the 'Robot Health Bar ' transparent?

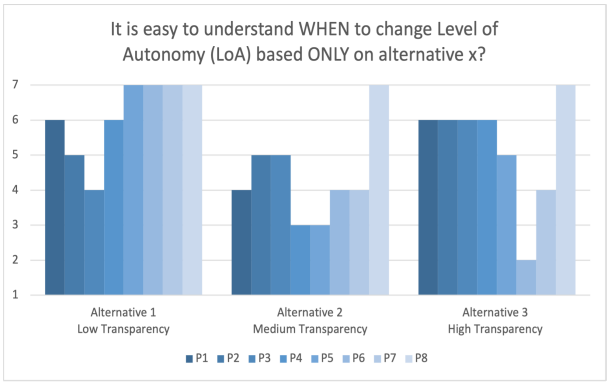
Q7: Is there anything else that you can think of, specific to the UI and UX to improve the use of LoA switching Robots?

A.2.3 Transparency/trust questionnaire

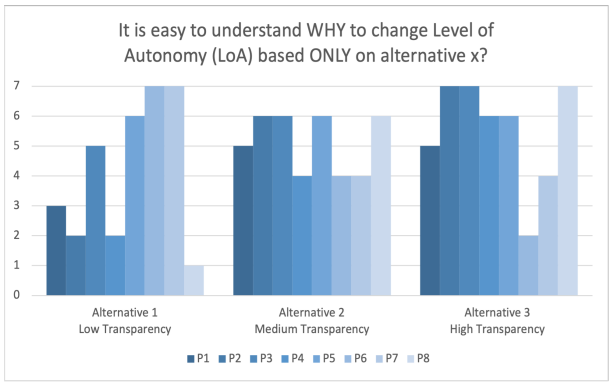
The transparency/trust questionnaire consisted of images of three different alternatives of a health bar ranging from low to high transparency and a 7-point Likert scale for each image (Figure A.3). The questions were the following pair, for each alternative:

Q1: It is easy to understand WHEN to change Level of Autonomy (LoA) based ONLY on alternative 1?

Q2: It is easy to understand WHY to change or not change Level of Autonomy (LoA) based ONLY on alternative 1?

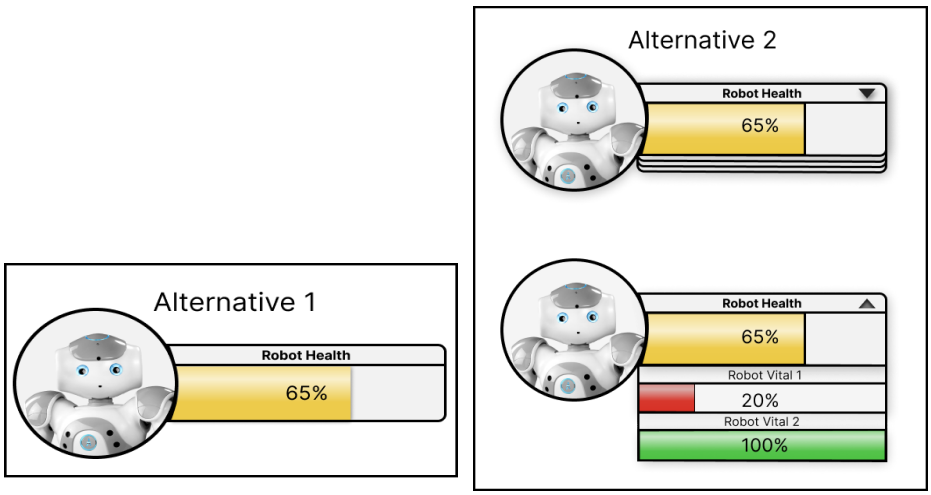


(a) Results for the questions regarding when to change LoA



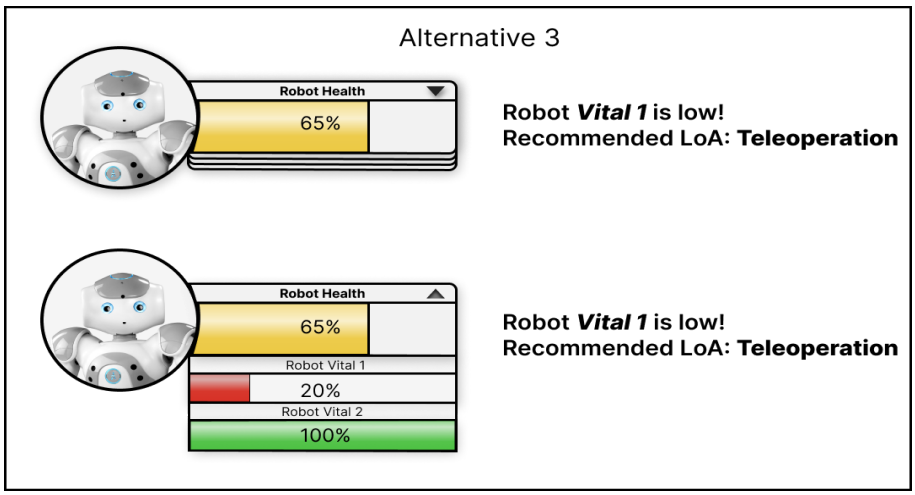
(b) Results for the questions regarding why to change LoA

Figure A.2: Results from questionnaire



(a) Alternative 1 - Low transparency

(b) Alternative 2 - Medium transparency



(c) Alternative 3 - High transparency

Figure A.3: Alternatives ranging from low to high transparency

A.3 Results

Table A.3.1: Correlation Between Background and Online Questionnaire

		Q1.1.1	Q1.1.2	Q1.1.3	Q1.1.4
It is easy to understand when to change Level of Autonomy (LoA) based ONLY on alternative 1?	Pearson's r	-0.874	0.045	-0.338	-0.336
	p-value	0.005**	0.916	0.413	0.416
It is easy to understand when to change Level of Autonomy (LoA) based ONLY on alternative 3?	Pearson's r	0.455	-0.277	0.802	0.120
	p-value	0.257	0.507	0.017*	0.778
It is easy to understand WHY to change Level of Autonomy (LoA) based ONLY on alternative 2?	Pearson's r	0.363	0.051	0.895	0.191
	p-value	0.376	0.905	0.003**	0.651
It is easy to understand WHY to change Level of Autonomy (LoA) based ONLY on alternative 3?	Pearson's r	0.433	-0.266	0.858	0.000
	p-value	0.284	0.524	0.006**	1.000

*p $<$ 0.05, **p $<$ 0.01

Note. Only selected results are displayed

A.4 Interaction Design for Variable Autonomy Multi-Robot Systems

A.4.1 Background information questionnaire

The background questionnaire consisted of the following questions:

Q1: How often do you operate or use to operate remote controlled vehicles (e.g. Robots, Drones, Heavy machinery)

Q2: How often do you play or used to play video games involving driving, flight simulation and third person shooters, RPG and sports?

Q3: Do you use AI (e.g. Autonomous Robots, Machine Learning Algorithms, AI Tools) for work?

Q4: Do you use AI (e.g. Personal Assistant) in your personal life?

The possible answers ranged between 1-5; i.e., least to most often.

A.4.2 Usability and Reliability Evaluation

For each trial, we first noted the experimental condition and participant number. The possible for the following questions ranged between 1-7; i.e. ranging from strong disagree to strongly agree.

Q1: The interface capabilities give me adequate information I need to perform the task

Q2: This interface was easy to use and understand

Q3: I have to spend too much time figuring out how to do the task from the information on the interface.

Q4: I could depend on the interface to work correctly every time.

Q5: The Visual Cues seem reliable.

Q6: The Audio Cues seem reliable.

Q7: The Haptic Cues seem reliable.

Q8: I can trust this interface to work every time.

Q9: If I did the same task with the interface again, it would be equally as helpful.

Q10: The interface is deceptive

Q11: The system behaves in an underhanded manner

Q12: I am suspicious of the system's intent, action, or outputs

Q13: I am wary of the system

Table A.4.1: Participant Feedback

Category	Notable Comments
Robot Health Bar is unreliable	• It increased my frrustration during runtime • Health bar increases mental demand, so irritates me all the more when it is wrong • I keep constantly comparing the health bar to the camera feed, and waiting for it to update correctly • Doesn't match expectations of what I thought it does based on the training • If it is unreliable, I would prefer my own judgement • Especially when multiple robots are facing problems, it stresses me out
Robot Health Bar is useful	• It helped me cut through the clutter of information • Felt complacent and out of the loop without the health bar and sound alert • I just saw the health bar instead of seeing the camera and map, so it was a simpler interface • I trusted the health bar to decide when to take control of the robot • Helps with deciding which robot to help, especially when it's a multi-robot system • Its sometimes unreliable, but its better if I am alerted than If I neglect the robot • It would be useful for high workload environments, but I could do without it for normal scenarios • It gives me the robot's perspective about the situation, which is useful when I can't see what the robot sees
Low Health Sound Alerts are helpful	• I can chat with people, and look at the screen only when there is a sound alert. • Incase I get distracted or miss something, it is bring my attention back • Sound alert helps with awareness, and keeps me in the loop • Makes me realise the robot is in pain and I can attend to it quicker
Low Health Sound Alerts are frustrating	• When multiple sound alerts are played in quick succession it is annoying and frustrating • Either have the sound alert, or the health bar. Both together makes me feel stressed • If I am already attending to a sick robot, I don't want a sound alert
Scope for Improvement	• Would prefer if I could toggle visibility the health bar for each robot • Would prefer to switch to the robot with low health automatically in Manual Control Mode

References

Abbink, David A et al. (2018). "A topology of shared control systems—finding common ground in diversity". In: *IEEE Transactions on Human-Machine Systems* 48.5, pp. 509–525.

Agrawal, Siddharth and Holly Yanco (2018). "Feedback methods in HRI: Studying their effect on Real-Time Trust and Operator Workload". In: *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. DOI: 10.1145/3173386.3177031.

Akail, Naoki, Luis Yoichi Moralesl, and Hiroshi Murase (2018). "Reliability estimation of vehicle localization result". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 740–747.

Ashcraft, C Chace, Michael A Goodrich, and Jacob W Crandall (2019). "Moderating Operator Influence in Human-Swarm Systems". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, pp. 4275–4282.

Audet, Charles and John E Dennis Jr (2002). "Analysis of generalized pattern searches". In: *SIAM Journal on optimization* 13.3, pp. 889–903.

Automation in horticulture review (July 2022). URL: https://www.gov.uk/government/publications/defra-led-review-of-automation-in-horticulture/automation-in-horticulture-review#executive-summary.

Backes, Paul G (1994). "Dual-arm supervisory and shared control task description and execution". In: *Robotics and autonomous systems* 12.1-2, pp. 29–54.

Barfoot, Timothy D (2024). *State estimation for robotics*. Cambridge University Press.

Basich, Connor et al. (2020). *Learning to Optimize Autonomy in Competence-Aware Systems*. DOI: 10.48550/ARXIV.2003.07745. URL: https://arxiv.org/abs/2003.07745.

Beer, Jenay M, Arthur D Fisk, and Wendy A Rogers (2014). "Toward a framework for levels of robot autonomy in human-robot interaction". In: *Journal of human-robot interaction* 3.2, p. 74.

Box, George EP (1958). "A note on the generation of random normal deviates". In: *Ann. Math. Statist.* 29, pp. 610–611.

Braun, Christian A, Michael Flad, and Sören Hohmann (2019). "A continuous and quantitative metric for the levels of automation". In: *IFAC-PapersOnLine* 52.19, pp. 37–42.

Braun, Christian Alexander, Aniketh Ramesh, Simon Rothfuß, et al. (2023). "Model predictive control of the degree of automation optimizing robot health". In: *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, pp. 000381–000386.

Braun, Christian Alexander, Aniketh Ramesh, Simon Rothfuss, et al. (2023). "Model predictive degree of automation regulation for mobile robots using robot vitals and robot health". In: *IFAC-PapersOnLine* 56.2, pp. 8345–8350.

Broggi, Alberto et al. (2015). "Proud—public road urban driverless-car test". In: *IEEE Transactions on Intelligent Transportation Systems* 16.6, pp. 3508–3519.

Brooks, Begum, and Yanco (2016). "Analysis of reactions towards failures and recovery strategies for autonomous robots". In: *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, pp. 487–492.

Bruemmer, David J, Donald D Dudenhoeffer, and Julie L Marble (2002). "Dynamic-Autonomy for Urban Search and Rescue." In: *AAAI mobile robot competition*. Menlo Park, CA, pp. 33–37.

Burns, Catherine M and John Hajdukiewicz (2017). *Ecological interface design*. CRC Press.

Calhoun, Gloria (2021). "Adaptable (Not Adaptive) Automation: The Forefront of Human–Automation Teaming". In: *Human Factors*. ISSN: 15478181. DOI: 10.1177/00187208211037457.

Calhoun, Gloria et al. (2021). "Performance-based adaptive automation: Number of task types and response time measures triggering level of automation changes". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 65. 1. SAGE Publications Sage CA: Los Angeles, CA, pp. 37–41.

Cao, Xuan et al. (2023). "Robot proficiency self-assessment using assumption-alignment tracking". In: *IEEE Transactions on Robotics* 39.4, pp. 3279–3298.

Capelli, Beatrice, Cristian Secchi, and Lorenzo Sabattini (2019). "Communication through motion: Legibility of multi-robot systems". In: *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, pp. 126–132.

Carlson, Jennifer and Robin R Murphy (2005). "How UGVs physically fail in the field". In: *IEEE Transactions on robotics* 21.3, pp. 423–437.

Carlucho, Ignacio et al. (2017). "Incremental Q-learning strategy for adaptive PID control of mobile robots". In: *Expert Systems with Applications* 80, pp. 183–199.

Cassel, David (Dec. 2021). *How drones and robots helped save Notre Dame*. URL: https://thenewstack.io/how-drones-and-robots-helped-save-notre-dame/.

Chamanbaz, Mohammadreza et al. (2017). "Swarm-enabling technology for multi-robot systems". In: *Frontiers in Robotics and AI* 4, p. 12.

Chavez-Garcia, R Omar et al. (2018). "Learning ground traversability from simulations". In: *IEEE Robotics and Automation letters* 3.3, pp. 1695–1702.

Chen, Jessie YC and Michael J Barnes (2014). "Human–agent teaming for multirobot control: A review of human factors issues". In: *IEEE Transactions on Human-Machine Systems* 44.1, pp. 13–29.

Chen, Jessie YC, Michael J Barnes, et al. (2011). "Effectiveness of RoboLeader for dynamic retasking in an urban environment". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 55. 1. SAGE Publications Sage CA: Los Angeles, CA, pp. 1501–1505.

Chen, Jessie YC, Shan G Lakhmani, et al. (2018). "Situation awareness-based agent transparency and human-autonomy teaming effectiveness". In: *Theoretical issues in ergonomics science* 19.3, pp. 259–282.

Chiou, Manolis, Goda Bieksaite, et al. (2016). "Human-initiative variable autonomy: An experimental analysis of the interactions between a human operator and a remotely operated mobile robot which also possesses autonomous capabilities". In: *2016 AAAI Fall Symposium Series*.

Chiou, Manolis, Georgios-Theofanis Epsimos, et al. (2022). "Robot-assisted nuclear disaster response: Report and insights from a field exercise". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4545–4552.

Chiou, Manolis, Nick Hawes, and Rustam Stolkin (2021). "Mixed-initiative variable autonomy for remotely operated mobile robots". In: *ACM Transactions on Human-Robot Interaction (THRI)* 10.4, pp. 1–34.

Chiou, Manolis, Nick Hawes, Rustam Stolkin, et al. (2015). "Towards the principled study of variable autonomy in mobile robots". In: *2015 IEEE international conference on systems, man, and cybernetics*. IEEE, pp. 1053–1059.

Chiou, Manolis, Rustam Stolkin, et al. (2016). "Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3581–3588. DOI: 10.1109/IROS.2016.7759527.

Chiou, Manolis, Mohammed Talha, and Rustam Stolkin (2019). "Learning effects in variable autonomy human-robot systems: how much training is enough?" In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 720–727. DOI: 10.1109/SMC.2019.8914558.

Comanici, Gheorghe and Doina Precup (2010). "Optimal policy switching algorithms for reinforcement learning". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*. AAMAS '10. Toronto, Canada: Inter-

national Foundation for Autonomous Agents and Multiagent Systems, pp. 709–714. ISBN: 9780982657119.

Committee, On-Road Automated Driving (ORAD) (2021). *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles*. SAE international.

Conlon, Nicholas, Nisar R Ahmed, and Daniel Szafir (2024). "A Survey of Algorithmic Methods for Competency Self-Assessments in Human-Autonomy Teaming". In: *ACM Computing Surveys* 56.7, pp. 1–31.

Cosenzo, Keryl et al. (2010). "Adaptive automation effects on operator performance during a reconnaissance mission with an unmanned ground vehicle". In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 54. 25. SAGE Publications Sage CA: Los Angeles, CA, pp. 2135–2139.

Cummings, Missy L (2004). "Human supervisory control of swarming networks". In: *2nd annual swarming: autonomous intelligent networked systems conference*, pp. 1–9.

Czaja, Sara J and Sankaran N Nair (2012). "Human factors engineering and systems design". In: *Handbook of human factors and ergonomics*, pp. 38–56.

Dahiya, Abhinav et al. (2022). "Scalable operator allocation for multirobot assistance: A restless bandit approach". In: *IEEE Transactions on Control of Network Systems* 9.3, pp. 1397–1408.

Dawson, Shameka et al. (2015). "Affecting operator trust in intelligent multirobot surveillance systems". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3298–3304.

De Visser, Ewart J et al. (2008). "Designing an adaptive automation system for human supervision of unmanned vehicles: A bridge from theory to practice". In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 52. 4. SAGE Publications Sage CA: Los Angeles, CA, pp. 221–225.

Delmerico, Jeffrey et al. (2019). "The current state and future outlook of rescue robotics". In: *Journal of Field Robotics* 36.7, pp. 1171–1191.

Desai, Munjal and Holly A Yanco (2005). "Blending human and robot inputs for sliding scale autonomy". In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* IEEE, pp. 537–542.

Dietterich, Thomas G (2000). "Hierarchical reinforcement learning with the MAXQ value function decomposition". In: *Journal of artificial intelligence research* 13, pp. 227–303.

Dietterich, Thomas G et al. (2002). "Ensemble learning". In: *The handbook of brain theory and neural networks* 2.1, pp. 110–125.

Dijkstra, E. W. (Dec. 1959). "A note on two problems in connexion with graphs". In: *Numer. Math.* 1.1, pp. 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390. URL: https://doi.org/10.1007/BF01386390.

Domingos, Pedro (2012). "A few useful things to know about machine learning". In: *Communications of the ACM* 55.10, pp. 78–87.

Doroodgar, Barzin, Yugang Liu, and Goldie Nejat (2014). "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims". In: *IEEE Transactions on Cybernetics* 44.12, pp. 2719–2732.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Endsley, Mica R (2017). "From here to autonomy: lessons learned from human–automation research". In: *Human factors* 59.1, pp. 5–27.

Englund, Madeleine (2023). *Unleashing the Power of AI Mobile Robots: A Journey into CognitiveLoad, Multimodal Signalling, and Trust in Multi-Robot Team*.

Ermacora, Gabriele, Stefano Rosa, and Basilio Bona (2015). "Sliding autonomy in cloud robotics services for smart city applications". In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pp. 155–156.

Etzi, Roberta et al. (2019). "Using virtual reality to test human-robot interaction during a collaborative task". In: *International Design Engineering Technical Conferences and Computers*

and Information in Engineering Conference. Vol. 59179. American Society of Mechanical Engineers, V001T02A080.

Evans, Dakota C. and Mary Fendley (Jan. 2017). "A multi-measure approach for connecting cognitive workload and Automation". In: *International Journal of Human-Computer Studies* 97, pp. 182–189. DOI: 10.1016/j.ijhcs.2016.05.008.

Finstad, Kraig (2010). "The usability metric for user experience". In: *Interacting with computers* 22.5, pp. 323–327.

Ganis, Giorgio and Rogier Kievit (Feb. 2016). "A new set of three-dimensional shapes for investigating mental rotation processes: validation data and stimulus set." In: DOI: 10.6084/m9.figshare.1045385.v13. URL: https://figshare.com/articles/dataset/A_new_set_of_three_dimensional_stimuli_for_investigating_mental_rotation_processes/1045385.

Goodrich, Michael A. and Alan C. Schultz (Jan. 2007). "Human-robot interaction: a survey". In: *Found. Trends Hum.-Comput. Interact.* 1.3, pp. 203–275. ISSN: 1551-3955. DOI: 10.1561/1100000005. URL: https://doi.org/10.1561/1100000005.

Gordon, Geoff and Ryan Tibshirani (2012). "Karush-kuhn-tucker conditions". In: *Optimization* 10.725/36, p. 725.

Gunning, David (2017). "Explainable artificial intelligence (xai)". In: *Defense Advanced Research Projects Agency (DARPA), nd Web* 2, p. 2.

Gutman, Dana, Samuel Olatunji, and Yael Edan (2021). "Evaluating Levels of Automation in Human–Robot Collaboration at Different Workload Levels". In: *Applied Sciences* 11.16. ISSN: 2076-3417. DOI: 10.3390/app11167340. URL: https://www.mdpi.com/2076-3417/11/16/7340.

Hardin, Benjamin and Michael A Goodrich (2009). "On using mixed-initiative control: A perspective for managing large-scale robotic teams". In: *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 165–172.

Hart, Peter E et al. (2017). "Shakey: From Conception to History." In: *AI Magazine* 38.1.

Hong, A et al. (2019). “Investigating human-robot teams for learning-based semi-autonomous control in urban search and rescue environments”. In: *Journal of Intelligent & Robotic Systems* 94.3-4, pp. 669–686.

Honig, Shanee and Tal Oron-Gilad (2018). “Understanding and resolving failures in human-robot interaction: Literature review and model development”. In: *Frontiers in psychology* 9, p. 351644.

Hou, Ming et al. (2010). “Optimizing operator–agent interaction in intelligent adaptive interface design: A conceptual framework”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41.2, pp. 161–178.

Howe, A Scott et al. (2020). “Maintenance-optimized modular robotic concepts for planetary surface ISRU excavators”. In: *2020 IEEE Aerospace Conference*. IEEE, pp. 1–15.

Hussein, Aya and Hussein Abbass (2018). “Mixed Initiative Systems for Human-Swarm Interaction: Opportunities and Challenges”. In: *2018 2nd Annual Systems Modelling Conference (SMC)*, pp. 1–8. DOI: 10.1109/SYSMC.2018.8509744.

Hussein, Aya, Leo Ghignone, et al. (2018). “Towards bi-directional communication in human-swarm teaming: A survey”. In: *arXiv preprint arXiv:1803.03093*.

Immerkaer, John (1996). “Fast noise variance estimation”. In: *Computer vision and image understanding* 64.2, pp. 300–302.

Janabi-Sharifi, Farrokh and Iraj Hassanzadeh (2010). “Experimental analysis of mobile-robot teleoperation via shared impedance control”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.2, pp. 591–606.

Jang, Inmo et al. (2019). “Omnipotent Virtual Giant for Remote Human-Swarm Interaction”. In: *arXiv preprint arXiv:1903.10064*.

Jian, Jiun-Yin, Ann M Bisantz, and Colin G Drury (2000). “Foundations for an empirically determined scale of trust in automated systems”. In: *International journal of cognitive ergonomics* 4.1, pp. 53–71.

Jiang, Shu and Ronald C Arkin (2015). “Mixed-initiative human-robot interaction: definition, taxonomy, and survey”. In: *2015 IEEE International conference on systems, man, and cybernetics*. IEEE, pp. 954–961.

Kaber, David B and Mica R Endsley (1997). “Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety”. In: *Process safety progress* 16.3, pp. 126–131.

Kalman, Rudolph Emil (1960). “A new approach to linear filtering and prediction problems”. In.

Kaufmann, Marcel et al. (2021). “Copilot mike: an autonomous assistant for multi-robot operations in cave exploration”. In: *2021 IEEE Aerospace Conference (50100)*. IEEE, pp. 1–9.

Kolling, Andreas et al. (2015). “Human interaction with robot swarms: A survey”. In: *IEEE Transactions on Human-Machine Systems* 46.1, pp. 9–26.

Kucukyilmaz, Ayse and Yiannis Demiris (2018). “Learning shared control by demonstration for personalized wheelchair assistance”. In: *IEEE transactions on haptics* 11.3, pp. 431–442.

Lee, John D and Katrina A See (2004). “Trust in automation: Designing for appropriate reliance”. In: *Human factors* 46.1, pp. 50–80.

Li, Huiyang et al. (2013). “Supporting human-automation collaboration through dynamic function allocation: The case of space teleoperation”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 57. 1. SAGE Publications Sage CA: Los Angeles, CA, pp. 359–363.

Lim, Kai Li et al. (2019). “Evolution of a reliable and extensible high-level control system for an autonomous car”. In: *IEEE Transactions on Intelligent Vehicles* 4.3, pp. 396–405.

Llanos, Stein C. and Kristine Jørgensen (2011). “Do Players Prefer Integrated User Interfaces? A Qualitative Study of Game UI Design Issues”. In: *DiGRA &#3911 - Proceedings of the 2011 DiGRA International Conference: Think Design Play*. DiGRA/Utrecht School of the Arts. URL: http://www.digra.org/wp-content/uploads/digital-library/11313.34398.pdf.

Losey, Dylan P et al. (2018). "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction". In: *Applied Mechanics Reviews* 70.1, p. 010804.

Mao, Zhenjiang, Siqi Dai, et al. (2024). "Zero-shot Safety Prediction for Autonomous Robots with Foundation World Models". In: *arXiv preprint arXiv:2404.00462*.

Mao, Zhenjiang, Carson Sobolewski, and Ivan Ruchkin (2023). "How safe am i given what i see? calibrated prediction of safety chances for image-controlled autonomy". In: *arXiv preprint arXiv:2308.12252*.

Marder-Eppstein, Eitan (2024). *move_base*. ROS Wiki. URL: https://wiki.ros.org/move_base.

Marder-Eppstein, Eitan, Eric Berger, et al. (2010a). "The office marathon: Robust navigation in an indoor office environment". In: *2010 IEEE international conference on robotics and automation*. IEEE, pp. 300–307.

— (2010b). "The office marathon: Robust navigation in an indoor office environment". In: *2010 IEEE international conference on robotics and automation*. IEEE, pp. 300–307.

Marr, Bernard (Oct. 2017). *The 4 DS of robotization: Dull, dirty, dangerous and dear*. URL: https://www.forbes.com/sites/bernardmarr/2017/10/16/the-4-ds-of-robotization-dull-dirty-dangerous-and-dear/.

MathWorks (2024). *Pattern Search*. https://www.mathworks.com/help/gads/patternsearch.html. Accessed: 2024-07-01.

Matsuzaki, Shigemichi, Hiroaki Masuzawa, and Jun Miura (2022). "Image-based scene recognition for robot navigation considering traversable plants and its manual annotation-free training". In: *IEEE Access* 10, pp. 5115–5128.

Mendoza, Juan Pablo, Manuela M Veloso, and Reid Simmons (2012). "Mobile robot fault detection based on redundant information statistics". In.

Methnani, Leila et al. (2021). "Let me take over: Variable autonomy for meaningful human control". In: *Frontiers in Artificial Intelligence* 4, p. 737072.

Mi, Zhen-Qiang and Yang Yang (2013). "Human-robot interaction in UVs swarming: a survey". In: *International Journal of Computer Science Issues (IJCSI)* 10.2 Part 1, p. 273.

Middleton, Stuart E et al. (2022). "Trust, regulation, and human-in-the-loop AI: within the European region". In: *Communications of the ACM* 65.4, pp. 64–68.

Miller, Christopher A (2018). "The risks of discretization: what is lost in (even good) levels-of-automation schemes". In: *Journal of Cognitive Engineering and Decision Making* 12.1, pp. 74–76.

Miller, Ian D et al. (2022). "Stronger Together: Air-Ground Robotic Collaboration Using Semantics". In: *IEEE Robotics and Automation Letters* 7.4, pp. 9643–9650.

Minculete, G and V Păstae (2023). "Essential approaches to combat the use of drones. Specific elements of the armed conflict in Ukraine". In: *Bulletin of" Carol I" National Defence University* 12.4, pp. 208–224.

Mizuno, Kei et al. (2011). "Mental fatigue caused by prolonged cognitive load associated with sympathetic hyperactivity". In: *Behavioral and Brain Functions* 7.1, p. 17. DOI: 10.1186/1744-9081-7-17.

Moniruzzaman, MD et al. (2022). "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey". In: *Robotics and Autonomous Systems* 150, p. 103973.

Morales, Cecilia G et al. (2019). "Interaction needs and opportunities for failing robots". In: *Proceedings of the 2019 on designing interactive systems conference*, pp. 659–670.

Morris, Ben (Apr. 2024). *How robots are taking over warehouse work*. URL: https://www.bbc.co.uk/news/business-68639533.

Mostafa, Salama A, Mohd Sharifuddin Ahmad, and Aida Mustapha (2019). "Adjustable autonomy: a systematic literature review". In: *Artificial Intelligence Review* 51, pp. 149–186.

Mukherjee, Debasmita et al. (2022). "A survey of robot learning strategies for human-robot collaboration in industrial settings". In: *Robotics and Computer-Integrated Manufacturing* 73, p. 102231.

Murphy, Robin R (2014). *Disaster robotics*. MIT press.

Murphy, Robin R and Satoshi Tadokoro (2019). "User interfaces for human-robot interaction in field robotics". In: *Disaster Robotics: Results from the ImPACT Tough Robotics Challenge*, pp. 507–528.

Musić, Selma and Sandra Hirche (2017). "Control sharing in human-robot team interaction". In: *Annual Reviews in Control* 44, pp. 342–354.

Nielsen, Jakob (1994). *Usability engineering*. Morgan Kaufmann.

Nistér, David, Oleg Naroditsky, and James Bergen (2006). "Visual odometry for ground vehicle applications". In: *Journal of Field Robotics* 23.1, pp. 3–20.

Nittala, Sai KR et al. (2018). "Pilot skill level and workload prediction for sliding-scale autonomy". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1166–1173.

Olsen Jr, Dan R and Stephen Bart Wood (2004). "Fan-out: Measuring human control of multiple robots". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 231–238.

Onur, KOCA, Ozgur Turay Kaymakci, and Muharrem Mercimek (2020). "Advanced Predictive Maintenance with Machine Learning Failure Estimation in Industrial Packaging Robots". In: *2020 International Conference on Development and Application Systems (DAS)*. IEEE, pp. 1–6.

Panagopoulos, Dimitris et al. (2022). "A Hierarchical Variable Autonomy Mixed-Initiative Framework for Human-Robot Teaming in Mobile Robotics". In: *2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6. DOI: 10.1109/ICHMS56717.2022.9980686.

Pappas, Pantelis et al. (2020). "VFH+ based shared control for remotely operated mobile robots". In: *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 366–373. DOI: 10.1109/SSRR50563.2020.9292585.

Parasuraman, Raja, Toufik Bahri, et al. (1992). “Theory and design of adaptive automation in aviation systems”. In: *Naval Air Warfare Center, Warminster, PA, Tech. Rep. NAWCADWAR-92*, pp. 033–60.

Parasuraman, Raja, Thomas B Sheridan, and Christopher D Wickens (2000). “A model for types and levels of human interaction with automation”. In: *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30.3, pp. 286–297.

Patel, Jayam, Prajankya Sonar, and Carlo Pinciroli (2022). “On multi-human multi-robot remote interaction: a study of transparency, inter-human communication, and information loss in remote interaction”. In: *Swarm Intelligence* 16.2, pp. 107–142.

Patle, BK et al. (2019). “A review: On path planning strategies for navigation of mobile robot”. In: *Defence Technology* 15.4, pp. 582–606.

Perille, Daniel et al. (2020). “Benchmarking metric ground navigation”. In: *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, pp. 116–121.

Pizzagalli, SL, V Kuts, and T Otto (2021). “User-centered design for Human-Robot Collaboration systems”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 1140. 1. IOP Publishing, p. 012011.

Pohlt, Clemens et al. (2018). “Effects on user experience during human-robot collaboration in industrial scenarios”. In: *2018 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, pp. 837–842.

Prágr, Miloš and Jan Faigl (2019). “Terrain Learning Using Time Series of Ground Unit Traversal Cost”. In: *International Conference on Modelling and Simulation for Autonomous Systesm*. Springer, pp. 97–107.

Prati, Elisa et al. (2021). “How to include User eXperience in the design of Human-Robot Interaction”. In: *Robotics and Computer-Integrated Manufacturing* 68, p. 102072.

Prewett, Matthew S et al. (2010a). “Managing workload in human–robot interaction: A review of empirical studies”. In: *Computers in Human Behavior* 26.5, pp. 840–856.

Prewett, Matthew S. et al. (2010b). "Managing workload in human–robot interaction: A review of empirical studies". In: *Computers in Human Behavior* 26.5. Advancing Educational Research on Computer-supported Collaborative Learning (CSCL) through the use of gStudy CSCL Tools, pp. 840–856. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2010.03.010. URL: https://www.sciencedirect.com/science/article/pii/S0747563210000506.

Qi, Juntong et al. (2016). "Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake". In: *Journal of Field Robotics* 33.3, pp. 290–321.

Ramesh, Aniketh, Christian Alexander Braun, et al. (2023). "Experimental evaluation of model predictive mixed-initiative variable autonomy systems applied to human-robot teams". In: *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 5291–5298.

Ramesh, Aniketh, Manolis Chiou, and Rustam Stolkin (2021). "Robot vitals and robot health: An intuitive approach to quantifying and communicating predicted robot performance degradation in human-robot teams". In: *Companion of the 2021 ACM/IEEE international conference on human-robot interaction*, pp. 303–307.

Ramesh, Aniketh, Madeleine Englund, et al. (2023). "Robot health indicator: A visual cue to improve level of autonomy switching systems". In: arXiv preprint arXiv:2303.06776.

Ramesh, Aniketh, Rustam Stolkin, and Manolis Chiou (2022). "Robot vitals and robot health: Towards systematically quantifying runtime performance degradation in robots under adverse conditions". In: *IEEE Robotics and Automation Letters* 7.4, pp. 10729–10736.

Reinmund, Tyler et al. (2024). "Variable Autonomy through Responsible Robotics: Design Guidelines and Research Agenda". In: *ACM Transactions on Human-Robot Interaction* 13.1, pp. 1–36.

Rigter, Marc, Bruno Lacerda, and Nick Hawes (2020). "A framework for learning from demonstration with minimal human effort". In: *IEEE Robotics and Automation Letters* 5.2, pp. 2023–2030.

Rizk, Yara, Mariette Awad, and Edward W Tunstel (2019). "Cooperative heterogeneous multi-robot systems: A survey". In: *ACM Computing Surveys (CSUR)* 52.2, pp. 1–31.

Rosenfeld, Ariel et al. (2017). "Intelligent agent supporting human–multi-robot team collaboration". In: *Artificial Intelligence* 252, pp. 211–231.

Rothfuß, Simon et al. (2022). "A Negotiation-Theoretic Framework for Control Authority Transfer in Mixed-Initiative Robotic Systems". In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 921–928. DOI: 10.1109/SMC53654.2022.9945196.

Ruan, Tianshu et al. (2022). "A Taxonomy of Semantic Information in Robot-Assisted Disaster Response". In: *arXiv preprint arXiv:2210.00125*.

Ruff, Heath et al. (2018). "Comparison of adaptive, adaptable, and hybrid automation for surveillance task completion in a multi-task environment". In: *Proceedings of the Human Factors and Ergonomics Society* 1, pp. 155–159. ISSN: 10711813. DOI: 10.1177/1541931218621036.

Ruff, Heath A, Sundaram Narayanan, and Mark H Draper (2002). "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles". In: *Presence: Teleoperators & Virtual Environments* 11.4, pp. 335–351.

S.L, Herogra Especiales (July 2023). URL: https://herograespeciales.com/en/fertilization-using-drones/.

Sánchez-Ibáñez, José Ricardo, Carlos J Pérez-del-Pulgar, and Alfonso García-Cerezo (2021). "Path planning for autonomous mobile robots: A review". In: *Sensors* 21.23, p. 7898.

Sanders, Tracy L. et al. (2014). "The influence of modality and transparency on trust in human-robot interaction". In: *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 156–159. DOI: 10.1109/CogSIMA.2014.6816556.

Santoni de Sio, Filippo and Jeroen Van den Hoven (2018). "Meaningful human control over autonomous systems: A philosophical account". In: *Frontiers in Robotics and AI* 5, p. 15.

Schelble, Beau G et al. (2022). “Let’s think together! Assessing shared mental models, performance, and trust in human-agent teams”. In: *Proceedings of the ACM on Human-Computer Interaction* 6.GROUP, pp. 1–29.

Schneider, H and PM Frank (1994). “Fuzzy logic based threshold adaption for fault detection in robots”. In: *Proc. 3rd IEEE Conf. Contr. Applicat*, pp. 24–26.

Sevastopoulos, Christos, Katerina Maria Oikonomou, and Stasinos Konstantopoulos (2019). “Improving traversability estimation through autonomous robot experimentation”. In: *Computer Vision Systems: 12th International Conference, ICVS 2019, Thessaloniki, Greece, September 23–25, 2019, Proceedings 12*. Springer, pp. 175–184.

Sheridan, Thomas B, William L Verplank, and TL Brooks (1978). “Human/computer control of undersea teleoperators”. In: *NASA. Ames Res. Center The 14th Ann. Conf. on Manual Control*.

Smith, Gary B et al. (2019). “The National Early Warning Score 2 (NEWS2)”. In: *Clinical Medicine* 19.3, pp. 260–260.

Soni, Jayesh, Nagarajan Prabakar, and Jong-Hoon Kim (2017). “Prediction of Component Failures of Telepresence Robot with Temporal Data”. In: *30th Florida Conference on Recent Advances in Robotics*.

Steinfeld, Aaron (2004). “Interface lessons for fully and semi-autonomous mobile robots”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*. Vol. 3. IEEE, pp. 2752–2757.

Stolkin, R. et al. (2022). *Status, Barriers and Cost-Benefits of Robotic and Remote Systems Applications in Nuclear Decommissioning and Radioactive Waste Management*. Tech. rep. OECD Nuclear Energy Agency.

Storms, Justin, Kevin Chen, and Dawn Tilbury (2017). “A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay”. In: *The International Journal of Robotics Research* 36.5-7, pp. 820–839.

Studenski, Stephanie et al. (2003). "Physical performance measures in the clinical setting". In: *Journal of the American Geriatrics Society* 51.3, pp. 314–322.

Swamy, Gokul et al. (2020). "Scaled autonomy: Enabling human operators to control robot fleets". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5942–5948.

Theodorou, Andreas, Robert H Wortham, and Joanna J Bryson (2017). "Designing and implementing transparency for real time inspection of autonomous robots". In: *Connection Science* 29.3, pp. 230–241.

Turin, GL (1995). "An introduction to matched filters". In: *SPIE MILESTONE SERIES MS* 105, pp. 128–128.

Tzafestas, Spyros G (2013). *Introduction to mobile robot control*. Elsevier.

— (2018). "Mobile robot control and navigation: A global overview". In: *Journal of Intelligent & Robotic Systems* 91, pp. 35–58.

UKRI (2020). URL: https://iuk.ktn-uk.org/wp-content/uploads/2020/09/ISCF-Robotics-Brochure-2020.pdf.

Vagia, Marialena, Aksel A Transeth, and Sigurd A Fjerdingen (2016a). "A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed?" In: *Applied ergonomics* 53, pp. 190–202.

— (2016b). "A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed?" In: *Applied Ergonomics* 53, pp. 190–202. ISSN: 18729126. DOI: 10.1016/j.apergo.2015.09.013.

Valero, Alberto et al. (2008). "Adaptative human-robot interaction for mobile robots". In: *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, pp. 243–248.

Valero-Gomez, Alberto, Paloma De La Puente, and Miguel Hernando (2011). "Impact of two adjustable-autonomy models on the scalability of single-human/multiple-robot teams for exploration missions". In: *Human factors* 53.6, pp. 703–716.

Valero-Gomez, Alberto, Paloma de la Puente, and Miguel Hernando (2011). "Impact of Two Adjustable-Autonomy Models on the Scalability of Single-Human/Multiple-Robot Teams for Exploration Missions". In: *Human Factors* 53.6. PMID: 22235531, pp. 703–716. DOI: 10.1177/0018720811420427. eprint: https://doi.org/10.1177/0018720811420427. URL: https://doi.org/10.1177/0018720811420427.

Van Turennout, P, G Honderd, and W Jongkind (1989). "Motion control of a mobile robot". In: *Proceedings. ICCON IEEE International Conference on Control and Applications*. IEEE, pp. 577–582.

Visinsky, Monica L, Joseph R Cavallaro, and Ian D Walker (1994). "Robotic fault detection and fault tolerance: A survey". In: *Reliability Engineering & System Safety* 46.2, pp. 139–158.

Visser, Ewart de and Raja Parasuraman (2011). "Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload". In: *Journal of Cognitive Engineering and Decision Making* 5.2, pp. 209–231.

Vitale, Jonathan et al. (2018). "Be more transparent and users will like you: A robot privacy and user experience design experiment". In: *Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction*, pp. 379–387.

Walker, Phillip et al. (2013). "Levels of automation for human influence of robot swarms". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 57. 1. SAGE Publications Sage CA: Los Angeles, CA, pp. 429–433.

Wandke*, Hartmut (2005). "Assistance in human–machine interaction: a conceptual framework and a proposal for a taxonomy". In: *Theoretical Issues in Ergonomics Science* 6.2, pp. 129–155.

Wang, Jijun, Michael Lewis, and Jeffrey S Gennari (2003). "Interactive simulation of the NIST USAR arenas". In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*. Vol. 2. IEEE, pp. 1327–1332.

Wedler, Armin et al. (2021). “Preliminary Results for the Multi-Robot, Multi-Partner, Multi-Mission, Planetary Exploration Analogue Campaign on Mount Etna”. In: *Proceedings of the International Astronautical Congress, IAC*.

Wickens, Christopher D (2002). “Multiple resources and performance prediction”. In: *Theoretical issues in ergonomics science* 3.2, pp. 159–177.

Wong, Choon Yue et al. (2010). “Single-human multiple-robot systems for urban search and rescue: Justifications, design and testing”. In: *2010 11th International Conference on Control Automation Robotics & Vision*. IEEE, pp. 579–584.

Wong, Cuebong et al. (2017). “An overview of robotics and autonomous systems for harsh environments”. In: *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, pp. 1–6.

Wray, Kyle Hollins, Luis Pineda, and Shlomo Zilberstein (2016). “Hierarchical approach to transfer of control in semi-autonomous systems”. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1285–1286.

Xu, Wei (2020). “From automation to autonomy and autonomous vehicles: Challenges and opportunities for human-computer interaction”. In: *Interactions* 28.1, pp. 48–53.

Yagoda, Rosemarie E and Douglas J Gillan (2012). “You want me to trust a ROBOT? The development of a human–robot interaction trust scale”. In: *International Journal of Social Robotics* 4, pp. 235–248.

Ye, C. (2007). “Navigating a Mobile Robot by a Traversability Field Histogram”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.2, pp. 361–372.

Yu, Lingli et al. (2011). “A particle filter and SVM integration framework for fault-proneness prediction in robot dead reckoning system”. In: *WSEAS Transactions on Systems* 10.11, pp. 363–375.

Zhou, Lifeng and Pratap Tokekar (2021). “Multi-robot coordination and planning in uncertain and adversarial environments”. In: *Current Robotics Reports* 2, pp. 147–157.

Zhu, Yuke et al. (2017). “Visual Semantic Planning Using Deep Successor Representations”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 483–492. DOI: 10.1109/ICCV.2017.60.