A framework for human-robot interaction with a mobile and telepresence enabled anthropomorphic robot

Pang, Wee Ching

2021

Pang, W. C. (2021). A framework for human-robot interaction with a mobile and telepresence enabled anthropomorphic robot. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/152630>

<https://hdl.handle.net/10356/152630>

<https://doi.org/10.32657/10356/152630>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 22:38:16 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**A FRAMEWORK FOR HUMAN-ROBOT INTERACTION
WITH A MOBILE AND TELEPRESENCE ENABLED
ANTHROPOMORPHIC ROBOT**

PANG WEE CHING

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2021

**A FRAMEWORK FOR HUMAN-ROBOT INTERACTION
WITH A MOBILE AND TELEPRESENCE ENABLED
ANTHROPOMORPHIC ROBOT**

PANG WEE CHING

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

19/01/2021

.....
Date

.....
Pang Wee Ching

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

19 January 2021

.....
Date

.....
A/Prof. Gerald Seet Gim Lee

Authorship Attribution Statement

This thesis contains material from [6 number] paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Section 4.1.1 is published as G. Seet, W. C. Pang, Burhan, I.-M. Chen, I. Viatcheslav, G. William, and C. Y. Wong, “A Design for a Mobile Robotic Avatar - Modular Framework,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, Zurich, Switzerland, 2012. DOI: <https://doi.org/10.1109/3DTV.2012.6365444>.

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet and Prof. Chen I-Ming provided the initial project direction.
- A/Prof Gerald Seet and I wrote the drafts of the manuscript. The manuscript was revised by A/Prof Gerald Seet.
- I designed and developed the software component of the work at the School of Mechanical and Aerospace Engineering. Dr. William Gu helped with the software implementation.
- Dr. Wong Choon Yue and Mr. Burhan designed and developed the hardware component of the work at the School of Mechanical and Aerospace Engineering. Dr. Viatcheslav helped with the electrical system.

Section 4.1.2 is published as W. C. Pang, C. Y. Wong, and G. Seet, “Design and development of EDGAR - A Telepresence Humanoid for Robot-Mediated Communication and Social Applications,” in *Proceedings of the International Conference on Control and Robotics Engineering (ICCRE)*, pp. 1–4, IEEE, 2016. DOI: <https://doi.org/10.1109/ICCRE.2016.7476147>.

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet provided the initial project direction and edited the manuscript drafts.
- I wrote the drafts of the manuscript. The manuscript was revised by Dr. Wong Choon Yue.

- I designed and developed the software component of the work at the School of Mechanical and Aerospace Engineering.
- Dr. Wong Choon Yue designed and developed the hardware component of the work at the School of Mechanical and Aerospace Engineering.

Section 4.5.4 is published as W. C. Pang, G. Seet and X. Yao. “A Study on High-level Autonomous Navigational Behaviors for Telepresence Applications.” *Presence: Teleoperators and Virtual Environments*, vol. 23, no. 2, pp. 155–171, 2014. DOI: https://doi.org/10.1162/PRES_a_00178.

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet provided the initial project direction.
- I wrote the drafts of the manuscript. The manuscript was revised by A/Prof Gerald Seet.
- I designed the study and co-implemented the work with Dr. Yao Xiling at the School of Mechanical and Aerospace Engineering. I also analyzed the data.

Section 4.5.4 is also published as W. C. Pang, G. Seet, and X. Yao, “A Multimodal Person-following System for Telepresence Applications,” in *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology (VRST)*, pp. 157–164, ACM, 2013. DOI: <https://doi.org/10.1145/2503713.2503722>

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet provided the initial project direction.
- I wrote the drafts of the manuscript. The manuscript was revised by A/Prof Gerald Seet.
- I designed the study and co-implemented the work with Dr. Yao Xiling at the School of Mechanical and Aerospace Engineering. I also analyzed the data.

Section 4.6.1 is published as W. C. Pang, Burhan, and G. Seet, “Design Considerations of a Robotic Head for Telepresence Applications,” in *Intelligent Robotics and Applications (Lecture Notes in Computer Science)*, vol. 7508, pp. 131–140, 2012. DOI: https://doi.org/10.1007/978-3-642-33503-7_14

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet provided the initial project direction and edited the manuscript drafts.

- I wrote the drafts of the manuscript. The manuscript was revised by A/Prof Gerald Seet.
- I designed and developed the software component of the work at the School of Mechanical and Aerospace Engineering.
- Mr. Burhan designed and developed the hardware component of the work at the School of Mechanical and Aerospace Engineering.

Section 5.2 and Section 5.5 are published as W. C. Pang, C. Y. Wong and G. Seet. “Exploring the Use of Robots for Museum Settings and for Learning Heritage Languages and Cultures at the Chinese Heritage Centre.” *Presence: Teleoperators and Virtual Environments*, vol. 26, no. 4, pp. 420–435, 2018. DOI:https://doi.org/10.1162/PRES_a_00306

The contributions of the co-authors are as follows:

- A/Prof Gerald Seet provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr Wong Choon Yue.
- I designed the study and implemented the work at the School of Mechanical and Aerospace Engineering. I also analyzed the data.
- I deployed the work at the Chinese Heritage Centre, with the assistance of A/Prof Gerald Seet and Dr. Wong Choon Yue.

19/01/2021

.....
Date

.....
Pang Wee Ching

ACKNOWLEDGMENTS

This research was supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Program.

First, I would like to express my sincere appreciation and gratitude to my supervisor, Prof. Gerald Seet, for his guidance and inspiring suggestions throughout my doctoral studies. His patience and insistent nudging me towards completion have been most helpful for the development of this dissertation. I would also like to thank the members of my thesis advisory committee, Prof. Cai Yiyu and Prof. Yeo Song Huat for their insightful comments and discussions.

The School of Mechanical and Aerospace Engineering at the Nanyang Technological University deserves a special thank you for providing a conducive environment in which I can maintain my full-time job and still work to complete the degree. I wish to thank Mr. Lim Eng Cheng, Mr. You Kim San and Ms. Agnes Tan from the Robotics Research Centre. They have provided me with much technical and administrative support during this study.

In addition, I am grateful to all fellow research staff, as well as research students and undergraduates, who have helped me with the experiments. I especially want to thank Mr. Burhan for building a series of MAVEN robots and Mr. Hendra Suratno Tju for debugging my code, as well as Dr. William Gu for sharing his programming experience with me.

I am also indebted to my family and friends for their love, support and understanding in my life. I wish to dedicate this work to the memory of my father. His unfailing faith in me has been my strength. I am most grateful to my sister, Ms. Pang Wee Ya and my mother. My deep appreciation also goes to my dearest friends, Ms. Francina Chan, Ms. Angela Wu and Ms. Liu Yimei for their constant encouragement and prayers.

I owe my deepest gratitude to my wonderful husband and colleague, Dr. Wong Choon Yue, who is always around to encourage me and keep me sane. Besides proofreading my thesis, he has provided me with a lot of assistance at home and at work. He is also the creator of the EDGAR robots! Finally, with the highest regards, I thank God, without whom nothing would have been possible.

CONTENTS

Acknowledgments	i
Abstract	vi
List of Figures	vi
List of Tables	viii
List of Algorithms	ix
List of Abbreviations	xi
1 Introduction	1
1.1 Towards a Social and Telepresence enabled Anthropomorphic Robot	5
1.1.1 Benefits of using an anthropomorphic telepresence humanoid	5
1.1.2 Apparatus for a robotic telepresence system	6
1.1.3 Interactions within a robotic telepresence system	8
1.2 Problem Definition	10
1.2.1 Inhabitor cognitive workload	10
1.2.2 Lack of autonomy for telepresence robots	11
1.3 Motivation	11
1.4 Objectives	12
1.5 Approach	12
1.6 Scope of work	13
1.7 Structure of the thesis	13
2 Literature Review	15
2.1 Telepresence and Teleoperated Systems	15
2.1.1 Teleoperated Telepresence Robotic Avatars	17
2.1.2 Advantages of Telepresence Robotic Avatars	18
2.2 Telepresence and Presence	19
2.2.1 Physical Presence	20
2.2.2 Social Presence	20
2.3 Adding Autonomy	21
2.3.1 Levels of Robot Autonomy	22
2.3.2 Levels of Autonomy in Navigation	22
2.3.3 Levels of Autonomy in Gesticulation/Manipulation	23
2.3.4 Levels of Autonomy in Conversation	25
2.4 Metrics to evaluate a telepresence robotic avatar system	26
2.4.1 Usability test of the inhabitor control station and GUI design	26
2.4.2 Task Performance Evaluation	27
2.4.3 Acceptance and Attitudes Measures	27
2.4.4 Quality of Human-to-human Interaction	28
2.5 Robotic System Architecture	29
2.5.1 Control Architectural Structure	29
2.5.2 Cognitive Architecture	31
2.5.3 Connecting Components	32

2.5.4	Examples of existing robotic system framework	33
2.6	Summary of the Literature Review	41
2.6.1	Chapter Conclusion	42
3	Autonomy Levels for Social and Telepresence Enabled Robot Framework	43
3.1	System Architecture of the ALSTER Framework	45
3.1.1	Hardware Components	46
3.1.2	Sensing Subsystem	47
3.1.3	Controllers	47
3.1.4	Action Manager	48
3.1.5	Perception Subsystem	49
3.1.6	Robot Autonomy	50
3.1.7	Memory Subsystem	51
3.1.8	Learning Subsystem	51
3.1.9	Session Supervision	52
3.1.10	Human-Robot Interfaces	52
3.2	Ontology for Social and Telepresence Robot	54
3.3	Software organization of the ALSTER framework	55
3.3.1	Possible usages of the framework	57
3.4	Features of the framework	57
3.5	Chapter Summary	58
4	Implementation of the ALSTER Framework	60
4.1	The Physical Layer — Robotic Prototypes	62
4.1.1	First Prototype — MAVEN	62
4.1.2	Second Prototype — EDGAR	64
4.2	The Controllers	66
4.2.1	Mobile Base Controller	66
4.2.2	Humanoid Motion Controller	67
4.2.3	Avatar Controller	70
4.3	Sensing Subsystem	70
4.4	The Perception Modules	71
4.4.1	Human Detection and Tracking	71
4.4.2	Sound Localization	74
4.4.3	QR Code Detection and Decoding	74
4.4.4	Localization	75
4.4.5	Map Server	77
4.5	The Navigation Service Elements	78
4.5.1	Teleoperation	78
4.5.2	Waypoint Navigation	79
4.5.3	Auto-Docking	80
4.5.4	Person Following	81
4.6	The Gesticulation Service Elements	88
4.6.1	Tele-Conferencing (Video)	88
4.6.2	Kinect Avateering	90
4.6.3	Scripted Movements	94
4.6.4	Reactive Gesticulations	94
4.7	The Conversation Service Elements	95
4.7.1	Tele-Conferencing (Audio)	95

4.7.2	Manual Text Input	96
4.7.3	Scripted Speech	96
4.7.4	Automated Dialog Engine	97
4.8	The Supervisory Layer	98
4.8.1	Memory Subsystem	98
4.8.2	Learning Subsystem	98
4.8.3	Session Supervision	99
4.9	The Social Layer	100
4.9.1	Network Link	100
4.9.2	Graphical User Interface	102
4.9.3	Text-to-Speech Synthesizer	106
4.9.4	Automated Speech Recognizer	107
4.10	Chapter Summary	107
5	Evaluation Studies, Results and Discussion	110
5.1	Study Objectives	111
5.1.1	Demonstrating the implemented framework	111
5.1.2	Evaluating the developed framework	111
5.2	Scenario 1 — A Robotic virtual tour guide and user experience study . . .	112
5.2.1	The Robot	112
5.2.2	Implementation of the Tour Guiding Application	112
5.2.3	Result	115
5.3	Scenario 2 — A Telepresence Robot and Workload Study	118
5.3.1	The Robot	118
5.3.2	The Telepresence Robot Application	118
5.3.3	The HRI Experimental Study	119
5.3.4	Result	123
5.4	Scenario 3 — A Telepresence Humanoid and Usability Study	125
5.4.1	The Robot	125
5.4.2	The Telepresence Humanoid Application	125
5.4.3	Report of Local User Experience	126
5.4.4	The Usability Study and Result	127
5.5	Scenario 4 — A Social Humanoid Assistant and Acceptance Study	130
5.5.1	The Robot	130
5.5.2	The Implementation of a Social Humanoid Assistant	131
5.5.3	Result	133
5.6	Discussion	136
5.6.1	Demonstration of ALSTER framework	136
5.6.2	Evaluation of ALSTER framework	137
6	Conclusion	139
6.1	Thesis Summary	139
6.2	Contributions	141
6.2.1	Proposal and design of a software framework for telepresence and autonomously-social robots	142
6.2.2	Implementation and demonstration of the proposed framework on non-anthropomorphic and humanoid robots	143
6.3	Future Work	145
6.3.1	Limitations regarding this study	145

6.3.2 Further improvements	146
6.4 Closing Remarks	147
Bibliography	150
Appendices	165
A Kinect for Windows Sensor	165
A.1 Comparison of two generations of Kinect	165
A.2 Skeletal Data of Kinect v1 and Kinect v2	166
B NASA-task load index (TLX)	168
C Mann-Whitney U Test Calculations	169
C.1 Measurement of navigation performance	169
C.1.1 Mann-Whitney U Test for Completion Time	170
C.1.2 Mann-Whitney U Test for Number of Collisions	171
C.2 Measurement of Workload (NASA-TLX Index)	172
D Kruskal-Wallis Test Calculations	174
D.1 Kruskal-Wallis Test (Completion Time)	174
D.2 Kruskal-Wallis Test (Number of Collision)	175
D.3 Kruskal-Wallis Test (Workload)	175
E Usability Questionnaire for Remote User	177
F Usability Questionnaire for Remote User (Completed by two users)	180
G Godspeed Questionnaire	183

ABSTRACT

Effectiveness in both personal and professional relationships lies in one's ability to communicate. In addition to words, nonverbal cues or "body language" is recognized to enhance interpersonal communication. Body language is the use of physical behavior, expressions, and mannerisms to communicate non-verbally. Telepresence robots, or robotic avatars, can be viewed as a tool that embodies gestures within a medium for distance interpersonal communication. Early designs of telepresence robots attempted to address this limitation by integrating a mobile robotic base to the screen that displays the robot's user. This permits the user to remotely drive the platform from point to point, extending its interaction space, and to interact with another moving human. Such robots, although not resembling the human form, can be recognized as an opportunity to exercise the user's presence, remotely.

A trend towards anthropomorphic telepresence robots is emerging. In addition to the basic functions like navigation and conversation that traditional telepresence robots are expected to perform, the humanoid form enables a user to gesture and manipulate objects remotely. With this increase in functionality, users may find the increase in cognitive workload overwhelming. The robot's user must drive the robot, manipulate the robot's arms for gesturing and attempt a meaningful conversation. This level of cognitive load increases the possibility of accidents. In addition, the secondary tasks relating to motion and gesturing hampers the primary task of effective interpersonal communication.

Attempts to mitigate this high human cognitive workload have taken the approach of providing autonomy for secondary functions. Autonomy is, however, not a panacea for all situations. Inappropriate autonomy can cause frustration, diminished situation awareness and unintended outcomes. Selecting the proper amount of autonomy to apply during an episode of robot deployment is necessary. The level of autonomy is required to be dynamically adjusted to support the interaction tasks. During situations, like navigating the robot along a cluttered hallway, autonomy in the form of automated obstacle avoidance is desirable, but during instances like moving the telepresence robot close to an object for a closer look may introduce an unwanted obstacle avoidance response.

Current telepresence robots lack a generic structure to assist dynamically in implementing autonomy. There is also an absence of consideration for what aspects of telepresence-robot-deployments to automate. To address this, the thesis identifies three aspects that characterize the autonomy needs in telepresence robots for navigation, gesticulation and conversation. With the goal of addressing these concerns, a framework called "Autonomy Levels for Social and Telepresence Robots" or ALSTER is proposed. It is a set of core software modules categorized in the identified aspects of autonomy required for telepresence robots. This array of software modules has also been cohesively arranged in a hybrid six-layered framework to incorporate an appropriate level of autonomous behaviors during the various stages of interaction. Past software frameworks for similar platforms were limited by a focus only on the navigation aspect, neglecting the gesticulation and conversation aspects or simply multiple iterations of the same robot.

This thesis describes the conceptualization of a framework that unifies the navigation, gesticulation and conversation aspects with adjustable autonomy. The framework has been implemented on platforms that support, platform motion, hand gestures, verbal communication, and video interaction. These implementations demonstrate the interaction of primary and secondary interactions between remote humans. Through the implementation of the framework on four evaluation studies, the software modules demonstrate the software modules as a tool for the realization of human-robot systems in communication tasks. In the limited trials, human interaction was demonstrated to support the objective of enhanced interpersonal interaction at lower cognitive loads and mitigate the issues of inappropriate autonomy. The addition of autonomous navigation behavior to a telepresence robotic avatar ensured enhanced performance in collision free navigating whilst engaging in a multiplicity of scenarios including one involving young children in a learning engagement. Without the varying levels of autonomy, the robot would have been less able to adjust to the needs for physical interaction, and the tasks of teaching or learning.

LIST OF FIGURES

Figure 1.1	Spectrum of telepresence applications for telecommunication	1
Figure 1.2	Elements of a robotic telepresence application	7
Figure 1.3	Illustration of data flow from the inhabitator to a mobile robotic avatar	8
Figure 1.4	Proposed Methodology	13
Figure 2.1	Some existing telerobotic machines.	16
Figure 2.2	Some existing commercial telepresence robots.	17
Figure 2.3	The <i>sense-plan-act</i> deliberative architecture. Adapted from [1]. . . .	30
Figure 2.4	The subsumption architecture. Adapted from [2].	30
Figure 2.5	An Example of a 3T architecture. Adapted from [1].	31
Figure 2.6	System framework of ROTI, adapted from [3]	33
Figure 2.7	System framework of ERICA, adapted from [4]	34
Figure 2.8	System framework of Robovie, adapted from [5]	35
Figure 2.9	System framework of Robonaut, adapted from [6]	36
Figure 2.10	System framework of Grace, adapted from [7]	37
Figure 2.11	System framework of Kismet, adapted from [8]	38
Figure 3.1	Overview of the ALSTER Framework	44
Figure 3.2	Schematic architecture of the multi-layered ALSTER Framework . .	45
Figure 3.3	Sensing subsystem of the ALSTER Framework	47
Figure 3.4	Action Manager subsystem of the ALSTER Framework	48
Figure 3.5	Perception subsystem of the ALSTER Framework	50
Figure 3.6	Software organization of the ALSTER framework.	56
Figure 4.1	Implementation of the ALSTER framework.	61
Figure 4.2	A suite of MAVEN non-anthropomorphic robotic avatars	63
Figure 4.3	Two versions of EDGAR humanoid robot	65
Figure 4.4	Construction of the Rear Projection Head Unit	66
Figure 4.5	Robot reference system and the basic motions for navigation	67
Figure 4.6	Definition of the Robot Coordinate System and Sign Convention . .	68
Figure 4.7	Illustration of the various human motions	69
Figure 4.8	Types of QR coded images that ALSTER framework currently detects	75
Figure 4.9	The Localization Module	76
Figure 4.10	Example of a 2D binary occupancy SLAM map	78
Figure 4.11	Gamepad joystick commands mapping for teleoperation	79
Figure 4.12	Different high-level navigational behaviors	81
Figure 4.13	Feedback control loop for robot speed calculation	82
Figure 4.14	Velocity Profiling for Person Following	83
Figure 4.15	Result of the Back-Following Mode.	85
Figure 4.16	Result of the Side-by-Side Accompaniment Mode.	85
Figure 4.17	Trajectory plots of various following mode	86
Figure 4.18	Result of back-following mode in physical setup	87
Figure 4.19	Result of side-by-side accompaniment mode in physical setup	87
Figure 4.20	Generation of the Reference Image	89
Figure 4.21	Calculation of the joint angles	91
Figure 4.22	Functionality of the Reactive Gesticulations Module.	95
Figure 4.23	A tour guiding robot state machine implemented using SMACH . .	100

Figure 4.24 Signaling and information exchange process of the Network Link .	101
Figure 4.25 Graphical User Interface	103
Figure 4.26 Text-to-Speech Module	106
Figure 5.1 General robotic software architecture used in all evaluation studies	110
Figure 5.2 ALSTER Framework applied to a robotic tour guide application . .	113
Figure 5.3 State diagram of the robotic tour guide	114
Figure 5.4 Sequence of interactions during a guided tour session	115
Figure 5.5 Box-and-whiskers plot for the questionnaire responses of the visitors.	116
Figure 5.6 ALSTER Framework applied to the telepresence robot application .	119
Figure 5.7 Schematic Layout of the test environment	121
Figure 5.8 ALSTER Framework applied to the telepresence humanoid application	126
Figure 5.9 Direct teleoperation of EDGAR to use it as a telepresence humanoid	127
Figure 5.10 Snapshots taken from the Machine Impossible show	128
Figure 5.11 Various Edgar Deployments	130
Figure 5.12 ALSTER Framework applied to the social humanoid assistant application	132
Figure 5.13 Children's Season 2016 at the Chinese Heritage Centre	133
Figure 6.1 A pictorial representation of the contributions.	144
Figure A.1 Kinect Audio Specifications	165
Figure A.2 Two Generations of Kinect sensor	166
Figure A.3 Body joints that are tracked by Kinect v1 and Kinect v2	167

LIST OF TABLES

Table 2.1	technical specifications of some existing commercial telepresence robots	18
Table 2.2	Summary of various robotic frameworks	41
Table 3.1	Different types of perceptual knowledge	49
Table 3.2	Ontology of the ALSTER framework	55
Table 4.1	List of human head and upper limb motions	70
Table 4.2	List of attributes that the robot keeps track of a person in people-list	72
Table 4.3	Parameters for Velocity Profiling for Person Following	84
Table 5.1	Overview of the evaluation studies	110
Table 5.2	Questions used to evaluate the robotic tour guide application	116
Table 5.3	Questionnaire result of the visitors	117
Table 5.4	Experimental Results	123
Table 5.5	Result of the Usability Questionnaire for Remote User	129
Table 5.6	Result of the Godspeed questionnaire on EDGAR version 2	135
Table 6.1	Comparison of ALSTER framework with the reviewed existing frameworks	145
Table A.1	Characteristics of Kinect v1 and Kinect v2 Sensors	166
Table A.2	Skeletal Data of Kinect v1 and Kinect v2 Sensors	167
Table B.1	Definitions of the NASA-TLX dimensions [9]	168

LIST OF ALGORITHMS

Algorithm 4.1	Algorithm to Update the People-List	73
Algorithm 4.2	Algorithm to perform Auto-Docking	81
Algorithm 4.3	Geometric Image Transformation Algorithm	90

LIST OF ABBREVIATIONS

AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
AMCL	Adaptive Monte Carlo Localization
APET	Amorphous Polyethylene Terephthalate
AV	Audio-Video
CAD	Computer-Aided Design
CSS	Cascading Style Sheets
DC	Direct Current
DOF	Degree of Freedom
EDGAR	Expressions Display and Gesturing Avatar Robot
EKF	Extended Kalman Filter
EM	Expectation Maximization
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HHI	Human-Human Interaction
HMD	Head-Mounted Display
HRI	Human-Robot Interaction
HTML	Hyper Text Markup Language
ICE	Interactive Connectivity Establishment
ICS	Inhabitor Control Station
IMU	Inertial Measurement Unit
IP	Internet Protocol
IPC	Inter-Process Communication
JSON	JavaScript Object Notation
KLD	Kullback-Leibler Divergence
MAVEN	Mobile Avatar for Virtual Engagement by NTU
NAT	Network Address Translation
NTU	Nanyang Technological University
OWL	Web Ontology Language
PID	Proportional-Integral-Derivative
POSIX	Portable Operating System Interface
QR	Quick Response
ROS	Robot Operating System
RTC	Real-Time Communication
SDP	Session Description Protocol

SLAM	Simultaneously Localization And Mapping
SPA	Sense-Plan-Act
STUN	Session Traversal Utilities for NAT
TEB	Time Elastic Bands
TURN	Traversal Using Relay around NAT
URDF	Unified Robot Description Format
URL	Uniform Resource Locator
WebRTC	Web Real-Time Communication
2D	2-Dimensional
3D	3-Dimensional

CHAPTER 1

INTRODUCTION

The commonly-held view regarding robots has been one of intelligent machinery to reduce human labour. Likewise, the traditional understanding of telecommunication is to use electronic means for communicating text and imagery. However, at least as far back as 1980, a combination of using a human-like robot with arms to communicate human actions over a distance has been described by Marvin Minsky [10]. In that article, Minsky described a user wearing a jacket arrayed with sensors to capture information about the user's movements. Robotic hands located remotely from the user reproduces each motion of the user's arms, hands and fingers. He adds: "Light, dexterous, and strong, these hands have their own sensors through which you see and feel what is happening." Thus, the jacket also permits the user to also have a tactile feel for the object which the robot hands are handling. He envisioned this telecommunicated motion to be possible between rooms, cities, countries and even planets.

Minsky was quite the visionary to have detailed such technology. It is present in some form today in 2021, but has not been fully realised. This thesis describes work done in the research and development of solutions to bring technology closer to fulfilling Minsky's vision. In addition to communicating physical actions and sensations between two locations, the work described in this thesis also addresses the communication of presence between the said locations. This is achieved by including the traditional communication of speech and video imagery.

Figure 1.1: Spectrum of telepresence applications for telecommunication, including (a) MAVEN-gray, (b) MAVEN-1, (c) EDGAR and (d) Nadine telepresence humanoid.

To facilitate discussion, a spectrum of telepresence applications is presented here in Figure 1.1, showcasing various methods of telecommunication from one of the simplest ¹, to telepresence robotic avatars as some of the most sophisticated forms. At this time, such avatars represent technology that is just shy of the richest and most nuanced communication, which is to converse face-to-face, represented at the right-most extreme of the spectrum.

The figure shows the evolving array of telepresence technologies designed to support collaborative interaction between people, especially in terms of natural conversation that involves the exchange of audible verbal information as well as visual nonverbal communication cues. The spectrum shows how current systems along this array have become progressively more effective in using telepresence for communication.

The arrangement of the telepresence technologies along the spectrum is inspired by the Media Richness Theory [11] and the Social Presence Theory [12]. The Media Richness Theory is a framework to describe the ability of these technologies to reproduce information which enables users to communicate and facilitate understanding. The Social Presence Theory classifies the technologies based on the degree of social presence. It rates the technology's ability to make users aware of each other's presence during telecommunication. According to both theories, face-to-face interaction is the richest of the telecommunication media.

Audio-mediated telepresence

On the other end of the spectrum, telephony is the least rich telecommunication media. It involves only the transmission of speech between two parties and richer information such as visual content is absent. However, telephony is a well-established telepresence service [13] that enables geographically separated callers to perceive each other's presence through the extension of human speech and hearing. For example, when a user calls a telephone helpline, service personnel can provide a remedy to a technical issue, even if the technician is miles away.

However, multi-party collaboration using audio conferencing has many shortcomings. Participants have sometimes displayed bad meeting etiquette, such as muting or ignoring the caller. They may also leave or join the conference without the other members being aware. Other shortcomings of audio conferencing include not knowing who is talking or participating in the call.

¹There are still simpler forms of telecommunication than the telephone. These include texting and even the telegraph. These are not within the spectrum's scope in order to maintain focus on discussing telecommunication with high experiential richness; i.e. telecommunication involving at least speech, video, or movement.

Video-mediated telepresence

The pandemic of 2020 has made such video-mediated telepresence technology widely popular. We note this in the pervasive use of software such as Zoom, MS Teams and Skype. Video conferencing improves the telepresence experience by enabling visual communication in addition to the exchange of audio. As people can see each other during a video conference call, it helps to ensure that the meeting participants are actively engaged with the discussion and interaction. Presently, telepresence through video-mediated communication ranges from mobile video chatting tools to multi-party video conferences as well as dedicated telepresence boardroom [14]. Other advanced video-mediated telepresence technologies that are envisioned for interaction include 3D reconstructed scenes streamed and displayed on life-size stereo displays [15].

The display of video-mediated telepresence is traditionally done on screens like those on computers, cell phones and tablets. Such screens also facilitate telepresence by displaying graphics like virtual avatars [16] or mixed-reality interaction [17]. Strictly speaking, these examples do not display video of the user, but instead provide a graphical representation of the user(s) in virtual reality. However, they are grouped with other examples of video-mediated telepresence here because like video, they are displayed on flat screens. When compared with face-to-face interaction, they suffer similar disadvantages in remote communication. These drawbacks [15, 18] include the difficulty in managing turn-taking, pointing at things and manipulating real-world objects remotely.

Robot-mediated telepresence

Now, there has been a new telecommunication method where a mobile robot is used to augment interaction, by integrating video-mediated communication tools with the movable platform. Such robot-mediated telecommunication, as shown in Figure 1.1, is commonly known as a telepresence robot. Some may refer to it as a remote presence system [19, 20], virtual presence system [21], embodied social proxy [22] or robotic avatar [23, 24, 25].

The use of a robotic avatar as an emerging telecommunication application is evidenced by the increasing number of telepresence robots available in the market as well as the expanding amount of related research efforts. These robots have been developed to perform in a plethora of applications such as performing medical tele-rounds in health care institutions [26, 27] and conducting ad-hoc conversations in office environments [28]. Commercial telepresence robots are also being used in crowded public places, such as museums [29] and academic conferences [30], to allow remote visitors to tour and interact with the local people.

The use of robots would further improve telecommunication as many researchers [31, 32] believed that a physical embodiment would project the social presence of the remote user.

It has been shown in [33] that having a physical embodiment for telecommunication would result in more positive collaboration outcomes than video-conferencing.

Humanoid-mediated telepresence

Most of the existing telepresence robotic avatars, if not all, have a non-anthropomorphic design. This common design includes a flat screen on a computerized movable platform — that is without arms or hands and thus is not human-like. However, when people communicate face-to-face, a wide variety of sensory and motor modalities are used to regulate the conversation or complement the verbal utterances. In particular, humans subconsciously exhibit proxemics [34] (behaviors related to spatial distance) and kinesics [35] (behaviors related to body movement and gestures) as nonverbal communication-cues during face-to-face interactions.

This motivates the design and development of anthropomorphic robots or humanoids for telepresence. One example is the ultra-realistic Geminoid androids [36], which are designed to represent the user with the highest fidelity. It can be teleoperated by its user for social interaction. This user, referred to as an *inhabitor*, has the objective of using the robot to mimic his/her speech and actions. For example, in Figure 1.1(c), the Nadine android avatar [37] is made to look exactly like its inhabitor and can be controlled by her for telecommunication. In that manner, the inhabitor is able to perform in-space movements to express gestures and to point at things during remote interaction. However, although the Geminoid androids have limbs and a human-like appearance, they are unable to navigate about a physical space. A major disadvantage of such a telepresence android is the fact that without some effort in altering its appearance, it can only accurately represent one individual person.

The Mebot [38] and Hasegawa's robot [39] are another type of telepresence robot with some inbuilt anthropomorphism. They have a small flat screen head unit attached to a pan-tilt mechanism and motorized arms for gesturing. However, these two robotic avatars are considered partially-anthropomorphic telepresence robots because they are toy-sized and can only move on a table-top. Hence they are unable to represent the inhabitor faithfully.

Besides the Geminoid androids, a number of non-anthropomorphic telepresence robots are also equipped with anthropomorphic functions to improve social interaction. For instance, both the PRoP robot [40] and the QB robot [21] have a laser point to perform deictic gestures. The RP-Vita [41] has a pan-tilt head screen unit that simulates gazing.

1.1 Towards a Social and Telepresence enabled Anthropomorphic Robot for Telecommunication

A spectrum of telepresence technologies for remote mediated communication has been discussed. The discussion provides an understanding of the various applications that have been implemented to support linkages between geographically separated people. These telepresence technologies, which include phone calls, video conferencing and telepresence robotic avatars are being adopted by corporations to support collaboration between remote users.

There is a desire to develop better telecommunication technologies and one way is to produce more sophisticated robotic avatars. The development of such avatars has been highlighted as an enhancement to static video conferencing. Moreover, a telepresence humanoid avatar is felt to be capable of facilitating a communication experience that is more natural when compared to non-humanoid or non-anthropomorphic avatars. This is because the aspect of kinesics can be transmitted in addition to other aspects such as mobility and visual information.

1.1.1 Benefits of using an anthropomorphic telepresence humanoid

If a robotic telepresence system “*was perceived as being human, then effectively it should be human and not a robot*” [42]. It is believed that employing anthropomorphism to a robotic system would facilitate human-robot interaction. This is because when “humanness” is integrated into a telecommunication system, it affects our dealings with the technology, making us regard it more than we would a typical machine. This is desirable in a telepresence system because when the remote inhabitator uses the robot for communication, he would expect the people with whom he is communicating via the robot, to see him as a person rather than a robot.

This has been evidenced in a study [43] when the addition of physical proxy motions favorably influenced the perceived dominance and involvement of the interaction. It is mentioned in the study that when physical gesticulations were shown, the remote user was perceived as having a greater involvement during interaction. The local on-site teammates were able to view the user as being more equal in stature.

Similarly, a remote inhabitator described that “*his presence was more felt*” [32] when using the telepresence robot in a meeting. His local teammates were also able to remember that the remote inhabitator was part of the meeting. They would make eye contact with him and gesture with their hands towards the telepresence robot. It is highlighted in the same article that informal interaction, such as hallway discussions before and after the meetings, are

important during a collaboration. This is achievable only via a mobile telepresence robot because unlike video-conferencing, it is not tied to a particular room or space.

Besides the benefits of having a physical embodiment and mobility, a telepresence humanoid is also equipped with anthropomorphic functions such as gesturing, gazing, pointing and touching, which improve social interaction. For example, in a study [44] involving two identical telepresence humanoids deployed for robot-mediated communication, it was observed that people who inhabited the humanoid would use the robot to gesture and convey their thoughts more effectively. In another study [45] exploring children-robot interaction, a correlation between robot-touch behaviors and social connectedness. This type of touch behavior between robot and children is akin to peer-to-peer contact between young children when they interact with each other.

Pointing, in particular, is a deliberate and intensively used gesture for disambiguating speech. For instance, when a speaker says, “The restroom is in that direction.”; it is usually unclear to the listener where the exact location of the restroom is unless the speaker also points to a particular direction as he speaks. As such, being able to replicate this gesture would increase a telepresence robot’s utility. It has been noted that pointing gestures by such robots can help listeners remember more object-spatial information [46].

1.1.2 Apparatus for a robotic telepresence system

In this subsection, the apparatus for a typical telepresence system is described. Figure 1.2 provides a pictorial overview of a robotic telepresence system and shows the relationship of each human or robot entity within this system. This thesis adopts specific terms to refer to the robotic system and its users. These terms are listed and described in the following paragraphs.

An **inhabitor**, also known as a *remote user*, an *operator* or a *teleoperator*, is the person who controls the robotic avatar over the internet. This person uses the avatar to represent him/herself at the location where the robot is situated. In the systems implemented within this thesis, this person inhabits the robotic avatar via the computer interface. An **inhabitor control station** (ICS) is the place hosting the hardware used by the inhabitor. Two roles are performed by the control station: 1) to display data acquired from the robotic avatars to the inhabitor and 2) to get control instructions as well as related data from the inhabitor. The data relayed from the avatar to the control station as well as that flowing vice-versa can include sounds and video. Much like the video-conferencing applications used today, this data as well as control instructions between inhabitor and robot avatar flows via the Internet. The computer interface need not be the same one and can be varied in sophistication. For instance, the wide-screen display used by Inhabitor B in Figure 1.2 would likely provide a richer experience that is more immersive than that used by Inhabitor A. Future systems may

Figure 1.2: Elements of a robotic telepresence application

enable the inhabitant to experience full embodiment through a full-body and multi-sensory interface.

A **robotic avatar** or an *avatar robot* is a robot that can be used to represent a human. Just like a digital avatar can be used to represent a person in online games or chat rooms, a robotic avatar represents a person in a physical location. To proceed, it is necessary to define two types of robotic avatar: 1) the non-anthropomorphic robot avatar and 2) the humanoid robot avatar.

The **non-anthropomorphic robot avatar** is a telepresence robot characterized by a video conferencing system mounted on a mobile robotic platform. As such, it is a mechanical surrogate used to represent the inhabitant in a remote location. The factor that differentiates it from a conventional tele-conferencing set up is its ability to be mobile, enabling the inhabitant to project his/her telepresence in motion. Since this physical form is not human-like, it is considered non-anthropomorphic. Such robot avatars are easily recognized as they typically have a flat screen mounted on a mobile platform. Figure 1.2 shows the non-anthropomorphic robot avatar in this form, and Inhabitor A is inhabiting it.

The **humanoid avatar** or *telepresence humanoid* is an anthropomorphic version of a robotic avatar. It is of greater mechanical complexity than the non-anthropomorphic robot avatar because the hardware required to express the human form in the real world, as well as to replicate human-like limbs and movement are sophisticated. In Figure 1.2, Inhabitor B is inhabiting such an anthropomorphic humanoid avatar.

The **interactant**, also known as the *confederate* or *local user*, is the person (or people) with whom the robotic avatar is interacting at the robot(s)' location.

It is worth mentioning that the robotic telepresence system may be simpler than what is depicted in Figure 1.2. The simplest of such a system would be one inhabitant using one avatar robot to interact with one interactant. It is assumed in this thesis that the ratio between inhabitant and robot avatar is 1:1. The robotic telepresence can be scaled to include more inhabitants (and thus an equal number of avatars) and more interactants.

1.1.3 Interactions within a robotic telepresence system

As mentioned in the previous section, using a robotic avatar for telecommunication requires a two-way transmission of data such as audio and video in real time. When using a humanoid avatar (as opposed to a non-anthropomorphic one), other contents needed for non-verbal communication may be relayed so that the benefits of using a humanoid can be realized. Figure 1.3 depicts the flow of data from the inhabitant to the mobile robotic avatar at the local site.

Figure 1.3: Illustration of data flow from the inhabitant to a mobile robotic avatar

At the ICS, control commands can be acquired from the inhabitant via traditional input devices such as a mouse, keyboard and joystick. Such data is used to direct the locomotion of the avatar robot. A web-camera system is used to capture the audio data as well as video imagery, which is relayed to the avatar robot and subsequently displayed. The user interface facilitates data input as well as presentation. Other variants of input and displaying data are also possible. For example, data can be displayed on a touch screen tablet or a

head-mounted display.

The description of the ICS in the preceding paragraph is true for instances of an inhabitant deploying either type of robotic avatar. However, in the case of a humanoid robotic avatar, more data may be needed for controlling the humanoid's limbs and for presentation on such a robot's graphical display. A research effort [47] related to that in the thesis describes such a deployment. In that work, a human motion tracking camera system (such as a Kinect sensor from Microsoft) was used to capture data of the inhabitant's head and arms at the ICS. The sensor also segmented the image of inhabitant's face for display on the humanoid's face, which was contoured according to that of a human. The humanoid robot was then able to mimic not only the inhabitant's movements but also the human's facial features and expressions.

From the information presented in subsection 1.1.2 as well as that discussed here, it is possible to deduce the following types of interactions involved in the system's use if we consider the inhabitant's perspective:

1. Human-Computer Interaction (HCI) — denoting the direct interaction between the ICS and the inhabitant.
2. Human-Robot Interaction (HRI) — the interaction between the inhabitant and the remote robotic avatar, via the ICS.
3. Human-Human Interaction (HHI) — the target interaction between the inhabitant and interactant via the robotic avatar.

On the other hand, when the interactant uses the telepresence humanoid avatar for telecommunication, two types of real-time interactions are involved:

1. Human-Robot Interaction (HRI) — the interaction between the interactant and the robotic avatar at the local site.
2. Human-Human Interaction (HHI) — the target interaction between the interactant and inhabitant via the robotic avatar.

It is worth pointing out here that even though we may consider the inhabitant to be interacting only directly with the ICS and thus experiencing HRI, such an assumption may be overly simplistic. This contends that HRI and HHI are present because the ICS is not a perfect interface which allows the inhabitant to be fully aware of the situation at the robot's locality. This imperfection arises due to reasons such as limitations in robot sensing and actuation technology, disruptions and latency due to telecommunication via the internet.

1.2 Problem Definition

1.2.1 Inhabitor cognitive workload

From the previous section, it is possible to discern that achieving telepresence via a robot avatar is a complex affair. For the inhabitor simultaneously performing HCI and HRI while also trying to achieve the goal of projecting presence remotely via the robotic avatar (HHI), high cognitive workload is likely to be experienced.

It is foreseeable that a number of issues will result from such a high level of workload.

Increased navigational errors and reduced safety

We may imagine a specific scenario of the inhabitor conversing with the interactant while attempting to pilot the robot safely and meaningfully alongside the interactant. Under normal circumstances, when two people are walking side by side and conversing, our peripheral vision and sense of feel provides awareness of our surroundings. This allows us to focus on chatting without fear of straying off a path or walking into a wall for instance. In a telepresence system, the sensors' field-of-view is limited, making the achievement of such a level of awareness improbable even if the inhabitor is fully focused. A study by Tsui et al. [32] showed that two-thirds of participants acting as the inhabitor caused the robot to collide with the environment that has been arranged like an office space. It was posited that the lack of situation awareness that stemming from the use of such remote operation caused the collisions.

Degraded conversation with interactant(s)

Distracted by safety concerns, the inhabitor may not be as focused on HHI with the interactant as would be the case when both parties are co-located or when using conventional video-conferencing applications. Some words may be missed, resulting in misunderstanding or having to repeat the missed words. The speed of conversation may also slow down.

Abandonment of subtask

The inhabitor may be required to do more than one task during a telepresence session, e.g. the walk-and-talk scenario. In cases of high workload, the inhabitor may opt simply to abandon one of the tasks, either navigation or conversation. This can cause undesirable disruption in human-to-human interaction.

Avoidance

The frustration and stress associated with using such technology can make users wary of using the robot telepresence system again, even as repeated use can result in improved familiarity and ease of use.

1.2.2 Lack of autonomy for telepresence robots

The use of autonomous behaviors or autonomy has often been advocated in instances where highly complex tasks (such as the one described in this thesis), are performed. Simply put, autonomous behaviors (or robot autonomy) is automation which allows the system to perform some sub-tasks like navigation, so as to alleviate workload and make the system easier to use. A good example of such automation is the autopilot used on airliners.

While autonomous navigation on mobile robots, for applications such as in logistic warehouses and factories, is relatively commonplace, telepresence robots are typically teleoperated without autonomous behaviors [48]. Yet telepresence robots should be equipped with the ability to deal with possible hazards autonomously even as the inhabitant is either unaware of the hazard, unable to avoid the hazard or when data transmission is interrupted. In the case of telepresence robots, automation is beneficial even when it is imperfect, and particularly in conditions of high inhabitant workload [49].

1.3 Motivation

It is therefore the goal in this research effort to develop autonomy for telepresence robots, so that the high workload that is experienced by the inhabitant when using the robotic avatar can be reduced to a manageable level.

The lack of autonomous features on telepresence robots leads to another issue. Robotic avatars, whether non-anthropomorphic or humanoid, are non-trivial in terms of financial cost. Without a person to teleoperate the robot, it would be redundant and wasteful since it serves no other purpose. Therefore, this work advocates developments encouraging the implementation of autonomy on telepresence robots such that they may be multi-purpose. Such robots when not serving as robotic avatars can act as concierge or indoor tour-guide robots for instance.

1.4 Objectives

This work aims to address the previously mentioned problem of workload-associated issues by equipping telepresence robots with autonomy. Yet, the work that this thesis shall describe is more than the development of a number of autonomous behaviors. Instead, the envisioned autonomy is to be implemented via cohesive structure. As such, a software framework to facilitate the organized application of different autonomous behaviors is proposed while also enabling a robot to perform its core function as a telepresence platform.

In addition, the software modules that facilitate autonomous behaviors are to span not only different behaviors but also degrees of autonomy. This is important because the level of autonomy can affect user engagement to the task; that is a high level of autonomy can decrease user engagement and vice versa. Hence, the proposed framework enables the adjustment or selection of autonomy levels.

Furthermore, this work aims to implement the proposed framework on two different types of robotic avatar. One implementation is to be performed on a non-anthropomorphic robot while another is to be made on a humanoid robot. This is due to the fact that while there have been research efforts proposing frameworks for autonomous behaviors, such work was done on a single robot or multiple iterations of the same robot [50]. This makes it difficult to differentiate such work between an implementation specific to one system or a framework that should be applicable in general to a variety of systems. The implementation of the same framework to different systems can aid in the comparison of systems with differing combinations of features, supplying results that can inform about the effectiveness of one system over another.

1.5 Approach

The overall approach in developing the framework is illustrated in Figure 1.4. Since the goal of this work is chiefly to mitigate workload levels via autonomy, various modules enabling such autonomous behaviors are to be developed.

In this process, it is crucial to consider the tasks that generate workload and to automate them; that is what to automate. Also important is the degree to which automation is invoked; meaning how high a level of autonomy to use and as a matter of consequence, how many levels of autonomy to make available. Furthermore, the inhabitant should get to select when autonomy is engaged or disengaged.

The framework is designed for adjustable autonomy to suit different scenarios. Figure 1.4 also reflects the application of adjustable robot autonomy for three key aspects of using telepresence robot systems: navigation, gesticulation and conversation. The framework is

Figure 1.4: Proposed Methodology: A framework to include level of autonomy for control in navigation, gesticulation and conversation aspects of a telepresence robotic avatar

thus designed around a need to provide an array of autonomous behaviors on each of these aspects.

1.6 Scope of work

This work is focused on developing a software framework for social and telepresence robotic applications. The framework will include adjustable autonomy in navigation, gesticulation and conversation aspects of a robotic avatar. The framework is implemented on two types of robots — anthropomorphic humanoid and non-anthropomorphic robot.

While the framework is primarily to address the dearth of autonomous behaviors for telepresence robots, it is not an in-depth examination of the relationship between autonomy and cognitive workload. Rather, the evaluation studies conducted in this effort are to serve to validate the framework's feasibility.

1.7 Structure of the thesis

Chapter 1 has provided some background information regarding the use of telepresence robotic avatars for telecommunication. This has led to the discussion on the challenges that are relevant to humanoid-mediated telepresence, which subsequently forms the objective of this research.

Chapter 2 is a review of the literature related to the concept of telepresence and teleoperated systems. Various examples and types of frameworks would be surveyed to provide a technical direction of the research work.

Chapter 3 presents the proposed solution to implement a framework for human-robot interaction with a social and telepresence-enabled robot. Individual components of the proposed framework would be described. Chapter 4 describes the work that has been achieved in designing and implementing the proposed telepresence robotic framework.

Evaluation studies will be conducted to evaluate the implemented framework as well as some of the implemented behaviors. The descriptions, as well as the result of these evaluation studies will be discussed in Chapter 5. The thesis is concluded in Chapter 6, which also presents the contributions of the thesis as well as the future work and a list of original publications.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the concept of telepresence is introduced, with a pertinence to teleoperation and telecommunication systems. After that, the enabler role of telepresence robot (a robot-mediated telecommunication system) is discussed. The concepts will form the foundation and motivations for the future research direction in the field of telepresence robotics. Discussion on the examples and types of telepresence systems and the necessary state-of-the-art telepresence robotic technologies are included to provide the background knowledge of the field. Surveying these technologies provided the technical direction and focus of the research work.

2.1 Telepresence and Teleoperated Systems

In the past, this term *telepresence* was used in reference to *teleoperation*. When Marvin Minsky [10] first coined the notion of telepresence in 1980, he was envisioning a set of teleoperation systems to facilitate the sensation of “being there”, where one can see and feel what is happening in another room or country. Sheridan [51] continued the trend and reserved “telepresence” only for cases involving teleoperation.

Sheridan further defined *teleoperation* as human controlling a set of sensors and actuators remotely [52]. Additionally, he explained *telerobotics* as a subclass of teleoperation, where the human is able to control a semi-autonomous machine (which has a set of sensors and actuators, as well as a computer) remotely in a supervisory manner. This implied that the telerobotic machine, or a *robot*, is one that is able to sense and act upon its environment autonomously for periods of time but is supervised by a human. The telerobotic machine is also configurable and controllable by the human operator remotely.

Telepresence is achieved through teleoperation because the human operator is able to perceive the distant environment visually and auditorily, via a camera system. The remote operator can also control the telerobotic machine remotely, from an interface, to perform some tasks. These tasks may include changing the position of the telerobotic machine or manipulating the objects remotely, in the environment where the telerobotic machine is situated.

For the past few decades, many such telerobotic machines have been implemented and used in various industries and fields, as shown in Figure 2.1. The main motivation of creating these telerobotic (or “telepresence”) machines is to perform tasks that are too dangerous or too repetitive for humans to carry out effectively.

Figure 2.1: Some existing telerobotic machines.

Some examples of the telerobotic machines include the Curiosity Mars Rover [53] (Figure 2.1(a)), which was implemented for Mars exploration and Nereus [54] (Figure 2.1(b)), which was an underwater robot used for hydro-thermal vents investigation. The Packbot robot [55] (Figure 2.1(c)) has been deployed by the US military to search Al Qaeda caves in Afghanistan as well as to hunt for chemical and nuclear weapons in Iraq. Telerobotics is also applied to perform repetitive works, like many of today's automated assembly systems are pipelined configurations of KUKA robotic manipulators [56], as shown in Figure 2.1(d). Telerobotic machines can be used to perform precise and dexterous operations, like in the case of the da Vinci surgical robot (Figure 2.1(e)) and the Robonaut humanoid (Figure 2.1(f)). The Robonaut [6] was built with dexterity similar to that of a human, in order to take over simple and repetitive tasks on the International Space Station.

While there have been many examples of teleoperated robots, few are established products or widely adopted. This is due likely to the technological limitations in solving issues such as communication latency, perception, actuation, and tactile feedback. These are elements of face-to-face communication or interaction that humans have grown accustomed to and deficiencies in any of these aspects will decrease user experience when it comes to teleoperated avatar robots.

Telepresence is studied, measured and evaluated with the motivation to develop better telerobotic machines or frameworks [51, 57], which ultimately improve the performance of the tasks that they are created for. By observation, it can be noted that these examples of telerobotic robots have minimum contact and interaction with the human. However, this will no longer be the case. The current research trend in consumer robotics shows that more robots will be created to work socially and interact closely with the human.

Figure 2.2: Some existing commercial telepresence robots.

2.1.1 Teleoperated Telepresence Robotic Avatars

A telepresence robotic avatar is a kind of telerobotic machine that allows the human user to perform the task of interpersonal communication (which includes verbal and nonverbal communication) remotely. It has been created to interact with humans socially and closely in the human environment.

Some of the commercially available telepresence robots, as shown in Figure 2.2, include QB from Anybots [21], BeamPro from Sutable Technologies [58], Double2 from Double Robotics [59], VGo from VGo Communication [60] and RP-VITA from InTouch Technologies [20] and iRobot Corporation [61] and Ava 500 from iRobot Corporation [61].

Table 2.1 lists the technical specifications of these robots, which include the sensors and status indicators that they carry, as well as any autonomous behaviors or gestural/manipulation functionality. Kristoffersson et al. [64] has presented a review of these telepresence robots as well as other research prototypes for robot-mediated telecommunication.

There are two basic ingredients for a telepresence robot [64][3][62], which include a remote-controllable mobile robotic system and two-way audio-video communication. The mobile robotic system is typically non-anthropomorphic in appearance and only a limited subset of human skills is possible [64]. For example, QB enables deictic reference via an articulated laser pointer [62]. Most of the current telepresence robots would also have a suite of sensors (see Table 2.1) installed on the mobile base to ensure safety, as well as to support the implementation of autonomous behaviors.

The two-way audio and video communication has been established in these telepresence robots to enable *social-presence* [65][66] for human-to-human interaction. This is unlike the

Robot	List of Sensors	Autonomy Technology	Gesticulation	Feedback Indicator for Interactants & Bystanders
QB [21][62]	Laser range finder (LIDAR)	Obstacle avoidance; Assisted doorway navigation; Dynamic balance	Pan-Tilt Laser Pointer	Blue LEDs around the “eyes”
BeamPro [58]	Downward-facing camera; Optional LIDAR	Collision prevention; Waypoint navigation; Assisted docking	None	LEDs under the screen
Double2 [59]	3-axis accelerometer and gyroscope	Dynamic balance	None	Purple/Green LED in base
VGo [60][62]	Front bumpers; Infrared distance sensors	Obstacle detection; Cliff detection; Auto-docking	None	RGB LEDs in “ears”
RP-VITA [20][41]	Electronic stethoscope; depth sensors; LIDAR; Sonar sensors	Waypoint navigation; Obstacle avoidance; Auto-docking; Person-following	Head display auto turns to face talking person	RGB LEDs in base
Ava500 [63]	Sonar sensors; Depth sensors;	Collision prevention, Obstacle avoidance; Auto-docking; Waypoint navigation	Adjustable height for head screen	LEDs around base

Table 2.1: Some existing commercial telepresence robots and their technical specifications

usage of video streaming in the usual teleoperated robots, which is typically one-way so as to provide the *spatial-presence* [65] for human-robot interaction (telerobotic control) [54].

2.1.2 Advantages of Telepresence Robotic Avatars

Several benefits of using a telepresence robotic avatar have been discussed in Section 1 and Section 1.1.1 in brief.

The first advantage is telepresence robotic avatars can improve the quality of remote team collaboration in workplace [32][67]. The survey in [67] reveals that using a telepresence robot for work can improve activities such as having impromptu meetings and attending planned social events. The remote inhabitant is able to make work-related and socializing conversations during these impromptu meetings or social events, via a telepresence robotic avatar. This would help the remote inhabitant to express commitment,

engage and give attention to the local co-workers, which consequently build affinity among geographically-separated team workers.

Telepresence robotic avatars, such as the Remote-Presence (RP) robots from InTouch Technologies [20], have been used to support health care in the hospitals. The usage of these robotic avatars, by remote doctors, to perform telerounds in the hospital has helped to reduce the length of stay [68] and has enabled faster face-to-face attending physician response time to intensive care unit patients [69]. This leads to earlier discharge, reduction in hospital cost and an increased in bed capacity at the hospital [68] [69].

Telepresence robots can also be used at home to provide aftercare or simply keep an eye on elderly or kids. In [70], a VGo telepresence avatar has been employed to allow physicians to monitor and give video consultations to the patient, who has been home after early discharge. It is feasible to equip these telepresence robots with medical equipment, such that health-related data can be collected [71] from the patients or elderly and be used to assist health care at home [72]. Besides medical devices, smart alerting systems can also be implemented on the robots to detect and alert the adults when someone has fallen down or has an accident at home [71]. Few telepresence robotic avatars have been implemented to enable people, who are prone to isolation and loneliness, to have social interaction

Lastly, telepresence robots can be used for remote teaching and consultation in the hospitals. For instance, RP-7 has been used for nursing education in [73], and the results indicate that both the faculty and students feel positively about using a robotic avatar as a teaching strategy. The same telepresence robot has been used for tele-monitoring and consultation during five minimally invasive surgery operations [74]. The remote consulting surgeon has expressed that he felt immersed in the operating room because he has been able to move around the room easily, without disrupting the surgery, and communicate with a junior attendee effectively.

It is believed that such telepresence robotic avatars can be used to provide medical mentoring and consultation worldwide in the future, enabling an expert physician to provide immediate consultation to time critical situations during a surgery. This will also enable the novice surgeons or nurses in the developing countries to receive medical advice from the more experienced physicians readily.

2.2 Telepresence and Presence

The concepts of presence and telepresence are complex and multidimensional [65][66][75][76][77] and have been used in various areas of research such as computer science, virtual reality, psychology, communication, education and sociology [75].

There have been many attempts to identify and classify the different types of presence.

Lombard and Ditton [76] have listed six different conceptualizations of presence: realism, immersion, transportation, social richness, social actor within a medium, and medium as a social medium. Commonly, in the discussion of presence in the telecommunication, virtual reality and telepresence robotic avatars domains, these conceptualizations have been simplified into two broad categories [64][65][66].

2.2.1 Physical Presence

Marvin Minsky [10] first used the term “telepresences” to describe remotely controllable tools, so is Sheridan [52], who used telepresence in reference to *teleoperation*, as well as *teleroobotics*. Ijsselstein [78] has considered teleoperation as one of the roots of presence research.

Therefore, the concept of telepresence is associated with the notion of *physical* (or *spatial*) presence that is created by technology [75, 79, 80]. Telepresence is defined in [81] as “the perception of presence within a physically remote or simulated site”, and in [52] as the “feeling like you are actually there”. In [79], telepresence is defined as “objectively present in a real environment that is physically separated from the person in space.” Physical presence is also known as tele-existence [80] or spatial presence [64][82][83]. It is a subjective feeling of being physically located in a mediated environment.

A person’s sense of presence can be labile such that one would variously feel present in either the natural-physical world, the mediated-virtual world or the imaginary world at one point of time. Therefore, to have physical presence, it involves the transportation of spatial cognition from the natural environment to the mediated one through evolving technologies of sensory engagement, motor engagement, and sensorimotor coordination [84]. These technologies would enrich the user’s feeling of being located in the mediated environment.

In video-based telecommunication, spatial presence is the ability to transfer auditory and visual spatial cues [83]. Although no telemanipulation or teleoperation is involved during a video-based telecommunication, spatial presence can be enhanced with immersive technologies (such as better displays [85][86] or tracking systems [83]) that match and support the spatial fidelity of the real world [81].

2.2.2 Social Presence

Communication scholars would describe telepresence as a mediated presence [87] or a mediated perception [77] of an environment, in which the feeling of presence is experienced by means of a *telecommunication* medium. The role that telecommunications has played in the social relations is significant and is becoming increasingly important [88][89]. In [81], telecommunication was considered an exceptional case of telepresence where the main

activity during a telepresence session is communicating as opposed to performing a physical task. Communication was not considered a necessary component to establish a sense of physical presence [66][81] but it is essential for *social* presence [66].

Social presence was originally defined in the telecommunication context [12][90], as “the degree of salience of the other people in the interaction and the consequent salience of the interpersonal relationships” [12]. It is posited that telecommunication means (such as video, telephone or email systems) vary in their degree of social richness, and social presence is a measurement of how much effect telecommunication media can have on communication. Later in [91], the definition of social presence has been refined to “the degree to which a person is perceived as ‘real’ in mediated communication.” Therefore, face-to-face interaction is a benchmark for the social presence theory.

However, this definition is applicable to a user’s experience to other humans only [75]. Many virtual reality researchers [65][66][75][92][84] argued that social presence can also be produced through computer-generated beings. Therefore, in the virtual reality domain, social presence is described as the mediated feeling of being together, of social interaction with other intelligences (i.e. computer-generated virtual agents, physical machines, live animals, or remotely located communication people). In this case, the amount of social presence is the intensity of the user’s empathetic simulation of another’s internal states, which is based on “bodily motions and cues” [84]. On the other hand, these authors (such as [65], [66] and [93]), tend to use Goffman’s notion of co-presence [94] to describe a sensation of two or more physically-separated humans being there together in a shared place.

2.3 Adding Autonomy

Currently, most of the commercial telepresence robotic avatars are manually controlled remotely; the inhabitant has to move the robot through an environment (and at the same time having a social conversation with the interactants), with all provided control. The inhabitant would need to pay attention to the surrounding where the robot is navigating, and a good user interface can help with situation awareness [95]. However, the inhabitant is still required to spend a great amount of time to consider the way to control the robot and this would lessen the experience of telepresence. In the attempt to enhance telepresence, levels of robot autonomy can be implemented to imbue these robotic avatars with some semi-autonomous or autonomous behaviors. These behaviors should strive to balance the inhabitant’s cognitive workload and engagement [96].

2.3.1 Levels of Robot Autonomy

Autonomy has always been considered along a continuum between zero autonomy to full automation. Several research, such as [97], [98] and [99], have proposed a spectrum of autonomy based on ten possible automation or autonomy levels. These autonomy levels begin from the lowest extreme where the human performs the whole task manually (Level 1), to autonomy at a middle level where the system executes an action upon human approval (Level 5). The highest autonomy level (Level 10) is where the autonomous system handles the whole task with no human intervention.

A robotic system can be designed such that its level of autonomy is fixed for the entire duration of a task. This kind of fixed autonomy is also known as static autonomy. However, due to the complex nature of HRI, it is likely that the level of robot autonomy needs to fluctuate or change throughout an interaction within any given environment, task or interaction. Therefore, with adjustable autonomy [100] (or sliding autonomy), the level of robot autonomy can be dynamically adjusted, either by the human user or the robot itself, according to the needs of the situation.

The confidence of using robots with autonomy can be gained when these robots are able to function accurately, and with technical reliability, for all situations [101]. Autonomous robots, at any level of autonomy, should have the capacity to perform within the human's expectation [62]. Furthermore, when designing autonomous behaviors for telepresence robots, it is essential to consider how these robots can navigate in the human environment in a socially accepted manner [102]. Besides safety, it is also important to consider the comfort level of the interactants and bystanders for every movement made by the robot [96]. The people around the robot will feel uncomfortable if the movement of the robot is abrupt, jerky or unstable [103]. When the interactants feel uncomfortable, they would try to move away from the robot, and this may lessen the inhabitant's telepresence. The interactants' comfort is also affected by social proxemic [34][104] between the robot and human as well as motion speeds [32][103] of the robot.

2.3.2 Levels of Autonomy in Navigation

It is desirable to implement autonomous navigation behaviors on telepresence robots for safety and for ease of use. Most of the commercial telepresence robotic avatars would have the capability to detect obstacles including negative obstacles such as cliffs or stages (see Table 2.1 "Sensor Suite"). However, only some would utilize this information to perform autonomous navigation behaviors such as collision prevention or obstacle avoidance (see Table 2.1 "Autonomous Behaviors").

With autonomy, the robot is allowed to take immediate reactive actions, based on its sensors' readings, to provide for a safer control under varying network conditions. For

example, the Texai robot [105] would continuously sense its surroundings with its on-board sensors and modify the inhabitant's teleoperation commands if there are obstacles, to prevent collisions. This behavior is known as assisted navigation and it helps to ensure that the movement of the robotic avatar is safe for the people who are physically present with the robot. Additionally, it helps to prevent damage to the robot, as well as to the environment. In this manner, the inhabitant is relieved from the technicalities of robot navigation and can concentrate on other tasks.

Another typical example of an autonomous navigational behavior is to command the robot to move to a defined goal destination. This will be especially useful if the robotic avatar is able to move to a specified location at a scheduled time on its own for a telepresence meeting. This functionality would save the inhabitant the effort to drive the robot from the charging station to the meeting area. Auto-docking behavior can also be implemented to command the robot to autonomously navigate to the charging station and dock for recharging, after usage.

Other desirable autonomous navigation behaviors can be implemented to allow the robot to navigate in a manner that is human-like and socially accepted by humans. For instance, a "follow a person" behavior can be implemented to allow the robot to autonomously keep pace and maintain a comfortable proxemic distance with the person physically present with it. The autonomous person-following capability would be useful for the "walk-and-talk" scenario, as it would reduce the workload of the inhabitant. The attention of the inhabitant can be dedicated to the conversation and not the navigation.

However, it has been argued that with autonomy, the teleoperator's sense of engagement may decrease when the robot is performing its autonomous behaviors [81] [106]. This is because, when the robot is moving autonomously, the teleoperator is like a passenger who waits for the robot to arrive at the goal position. The attention of the teleoperator may be switched away from the robot's environment, and this would impact the sense of presence negatively. However, in the case of a telepresence robotic avatar that is used for telecommunication, this sense of presence can be affected by other factors, such as the social interactions with the people in the robot's environment. The task that the inhabitant has to perform is not limited to navigational tasks, hence autonomous behaviors are expected to reduce cognitive workload but not the sense of telepresence.

2.3.3 Levels of Autonomy in Gesticulation/Manipulation

Since the telepresence robotic avatars are used for communication, the ability to gesture (or tele-manipulate the telepresence humanoid arms) can contribute to a richer telepresence experience for both the inhabitant and interactant.

This is because gesticulation (or non-verbal cues) is naturally a part of spoken language

during the conversations between people. However, no commercial telepresence robotic platform is able to receive and reproduce the physical gesticulation exhibited by the inhabitant yet. Although research platforms such as Geminoid androids [36] and Mebot [38] allow the physical gesticulations to be projected on their robots, these robots are unable to navigate, from room to room, like a typical telepresence robot.

Direct control of the robotic arms (or manipulators) is provided to the Geminoid robotic avatar by using an optical tracking system [36]. This tracking system can be expensive and the inhabitant is required to wear the optical markers, which can be uncomfortable. Furthermore, there is a difficulty of managing which movements should be mapped onto the robot and which ones should not (example hand motion for typing on a keyboard). On the other hand, direct tele-manipulation is applied to Mebot [38] by using a passive set of manipulators as its controller. The controller has exactly the same hardware joints configuration as the robot's manipulator, such that when the inhabitant moves the joint of the passive controller, the same joint at the robot moves correspondingly. The Huggable robot [107] and Robothespian robot [108] are another type of teleoperated social robots, made especially for interaction with humans. The gestures on both the Huggable and Robothespian robots can be pre-programmed. This allows the teleoperator to select which gestures to perform, via clicking on a user interface, and the robots will produce the selected gesticulation autonomously. Non-verbal gestures for social interactions can also be generated autonomously via a non-verbal behavior generator [109]. This can help to create a more immersive experience for users and improve the effectiveness of communication.

Commercial telepresence robotic avatars are not yet capable of tele-manipulation and only a few telepresence systems have pointing capability via a laser pointer (i.e. ProP [40] and QB [21]) or via a motorized manipulator (i.e. MantaroBot TeleTrak [110]). The laser pointer would give the inhabitant the ability to point at a physical object in the robot's environment. This is more specific and precise than just rotating the robot (or the robot's head) to face the object. The laser pointer on QB is manipulated semi-autonomously when the inhabitant performs a click-and-hold on the location to point at in the robot video.

Besides gesticulation and pointing, tele-manipulation can be applied to grasp and manipulate objects at the robot's site. It has been reported in [67] that one inhabitant actually felt disabled when using a non-anthropomorphic telepresence robot because he has to keep asking the interactant to perform the physical manipulation (e.g opening doors and picking up things) for him. In such a case, the limitations in robot hardware have degraded the user's telepresence experience. It is more difficult to perform full tele-manipulation than to teleoperate a wheeled robot to navigate around.

Therefore, allowing the robot to have shared control, in the form of low level skills and functions, would reduce the inhabitant's workload significantly for many tasks. For example, shared control have been implemented to perform several hand primitives for the NASA

robots [111][6] such that the remote user's hand motions for specific grasps are simplified. There are also research works on combining telepresence and grasp planning to achieve shared grasping [112], as well as utilizing 3D point cloud for mobile manipulation in domestic environments [113, 114]. However, none of these technologies have been deployed in commercial telepresence robots.

Effective communication using a current telepresence robotic system can be hindered by the lack of nonverbal communication cues, such as physical gesticulations and mutual gaze awareness. Gaze or eye contact is a critical aspect of human communication [115]. However, during a telepresence session, the users (both inhabitator and interactant) will naturally look at the face of the other user, which is rendered on a screen. They will not consciously look at the camera, which is typically located at the top or bottom of the screen. In this manner, the users would have an illusion that their conversation partner is always looking somewhere above or below them and not at them. This causes a lack of eye contact and hence makes the communication awkward and unnatural.

An automated gaze correction system [116] can be implemented within a video conferencing software to process the images that are captured by a camera in real-time. The system would extract the face segment from the original video frame, synthesize a gaze-corrected face from it and transfer the synthesized segment onto the original video frame seamlessly. This capability rectifies the awkward gaze problem in video-mediated communication by synthesizing eye-contact autonomously at the background. This enhances communication and hence the telepresence experience.

2.3.4 Levels of Autonomy in Conversation

The advancement in technologies has also enabled automation in many human-to-human communication services. Some of these technologies are even replacing many face-to-face interactions in educational [117] and business setting [118].

One example of how autonomy has been implemented in conversational service is the automation of the operator assistance calls with a menu telephony system. Such a telephony automation system offers a list of menus, and it will prompt the human callers to navigate to appropriate destinations to obtain a service or resolve a problem, with a synthesized voice. Although it is not as interactive as talking with a real human, such an automated system can provide the desired service in many cases, such as supplying an account balance or waiving off some charges from a credit card, autonomously. In the attempt to improve automated communication services, spoken dialog systems [119][120] and speech interfaces like "Siri", have been developed to understand speech inputs and provide the same services as though one is speaking to a real person.

Another example of automation in conversational service is the replacement of class meetings

(where the mentor imparts knowledge to the learners via face-to-face communication) with online learning (where knowledge is transferred through the use of video-mediated communication) [117]. The task of conversing in teaching has been automated because the action of presenting the slides verbally (or teaching) is pre-recorded as video and can be played anytime, anywhere without the teacher's involvement. Such *autonomous* teaching or presentation of slides can be implemented by using virtual reality (where the presenter is a virtual character in a software application) [121][122]. Robots, such as a social robot [123] or a telepresence robot [3] can also be used to represent the presenter, and autonomous communication behaviors can be implemented to deliver the presentation.

2.4 Metrics to evaluate a telepresence robotic avatar system

Evaluation of a telepresence robotic avatar system is particularly complex because many real-time interactions, as shown in Figure 1.3, are occurring at the same time. As discussed in Section 1.1.3, three types of interactions, such as HCI, HRI and HHI, are involved when a robotic avatar is used for telecommunication.

Various metrics can be used to evaluate these interactions. For instance, to measure the quality of HCI, which is the interaction between the remote inhabitator and the computer interface that controls the robot, the usability of the user interface can be tested. The quality of HRI, which is the geographically-separated interaction between the inhabitator and the robotic robot via an interface, can be measured by evaluating the task performance of the system. Another type of HRI also occurs when an interactant interacts with the robot directly in the local environment. This type of interaction can be evaluated by evaluating robot design as well as measuring the level of acceptance and attitude towards the robotic avatar. Finally, the quality of HHI is determined by how socially-interactive is the communication between the interactant and inhabitator via a robotic avatar.

2.4.1 Usability test of the inhabitator control station and GUI design

The inhabitator station is critical for creating the conditions for successful interaction. The usability of the inhabitator station would be the yardstick to evaluate the quality of human-computer interaction involved when using a robotic avatar. The performance is typically affected by how the *visual information* is being displayed on the user interface and how the robot is being controlled by various *control mechanisms* [124] [125].

As a telepresence robotic avatar is being teleoperated to navigate and manipulate remotely, visual information is required for providing the situation awareness to perform these tasks [124] [125]. These visual information, such as video feed, map, system and robot status, can be displayed on a video-centric interface, a map-centric interface, an augmented-reality

interface or a mixed-perspective exocentric interface. These interfaces have been compared in [126], and it was concluded that the mixed-perspective exocentric display interface was the most efficient as it provided the highest ease of use and task performance. On the other hand, the augmented-reality interface is useful for precise navigation tasks.

When using a telepresence humanoid avatar for telecommunication, there is a need to control the arms and fingers of the robot to perform some gestures or manipulation. The gestural or manipulation task can be controlled by using a mouse (or joystick) or by hand gesturing through the user interface. The advantages and disadvantages of using hand gesturing control in comparison with mouse or joystick control has been discussed in [127]. The main disadvantage of using hand gesturing control is the need to wear a data glove, and that the user would need to remember the set of available gestures for controlling the robot. On the other hand, it is advantageous to use hand gesturing control for operations such as grasping.

Subjective usability measurements, such as the TAM questionnaire [128], USE questionnaire [129] and the IBM computer usability satisfaction questionnaires [130], can be used to measure the usefulness, ease-of-use as well as the users' satisfaction with the usability of the inhabitator station and GUI.

2.4.2 Task Performance Evaluation

One of the main tasks that a telepresence robot can perform is to move around in a human environment. Navigation strategies can be evaluated by time-based or error metrics [131]. Time-based metrics are used to measure the speed of the task performance, while the error metrics measure the mistakes or damages made during the performance.

Another way to evaluate how different navigation strategies can affect telepresence is to measure cognitive workload [57] [132]. One common way to subjectively evaluate the inhabitator's mental workload is by using the six scales of the NASA-task load index (TLX) [9]. It is a questionnaire that was administered to rate workload according to six dimensions: mental workload, physical demand, temporal demand as well as the inhabitator's perceived level of task performance, effort and frustration. Other subjective measures may include the Cooper-Harper scale [133], the Workload Profile [134] and the Subjective Workload Assessment Technique (SWAT) [135]. A comparison of these subjective mental workload measures has been made in [136].

2.4.3 Acceptance and Attitudes Measures

The acceptance and attitude measures would help to investigate people's perceptions and emotions towards the usage of a robot. Typically, the Negative Attitudes toward Robots

Scale (NARS) [137][138] and the Robot Anxiety Scale (RAS) [137] is used to determine human attitudes toward robots – that is related to the feelings and opinions people have about robots. The NARS measures negative attitudes towards robots in three dimensions: interaction with robots, social influence of robots and emotional interactions with robots, while the RAS measures the anxiety toward the communication and behavioral capabilities of the robots.

In particular, the NARS measures have been used in [139] to evaluate the attitudes toward telepresence robotic avatars. The authors have performed an evaluation on video conferencing, a study on piloting a robotic avatar as well as an assessment of using the telepresence robot for interaction. They concluded that NARS is a suitable tool to measure attitudes in the telepresence robotic system domain.

Several factors, such as the robot's appearance, size, gender as well as the human-likeness behaviors and emotions of the robot, would influence the level of acceptance a person has on the robot [140]. Therefore, a series of questionnaires called "Godspeed" [141] has been developed to measure the users' perception of robots by assessing the five concepts – anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety – of robots. This series can be useful for robot creators to evaluate their creations without the prerequisite in experimental psychology.

2.4.4 Quality of Human-to-human Interaction

Besides the navigation task, a telepresence robotic avatar is mainly used by an inhabitant to perform the task of communication and interaction with another person remotely. The quality of this distant human-to-human (H2H) communication or interaction is dependent on the degree of telepresence and the quality of communication.

Several quantitative and qualitative performance measures for the communication part of the H2H interaction, such as the audio and video signal, have been discussed in [142]. It was concluded that the ITU-T Recommendation P.805 [143] can be used to evaluate the audio quality subjectively. On the other hand, the ITU-T Recommendation P.910 [144] can be used to rate the video conferencing quality based on the image's color, contrast, frame latency and etc. The authors in [142] had also applied the Presence Questionnaire [145] as well as the ABCCT (Affective Benefits and Cost of Communication Technologies) survey [146] to evaluate the quality of H2H interaction using a telepresence robotic avatar as a communication medium.

The quality of social presence, which is the feeling of presence toward people that are physically separated, can also be measured with quantitative instruments. Some of these Likert Scale examples that measures social presence or co-presence include the Networked

Minds Social Presence [93], the Social Presence Scale [12] and the Temple Presence Inventory [147].

H2H interactions utilizing a telepresence robotic avatar can likewise be examined objectively. In [79], the degree of objective presence is measured by calculating the expected value of the inhabitant's performance. On the other hand, the theory of spatial formations [148] can be used to analyze the way that the inhabitant orients the robotic avatar while interacting with the interactants.

2.5 Robotic System Architecture

An anthropomorphic telepresence robotic avatar, which is also capable of autonomous interaction, would require a large number of functions. Hence, a robotic system architecture is required to integrate these functions. The system architecture, or framework, refers to the entire system that has been divided into smaller sub-modules, as well as the interactions between these sub-modules [1]. These modules provide the functional capabilities of the framework [149].

The robotic framework would provide the connecting components to make various complex modules to work together to achieve the desired capabilities [1]. Besides system integration, it also serves as the cognitive architecture to allow the robotic avatar to produce all aspects of behavior using a fixed set of knowledge infrastructure, such as memory module [149] and functional processes [150]. Additionally, it also serves as the control architecture [1], which helps the robotic avatar to achieve real-time performance and concurrency so that the robot is able to react to dynamic and uncertain environments intelligently. Lastly, the architecture, being hierarchy and modular [1], can be used as a way to perform function decomposition. This simplifies development works.

2.5.1 Control Architectural Structure

The earliest robotic control architecture has three functional elements: sensing, planning and executing as shown in Figure 2.3. This approach is known as the *sense-plan-act* (SPA) or the deliberative paradigm. The core of many robotic frameworks and programming has been based on this paradigm for many years. The main features of SPA framework are that sensing would create and update a global world model, which is then used by the planner to produce a plan. The plan will be executed but the sensor's information will not be used directly during the execution.

However, the SPA robots have been slow to react in a dynamic environment [1], because it is time-consuming to build models and plan in the real world. Furthermore, task execution

Figure 2.3: The *sense-plan-act* deliberative architecture. Adapted from [1].

without using the sensors can be dangerous in a dynamic environment. This brought about reactive planning, such that plans can be generated fast and reactively, based on sensory information instead of internal models. The most influential work was Brook's subsumption architecture [2], which was also known as the *behavior-based* framework. A popular example of such architecture is Arkin's motor-control schema [151].

Figure 2.4 illustrates the subsumption architecture, which comprises a hierarchy of behavior-based modules organized into layers. Since multiple behaviors can occur concurrently, subsumption architecture has an arbitration mechanism that enables the higher-level behaviors to subsume the signal from the lower-level ones when the sensory information determines it.

Figure 2.4: The subsumption architecture. Adapted from [2].

Subsumption robots would constantly sense the world and react quickly using the sensory information. They have been considered fast and reactive to dynamic environments but not capable of long-term goal planning. On the other hand, SPA architecture is necessary for long-term goals, but it is slow [1]. To combine the strength of the two without their shortcomings, hybrid frameworks like 3T architecture can be used to integrate reactivity and deliberation.

3T architecture, as shown in Figure 2.5 is named after its three tiers of separated but interacting control processes, which consists of a reactive behavioral control mechanism, an executive plan-to-execution mechanism and a mechanism for performing the time-consuming deliberative computation. This architecture has been used on many

generations of robots and has named the 3 layers as the *reactive skill layer*, the *sequencing layer* and the *planning layer* respectively [152]. In the ATLANTIS project [153], these layers are called the *controller*, the *sequencer* and the *deliberator*.

Figure 2.5: An Example of a 3T architecture. Adapted from [1].

2.5.2 Cognitive Architecture

Cognitive architectures have been frequently used to model human performance in multitasking situations [154][155], such that the results of the human cognitive psychology are formalized in a computer model. Some formalized models are further being used to refine the theories of cognition [154] which contribute to the development of systems or agents with artificial intelligence. Hence, a cognitive architecture can be the underlying infrastructure that supports the acquisition and usage of knowledge to connect different

functional modules together, in order to achieve certain behaviors [150].

SOAR (State, Operator And Result) [156] is the classic example of a rule-based cognitive architecture and it has been the first to be applied in robotics. The SOAR architecture can help to subdivide information processing into rule-based elements rationally, but it does not consider the affective (such as social and emotions) elements within the system. Other cognitive architectures, such as the BDI (Belief-Desire-Intention) framework [157] can be used to integrate the affective elements as motivational system. ACT-R (Adaptive Components of Thought-Rational) [158] is another cognitive architecture that provide a theoretical framework for emulating human cognition, and it has been implemented in robotics for human-robot interaction [159].

The two main design properties that underlie the development of any cognitive architecture are memory and learning. There are various types of memory, which serve as a repository for background knowledge about the world and oneself, as well as about the current episode of activity. On the other hand, learning is the key process that generates this knowledge. Memory and learning are the fundamental components of cognition on which higher-level capabilities, such as deliberative reasoning and planning, are developed.

2.5.3 Connecting Components

All of the components within the framework are required to communicate, which includes exchange data and send commands, with each other. The choice of how components communicate, also known as the middle-ware, is one of the critical and most constraining decisions that one has to make while designing a robotic framework. There are two basic approaches to enable different modules or components to communicate: client-server and publish-subscribe.

The client-server protocol is similar to end-to-end communication, where individual components talk to one another directly. The remote procedure call (RPC) is an example of the client-server communication protocol, where the client component can call functions and procedures of the server component.

In a publish-subscribe (broadcast) protocol, the publishers do not send the data directly to specific receivers, called subscribers. Instead, the publisher component would publish data to a message queue, without knowing whether there will be any subscriber interested in that data or not. Any other subscriber component, which needs that data, can subscribe to that data without establishing a direct connection to the publisher. Basically, a centralized process would route the data between publishers and subscribers. Some of the popular publish-subscribe middle-ware solutions for robotics are the Inter-Processes Communication (IPC) [160] and Robotic Operating System (ROS) [161].

2.5.4 Examples of existing robotic system framework

A number of robotic system frameworks that support telepresence and social interaction between human and robotic avatar are discussed in this section.

System framework of the ROTI telepresence robot

The first example is ROTI [3], as shown in Figure 2.6. It is a non-anthropomorphic telepresence robot, which has been used by a remote tutor to conduct English lessons in Korea classroom. The robot has a static head screen and the tutor's face is projected on it. This helps to facilitate various conversational-related tasks, such as talking and displaying facial expressions. The remote tutor can also control the robot to perform various navigational tasks.

Its robotic architecture depicts a typical framework of a telepresence robotic avatar, which comprises the hardware components as well as the software controllers to provide for the navigational and conversational aspects of telepresence. No manipulator is installed on ROTI, hence the gestural/manipulation aspect of telepresence is unavailable.

In addition to the basic modules of a regular telepresence robot, ROTI has a subsumption model, which allows inhabitant to adjust the level of autonomy in navigation for collision prevention. Hence, it also has sensors and a sensing module to detect for obstacles.

Figure 2.6: (a) ROTI (b) System framework of ROTI, adapted from [3].

System framework of the ERICA and Geminoid HI-1 robots

The second example is the framework of ERICA [4], as shown in Figure 2.7. ERICA is a type of Geminoid robotic humanoid. Besides anthropomorphic, it looks exactly like a human being, as seen in Figure 2.7(a). However, unlike other Geminoid humanoids, ERICA is not a telepresence humanoid, hence no teleoperation functionality has been developed.

In another word, ERICA is a fully autonomous social robot. It is used mainly for having a social conversation with the people around her. The humanoid is capable of performing conversational tasks such as talking, as well as gestural tasks such as non-verbal gesturing with its hands. However, as ERICA is pneumatic driven and has to be connected to a compressor. It is not designed to move around in the human environment.

Therefore, the ERICA architecture only has controllers to provide for the conversational and gestural aspects of actuation. Besides the controllers, the components of ERICA architecture include sensing, perception, memory, as well as autonomous behaviors for conversational and gestural tasks. No navigation-related behaviors or controls are included.

On the other hand, the predecessors of ERICA, such as the Geminoid HI-1 [36], as shown in Figure 2.7(b), and Geminoid F are telepresence humanoid avatars that have similar appearance of the original person. The framework of the telepresence Geminoid [36] is different from ERICA's framework because it consists of the two basic modules for telepresence, that is a motion capture teleoperation interface as well as the controllers for conversation and navigation.

Here, it may be worth noting that robots such as these Geminoids have been considered by many to cause mental discomfort. Such experiences have been described in the Uncanny Valley theory [162] as human aversion to objects which achieve a high degree of physical

Figure 2.7: (a) ERICA (b) Geminoid HI-1 (c) System framework of ERICA, adapted from [4].

life-likeness but lack the full complement of nuances to make them indistinguishable from the same living thing which they imitate. In particular, it is felt that characteristics like the use of skin-like silicone as well as hair, may project some emotions of eeriness or fear. However, there are studies [163, 164] which also dispute the Uncanny Valley Theory. Regardless, the avatar robots used in this research effort attempt to avoid this issue by not attempting to copy human physical traits such as hair and skin.

System framework of the Robovie robot

The third example is the robotic framework of Robovie [5], as shown in Figure 2.8. This anthropomorphic humanoid system has been used as a conversational robot in a shopping mall for a research study in Japan. The architecture, as shown in Figure 2.8(b), consists of modules for functions like sensing, perception, memory, autonomous behaviors and controllers for conversational, navigational and gestural tasks.

This makes Robovie a social humanoid that is capable of performing simple conversation, gesticulation as well as navigation. The memory modules store the necessary information of the shopping mall and also the history of the interaction between the robot and human. RFIDs have been used to identify the shoppers. These help to establish a long-term HRI between the robot and the shoppers.

Robovie is not a telepresence robotic avatar, because it is not used for the purpose of telecommunication. However, it is a social humanoid avatar that has the teleoperation functionality implemented on it. A remote user is able to observe the environment where Robovie is, and share control with Robovie to select suitable behaviors for a certain scenario. According to its framework, speech recognition is performed by a remote user.

Figure 2.8: (a) Robovie (b) System framework of Robovie, adapted from [5].

System framework of the Robonaut robot

The fourth example is Robonaut [6], which is an anthropomorphic humanoid, as seen in Figure 2.9. This humanoid has been developed by NASA/DARPA to reduce extravehicular activity on-board the International Space Station. Robonaut started off as a purely teleoperator controlled robot, but throughout its development, new sensors and software have been included to allow for more shared control with the teleoperator.

Three different control modes — teleoperation, shared control and automation, have been implemented to form a general distributed control model, as shown in Figure 2.9(b). Within this framework, a teleoperator can directly control Robonaut or a simulated Robonaut, while data is collected for the robot to build learning algorithms for new skills. These learned skills or pre-programmed skills can be used later in a shared control mode to reduce the teleoperator's workload. In the automation control, Robonaut is able to autonomously perform any subtasks that it has already mastered. The human operator can monitor the system and intervene when Robonaut indicates it needs assistance.

This is the framework of the first version Robonaut, which can only be used to perform manipulation-related tasks. It is distributed into modules for functions like sensing, perception, controllers, as well as teleoperation behaviors for manipulation tasks. These are the core modules for teleoperation control. Subsequently, the framework has been enhanced to include the supervisory, memory, learning and autonomous behaviors modules to allow

Figure 2.9: (a) Robonaut (b) System framework of Robonaut, adapted from [6].

for more shared control with the human and with ever increasing levels of autonomy. The other version of Robonaut [165] may include the navigational aspect of control, but since Robonaut is never meant to be a social robot, it lacks the conversational aspect.

System framework of the Grace robot

The fifth example is the Grace (Graduate Robot Attending ConferencE) robot [7], as shown in Figure 2.10. This robot has a computer-animated face projected on a flat screen, as well as a mobile platform with a large array of sensors. It has similar hardware configuration as some of the commercial telepresence robots, as shown in Figure 2.2. However, Grace is an autonomous social robot, it requires more sensors such as laser range finder, pan-tilt unit, cameras, touch, infrared and sonar sensors.

The framework of Grace has modules for sensing, perception, memory, as well as autonomous behaviors and controllers to perform the aforementioned navigational and conversational tasks. It was mentioned that Grace is able to plan and perform various conversational tasks such as having conversations with the humans as well as giving a presentation talk at the conference. The robot is also able to plan and execute navigational tasks, such as moving around the building and taking elevator autonomously. Grace does not have robotic arms hence it is not capable of manipulation or performing any gestural tasks.

Figure 2.10: (a) Grace (b) System framework of Grace, adapted from [7].

System framework of the Kismet and Huggable robots

The last example is Kismet [8], as shown in Figure 2.11. Kismet is one of the pioneering works in social robotics. It is a social robot that is able to perform conversational tasks (such as speaking with varying pitch) and gestural tasks (such as making facial expressions) with emotions.

Besides the hardware components, the modules of the framework include sensing, perception, memory as well as controllers and autonomous behaviors for conversation and gesticulation. The major part of the framework is the inclusion of an emotion system for autonomous conversational and gestural behaviors. It allows the robot to simulate emotions (via speech or facial expressions) to respond to the stimuli that it perceived during an interaction.

Kismet is not a telepresence robot, however the research made during its development have been used to implement various teleoperated robots, such as Huggable [107] and DragonBot [166]. Huggable, as shown in Figure 2.11(b) and DragonBot are small soft-toy looking teleoperator-controlled robots that have been developed to interact with children.

Figure 2.11: (a) Kismet (b) Huggable (c) System framework of Kismet, adapted from [8].

Huggable was used in the pediatrics hospitals to mitigate stress and anxiety of child patients [167]. DragonBot, on the other hand, was developed to have long-term interactions with children either at home or in school.

Both teleoperator-controlled robots have similar framework design of a typical telepresence robot, which comprises rudimentary modules for teleoperation, sensing and controllers for conversational and gestural tasks. The design of the Mebot telepresence robot [38] was inspired by the method of control of Huggable.

Besides the basic module for telepresence, an emotion system was implemented on these platforms and the teleoperator is able to map a certain emotion to a primitive action. For example, if the primitive action is “wave hello”, the robot will move much quicker if it is “excited” vs “bored”. Additionally, a share control system has been implemented so that the human teleoperator is able to focus on the social portions of an interaction and the robot is given more autonomy.

Summary

Table 2.2 summarizes the various robotic frameworks, their special modules and the missing modules to support human-robot interaction when using an animatronic robot for social and telepresence interactions. Besides the hardware components, these modules include sensing, perception, controllers, memory, teleoperation, autonomy and learning.

Most of the example robots, except ROTI and Grace robot, have a set of robotic arms and hands. Social robots, such as ERICA, Robovie and Kismet (as well as Huggable and DragonBot), would use these robotic arms and hands for non-verbal communication, so as to enhance the user experience when interacting with these social robots socially. On the other hand, teleoperated robots, such as Robonaut, would use the robotic arms and hands for a wide array of task-oriented manipulations. Therefore, it is essential to consider what non-verbal gesticulation behaviors as well as what task-oriented manipulations when developing a robotic avatar for telepresence and social applications.

Robots	Type (Aspects)	Modules	Missing Modules	Remarks
ROTI	Telepresence (C,N)	Sensing, Controllers, Teleoperation, Autonomy	Perception, Memory, Learning	Adjustable Autonomy; Non-anthropomorphic

ERICA	Social (C,G)	Sensing, Perception, Controllers, Memory, Autonomy	Teleoperation, Learning	Human Appearance; Anthropomorphic
Geminoid	Telepresence (C,G)	Sensing, Controllers, Teleoperation	Perception, Memory, Autonomy, Learning	Human Appearance; Use motion capture for teleoperation; Anthropomorphic
Robovie	Social (C,G,N)	Sensing, Perception, Controllers, Memory, Teleoperation, Autonomy	Learning	Use RFID to track humans; Remote monitoring; Speech recognition via teleoperation; Anthropomorphic
Robonaut	Teleoperated (M)	Sensing, Perception, Controllers, Memory, Teleoperation, Autonomy, Learning	Lacks social-related Autonomy	Adjustable Autonomy; Anthropomorphic
Grace	Social (C,N)	Sensing, Perception, Controllers, Memory, Autonomy	Learning, Teleoperation	Virtual character on mobile platform; Non-anthropomorphic
Kismet	Social (C,G)	Sensing, Perception, Controllers, Memory, Autonomy	Learning, Teleoperation	Capable of having affective conversation; Non-anthropomorphic

Huggable	Teleoperated (C,G)	Sensing, Perception, Controllers, Memory, Teleoperation, Autonomy	Learning	Toy-size teddy bear robot; Shared control for affective conversation; Non-anthropomorphic
----------	-----------------------	--	----------	---

Table 2.2: Examples of robot with special modules and the missing modules to support human-robot interaction. Aspects: C denotes Conversation, G denotes Gesticulation, and N denotes Navigation.

2.6 Summary of the Literature Review

In this chapter, the concept of telepresence has been reviewed, with a pertinence to teleoperated robotic systems and telecommunication systems. Some of the existing commercial telepresence robotic systems and the necessary state-of-the-art technologies have been discussed in Section 2.1.1 to provide the background knowledge of the field. The advantages of using the telepresence robotic avatars for communication has been highlighted in Section 2.1.2 and this stipulates a motivation for conducting research that would enhance the telepresence robotic system.

Next in Section 2.2, the concept of presence has been discussed and one way to improve a telepresence robotic system is to enhance the experience of presence in form of physical presence (Section 2.2.1) and social presence (Section 2.2.2).

The other way that can crucially improve a telepresence robotic system is by adding autonomy. Section 2.3 presented the applicability of using robot autonomy to alleviate workload and Section 2.3.1 provided some guidelines to aid in the development of a robotic system that have levels of robot autonomy. The subsequent sub-sections discussed some of the ways of how automation have been or can be added to a telepresence robotic system, in the aspects of navigation, gesticulation/manipulation and conversation.

In Section 2.4, various metrics to evaluate a telepresence robotic system have been reviewed. Some of these metrics can be used later in this research to evaluate the developed telepresence robotic framework.

In designing and developing a robotic framework, issues like control paradigms and architectural styles (Section 2.5.1) as well as cognitive functions (Section 2.5.2) and middle-ware (Section 2.5.3) should be considered. Lastly, in Section 2.5.4, the survey of a few existing robotic frameworks have helped to identify the various modules needed to implement a teleoperated, yet social, robotic avatar.

2.6.1 Chapter Conclusion

Considerations for a robotic framework for social telepresence have been discussed. As the development of social telepresence robots progresses, robots have been imbued with more and higher levels of autonomy. This would reduce the effort spent on teleoperation and increase the opportunity for communication and social interactions. The functionality of a telepresence robotic avatar system can be categorized into three aspects: the navigation, verbal conversation as well as gestural interaction. If robot autonomy is to be implemented in a robotic framework that increases safety as well as to enhance telepresence experience, the various aspects of autonomy need to vary according to the desire of the inhabitant as well as the needs of the situation.

In a review of social robotic capability, current robots have been developed with the various capabilities, albeit on different robots with different levels of autonomous capability. Many social robots are able to have a conversation with a human autonomously. Some of them even have the capability to exhibit non-verbal gestures with their robotic arms and hands when they speak. There are also researches that allow social robots to express emotions as well as to move around autonomously in human environments, with social awareness. It is envisaged that the limited prevalence of sophisticated social robotics stems from ignoring that the success of deployment is beyond merely enhancing autonomy but in the understanding of a need of varying autonomy with respect to the needs of the situation and nature of interaction intended.

CHAPTER 3

AUTONOMY LEVELS FOR SOCIAL AND TELEPRESENCE ENABLED ROBOT FRAMEWORK

The ALSTER (Autonomy Levels for Social and Telepresence Enabled Robot) framework is first described in this chapter. The design of this framework stems from an attempt to advance technology for social and telepresence robots according to the research needs identified within the literature survey shown in the preceding chapter. While the prevalence of social and telepresence robots appears to be at an inflexion stage, with numerous capabilities accompanying such robots, these abilities seem isolated to their own hardware platforms. As such, the framework proposed here is to structure these useful abilities cohesively, according to the aspects key to such robots, namely in navigation, gesticulation and conversation. Furthermore, the framework's design also adheres to the need for various levels of autonomy associated with each of these three aspects.

The ALSTER framework, as shown in Figure 3.1, is decomposed into components and sub-components to encourage modularity in software development. This modular approach allows effective implementation, testing and validation of the components of similar functionality. This is an important consideration during the development of each component. It also allows easy replacement or inclusion of components without affecting the rest of the framework.

In order to provide a technology road map of how this social telepresence humanoid avatar system can be developed, the framework has been classified into six main components:

- Hardware Component (purple) — that comprises the robot hardware components such as sensors and actuators, as well as their respective software driver interfaces.
- Navigation Service Element (green) — that concerns the movement of the mobile robotic avatar.
- Gesticulation/Manipulation Service Element (orange) — that concerns the motion of the upper body parts of the humanoid avatar, including the head, arms, hands as well as waist manipulators, in this case for gesticulation. This can be extended to include manipulation motions.
- Conversation Service Element (blue) — that concerns the conversational aspects of using the robotic avatar for communicating with one or more humans.
- Social and Telepresence Tasks Management Component (yellow) — that concerns the human-to-robot interfaces as well as the higher level cognitive systems, which enable

Figure 3.1: Overview of the ALSTER Framework

the robot for various social (autonomous) and telepresence (human-in-the-loop) applications.

At the remote site, where the inhabitator is, there is an

- Inhabitor Control Station (gray) — that comprises an inhabitator-robot interface for controlling the humanoid robotic avatar and a graphical interface to display the information received from the robot.

The remote inhabitator or teleoperator is able to interact with the framework via the inhabitator-robot interface. This enables the teleoperator to adjust configurations of the framework as well as to teleoperate the robotic avatar remotely.

The objective is to provide a framework such that the robot adopting the framework can be easily developed to perform any humanoid-mediated telepresence or social tasks. In comparison with other telepresence robotic systems and frameworks, the key contribution of the ALSTER framework is that it has a layered structure and that it is versatile.

The ALSTER framework has been organized in a layered architecture to support manual and autonomous navigational, gestural and conversational behaviors within a complex robot for telepresence and social applications. These layers can be implemented independently,

as each layer comprises modules that are specialized and reusable. Modules such as the hardware configurations, motor controllers, sensing and basic perception form the lower-level layers of the framework. On the other hand, behavioral modules with some degree of autonomy are implemented in the high-level layers.

The ALSTER framework is also versatile. It can be tailored to create different applications on different robots, ranging from teleoperating a humanoid for telecommunication to deploying a highly autonomous robot for social interaction. The framework is also hardware independent. It is meant to be a software component of a functional robotic system, with appropriate hardware and middle-ware software installed.

3.1 System Architecture of the ALSTER Framework

Schematically, the components of the ALSTER framework in Figure 3.1 can be organized into a hybrid six-layers system architecture as depicted in Figure 3.2. The framework is hybrid because both the sense-plan-act strategy for deliberation and behavior-based scheme for reactive actions are used. The presented architecture has the following six layers: physical, reactive, executive, deliberative, supervisory and social.

Figure 3.2: Schematic architecture of the multi-layered ALSTER Framework. The operator control station has been omitted from this schematic architecture. A rectangle represents a module and a dotted line rectangle means that there can be multiple instances of such a module. The blue, orange and green bands correspond to the conversational, gestural and navigational aspects of the framework respectively. This same layered framework can be implemented on different robots although each robot can have different component implementations and hardware configurations.

Starting from the right, the first layer is the physical layer, which comprises the physical hardware components of the robot, as well their respective software interfaces. This

includes sensors, a mobile platform for navigation, a humanoid upper body for gesticulation, and various output devices to make up the telepresence or conversational aspects of using the avatar for interacting with people.

The next three layers (reactive, executive and deliberative layers) correspond to the popular 3T architecture [152]. The reactive layer handles the low-level control of the sense-and-act loops, while the deliberative layer utilizes prior knowledge and future projections to plan a global solution for a given task. The executive layer is the layer between these two layers, which converts a deliberative solution into a sequence of commands for the reactive layer.

To include higher levels of autonomy for social and telepresence enabled robots, the ALSTER framework comprises a supervisory layer, which is based on the cognitive architectures described in [168][169]. The purpose is to check the progress of other layers, making sure that the robot is performing its task accurately. If problems or unforeseen issues arise, it would respond with some recovery actions.

Additionally, the ALSTER framework includes a social layer to deal with the various inhabitator-robot and interactant-robot interactions, as mentioned in Chapter 1.1.3. It establishes and handles the communication with the human inhabitator and interactants.

Lastly, the blue, orange and green bands correspond to the conversational, gestural and navigational aspects of the framework respectively. Each of the three systems for conversation, gesticulation and navigation service elements, as seen in Figure 3.1, are represented in the schematic of the ALSTER architecture. They are highlighted by their correspondingly-colored band in Figure 3.2. It shows the sense-plan-act function within these three systems as participating within the physical, reactive, executive and deliberative layers.

In summary, as seen in Figure 3.2, there are ten subsystems, arranged within the six layers of the ALSTER framework. They are the hardware components, sensing, perception, controllers, action manager, memory, learning, robot autonomy, session supervision and the human-robot interfaces. The description of each subsystem is provided in the following sections.

3.1.1 Hardware Components

The robot hardware components include various sensors and actuators. Information about the robot's internal state and its surrounding, is acquired through proprioceptive sensors (such as battery sensor, encoders, gyroscopes, etc.) and exteroceptive sensors (such as camera, distance sensors, microphone, etc.) respectively.

The conversational avatar component includes the actuators as well as the peripherals to display information for human-robot interactions. For example, you can output the

telepresence-related data such as audio speech and video imagery to a loudspeaker and a flat screen display respectively. The video imagery can also be projected on a rear projection head unit, if it is available.

Gestures or arm movements of the inhabitator can be output to a humanoid avatar that has an upper body torso. Lastly, the control commands produced by the navigation modules can be sent to the mobile platform component so that the robotic avatar can be steered around in the local environment.

Here, it should be noted that physical hardware is categorized for the framework only within this module; that is, all other modules in the framework are software modules. Examples of hardware components, which are used for the implementation of the ALSTER framework, can be seen in Section 4.1.1 and Section 4.1.2.

3.1.2 Sensing Subsystem

Figure 3.3 depicts the sensing subsystem of the ALSTER framework. The sensing subsystem collects the raw signals from the environment and performs signal conditioning. The robot can gather information from the environment by using various sensors such as cameras or a laser rangefinder. The intended information is acquired in form of raw electric signal, together with other irrelevant data and noise. Therefore, the sensing subsystem not only provides a driver or interface to read the raw signals from a specific hardware, but it also performs signal conditioning to remove noise.

Figure 3.3: Sensing subsystem of the ALSTER Framework

3.1.3 Controllers

In the ALSTER framework, the controller subsystem includes the motion controllers for the navigational, gestural and conversational aspects of the humanoid avatar. In this case, the framework has the motion controller to control the mobile platform, a joint controller for

controlling the humanoid upper body torso and the avatar controller for the avatar system respectively.

The motion and joint controllers contain the motion profiles and implement a control loop mechanism to move the motors and actuators. They are usually closed loop, and combinations of Proportional-Integral-Derivative (PID) controllers are implemented for the control loop.

The avatar controller controls the sound and display quality of the avatar system. If the avatar system includes a motor-actuated face, then the actuator control of the face muscles will be implemented in the avatar controller.

3.1.4 Action Manager

The action manager is the interface between the deliberative layer and the reactive layer, as shown in Figure 3.2.

The components of the action manager are as illustrated in Figure 3.4. It takes in the teleoperated actions from the human-robot interface, as well as directive plans from any existing planners concurrently. Each directive plan comprises a list of actions to perform a subtask (such as obstacle avoidance, move along a path, nod the robot's head).

Figure 3.4: Action Manager subsystem of the ALSTER Framework. The green, orange and blue squares represent the navigational, gestural and conversational actions respectively.

The action manager would coordinate the plans and schedule these actions, which can be navigational, gestural or conversational type of actions, into a specific order based on their execution time. For example, in Figure 3.4, at time t , the action manager will send to the controllers two actions for execution — a navigational action and a conversational action. For illustration purposes, navigational action 1 can be rotate left and conversational action 1 can be say “hi”. Subsequently, at time $t + 1$, it will send three actions to the controllers — move forward (navigational action 2), say “how can i help you?”(conversational action 2) and move the robot head down (gestural action 1).

3.1.5 Perception Subsystem

In order for the robotic avatar to safely share the environment with humans, it is necessary for the robot to extract information from the events, people and objects of its environment, and infer relations between them. The ALSTER framework provides a perception subsystem, which includes multiple feature extractors, target and event detectors as well as a localization and mapping module.

The product of the perception subsystem is perceptual knowledge, which can be used for deliberative planning, reactive control or be stored in the memory subsystem, as shown in Figure 3.2. Table 3.1 shows the different types of perceptual knowledge that can be generated.

Type of Perceptual Knowledge	Description
Extracted Features	Desired portions or shapes (features) extracted from the sensor data to provide understanding of the situation or environment that the robot is in. For instance, lines, distinctive sounds, estimated pose, visual markers, etc.
Situational Knowledge	This is the knowledge of the situation that the robot believed that it is in exteriorly. This includes the layout of the environment (such as walls and obstacles), events (such as a loud noise) or the presence of other objects, robots and people near its surroundings.
Spatial Knowledge	The location as well as kinematics that the robot believed that it is in with respect to the environment. For instance, pose, distance to obstacles or target objects.

Table 3.1: Different types of perceptual knowledge

Multiple low-level feature extractors can be used to recognize features from the sensor data. Some of the feature recognition processes can be cascaded, meaning that the products of the low level feature extraction processes can be combined and used as inputs to a high level recognition process, as seen Figure 3.5. They can be combined to provide understanding of the situation or environment that the robot is in. Examples of extracted features include lines, distinctive sounds, color, and etc.

The localization & mapping module in ALSTER practically produces the spatial knowledge to enable the robot to perceive its own location, with respect to the extracted features that it acquired about the geometry of the environment. This localization process is commonly known as simultaneous localization and mapping (SLAM).

Multiple event and target detectors can be implemented to generate different situational knowledge about the relations between the geometry of the environment, the dynamics of the objects and the behaviors of the humans within the environment. For instance, an

Figure 3.5: Perception subsystem of the ALSTER Framework

target detector can be used to detect human in the environment, and provide situation knowledge, such as the number of people, the location of these people, The event detector can be used to determine whether they are walking or seated down.

In ALSTER, it will be configured in the task model what situation knowledge is needed for the task, and the perception subsystem would gather and provide the relevant situation knowledge in one symbolic representation.

3.1.6 Robot Autonomy

The humanoid telepresence avatar is a hybrid of a social and telerobotic robot. Therefore, besides planning for the telerobotic tasks, there is an additional requirement to enable the robot to have social interaction with its interactant.

The robot autonomy subsystem of the ALSTER framework, as shown in Figure 3.2, is a planning engine to create autonomous behaviors for the robot to perform. It has an overall task-oriented planning coordinator and three other more specific planners. The plan coordinator manages and regulates three different types of planners to automatically generate the plans (or a list of actions) in order to accomplish a particular task, such as being a tour guide or telling a story. It also handles notified problems by requesting the planner to perform a re-plan.

The framework allows additional planners to be implemented and added to the planning engine, if necessary. Currently, three planners, which correspond to the navigational, gestural and conversational aspects of the framework, have been implemented. They can be activated or deactivated depending on the configurations of robot and task.

The three planners include the dialogue planner, which understands the flow of the conversation and generates speech or dialog in response to the state of the conversation. The trajectory planner, which creates a motion plan for the upper body torso. In our case, our humanoid robot EDGAR has a 3-axis waist, 2-axis head module, a pair of 5-axis robotic arms and two 6-axis robotic hands. Currently, the trajectory planner plans for gesticulation motions, however it is expandable for manipulation motions as well. Lastly, the path planner that produces routes for the robotic avatar to navigate around its surrounding safely.

3.1.7 Memory Subsystem

The memory subsystem stores and provides procedural information and constraints to be executed by the perception and robot autonomy subsystems. This includes knowledge or facts about the external world of the robot (world model), the information needed to perform the task (task model), as well as the geometric configuration of robotic avatar (robot model).

This helps the robotic avatar in its reasoning process while planning or assessing the current situation, by linking the stored knowledge in the memory subsystem with the ongoing knowledge acquired from the surrounding. New knowledge acquired by the robot can be used to update the stored data in the memory subsystem.

3.1.8 Learning Subsystem

The robotic avatar can be equipped with a learning capability that learns and adapts, by recording and reusing the new data or information that the ALSTER framework generated. This can be achieved with data logging and subsequently, mining, processing and representing any useful dataset from the logs, for learning or reproduction. The learned data includes learned facts, configurations, tasks and flows of actions in the knowledge base.

The learning of new tasks and flows of actions can also be done in an off-line manner, by programming and setting the procedures manually. These are skills that are stored as procedural memory, in the form of a task model. On the other hand, new facts and configurations can be defined manually and stored as semantic information in the memory subsystem as data models. Incremental learning techniques can be applied to extend the existing learned models or retrain the existing model with the new input data acquired with data logging.

Additional machine-learning methods, such as symbolic, statistical, chunking and reinforcement learning, can also be implemented within the learning subsystem to enable the robot to learn and adapt in an unsupervised and on-line manner.

3.1.9 Session Supervision

The telepresence robotic avatar is predominantly a tele-operable robot but imbued with autonomous capabilities, such as autonomous person-following (Section 4.5.4) and reactive gesticulations (Section 4.6.4), when using the proposed framework. Thus, the supervisory layer is a pivotal package that (1) enables the remote inhabitator to assume a supervisor role when controlling the robot, and (2) ensures the robot to perform a task autonomously and accurately.

To do so, the ALSTER framework is designed with a session supervision subsystem that will ensure a telepresence session or an autonomous social application can function properly during run-time. It includes a task manager, a system monitor as well as a problem handler, as shown in Figure 3.2. A task manager tracks the status or progress of the task plans received from the planners and provides notification when a problem is detected.

The system monitor is a watchdog process, which monitors the performance of all actions and processes, and adapts to any unexpected changes during the execution of these actions and processes. If the robot is unable to adapt to the changes, the situation will be considered as a problem.

The problem handler manages the detected problems in a queue system for fault recovery, so that the robot is able to recover, from any deviation to its current plan. If the problem is unsolvable within the system itself, it would send a notification to the human-robot interfaces and request for human intervention.

3.1.10 Human-Robot Interfaces

The social layer is an important layer in the ALSTER framework, as it provides the human-robot interfaces for various social and telepresence applications. Two types of human-robot interfaces have been included to handle the communication between the robot and its two different types of human users.

Inhabitor-robot telepresence interface. The first type is the inhabitator-robot telepresence interface, which establishes the network link between the remote human inhabitator and robot, as shown in Figure 3.1. Unlike a teleoperated system, a telepresence system is used to provide for a long-distance communication between geographically separated people. The conventional network links that were used in teleoperated systems, including communication tether or local-area wireless network (such as Radio Frequency and Bluetooth) would not be feasible. Therefore, the network link for our inhabitator-robot telepresence interface is Internet-based. To emulate the characteristics of face-to-face communication in telepresence applications, the network link is also full-duplex. This is so

that both sides can experience real-time-like two-way communication, via the audio and video mediums.

Data received from the remote inhabitator side can include the the teleoperated action commands (e.g. gesturing a wave) and behavior (e.g. following a person) commands, as well as the task commands (e.g. delivering an item to a location). The received data will be decoded and disseminated to the various modules within the framework, as shown in Figure 3.2.

On the other hand, the robot will send the following data back to the remote operator via the inhabitator-robot telepresence interface. As shown in Figure 3.2, data from the robot may include the internal states of the robot or any data relevant to the task (such as the current task status and detected problems) or perceptual knowledge acquired from the robot's environment (such as map and detected events), as well as the raw conversational data (such as video and audio data), which is acquired from the camera and microphone directly.

Interactant-robot social interface. The second type of human-robot interface is the interactant-robot social interface. It provides the info-communication interface for the interactants (or the local users) to know the internal states of the robot, as well as the progress of the task and especially any detected problems. This information can be displayed on the robot's screen visually, or through auditory means, such as speech output and sound notifications.

The interactant-robot social interface also provides input methods for the local users to "send" action, behavior or task commands to the robot for social applications. For instance, an interactant can activate the robot's person following behavior, by telling the robot (through speech input) to follow him or her.

3.2 Ontology for Social and Telepresence Robot

An ontology for social and telepresence robot, as shown in Table 3.2, has been defined specifically for the ALSTER framework. This is to facilitate the semantic interoperability of the different systems within the framework as shown in Figure 3.2. The ontology defines a set of shared concepts that are categorized based on the inputs as well as the outputs of the components in the framework.

Ontology Concept	Description
Raw Signal	Raw reading obtained or measured directly from a sensor. Typically, the sensor would provide a sample software driver interface to acquire the raw signal.
Sensor Data	Processed data from the sensor.
Internal States	This includes the communication status (e.g. online/offline, signal strength, etc.), component status (e.g. battery level, temperature, etc.), robot status (e.g. moving, gesturing, speaking, autonomous/manual, etc.), perception status (e.g. a tracked target is within line of sight), etc.
Perceptual Knowledge	This is the output of the perception subsystem, to enable the robot to perceive the environment and situation that the robot is in. Three types of perceptual knowledge have been included in ALSTER: extracted features, situational knowledge and spatial knowledge. (Refer to Table 3.1 for their descriptions)
Stored Knowledge	This is the prior known information stored in the memory system, which includes the robot configuration, world model such as map, and task-related data.
Learned Knowledge	This is the output of the learning subsystem, which can include learned facts, configurations, tasks procedures and flow of actions. This knowledge is then stored in the memory as models and is retrievable for future usage.
Task	A task is a complex role to be performed by the robot (e.g. being a tour guide, delivering an object, etc.). It can be specified by the remote inhabitator or requested by the local interactant. A task can be broken down in a set of behaviors or subtasks.
Plan	A plan is a list of actions, produced by the planner, to achieve a complex task or subtask.

Action	An action is a primitive process that the robot can execute. In ALSTER, there are three kinds of action — navigational, gestural and conversational action. Some examples include moving forward, nodding the head and speaking. Teleoperated actions are actions that are produced by the remote inhabitator manual via the inhabitator-robot interface. Multiple actions can be performed simultaneously.
Behavior (or subtask)	A smaller task or a particular capability, such as the ability to avoid obstacles, to follow a person, to understand speech, and etc. Combination of behaviors can form a global solution for a task. Similarly, there are three kinds of behavior — navigational, gestural and conversational behavior.
Control commands	The control commands are values to be sent to the actuators to actuate the actions.
Problem	A problem regards a situation as undesirable or harmful for the robot to carry on. It needs to be dealt with and overcome. Examples of a problem include unachievable action, hardware or software failure, and etc.
Task Status	The information to track the progress of the task, including the state of processes and unexpected execution error.

Table 3.2: Ontology of the ALSTER framework

Currently, the ontology is formalized using the common data representations. One future work of this dissertation is to use a proper ontology language such as the Web Ontology Language (OWL) to specify this ontology.

3.3 Software organization of the ALSTER framework

The presented ALSTER framework has been implemented, and software organization of the framework is as shown in Figure 3.6. The subsystems are grouped into five main sets of software packages to cater for different possible usages of the framework. The details of the implementation will be described in the following chapter.

The first set of software packages, *ALSTER Libraries*, comprises software programs that implement specific algorithms (such as SLAM algorithms, pattern recognition algorithms, computer vision algorithms and event detection algorithms). These algorithms are

Figure 3.6: Software organization of the ALSTER framework. The dark-gray rectangles represent the operating system libraries, the gray rectangles represent the five main software packages of the framework, and the light-gray rectangles denote the different possible applications that can be implemented using the software packages of the framework.

standalone programs, implemented using cross-compilable languages such as C++ and Python. Hence, they are not dependent on the operating systems or the ROS system, and can be reused directly to build new social or telepresence robotic applications in Windows or Linux systems.

A set of software packages called the *ALSTER ROS Wrappers*, which are wrapper programs to adapt the existing ALSTER programs to use the ROS infrastructure and functions, was also included. The ALSTER ROS wrappers can also include programs that implement new algorithms within the ROS infrastructure.

The *ALSTER Hardware Interfaces* include software interfaces for various robot hardware components, sensors and actuators. For example, ALSTER includes software interfaces for Galil motion controller, Dynamixel motors, Hokuyo LIDARs, Microsoft Kinect RGB-D camera, and more.

The *ALSTER Conversational, Gestural, and Navigational Processes* include the deliberative processes, and especially the behavioral processes corresponding to the three aspects of the framework, as highlighted with the color bands in Figure 3.2. The processes are related to the planning system, the action manager, the controllers and the motion system of the ALSTER framework. This set of software packages integrates various general algorithms and modules of the ALSTER Libraries, with or without the ROS wrappers and libraries, to form functioning processes to perform a specific conversational, gestural or navigational task.

Lastly, the *ALSTER Social & Telepresence Central Processes* are the necessary software processes

that form the layered framework. They are the memory, learning, and supervision subsystem as well as the operator-robot telepresence and interactant-robot social communication interfaces.

3.3.1 Possible usages of the framework

This section describes different types of application that can be implemented using the ALSTER framework. As depicted in Figure 3.6, the light-gray blocks denote a few types of the application that can be developed. For instance, developers can use the ALSTER libraries to build an application of a certain algorithm, such as speech recognition or face detection, which can run in the Windows platform or a Linux platform.

However, the strength of the framework is that developers can use or combine any of its three service elements — navigational, gestural, and conversational — to build new social or telepresence applications quickly. For example, one developer can utilize the ALSTER libraries and processes, to build a mobile robot that is capable of autonomous navigation, such as a cleaning robot working in the human environment. The other developer can develop a humanoid that is capable of gesturing and conversing with humans, such as a therapeutic robot. And, when the developer combines navigational and conversational processes of the framework, the developer can build a mobile information kiosk. As the framework is scalable, gestural processes can be extended to possess manipulation capability; and by combining navigation with gestural processes that have manipulation capability, the developers can implement a pick-and-go or delivery robot.

Nevertheless, the goal is to utilize the complete framework to build a humanoid avatar that can move around, gesticulate and have a conversation with humans. And we have provided a framework that allows developers to do so platform-independently.

3.4 Features of the framework

In summary, the ALSTER framework is fully developed in C++ and Python, and from the viewpoint of a robotic software developer, the following features of the framework are considered more prominent among the others.

Navigational, gestural and conversational service elements. One important feature of the framework is that the collection of software modules is characterized by these three main classes of service elements. An assortment of social and telepresence robotic operations can be set up based on framework, ranging from fully manual to fully autonomous operations, from a mobile robot that is capable of navigation only to a complex humanoid system that is capable of navigation, gesticulation, and having conversation with humans.

Modularity. The other key feature of the framework is modularity. The components of the framework are individual software modules created for specific functionalities. The software modules can be tested individually and can be reused for different applications. Importantly, it enables us to compartmentalize the framework into modules related to navigation, gesticulation and conversation. In this manner, existing modules, or newly developed modules, can be assembled to construct different types of social or telepresence robotic systems quickly. This helps to ease the development process.

Scalability. The fundamental unit of the framework's software architecture is a process, and a module can be made up of one or multiple processes. Each process is an atomic executable unit that serves a specific functional role, and is able to run simultaneously and separately from other processes. This enables scalability, because developers can build upon the software packages by modifying, replacing or including new processes, without affecting the rest of the framework.

Versatility. The middle-ware that was used to support multi-processing in the ALSTER framework is the Robot Operating System (ROS) [161]. It is a widely used robotics middle-ware that enables message-passing between the processes. Besides providing the open-source libraries and tools for developing robot software, the ROS is also an ecosystem, which supports collaborative robotics software development. Hence, software engineers can use the existing ALSTER's processes and libraries, in conjunction with the ROS message handling interfaces, to create new processes.

Platform-independence. The platform-independence of the framework is two-fold. First, it can be run on different hardware robotic platforms and is compatible with a variety of sensors, owing to the advantages of the ROS ecosystem. Secondly, the framework's software packages can be used on both Linux and Windows operating systems. This is because the framework is implemented using cross-compilable languages such as C++ and Python, and ALSTER-ROS wrappers have been developed to guarantee interoperability between the operating systems.

As the framework uses the ROS middle-ware for its inter-processes communication, the ALSTER-ROS wrappers on Windows have included a set of client libraries that get ROS processes up and running on Windows. Alternatively, the Windows Subsystem for Linux (WSL) can be enabled to set up the entire ROS workspace within the Windows platform.

3.5 Chapter Summary

The conceptual design of the Autonomy Levels for Social and Telepresence Enabled Robot (ALSTER) framework has been described in this chapter. It comprises the hardware component, the navigational, gestural and conversational service elements components

as well as the management component. The components of the ALSTER conceptual framework are then organized schematically into a hybrid six-layers architecture, and the six layers are the physical, reactive, executive, deliberative, supervisory and social layers. The physical layer comprises the physical hardware components and their respective software interfaces. The reactive, executive and deliberative layers correspond to the popular three-tiers architecture. The supervisory layer supervises other layers to ensure safe robot autonomy. Lastly, the social layer handles the various human-robot interactions.

Ten subsystems are arranged within the six layers of the ALSTER framework. They are the hardware components, sensing, perception, controllers, action manager, memory, learning, robot autonomy, session supervision and the human-robot interfaces.

The hardware components, which constitute the physical layer, include various sensors and actuators. The reactive layer comprises the sensing subsystem as well as the controllers for navigational, gestural and conversational aspects of a robotic avatar. The action manager forms the executive layer and the perception subsystem, which produces perceptual knowledge for deliberative planning and reactive control, spans across the executive and deliberative layers. At the deliberative layer, the robot autonomy subsystem is the planning engine of the various tasks of navigation, gestures and voice interactions of a robotic avatar. Together with the supervisory layer (which comprises the memory, learning and session supervision subsystems) and the social layer (which comprises human-robot interfaces such as the inhabitator-robot telepresence interface and the interactant-robot social interface), the framework allows for the dynamic selection and management of autonomy in various navigational, gestural and conversational tasks of interactions.

The ontology of the framework has also been presented and the software organization of the framework as well as its features and possible usages is also described in this chapter. The following chapter will focus on the software implementation of the ALSTER framework.

CHAPTER 4

IMPLEMENTATION OF THE ALSTER FRAMEWORK

This chapter describes the implementation of the software modules and the processes that constitute the ALSTER framework. Figure 4.1 shows the comparison between the conceptual framework described in Chapter 3 and the implemented framework.

Here, the architecture is discussed from the perspective of sub-system implementation. This paragraph, together with Figure 4.1(a) describes the arrangement of nine subsystems within six independent software layers. Firstly, the physical layer consists of the drivers handling the hardware components (S1), including various sensors and output devices. Within the reactive layer, the controllers (S2) for the navigational, gestural and conversational aspects are represented. The sensing subsystem (S3) is also part of this reactive layer. The navigational, gestural, conversational service elements, (S4) are the result of integrating the action manager and the robot autonomy subsystems. The perception subsystem (S5) has been implemented to provide perceptual knowledge for the service elements. The subsystems for memory (S6), learning (S7) and the session-supervision (S8) are encapsulated within the supervisory layer. The final layer is the social layer, which contains the interfaces (S9) for inhabitator-robot interaction, as well as interactant-robot interaction.

Both diagrams (Figure 4.1(a) and Figure 4.1(b)) have been labeled accordingly so that corresponding components between the two can be compared easily. Figure 4.1(a) is the conceptual ALSTER framework described previously in Chapter 3, while Figure 4.1(b) illustrates the implemented software framework of ALSTER, re-organized yet still adhering to the 6-layered architecture of the conceptual framework. The layers have been labeled as L1 to L6.

In this chapter, the focus is on the implementation of the autonomy levels in the navigational, gestural and conversational service elements for a social and telepresence enabled robot. Hence, the deliberative, executive and reactive layers (L2 to L4 in Figure 4.1(a)) have been expressed collectively in the implementation as subsystems S2 to S6 in Figure 4.1(b). This is because, as described in Section 3.1, the service elements span across these three layers, which is also commonly known as the “sense-plan-act” paradigm.

Of particular interest in this project is the subsystem S4, which performs the roles of planning and preparing actions for the hardware controllers. S4 plans and prepares these actions across the *navigational*, *gestural* and *conversational* aspects of social and telepresence applications. This feature is highlighted in Figure 4.1(b) using the same three colors, each depicting the navigational (green), gestural (orange) and conversational (blue) aspects in

(a) Schematic architecture of the multi-layered ALSTER Framework, labeled for implementation.

(b) Implemented software architecture of the ALSTER Framework, labeled for comparison with the schematic architecture of ALSTER.

Figure 4.1: Implementation of the ALSTER framework, with a comparison between the software architecture and the schematic architecture.

Figure 4.1(a). S4 performs the planning and managing of actions via multiple autonomy levels within each aspect. In Figure 4.1(b), examples of these actions within each aspect have been listed according to autonomy level, with those of lower autonomy at the bottom of each colored block.

The implementation of the framework was executed in two stages. The first stage developed the prototype of a typical telepresence non-anthropomorphic robotic avatar such that the user is able to hold a video conversation whilst also control the robot to move around remotely. Iterative improvements were made to the software to increase the functionality of the non-anthropomorphic robotic avatar. These functionalities, such as the autonomous navigational behaviors and conversational behaviors, were added to the framework as various service elements in a modular fashion.

The second stage developed the prototype of a novel telepresence humanoid avatar such that the user is able to hold a video conversation whilst also controlling the robot to perform physical gesticulations remotely. We continued to use the ALSTER framework of the first stage to build the telepresence humanoid avatar. Additional functionalities were added to the framework in a modular fashion, as the software was developed to accommodate this. Autonomous gestural and conversational behaviors have been added to the framework as service elements subsequently to allow the inhabitator to use the humanoid with increased social presence and decreased workload.

Figure 4.1(b) is the resultant of the two-stage implementation of the ALSTER framework. It comprises the hardware configurations as well as the software modules that have been used or implemented while developing the two prototypes. In the following sections, the details regarding the implementation of the capabilities within the framework are presented. A total of 27 modules are presented here.

4.1 The Physical Layer — Robotic Prototypes

In this section, the appearances as well as the hardware components of the two robotic prototypes are described. This constitutes the physical layer of the implemented ALSTER framework.

4.1.1 First Prototype — MAVEN

The first robotic avatar is a typical telepresence non-anthropomorphic robotic avatar that carries a flat screen on a mobile platform. It is called MAVEN, which is an acronym for “Mobile Avatar for Virtual Engagement by NTU” [24][25]. A suite of MAVEN, as shown in Figure 4.2, had been built to evaluate the different configurations of flat screen display. They are labeled as MAVEN-1, MAVEN-mini and MAVEN-gray.

Figure 4.2: A suite of MAVEN non-anthropomorphic robotic avatars

MAVEN-1 has a 29-inches flat screen monitor mounted in the portrait manner. Its base is rectangular with side dimensions of 65 cm by 85 cm and it is 170 cm tall. The drive train is composed of four 12-volts geared motors paired with 15 cm mecanum wheels. Three webcams are mounted in the portrait manner to enhance the inhabitator view by providing for a wider field of view.

MAVEN-mini has the smallest footprint as the dimensions of its base are 30 cm by 30 cm and it is 110 cm tall. The robot comprises four 12-volts geared motors paired with 13cm mecanum wheels. The MAVEN-mini carries a 29-inches flat transparent screen, mounted in a portrait manner. The screen has been tilted at an angle of 60 degrees. This is because the footprint of the MAVEN-mini is small, therefore its center of gravity is low. Mounting the transparent display screen at human's eye-level is not feasible and so the screen is mounted at an angle so that the interactants can see the screen during video conferencing. A webcam has been mounted on top of the screen for video conferencing.

Lastly, MAVEN-gray is the latest version of the MAVEN series. It has a metal-casted base with side dimensions of 60 cm by 55 cm and it is 110 cm tall. The screen is a 29-inches flat screen monitor and it is mounted in a portrait manner, tilted at an angle of about 65 degrees.

Hardware Components of MAVEN

The MAVEN robots are holonomic robots. All of them have 4 mecanum wheels, and each wheel is driven independently by a drive motor. An on-board Galil motor controller has

been configured for controlling the drive motors. The robots are able to move forward, skid sideways and use differential steering to turn.

Each MAVEN robot is equipped with three types of sensors: a forward-looking camera, a 2D laser scanning rangefinder as well as an odometer. The camera would provide a view of the robot's surroundings to the inhabitator, via the network link. This allows the teleoperator to be aware of the robot's environment, facilitating teleoperation and autonomous robot control. Additionally, the camera comes with an inbuilt microphone. The video imagery and the audio data captured from the camera can also be used for video conferencing. As seen in Figure 4.2, the camera can be made up of a single webcam, a set of webcams, a Kinect sensor or a combination of webcams and Kinect sensor.

The 2D laser scanning rangefinder is used to scan the space in front of the robot and generate an occupancy map for navigation. This allows the robot and the human inhabitator to ascertain the presence of objects in front of the robot, which block the path of the robot. Hence, this sensor is crucial to both autonomous navigation as well as remote teleoperation.

Finally, the odometer monitors the wheel revolutions on the robot. Through this, it is possible to localize a robot by tracking the number of wheel revolutions which have occurred after the robot has traveled from a known location. Such a method is known as dead reckoning. Dead reckoning is unfortunately susceptible to cumulative errors due to wheel skidding and travel over uneven surfaces. The robot software, however, is able to address such errors by employing the use of the laser scanning rangefinder to help to localize the robot. This is presented further during the discussion of robot localization (Section 4.4.4) and navigation (Section 4.5).

4.1.2 Second Prototype — EDGAR

The second prototype is a novel telepresence humanoid avatar. It has an anthropomorphic upper torso and a person's face can be projected on the humanoid for telecommunication. This telepresence humanoid avatar is named EDGAR, which is an acronym for "Expression Display & Gesturing Avatar Robot" [47]. EDGAR has been designed for mimicking human movements from the waist up. The humanoid can twist, bow as well as lean to the side. With these capabilities EDGAR is able to perform gestures with its waist and torso, its head, as well as with its arms and fingers.

Figure 4.3 shows two versions of the EDGAR humanoid avatar. Both humanoids are 180 cm tall, with the footprint dimensions of 70 cm by 70 cm. Each EDGAR humanoid is equipped with two arms, each with 5 degrees of freedom (DOFs). Each finger on EDGAR's two hands is capable of independent movement, with the thumb having 2 DOFs. The first version of EDGAR has a rear projection head module with 3 DOFs while the second version of EDGAR

Figure 4.3: Two versions of EDGAR humanoid robot

has a 3D-printed head module with 2 DOFs. Both humanoids have no mobile platform base and hence will not provide any navigational service elements.

Hardware Components of EDGARs

EDGAR-1 and EDGAR-2 have a total of 28 DOFs and 27 DOFs respectively. All of the actuators on the humanoid are electromechanical motors running on DC power. EDGAR-1 is equipped with a forward-looking camera, while EDGAR-2 is equipped with a forward-looking Kinect camera. Both cameras come with in-built microphone, that can be used for video conferencing or social interaction.

The novelty of EDGAR-1 is that the head of this telepresence humanoid avatar is a rear-projection unit, which comprises an anthropomorphic head screen, a pan-tilt-roll unit and a projection unit, as shown in Figure 4.4(b). The remote user can use a normal webcam for live-streaming interaction as the face image can be processed to fit the contour of the face. This means that a person's facial features and expressions can be displayed on the robot, making EDGAR-1 suited for portraying any person.

Our very first prototype of the rear-projection head unit has been described in [170]. Similar method has been deployed to construct the head module for EDGAR.

Construction of the Rear Projection Head Unit

The steps to construct the head screen is as depicted in Figure 4.4(a). The head screen is developed using the vacuum forming technique on translucent APET (Amorphous Polyethylene Terephthalate) sheets. Therefore, the first step to make the head module is to acquire a mould of a head for vacuum forming. In the attempt to customize the face of the EDGAR robot to look like a particular inhabitant, the mould of the head has been created using a Kinect sensor and a 3D printer.

The Kinect sensor is used to scan a male inhabitant's head. KinectFusion algorithm [171] has been applied to reconstruct the 3D scanned data to form a digital 3D head model. After acquiring the 3D scanned head model from the Kinect sensor, post-processing techniques are applied manually to improve the head model for 3D printing. Subsequently, the life-size head mould of the inhabitant is reproduced cost-effectively using a 3D printer.

After the vacuum forming process with the face mould, a coat of matte paint is applied to the inner surface of the head. This is to diffuse the light from the projector. As the APET sheets are thin, a frame in the shape of the head has been cut out of sheet plastic of approximately 2 cm in thickness to provide rigidity. Both the head screen and the projector unit have been mounted onto this frame. Finally, the frame is attached to the pan-tilt unit, which allows the head module to perform gestures such as nodding, as well as shaking left and right.

Figure 4.4: Construction of the Rear Projection Head Unit. (a) Steps to construct EDGAR's rear-projector head module (b) The resultant rear-projector head module

4.2 The Controllers

In this section, the controllers that have been implemented for the MAVEN and EDGAR robotic avatars are described. This includes the motion controller (Section 4.2.1), joint (Section 4.2.2) controller, as well as the avatar controller (Section 4.2.3).

4.2.1 Mobile Base Controller

The mobile base controller determines how the robotic avatar should move from one point to another, in response to the sensors concerning the state of the robot's environment.

Figure 4.5: Robot reference system and the basic motions for navigation

Figure 4.5 depicts an overview of the robot reference system as well as the basic robot motions that the navigation service elements would support. A right hand coordinate system has been defined for the robot reference system, such that the robot faces front in the direction of the positive x-axis and the left direction of the robot corresponds to that in the positive y-axis.

As seen in Figure 4.5, six basic motions have been designed for the mobile robotic avatar to move around in cluttered human environments. These motions have been chosen to enable the robotic avatar to move like a human does — being able to move forwards, sideways, diagonally and turn on the spot as well as to move along an arc. These six motions are only feasible using holonomic or omni-directional mobile platforms, since the robot needs to produce the sideways drive motion.

To produce the six basic motions in Figure 4.5, three fundamental navigation actions are defined — translation along the x-axis, rotation about the z-axis as well as strafing along the y-axis. These navigation actions can be represented as command velocities and in this thesis, they are named as translation velocity (t_v), rotation velocity (r_v) and strafe velocity (s_v). The basic motions are then the combinations of these three velocities. For instance, to move along an arc (Figure 4.5(c)), the robot needs to move forward ($+t_v$) and rotate clockwise ($-r_v$) at the same time, while s_v is set to zero.

4.2.2 Humanoid Motion Controller

The humanoid motion controller determines how the joint rotations take place, and currently in this work, this controller is used by the gestural service elements for gesticulations only. The implementation of the humanoid motion controller includes defining a local

coordinate system for the robot and a sign convention for joint rotations. Subsequently, human anatomy is studied and a number of basic human motions are identified. As this is for gesticulation, only upper body human motions are required.

Robot Coordinate System and Sign Convention

The anatomical planes of the body are as shown in Figure 4.6(a). The coronal plane is the vertical plane that divides the body into front and back sections. It is made up of the left-right axis and craniocaudal axis. The shoulder vector, which is the vector from the left shoulder to the right shoulder, is parallel to the left-right axis. Similarly, the trunk vector, which is the vector from the spine-shoulder to the spine-base, can be used to represent the craniocaudal axis.

Figure 4.6: Definition of the Robot Coordinate System and Sign Convention. (a) Anatomical Planes of the human body (b) Local Coordinate System of Edgar (c) Local Coordinate System of Pepper. Image taken from [172]

The sagittal plane is a longitudinal plane that divides the body into left and right sections. It is made up of the anteroposterior axis and the craniocaudal axis. A forward vector, which is the cross product of the shoulder vector and trunk vector, is parallel to the anteroposterior axis and can be used to represent it during calculation.

A convention based on a Cartesian coordinate system is adopted in this part of the report to describe the relationship between actual human movements and those made by the robot. A right hand coordinate system has been used for the robot reference system, as shown in Figure 4.6(a) and Figure 4.6(b). The X-axis of this system is the horizontal axis extending out from the robot's front. The Y-axis extends out laterally to the side, with the left side of the robot being the positive half of the Y-axis. The Z-axis extends vertically from the robot.

To perform the rotation of the body parts, a frame, as shown in Figure 4.6, is placed at

each joint. When the robot is at the zero pose, all joint frames have the same orientation. Then, roll rotations take place around the X axis, pitch rotations around the Y axis and yaw rotations around the Z axis.

Basic Joint Motions for Gesticulations

Figure 4.7 illustrates a number of basic human joint motions that the gesticulation service elements would provide in ALSTER. This includes head motions, such as head flexion, extension and rotation, forearm flexion, extension, pronation and supination as well as upper arm flexion, extension, abduction, adduction, lateral rotation and median rotation.

Table 4.1 lists the relevant head and upper limb motions, as well as their corresponding robot joint motions with respect to the robot local coordinate system.

These basic joint motions are created for the EDGAR humanoid, which has a pair of 5 DOFs arms, 3 DOFs waist and a 3 DOFs neck mechanism. However, it is adaptable to any robot, for instance a Pepper robot [172], as shown in Figure 4.6. Developers, who want to use the gesticulation service elements in the ALSTER framework, have to determine the correspondences between the robot motions listed in Table 4.1 to the joint motions available on their anthropomorphic robots.

Figure 4.7: Illustration of the various human motions that are included in the framework

Body Part	Human Motions	Corresponding Robot Motions	Angle Label
Upper Arm	(i) Flexion (ii) Extension	Upper arm swings up or down about y-axis at shoulder joint	shoulder_pitch
Upper Arm	(iii) Abduction (iv) Adduction	Upper arm rotates about x-axis at shoulder joint	shoulder_roll
Upper Arm	(v) Lateral Rotation (vi) Median Rotation	Upper arm rotates about z-axis at shoulder joint	shoulder_yaw
Lower Arm	(vii) Forearm Flexion (viii) Forearm Extension	Lower arm swings up or down about y-axis at elbow joint	elbow_pitch
Lower Arm	(ix) Forearm Supination (x) Forearm Pronation	Lower arm rotate about x-axis at elbow joint	elbow_roll
Head	(xi) Extension (xii) Flexion	Head rotates up or down about the y-axis at the neck joint	head_pitch
Head	(xiii) Rotation	Head rotates about the z-axis at neck joint	head_yaw
Head	(xiv) Lateral Flexion	Head tilts left or right about the x-axis at the neck joint	head_tilt
Waist	(xv) Trunk Rotation	Waist rotates about the z-axis at trunk joint	waist_yaw
Waist	(xvi) Trunk Extension (xvii) Trunk Flexion	Waist bends up or down about the y-axis at the trunk joint	waist_pitch
Waist	(xviii) Lateral Trunk Flexion	Waist side bending to the left or right about the x-axis at the trunk joint	waist_tilt

Table 4.1: List of human head and upper limb motions and their corresponding robot motions

4.2.3 Avatar Controller

The avatar controller is a collection of tools and libraries provided by the Virtual Human Toolkit [173]. It has been integrated in the ALSTER framework to display a controllable virtual human conversational character for autonomous social interaction on the MAVEN robots.

The Virtual Human Toolkit (VHT) provides modules to enable the virtual character for natural language interaction and nonverbal behaviors. The avatar controller in the ALSTER framework uses the VHT messaging protocol to control the speech of the virtual character as well as its gesticulations.

4.3 Sensing Subsystem

The ALSTER framework allows for flexible configuration of the robotic avatar's sensor suite. However, during the implementation of the framework, the sensors are limited to laser rangefinder, as well as RGB and RGB-D cameras to provide for ground-truth depth

and semantic perception. These are as depicted in Figure 4.1b. The number of cameras and rangefinders, as well as their 3D location and orientation with respect to the robot's coordinate system and be specified in the robot model, as described in Section 4.8.1.

Additionally, a range of software drivers has been implemented in ALSTER to read the proprioceptive sensors and provide measurements associated with the robot's internal state and its surrounding. The proprioceptive sensors are also illustrated in Figure 4.1b. Measurements of the robot's internal states include battery health, robot's location and orientation as well as speed and acceleration vectors.

4.4 The Perception Modules

In the ALSTER framework, the perception modules are software packages that enable the robot to extract information from the events, people and objects of its environment, and infer relations between them. For the robot to interact with humans, and to function safely in the human environment, a number of perception modules have been implemented in ALSTER. This includes a human detection and tracking module (Section 4.4.1), a sound localization module (Section 4.4.2), a QR code detection and decoding module (Section 4.4.3), as well as the localization (Section 4.4.4) and mapping (Section 4.4.5) modules.

4.4.1 Human Detection and Tracking

The human detection and tracking module provides the information for the robot to keep track of the people around it. It mainly requires visual data from a RGB-D camera and a 2D range sensor (such as laser or sonar) if available.

In the ALSTER framework, the Kinect RGB-D camera is used to provide the visual information for human detection and tracking. The specification of the sensor can be found in Appendix A. The Kinect sensor has the capability of detecting a maximum of six people in its field of view. There are currently two versions of the Kinect sensor. Kinect v1 is able to provide detailed skeletal information (such as joints' position and orientation) for up to two detected people while Kinect v2 is able to provide the skeletal information for all six detected people.

The robot has a people-list stored in its memory module, such that all new people detected in the current frame will be associated with previously detected people, if possible. This would help the robot to keep track of the people around it, even when the person is temporarily out of view. A previously known person will only be removed from people-list when he or she has not been seen for a set amount of time.

Person's Attribute	Description
KinectID	The ID that Kinect assigned to the detected person
HeadingAngle	Angle (in degrees) at which the person is facing
Distance	Measured depth distance (in meters) from the robot
PositionInRobotFrame	3D coordinates of the person's centroid in robot frame
PositionInImageFrame	2D pixel coordinates of the person's centroid on image
ShirtColorHistogram	The HSV histogram of a region around the person's centroid
IsFaceDetected	An indication whether the person's face is detected or not
IsVisible	An indication whether the person is visible or not
PresentSince	The time when the person is first detected
NotSeenSince	The time when the person is first not seen

Table 4.2: List of attributes that the robot keeps track of a person in people-list

An algorithm, which is based on the calculated Euclidean distance between the positions and a color histogram matching technique, has been included to associate each detected person with the known population in the people-list. It is as described in Algorithm 4.1. The newly detected person's current position and HSV color histogram will be compared against the stored position and histogram data. With this comparison, the person with the closest matching position and histogram data will be associated as the previously tracked person, and the module will automatically update the person's attributes. Table 4.2 illustrates the list of attributes that the robot would keep track of a person. If the newly detected person is unable to associate with any previously tracked person in the people-list, he or she will be added to the population.

The position matching measures the Euclidean distance between positions (d_{pose}) as shown in Equation 4.1, where C_p is the centroid position of the previously tracked person and C_d is the centroid position of the currently detected person. Either the `PositionInRobotFrame` or `PositionInImageFrame` can be used for position matching. If the 3D coordinates of the person's centroid in the robot frame (`PositionInRobotFrame`) is used for the calculation, then $n = 3$, otherwise $n = 2$.

$$d_{pose} = \sqrt{\sum_{i=1}^n (C_{p,i} - C_{d,i})^2} \quad (4.1)$$

The histogram matching technique compares the histogram of the previously detected person, H_p and the histogram of each detected person, H_d with four types of histogram comparison methods [174] : Correlation ($d_C(H_p, H_d)$), Chi-square ($d_S(H_p, H_d)$), Intersection ($d_I(H_p, H_d)$) and Bhattacharyya ($d_B(H_p, H_d)$). The final histogram matching distance result is an arithmetic combination of all four histogram distances obtained from these

methods as shown in Equation 4.2.

$$d_{hist} = (1 - d_C(H_p, H_d)) + d_S(H_p, H_d) + (1 - d_I(H_p, H_d)) + d_B(H_p, H_d) \quad (4.2)$$

Subsequently, a score S will be computed for each detected person based on their position and histogram matching result, as shown in Equation 4.3. The previously tracked person in the people-list with the lowest score is most likely to be the currently detected person. The rest of the person's attributes, such as `HeadingAngle`, `Distance` and etc, are then calculated based on the current frame and people-list would be updated accordingly.

$$S = \omega d_{pose}(C_p, C_d) + \frac{(1 - \omega) d_{hist}(H_p, H_d)}{(\max(\forall d : d_{hist}(H_p, H_d))} \quad (4.3)$$

However, if the computed minimum S is more than a certain threshold, it implies that the currently detected person is not a close match to previously tracked person, despite having the lowest score. In this case, the newly detected person is added to the people-list as a new person instead.

Algorithm 4.1 Algorithm to Update the People-List

Input: Kinect RGB-D frame, I , and the people-list, V_p

Output: Updated people-list V_p

```

1: procedure UPDATEPEOPLELIST( $I, V_p$ )
2:   Extract centroid positions and histograms of all detected people in  $I$ :
3:    $V_c \leftarrow \text{DetectedPeopleCentroids}(I)$ 
4:    $V_h \leftarrow \text{CalculateHistogramAroundEachCentroid}(I, V_c)$ 
5:   Score each detected person:
6:   for each  $P$  in  $V_p$  do
7:      $C_p \leftarrow P[\text{PositionInRobotFrame}]$ 
8:      $H_p \leftarrow P[\text{ShirtColorHistogram}]$ 
9:     for each  $C_d$  in  $V_c$  do
10:      for each  $H_d$  in  $V_h$  do
11:         $d_{pose} \leftarrow \text{PositionMatching}(C_p, C_d)$ 
12:         $d_{hist} \leftarrow \text{HistogramMatching}(H_p, H_d)$ 
13:         $S_{(p,d)} \leftarrow \text{ScoreFunction}(d_{pose}, d_{hist})$ 
14:   Find associate of each detected person:
15:   for each  $d$  in  $S_{(p,d)}$  do
16:      $\min\_score, p \leftarrow \min(\forall s : S_{(p,d)})$ 
17:     if  $\min\_score < \text{threshold}$  then
18:       Calculate the rest of the attributes
19:       Update  $V_p[p]$ 
20:     else
21:       Calculate the rest of the attributes
22:       Add new entity in  $V_p$ 
23:   return  $V_p$ 

```

4.4.2 Sound Localization

The ability to localize the direction of a sound is essential for a social robot to interact with its environment. It helps to direct the robot's attention to the events happening around it autonomously. It is especially useful for enabling the robot to face and give attention to the person who is speaking in front of it.

On the other hand, during a telepresence session, the inhabitant's view of the environment can be limited by the perspective of the camera ("soda straw effect"), and sound localization information may help to increase the inhabitant's situational awareness of the robot's environment.

In the ALSTER framework, a Kinect sensor has been utilized to identify the direction of any loud enough sound or human speech. The sound wave emitted by a source is received at slightly different times on each of the Kinect sensor's four microphones, from the closest to the farthest. By computing the inter-aural phase differences between each of the four audio streams, the module is able to retrieve the direction of the emitting source.

4.4.3 QR Code Detection and Decoding

Over the past few years, there has been a significant development in the adoption of using QR (Quick Response) coded images for many industrial, commercial and robotic applications [175]. For instance, QR codes have been used for guest registration in various events, patient identification in hospitals, as well as monetary transactions via a mobile phone. For robotic applications, QR coded images have been used for robot localization and navigation [176], target (such as charging station, stairs, items and etc) localization [177], augmented reality [178] and many more.

Currently, three algorithms have been integrated in the ALSTER framework and they are running concurrently to detect and decode three types of QR coded images. They are the ArUco marker, the ARToolkit marker as well as the standard QR code (ISO/IEC18004), as shown in Figure 4.8. In the future, this module can be extended to include other types of 2D coded images.

ArUco marker. A small 2D squared marker, which corresponds to a number encoded into a small grid of black and white pixels. It can be used as tags, for instance, to identify the location of the charging station or an item that the robot should manipulate. The ArUco detection and decoding algorithm, used in ALSTER, is one of the OpenCV_contrib modules [179], which is based on the ArUco library [180]. The algorithm is able to detect multiple ArUco markers from a camera view, decode each marker's number, and estimate the 3D position and orientation of each marker with respect to the camera.

Figure 4.8: Types of QR coded images that ALSTER framework currently detects — (a) ArUco marker (b) ARToolkit marker (c) Standard QR code

ARToolkit marker. A patterned QR code that is used in the ARToolkit [181], typically for augmented reality and robotic applications. Similar to the ArUco marker, it can be used as tags to identify the 3D pose of a target. The ARToolkit marker detection algorithm used here is one of the ROS modules, `ar_pose` [182], which is a wrapper of the ARToolkit. It is able to detect and track multiple ARToolkit markers relative to the location of a camera.

Standard QR code. The commonly used matrix barcode that contains embedded information, such as an URL or a string of text. It functions like a barcode where coding information can be read and decoded by a machine. It can be used by a social robot to facilitate smart check-in, information dissemination and even mobile payments. Currently, the QR codes and other barcodes are detected and decoded in ALSTER, using the ZBar library [183]. It is able to detect different types of standard QR codes and barcodes (such as EAN13, EAN-8, ISBN10, etc.) and decode the embedded content.

4.4.4 Localization

Localization is fundamental to mobile robotics. The position and orientation of the mobile robotic avatar will be estimated in the localization module. An accurate estimate of the robot pose is necessary to perform navigational tasks, such as waypoint navigation. In this work, three localization methods — relative positioning, map-based and human assistance — have been integrated in the ALSTER framework, as shown in Figure 4.9.

Relative Positioning Localization

Relative positioning, which is sometimes known as dead-reckoning, is a positioning technique that estimates a relative position from an initial starting position. Typically, it fuses pose information from two or more inertial sensors, which measure the robot's orientations, velocities or displacements from the initial pose.

In the ALSTER framework, an extended Kalman filter (EKF) [184] has been used to estimate the position of the telepresence robotic avatar, by fusing the odometry and IMU signals.

Figure 4.9: The Localization Module

The odometry signals, such as the 2D pose (xy-position and the heading of the robot) as well as the pose covariance, are derived from the wheel encoders. Relative displacements and velocities are then computed to update the EKF. The IMU provides the roll, pitch and yaw angles of the robot frame with respect to the world frame. The roll and pitch angles acquired from the IMU are regarded as absolute readings and the yaw angle is used as a relative angle. Only relative pose differences of each sensor is used to update the EKF.

Map-based Localization

Although EKF relative positioning is useful and accurate in keeping track of the robot's position, it has a number of limitations [185][186]. First, it is dependent on the sensor models that generate estimates with Gaussian uncertainty, hence it is prone to sensor modeling errors. For instance, any discrepancy between the actual displacement traveled by the robot and the estimated displacement traveled per rotation derived from the odometry model (due to maybe unequal wheel diameters or slippage) will be a source of error. It is also unable to represent the robot's position robustly for a long period of time because the errors of relative positioning are cumulative. To overcome these limitations, a map-based localizer can be used to correct small, incremental pose errors produced by the relative positioning localization module.

A probabilistic localization algorithm, known as Adaptive Monte Carlo Localization (AMCL) [185], has been adopted as the map-based localizer of the ALSTER framework. It uses an adaptive particle filter (or KLD-sampling) [187] to estimate the robot's position against a known map, as the robot moves and senses the environment with a rangefinder sensor.

Human Assistance Localization

Lastly, both the relative positioning and map-based localization approaches would require the starting position of the robot to be known. The human user has to provide the initial position of the robot to the localizers, through the human assistance localization approach. The human-assisted localization is also useful in scenarios when the map-based localizer estimates the position of the robot wrongly, causing the “robot to be lost”.

The approach to this is to teleoperate the robot to a known position and initialize (or reinitialize) the relative positioning and map-based localizers accordingly. The human assistance localization approach reduces the complexity of the localization problem by using the human capacities in perception and decision-making to diagnose a failure [188].

4.4.5 Map Server

In order to perform map-based navigation, a 2D grid map is required for localization and path planning. Each grid cell of the map will contain a value that indicates the likelihood of occupancy. For instance, Figure 4.10(b) illustrates the map of the front area in the Robotics Research Centre (RRC). The occupied and unoccupied spaces are represented in black and white color respectively, while gray color indicates the unexplored spaces. The grid map is stored as an image of PGM file format.

The map server module would load the map image as well as its corresponding map description file, and publish the map to other modules that require it. The map description file contains information such as the folder path to the image file, the resolution of the grid map in meters/pixel, the world coordinates of the origin, which is represented as the lower-left pixel in the map, as well as the upper and lower bound thresholds for occupied and unoccupied probability respectively.

A grid map can be produced either by manually drawing a to-scale map out using an image editor or by driving the robot around and applying the GMapping algorithm [189] to its sensor data. The map in Figure 4.10(a) is produced using the second method, which uses the robot’s sensor data, hence it can be quite “messy”. It needs to be manually edited to correct errors and improve on the overall appearance. Changes made to the map would influence path planning. For example, drawing some black areas or lines would prevent the robot from planning paths to and through these areas, while white areas indicate unoccupied spaces that are safe to navigate. Figure 4.10(b) shows the final version of the map that is used in the ALSTER framework after editing. Besides removing things that shouldn’t be there, existing spaces and furniture, which are not mapped by the robot, are also being added into the map, using an image editor.

Figure 4.10: Example of a 2D binary occupancy SLAM map. The map has been digitally edited to include existing spaces and furnitures that the robot didn't map. The reference environment is the front area of the Robotics Research Centre.

4.5 The Navigation Service Elements

The navigation service elements are modules that get the robot avatar to move around its environment. One way to move the robot around is through manual teleoperation control, as described in Section 4.5.1. This is of the lowest autonomy as it requires a continuous stream of data for sensors and controls from the inhabitator.

The navigation service elements that are of a higher autonomy get the robot to move around autonomously to perform a task specifically. In ALSTER, waypoint navigation (Section 4.5.2) has been implemented to allow the robot to plan and move from one point to another autonomously. Furthermore, an auto-docking routine has been included to enable the robot to move and dock at the docking station when its battery level runs low (Section 4.5.3). Lastly, a person following module (Section 4.5.4) has been implemented to generate navigational behaviors for the robot to follow a person, to walk side-by-side with a person or to guide a person.

4.5.1 Teleoperation

The teleoperation service element offers two software modules in the ALSTER framework for the manual navigational control of the robotic avatar. They are the joystick-based control node using a gamepad, gaming pedals or joystick, and a keyboard-mouse control node. Each control node reads inputs from the device and provides the mapping configuration to map the inputs to the velocity commands of the robot. The difference is that the keyboard-mouse control node has no dead-man switch.

As mentioned in Section 4.2.1, there are six basic motions in ALSTER — translate forward or backward, strafe left or right and rotate clockwise or anti-clockwise. Therefore, there are three velocity commands — the translation velocity (t_v) to move the robot along the x-axis, the rotation velocity (r_v) to turn the robot about the z-axis and the strafe velocity (s_v) to

strafe the robot sideways along the y-axis.

For instance, a Logitech gamepad joystick has been configured in ALSTER with the following mapping as shown in Figure 4.11. In this case, the $\pm sv$ has been mapped to the left analog joystick, while the $\pm tv$ and $\pm rv$ are mapped to the right analog stick. The L1 button has been configured to be the dead-man switch.

Figure 4.11: Gamepad joystick commands mapping for teleoperation

To drive the robotic avatar around with this gamepad configuration, one would need to hold down L1 and use the two joysticks to control its direction of movement. This configuration can be modified to suit personal preferences. The mapping of the three velocity commands can be configured accordingly when different devices, such as the game pedals or keyboard, are being used.

4.5.2 Waypoint Navigation

The waypoint navigation service element is basically a path planning module, which plans and generates a path for the robot to navigate from a start position to a goal destination. The aim is not just to determine a path but an optimal path with the shortest distance, the smoothest maneuvers and in the safest manner.

To do so, the path planning module would first perform a global navigation planning, where a prior knowledge (such as an user-defined map) of the environment is used to generate the shortest feasible global path. The human user will define a goal position in the grid map, and the global planner will generate a safe and efficient route from the robot's current position to that goal position. The output of the global planner is a sequence of waypoints, $P_i = (x_i, y_i, \theta_i)^T, \in \mathbf{R}^2 \times S^1$, where S is the navigation plane where the robotic avatar is able to move.

Dijkstra algorithm has been used to generate the optimal path. For each empty cell, the Dijkstra algorithm would assign a cost, such that the cell around the starting point is of cost unit one and for the second generation neighbors of cost unit two and so on. The minimum value of the sum of cell costs from the starting point to the goal is then the shortest path.

The most optimal global path, P_i , is subsequently passed to a local obstacle avoidance module, or a local planner, as described in Section 4.5.2. The local planner would consider the local sensors' readings as well as the global path to determine an intermediate local path, which enables the robotic avatar to avoid any dynamic obstacles.

Obstacle Avoidance

The obstacle avoidance module, or the local planner, would try to drive the robot along the global planned path, by also considering the local observations in real time. First, a local map of the robot's surroundings will be built using the information published from the range sensors to detect any dynamic obstacles, which appear in front of the robot but not indicated on the map. The local map will be used to recalculate the global path in response to a regional change in the environment, such as dynamic or new obstacles. The map is constantly being updated as the robot moves around. Subsequently, with the updated local map and the generated global path, the local planner performs obstacle avoidance for dynamic obstacles while attempting to match the global trajectory as close as possible.

The Time Elastic Bands (TEB) algorithm [190][191] has been adopted for local planning. It takes in the robot's velocity and acceleration limits as well as its kinematic and dynamic constraints. This information is then used, together with the safe distances from obstacles, to generate the intermediate waypoints. The intermediate waypoints are represented as a set of commands, which comprises speed and steering angle, for the robot to move along the path.

4.5.3 Auto-Docking

An auto-docking service element allows an autonomous robot to monitor its battery level and perform the auto-docking action for charging when its battery level is low. When the robot senses that it requires charging, it will perform waypoint navigation to the docking station and align itself to the dock for charging. As a QR code has been used to mark the position of the dock and is visible from a distance of 3 meters, the robot needs to navigate to a location that is within the visibility of the docking station first.

The docking sequence is as depicted in Algorithm 4.2. The robot would travel to a position (P_d) near the docking station, take a picture and check whether a QR code is within the field of view, using the QR code detector as described in Section 4.4.3. If the QR code is not found, the robot turns 30° clockwise and tries again. Once the QR code is found, the robot will align itself by turning and strafing sideways (adjusting rv and sv) little by little until the QR code is in the center of the image. When the robot has aligned itself to the QR code, it approaches the docking station slowly until it is docked to the charging element.

Algorithm 4.2 Algorithm to perform Auto-Docking**Input:** Kinect RGB-D frame, I , and a waypoint position near the docking station, P_d

```

1: procedure AUTO-DOCKING( $I, P_d$ )
2:   Waypoint Navigation to  $P_d$ 
3:    $bDocked \leftarrow False$ 
4:   while not  $bDocked$  do
5:      $I \leftarrow \text{TakePicture}()$ 
6:      $Q \leftarrow \text{ScanForQRCode}(I)$ 
7:     if  $Q$  is found then
8:        $QPose \leftarrow \text{GetQRPose}(Q)$ 
9:        $bAligned \leftarrow \text{AlignRobot}(QPose)$ 
10:      if  $bAligned$  then
11:         $bDocked \leftarrow \text{ApproachDock}(QPose)$ 
12:      else
13:        Turn( $30^\circ$ )

```

4.5.4 Person Following

The person following service element generates higher-level, human-like navigational behaviors, such as *following*, *accompanying* and *guiding* a person autonomously, as shown in Figure 4.12.

In the following behavior, the robotic avatar moves behind the person whom it is following. For instance, a human inhabitant, who is using the telepresence robot to visit a museum, can choose to use the robot to autonomously follow a human guide. This behavior can also be used to enable the robotic avatar to follow an interactant and provide assistance as shown in Figure 4.12(a).

In the accompaniment behavior, as seen in Figure 4.12(b), the telepresence robot will move at the side of the interactant whom it is tracking. This is typical of a scenario where two people are having a conversation while walking side-by-side.

In the guiding behavior, as illustrated in Figure 4.12(c), the robotic avatar will be moving in

Figure 4.12: Different high-level navigational behaviors, including (a) following, (b) accompanying and (c) guiding a person autonomously

front of its interactant, so as to guide the interactant around. This behavior is useful when the inhabitant is more familiar with the robot's environment than the interactant.

During these three autonomous behaviors, the robot would try to maintain an appropriate standoff distance from the interactant. The obstacle avoidance capability has been implemented within the module to ensure safe navigation in the human environment. Furthermore, as these behaviors are implemented for a telepresence robot, it is essential for the robot to move at a speed that is similar to that of a real human. Three main components: human detection and tracking (see Section 4.4.1), velocity profiling (see Section 4.5.4) and obstacle avoidance (see Section 4.5.2) are required to achieve these social navigational behaviors.

The person-following module has been implemented earlier as a standalone module, and has been described in the paper [192]. The results are documented in [193]. In this thesis, the person following module has been modified to integrate into the ALSTER framework.

An PID controller, which is a control loop feedback mechanism, has been used to calculate the translation velocity (tv), rotation velocity (rv), as well as the strafe velocity (sv) for the robot avatar to move in order to follow a person. The output of the PID controller, which is the linear and angular speeds command (tv, rv, sv), is then passed to the obstacle avoidance module, as described in Section 4.5.2. This would ensure that the velocity command is safe to execute for navigation.

Velocity Profiling for Person Following

Each behavior within the multimodal person following module is a proportional feedback control loop. It measures the error between the current robot position and the desired robot position with respect to the interactant. The error is then used to calculate the velocity commands of the robot. The control loop is depicted in the control diagram in Figure 4.13.

Figure 4.13: Feedback control loop for robot speed calculation

The controller block represents the calculation of the robot's forward, lateral and rotational velocities. The robot will try to move in a manner such that it maintains a desirable distance from the interactant. It will also adopt an orientation such that the interactant is always at the center of the Kinect's field of view. The desired distance and direction angle of the interactant relative to the robot is denoted by the upper and lower limits in Figure 4.14(a) and Figure 4.14(b) respectively, which will be used as the reference for velocity calculation.

Figure 4.14: Velocity Profiling for Person Following (a) Linear Velocity and (b) Angular Velocity

The linear velocity is dependent on the distance between the robotic avatar and interactant, as shown in Figure 4.14(a). When this distance is smaller than β , the robot will try to move backward in order to increase the distance between the robot and interactant. On the other hand, when the distance is larger than λ , the robot will increase its velocity to catch up with the interactant.

Similarly, the angular velocity is dependent on the angular difference between the robotic avatar and interactant. As shown in Figure 4.14(b), if the angular difference is near zero, then the robot will not rotate. If the interactant turns to face another direction, the robot will rotate at a speed which is dependent on the angular difference at that point.

The velocity profiling for both the linear motion and angular motion is similar and is represented by a piecewise linear function, as shown in Equation 4.4. The variable x denotes the distance between the robot and human when the function is used for linear velocity profiling. On the other hand, the variable x represents the angular difference between the robot and human when it is used for angular velocity profiling. The rest of the parameters can be tuned differently to suit the different following behaviors and is tabulated in Table.

$$V = V_{\max} f(X) \text{ where } f(X) = \begin{cases} -1 & \text{if } X < \alpha \\ \frac{\beta-x}{\alpha-\beta} & \text{if } \alpha \leq X < \beta \\ 0 & \text{if } \beta \leq X \leq \gamma \\ \frac{x-\gamma}{\delta-\gamma} & \text{if } \gamma < X \leq \delta \\ 1 & \text{if } X > \delta \end{cases} \quad (4.4)$$

The linear velocity is dependent on the distance between the robotic avatar and interactant, as shown in Figure 4.14(a). When this distance is smaller than β , the robot will try to move

backward in order to increase the distance between the robot and interactant. On the other hand, when the distance is larger than λ , the robot will increase its velocity to catch up with the interactant.

Similarly, the angular velocity is dependent on the angular difference between the robotic avatar and interactant. As shown in Figure 4.14(b), if the angular difference is near zero, then the robot will not rotate. If the interactant turns to face another direction, the robot will rotate at a speed which is dependent on the angular difference at that point.

The velocity profiling for both the linear motion and angular motion is similar and is represented by a piecewise linear function, as shown in Equation 4.4. The variable x denotes the distance between the robot and human when the function is used for linear velocity profiling. On the other hand, the variable x represents the angular difference between the robot and human when it is used for angular velocity profiling. The rest of the parameters can be tuned differently to suit the different following behaviors and is tabulated in Table 4.3.

	Back-Following		Side-by-side Accompaniment			Front Guiding
V	TV	RV	TV	RV	SV	TV
X	d_b	θ_b	d_1	θ_s	d_2	d_g
$V_{\max}$	0.6	0.9	0.6	0.2	0.6	0.6
α	0.5	-0.45	-0.55	-0.81	1.2	1.2
β	1.2	-0.09	-0.15	-0.52	1.6	1.5
λ	1.5	0.09	0.15	0.52	1.8	12
δ	2.2	0.45	0.55	0.81	2.2	2.0

Table 4.3: Parameters for Velocity Profiling for Person Following

Result of the Person Following Module

The multimodal person following module has been implemented and tested in both simulation and physical setup. Figure 4.15 and 4.16 illustrates the result of the person-following module, executed in a simulator. In the simulator, a person is programmed to walk along a 6-meter square shape twice, at a speed of 1.2 meters per second. The trajectory of the person is depicted as the blue square.

The robot follows the human using the back-following mode in Figure 4.15(a) and side-by-side accompaniment mode in Figure 4.16(a). The trajectory of the robot is as depicted in red. It has been illustrated in Figure 4.15(a) that the robot is not following the actual path that the human is taking. Instead, the robot adjusted its direction continuously to ensure that the followed person is always centered in the sensor's field of view. This would reduce the distance the distance traveled by the robot during the process of person-following.

Figure 4.15: Result of the Back-Following Mode. (a) Trajectories of the robot (in red) and human (in blue) in back-following mode and (b) Distance between the robot and human during back-following

Figure 4.16: Result of the Side-by-Side Accompaniment Mode. (a) Trajectories of the robot (in red) and human (in blue) in side-by-side accompaniment mode and (b) Distance between the robot and human during side-by-side accompaniment

According to Hall's Proxemics Theory [34], the interpersonal distance between people in social space is between 1.2 m to 2.1 m. Therefore, the robot has been designed to maintain a predefined distance of 1.5 m from the person. Figure 4.15(b) and Figure 4.16(b) describe the distance between the human and the robot when it is performing the respective human-following behavior. The periodic pulses are reflective of the rapid decrease and subsequent increase of the separation between human and robot as the human turns the corner when walking the square shape. After turning the corner, the separation between the two becomes more steady until the next corner is reached. The blue line, in Figure 4.15(b) and Figure 4.16(b), shows that the mean separation is approximately 1.546 m and 1.509 m during back-following and side-by-side accompaniment respectively. Meanwhile the standard deviation, shown by the blue band, is ± 0.069 m and ± 0.028 m respectively.

Figure 4.17 illustrates the trajectories of the robot and human when the robot follows the human in the (a) back-following mode and (b) side-by-side accompaniment mode, with obstacles along the way. The same test as before has been performed, where the human

Figure 4.17: Trajectories of the robot (in red) and human (in blue) in (a) back-following mode and (b) side-by-side accompaniment mode with obstacles along the way

walks around the square shape. In this test however, there are slight differences from the previous test — that is, the human only walks once around the square and the presence of obstacles along the robot’s path as denoted by the shaded squares in both figures. The aim of this test is to demonstrate the robot’s capability for obstacle avoidance. As the robot follows the human, it will encounter these obstacles as well, and will plan a path around each obstacle that it encounters. The robot attempts to optimize this path around each obstacle by determining if there is less of the obstacle to avoid by turning towards the left or the right.

The implemented person following module was also tested in a physical setup. Figure 4.18 shows a sequence of images that depicts a robot following a person from his back. Static obstacles had been set up along the robot’s path and were denoted by the yellow triangles in the trajectory plot. As seen in Figure 4.18, the robot did not follow the person by moving on the same path taken by the person. Similar to the simulation, the robot adjusted its orientation continuously to ensure that it is always facing the person it is following. This behavior resembles the behavior of a real human when he or she is following another person. This is because a human would not typically attempt to follow the exact path taken by the leader. Additionally, the robot would skid sideways to avoid the obstacle in front of it, while maintaining its orientation that was facing the followed person.

Figure 4.19 shows a sequence of images that depicts a robot following a person using the side-by-side accompaniment mode. As depicted, the robot was able to perform side-by-side accompaniment by moving to stay at the side of the targeted person. However, the motion of the robot was not always stable due to the fluctuation and inaccurate calculation of the followed person’s orientation. The side-by-side accompaniment mode was less robust when

compared with the back-following mode, during the physical testing. This is because of the higher likelihood of losing track of target and the higher propensity for the robot to move in an unsteady manner, when the robot was in the side-by-side accompaniment mode.

Figure 4.18: Result of back-following Mode in physical setup: (top) Sequence of images showing a robot following a person with the back-following mode (bottom) Trajectories of the robot and human in back-following mode

Figure 4.19: Result of side-by-side accompaniment mode in physical setup: (top) Sequence of images showing a robot following a person with the side-by-side accompaniment mode (bottom) Trajectories of the robot and human in side-by-side following mode

4.6 The Gesticulation Service Elements

Gesticulation represents a crucial feature of human communication. Human speakers often use facial expressions as well as hand and arm gestures concurrently with their speech to illustrate what they wish to express during communication. More importantly, gestures help to convey spatial or referential information, which speech alone cannot provide.

Therefore, given the design of a humanoid used for telepresence, it is expected to express human-like communicative behaviors. The body of the humanoid can be used to display non-verbal gesticulations, in a fashion that is similar to a human, during a telecommunication session.

In this section, a number of gesticulation service elements, which are implemented within the ALSTER framework, are described. This includes an approach of displaying the live streaming video texture, acquired from a conferencing session, on the telepresence humanoid. This enables the display of facial expressions on the robot (Section 4.6.1). A method of teleoperating the humanoid robot for gesticulation during a telepresence session is also presented in Section 4.6.2. Furthermore, autonomous gestural service elements, such as scripted gesturing movements (Section 4.6.3)) and reactive gesticulations (Section 4.6.4), are included to imbue the humanoid with the capabilities to perform gestures on its own. This helps to mitigate workload and dissolve any ambiguity typical of unimodal communication.

4.6.1 Tele-Conferencing (Video)

The novelty of EDGAR is that the head of this telepresence humanoid is a rear-projection system, which comprises an anthropomorphic head screen, a pan-tilt-roll unit and a projection unit, as shown in Figure 4.4(b). This enables a face texture to be projected on the robot's face so that the inhabitant's face and facial expressions can be displayed on EDGAR in real-time.

During a video conferencing session, the streaming video texture of the inhabitant's face will be rendered for display from the rear of the head screen. The head screen is not a flat 2D screen, as it is in the shape of a human face. Hence there is a need to warp the texture of any human face 2D image to fit the screen's contours. This process would involve two key stages.

Reference Image Generation

The first stage involves the generation of a reference image. This requires three steps. Firstly, drawings are made on the outer surface of the head screen with a non-permanent

marker, as shown in Figure 4.20(a). These drawings will mark locations on the screen that correspond to the facial features such as hairline, eyebrows, eyes, cheekbones, the nose and the mouth.

The second step involves the implementation of a software that projects a clickable grid on the head screen. As shown in Figure 4.20(a), the clickable grid has been projected on the head screen from the rear.

The third step in generating the reference image is to note down the thirty-four 2D facial feature points, as shown in Figure 4.20(b), on the displayed grid. This is done by clicking on the grid sequentially, and the software will draw circles and record the coordinates of the clicked 2D points to a text file. The aim is to match the 2D points on the grid to correspond with those made previously on the head screen. The resultant grid, which has been clicked accordingly, then becomes the reference image, as shown in Figure 4.20(c).

Figure 4.20: Generation of the Reference Image. (a) Hand-drawn facial features on the head screen (b) 2D facial feature points for face tracking (c) Reference image of the projected facial features

Geometric Image Transformation

The second stage, as depicted in Algorithm 4.3, is to warp the source image to fit the reference image so that the source image can be displayed on the head screen accurately. A source image is the texture image of a person's face. It can be a static image or it can be an image frame extracted from a live video stream.

First, a face is detected from the source image frame. Subsequently, a face tracking algorithm [194] is used to extract the thirty-four facial feature points that corresponded to those that were marked on the reference image.

After acquiring the facial feature points from the source image, a piecewise affine transformation matrix, which is the relation between the source and reference images, is calculated. The source image is then warped, via linear mapping, to produce the resultant image.

To display another person's face, only stage two (the geometric image transformation stage) has to be repeated. Therefore, features on the new source image are then to be

corresponded to those on the reference image and a new resultant image is produced.

Algorithm 4.3 Geometric Image Transformation Algorithm

Input: RGB image, I_s , and a list of facial feature points from the reference image, V_{ref}

Output: Transformed image I_d

```

1: procedure GEOMETRIC IMAGE TRANSFORMATION( $I, P_d$ )
2:    $FaceBoundingBox \leftarrow$  Face Detection( $I_s$ )
3:   if  $FaceBoundingBox$  is found then
4:      $V_{src} \leftarrow$  ExtractFacialFeatures( $FaceBoundingBox$ )
5:     if  $V_{src}$  is found then
6:        $I_d \leftarrow$  PiecewiseAffineWarp( $I_s, V_{ref}, V_{src}$ )
7:   return  $I_d$ 
  
```

4.6.2 Kinect Avateering

A Kinect RGB-D camera sensor has been used to detect body segments of the inhabitant at the control station. It is placed at a fixed height of 1.4 meters and the inhabitant would stand at the distance of 2 meters from it, according to the specifications of the Kinect sensor. Since the inhabitant is standing in front of the camera, only the frontal body surface is detectable during skeletal tracking.

The Kinect sensor is able to capture imagery data and track body segments at a frame rate of 30 FPS continuously. The Microsoft Kinect for Windows SDK is used to extract real-time 3D coordinates of the tracked body's joints. The first version of the Kinect sensor returns a total of 20 skeletal joints data while the second version of the Kinect sensor returns 25 skeletal joints data. The skeletal joint map information of both Kinect v1 and Kinect v2 can be found in Appendix A.2.

Processing Kinect skeletal information to calculate joint angles

To calculate the orientation angles listed in Table 4.1, the Kinect joint information at the left and right shoulder, elbow wrist, thumb joints as well as the spine-shoulder and spine-base joints are extracted. The Microsoft Kinect Face API has been utilized to perform face tracking and to calculate the head yaw and pitch angles.

The flexion/extension angle of the shoulder (shoulder_pitch) is calculated as the angle between the arm vector and the trunk vector, projected on the sagittal plane, as shown in Figure 4.21(a). For instance, the calculation of the pitch angle at right shoulder involves the following equations. First, the normal vector of the sagittal plane, $\mathbf{n}_s$, which is the shoulder vector is determined. In this case, the shoulder vector, $\mathbf{v}_s$ is the vector from the

Figure 4.21: Calculation of the joint angles — (a) shoulder_pitch angle (b) shoulder_roll angle (c) shoulder_yaw angle (d) elbow_pitch angle (e) elbow_roll angle

right shoulder, P_{rs} to the left shoulder, P_{ls} .

$$\mathbf{n}_s = \mathbf{v}_s = P_{ls} - P_{rs} \quad (4.5)$$

Subsequently, the arm vector, $\mathbf{v}_a$ and the trunk vector, $\mathbf{v}_t$ at the right shoulder are calculated. The arm vector is the vector from the right shoulder, P_{rs} to the right elbow, P_{re} , while the trunk vector is the vector from spine-shoulder, P_{ss} to the spine-base, P_{sb} .

$$\begin{aligned} \mathbf{v}_a &= P_{re} - P_{rs} \\ \mathbf{v}_t &= P_{sb} - P_{ss} \end{aligned} \quad (4.6)$$

The arm vector and trunk vector are then projected onto the sagittal plane using Equation 4.7, resulting in $\mathbf{v}_{proj}(v_a, n_s)$ and $\mathbf{v}_{proj}(v_t, n_s)$ respectively.

$$\begin{aligned} \mathbf{v}_{proj}(\mathbf{u}, \mathbf{n}) &= \text{Projection of } (\mathbf{u}) \text{ on plane with normal } (\mathbf{n}) \\ &= \mathbf{u} - \text{proj}_{\mathbf{n}}(\mathbf{u}) \\ &= \mathbf{u} - \frac{\mathbf{u} \cdot \mathbf{n}}{\|\mathbf{n}\|^2} \mathbf{n} \end{aligned} \quad (4.7)$$

The right shoulder pitch angle, θ_{rsp} is then the angle between the projected arm and trunk vector, calculated using Equation 4.8.

$$\theta_{rsp} = \arccos\left(\frac{\mathbf{v}_{proj}(v_a, n_s) \cdot \mathbf{v}_{proj}(v_t, n_s)}{\|\mathbf{v}_{proj}(v_a, n_s)\| \cdot \|\mathbf{v}_{proj}(v_t, n_s)\|}\right) \quad (4.8)$$

For the calculation of the left shoulder pitch, Equation 4.5 and 4.6 are modified to use the left shoulder and left elbow pose reading instead.

The abduction/adduction angle of the shoulder (shoulder_roll) is calculated as the angle between the same arm vector and trunk vector, but projected on the coronal plane, as shown in Figure 4.21(b). The normal vector of the coronal plane, $\mathbf{n}_c$ is the forward vector,

$\mathbf{v}_f$, which is also the cross vector of the shoulder and trunk vectors.

$$\mathbf{n}_c = \mathbf{v}_f = \mathbf{v}_s \times \mathbf{v}_t \quad (4.9)$$

Hence, by using the variables of Equation 4.6 and Equation 4.7, the right shoulder roll angle, θ_{rsr} can be calculated, as shown in Equation 4.10.

$$\theta_{rsr} = \arccos\left(\frac{\mathbf{v}_{proj}(v_a, n_c) \cdot \mathbf{v}_{proj}(v_t, n_c)}{\|\mathbf{v}_{proj}(v_a, n_c)\| \cdot \|\mathbf{v}_{proj}(v_t, n_c)\|}\right) \quad (4.10)$$

As shown in Figure 4.21(c), the median/lateral rotational angle of the shoulder (shoulder_yaw) is calculated as the angle between the forearm vector, $\mathbf{v}_{fa}$ and the forward vector ($\mathbf{v}_f$ in Equation 4.9), projected on the plane where the normal is the arm vector $\mathbf{v}_a$. The right forearm vector is the vector from the right elbow, P_{re} to the right wrist P_{rw} .

$$\mathbf{v}_{fa} = P_{rw} - P_{re} \quad (4.11)$$

As shown in Equation 4.12, the right shoulder yaw angle, θ_{rsy} can be calculated using Equation 4.7, as well as the variables such as $\mathbf{v}_a$ from Equation 4.6, $\mathbf{v}_f$ from Equation 4.9 and $\mathbf{v}_{fa}$ from Equation 4.11.

$$\theta_{rsy} = \arccos\left(\frac{\mathbf{v}_{proj}(v_f, v_a) \cdot \mathbf{v}_{proj}(v_{fa}, v_a)}{\|\mathbf{v}_{proj}(v_f, v_a)\| \cdot \|\mathbf{v}_{proj}(v_{fa}, v_a)\|}\right) \quad (4.12)$$

The flexion/extension angle of the elbow (elbow_pitch) is calculated as the angle between the arm vector $\mathbf{v}_a$ and forearm vector $\mathbf{v}_{fa}$, as shown in Figure 4.21(d). In this case, the right elbow pitch angle θ_{rep} can be represented as

$$\theta_{rep} = \arccos\left(\frac{\mathbf{v}_a \cdot \mathbf{v}_{fa}}{\|\mathbf{v}_a\| \cdot \|\mathbf{v}_{fa}\|}\right) \quad (4.13)$$

Lastly, the pronation/supination angle of the elbow (elbow_roll) is calculated as the angle between the hand vector and the forearm vector, as shown in Figure 4.21(e). The hand vector (in this case, the right hand vector) $\mathbf{v}_h$ is the vector from the right thumb tip P_{rt} to the right arm wrist P_{rw} .

$$\mathbf{v}_h = P_{rt} - P_{rw} \quad (4.14)$$

Hence, the right elbow roll angle θ_{rer} can be calculated using the following equation.

$$\theta_{rer} = \arccos\left(\frac{\mathbf{v}_t \cdot \mathbf{v}_{fa}}{\|\mathbf{v}_t\| \cdot \|\mathbf{v}_{fa}\|}\right) \quad (4.15)$$

A total of 12 joint angles are calculated — 2 head orientation angles and 5 arm orientation angles per arm. The calculated angles are then averaged over ten consecutive frames

to smooth out any sharp fluctuations. The resultant angles are encapsulated into a data structure, which is subsequently being transferred to the robot via a server-client network.

Motion Remapping Module

A motion remapping module is implemented within the ALSTER framework to receive the packet of calculated angles through the network, process the angles to fit the limits of the robotic arms and perform the rotation at each joint accordingly.

The skeletal structure used by the Kinect tracking API is completely different from the internal structure of the robots used, and that different robotic avatar will have different actuator type and dimensions in terms of anatomy and humanoid design. The motion remapping module or library has been developed to map the Kinect skeleton information to many different types of humanoid skeletons, for both virtual avatars and physical humanoid avatars. It would provide the specification of the coordinate system as well as the rotational offsets and limits, if required. Besides facilitating the mapping of human movement to the virtual or robotic avatar in real-time, the motion remapping module also allows control of more than one avatar simultaneously.

The motion remapping module utilizes a server-client network to receive calculated Kinect joint angles. As various robots are built with different types of motorized actuators, motions may be represented differently for each robot and there will be different movement constraints of the robotic limbs. The DOFs associated with a virtual or robotic avatar's limb are going to dictate the degree of accuracy and control the user has while controlling it. The more closely a virtual or robotic avatar's limbs have been built to the likeness of a human limb, the more precise the motion can be.

Although a human arm has a total of 7 DOFs, only 5 orientation angles, per arm, are determined in our gesticulation engine, as mentioned in Table 4.1. There are positional limits to each axis of movement on the robot's arm. The positions between the maximum and minimum limits represent the range of movement for each axis. Similarly, there are maximum and minimum angles on each joint of the human user's arm. To map the joint angles on the human's arm to the robot's arm, a mathematical relationship was established such that the range of positions on the robot arm is represented as a percentage, with the minimum limit as 0%. Likewise, the joint angles of the human arm are converted to a percentage of the predefined range. To position the robot's arm during runtime, the motor for each axis moves to the position that corresponds to the percentage value from the human, as determined by the Kinect.

4.6.3 Scripted Movements

To enhance the richness of the robot's conversational abilities, a scripted movements module is present. This module contains basic gestural elements such as hand waving, pointing, and nodding. These gestural elements can be triggered by the inhabitator via the GUI as and when necessary or be placed one after another in a series to be called upon to accompany scripted speech. This allows the teleoperator to gesticulate without Kinect Avateering (Section 4.6.2).

Currently, the ALSTER framework does not have a conventional keyframe timeline to define and animate scripted movements. Instead, ALSTER utilizes the procedural animation methods, which apply simple scripting to puppeteer the humanoid avatar for gesticulation. The difference between procedural animation and traditional keyframe animation is that instead of controlling the orientation and position of each body part, we control the motion of each joint motor. Therefore, each scripted movement is a C code that tells the motor the desired angle for the joint to reposition to at a particular time. While the joint controller is executing the script, it ignores other joint commands until the current script is completed. This is to avoid any conflicting commands to be given to the same limb.

4.6.4 Reactive Gesticulations

The reactive gesticulations module has been included in the ALSTER framework as one of the autonomous gesticulation service elements. It provides the functionality that enables a social robot to be aware of the stimuli coming from its surrounding , and perform an appropriate response autonomously. This is always active by default to provide liveliness in the robot.

When no one is interacting or using the humanoid avatar, the robot will be in an idling mode where it actively performs some small random motions to indicate that it is “alive” and functioning. When the robot gets one stimulus (together with the source position of the stimulus), it will turn and face the origin of the stimulus and check if there is a human there. Currently, the robot can respond to two types of stimulus — “sound” and “face”.

Using the sound localization module, as described in Section 4.4.2, an autonomous robotic avatar is able to “hear” and determine the direction of any loud enough sound or human speech with a Kinect sensor. The module then generates the motions that cause the robot to face and give attention to the person who is speaking in front of it, as shown in Figure 4.22. This gives an impression that the robot is interested in the sound or the person who is speaking.

With the same Kinect sensor, the social robot is able to detect humans, using the human detection and tracking module described in Section 4.4.1. In this manner, the robot is able

to deduce the number of people who are standing within the field of view of the Kinect sensor. If there is only one person in the view, the robotic humanoid will always turn its torso to face and stay engaged with the person. When there are more than one person in the field of view, the robot will calculate the mean position among the people in view and turn the torso to face the centroid of the people cluster, as shown in Figure 4.22. The humanoid avatar will then turn its head to face each individual in the group randomly, if no one is speaking. However, when someone in the human group speaks, the robot will get this “sound” stimulus and turn its head to face the person who is speaking.

Figure 4.22: Functionality of the Reactive Gesticulations Module. Robot looking at the speaking person, while keeping its torso facing the people cluster.

4.7 The Conversation Service Elements

The conversation service elements allow the robot to interact verbally with interactants. A number of conversational service elements have been implemented in the ALSTER framework, and will be discussed in this section. The discussion of these conversational service elements have been arranged to begin first with those of low autonomy such as audio tele-conferencing (Section 4.7.1) and manual text input (Section 4.7.2). The discussion then follows by describing scripted speech conversation service element (Section 4.7.3), which operates with greater autonomy before finally describing the automated dialog module (Section 4.7.4) which affords the greatest autonomy level during conversation.

4.7.1 Tele-Conferencing (Audio)

Tele-conferencing has become a very familiar term since the past couple of decades. Typically, it refers to the two-way and often real-time electronic transmission of video as well audio content between two or more geographically separated parties. The tools for this usually involve a computer, an internet connection, microphones, speakers and a screen.

With a telepresence humanoid avatar, the set up for tele-conferencing is slightly different in that the screen may be the humanoid's face rather than the traditional flat-screen display. The humanoid's face is contoured like a human's face and would display the inhabitant's facial features and expressions. The details of this face display will be presented in Section 4.6.1 since this section is reserved for discussing the conversational or audio aspect.

The inhabitant's speech is captured from a microphone and transported to the robot via a teleconference link for playback on the robot's onboard speakers. The teleconference link can be established using WebRTC, as discussed in Section 2.8.1 or by using third-party tele-conferencing software such as Skype. Audio data can also be streamed within the ROS framework, using the `audio_common` package [195].

This form of service element is zero in autonomy because both the sound and the content of the speech is produced by the inhabitant manually during a conversation session.

4.7.2 Manual Text Input

In response to something said by the interactant, the inhabitant may choose to communicate verbally via a real-time text input instead of via audio tele-conferencing. This is especially useful when the teleoperator is using and controlling the social robotic avatar to perform a Wizard-of-Oz experiment or to build a translation application where the robot will "speak" a translated language, on behalf of the teleoperator. That is the purpose of the manual text input module for conducting verbal conversation via the social robot.

In ALSTER, a text interface has been provided as a client application. It takes in keyboard input. The inputted text can be processed first or be directly sent to the text-to-speech synthesizer, as described in Section 4.9.3, for audio playback.

Although the content of the speech is produced by the inhabitant, this manual text input form of conversation element is an increase in autonomy level from tele-conferencing because the sound of the speech is not reliant on the human user.

4.7.3 Scripted Speech

The scripted speech element is a slight increase in autonomy level from manual text input. This feature is useful when the social robot acts out a pre-determined script with another human actor. Therefore, rather than inputting a response via text in real time, a script is prepared in advance. The speech script is typically divided into chunks containing at the maximum a few sentences. Each speech chunk is invoked via a menu at the control station.

In ALSTER, the scripted speech can be designed to have accompanying gesticulations. For instance, when the humanoid avatar is scripted to say "Hello", an accompanying hand

waving gesture can be animated. This helps to liven up the scripted content with body language. To do so, the speech content can be incorporated with the scripted movement gesticulation service element, as described in Section 4.6.3.

4.7.4 Automated Dialog Engine

A dialog engine is a software module that has been included in the ALSTER framework to enable an autonomous social robot to converse with a human, with a coherent structure. The automated dialog engine takes in a text input that is then analyzed to produce a general flow of conversation. The text input can be derived from an automatic speech recognizer, which converts the human speech input to text, as described in Section 4.9.4. Alternatively, the text input can be obtained directly from a keyboard, via an interface.

In ALSTER, the dialog engine was developed using the Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) technology [196]. The main component of the A.L.I.C.E. project is the development of the Artificial Intelligence Markup Language (AIML) [197], which is an XML-based programming language that constitutes the knowledge database.

In the same manner, a set of AIML files have been included in the ALSTER framework, within the memory subsystem. They form the “categories” of the fundamental knowledge base. Each AIML file (or category) works in a stimulus-response model, and it consists of at least two elements: the “pattern” and the “template”. The stimulus is the human speech-to-text or typed-in text input and it may correspond with a defined “pattern”. The response will then be the “template” associated with the “pattern”. The process of retrieving the matched pattern and showing the related template will be carried out by an graph-based AIML interpreter engine.

Additionally, plug-in APIs are implemented within the automated dialog engine to acquire more content from the Internet. For instance, a Wikipedia API has been included in the dialog engine to search Wikipedia for content should a suitable response not be found on the AIML knowledge database. Another example is the Weather API, which has been implemented to acquire weather data and forecasts from the Internet, so as to provide accurate responses to weather-related queries.

The final output of the automated dialog engine is a text response. It can be directly displayed on a screen or be sent to the speech synthesizer module (as described in Section 4.9.3) for conversion to audible human speech.

4.8 The Supervisory Layer

The supervisory layer is the pivotal layer to enable the robot to perform a complex task autonomously and accurately. This allows the remote user to assume a supervisor role while the autonomous robot is performing its task. This is done so with the provision of a session supervision subsystem. The layer also consists of a memory subsystem, which allows the remote user to create or modify the task, world or robot configuration before the start or after the end of the session. Lastly, with the aim for long-term autonomous deployment, a learning subsystem is included to enable the robot to learn and adapt to new task sessions.

4.8.1 Memory Subsystem

The memory subsystem of the ALSTER framework is fundamentally a data storage and management system. It is made up of a database of information that various modules, especially the visual and planning algorithms, can access or alter repeatedly. The database has been organized into three different models within the implemented memory subsystem.

The robot model represents the specifications of the robot, which include sensor configurations, visual descriptions of the robot, collision model and etc. Currently, the Unified Robot Description Format (URDF) [198] has been used to represent the robot model in ALSTER.

The world model stores and provides facts or knowledge about the external world of the robot. It is platform-independent and can be reused on different robotic platforms. Examples of the world model include maps of the environments, properties of items, allowable actions or constraints.

The task model manages procedural information and data that are required to perform various tasks. It is made up of scripts that describe the desired states over the properties of robots and worlds, the local parameters, as well as the logic to achieve the desired states. It is also platform-independent, and multiple subtasks can be combined to provide a solution for an application.

4.8.2 Learning Subsystem

A learning subsystem has been included in the ALSTER to enable the framework to improve its performance over time. Any robot learning requires data. Hence, a simple data logging module has been implemented to log various variables or sensor data, which are generated during a session, to files.

The logs are then processed offline manually, to create a new model or to enhance an existing model, so that a desirable action or task can be produced by the robot autonomously. Currently, the following information has been logged for learning.

- The gestural actions, such as joint movements and angles, produced by the remote inhabitator while using the Kinect Avateering gesticulation service element, as described in Section 4.6.2.
- The speech audio input, produced by the interactant when he or she uses the automated speech recognizer (Section 4.9.4) to “talk” to the robot.
- The text input as well as the output of the automated dialog engine (Section 4.7.4), when an interactant engages in a conversation with the autonomous robot.

However, due to personal data protection act, data logging can only be used when the inhabitator or interactants give their consent for the collection.

4.8.3 Session Supervision

To enable the robot to perform a complex task autonomously and accurately, the session supervision subsystem of the ALSTER framework utilizes state machines when creating a complex autonomous task or behavior for the robot to perform. Failure recovery routines are also incorporated to the state machines to enable the robot to perform a complex task robustly.

State Machines

SMACH [199], which stands for “State Machine”, is a ROS-independent Python library that can be used to create hierarchical state machines for a complex autonomous robot task or behavior. It has been utilized in the ALSTER framework to help in designing, maintaining and debugging state machines for complex robotic tasks.

An example of a complex task that was implemented using the SMACH is as shown in Figure 4.23. It is a state machine of a tour guiding robot that has been developed in ALSTER for an evaluation study. The study will be further described in 5.2.2.

The image of Figure 4.23 is generated by `smach_viewer`, which is provided by SMACH to allow a full introspection in the implemented state machines, transitions as well as data flow. Hence, besides showing the state machine, the `smach` viewer also highlights (in green) the currently active state and shows the data that is passed between states during execution, as seen in Figure 4.23.

Failures Recovery

In order to perform a complex task robustly for a long period of time, it is necessary for the autonomous robot to know how to recover from failures or when to ask for human

Figure 4.23: A tour guiding robot state machine implemented using SMACH

intervention. The Recovery SMACH package [200] has been reviewed and included in the ALSTER to handle failure recovery by transitioning the state machine from a failed state to any other state in SMACH.

4.9 The Social Layer

The social layer in the ALSTER framework provides the human-robot interfaces for various social and telepresence applications. Two types of human-robot interfaces have been included to handle the communication between the robot and its two different types of human users. The first type is the operator-robot telepresence interface, which includes the network link between the remote human operator and robot (Section 4.9.1), as well as the graphical user interface for the remote operator to control the robot (Section 4.9.2). The second type of human-robot interface is the interactant-robot social interface, which include the text-to-speech synthesizer (Section 4.9.3), as well as the automated speech recognizer (Section 4.9.4).

4.9.1 Network Link

The network link provides the underlying means for the user to establish human-to-human communication and human-and-robot communication remotely. The main technology for

human-to-human communication is audio and video conferencing. And, to communicate with the robot remotely, an additional data stream is required for the transmission of control and monitoring information. Figure 4.24 illustrates the overview of the network link implemented in the ALSTER framework, which consists of a WebSocket data communication link and a WebRTC video conferencing communication link.

Figure 4.24: Signaling and information exchange process of the Network Link

Data Communication Link via Websocket

As ALSTER uses the ROS as its middle-ware, a number of software modules have been developed as ROS nodes to acquire audio, video as well as various sensor readings from individual hardware. The acquired readings are then published as data messages using the ROS Inter-Process Communication (IPC) Protocol.

Therefore, the ROSBridge package [201] has been utilized to provide a server for data communication, so that a variety of front-end interfaces can connect and interact with ALSTER. It is able to run on top of HTML5 WebSocket and standard POSIX IP sockets, such as TCP and UDP.

The ROSBridge also provides a protocol to communicate with the IPC data messages in ROS and ALSTER. The protocol is based on JSON (JavaScript Object Notation), and it covers topic publishing and subscribing, parameters setting as well as service request and response. The advantage is that it allows any implementation, regardless of its transport protocol or programming language, to interact with ALSTER by sending and receiving serialized JSON objects.

In this work, the transport protocol used for data communication is WebSocket, and the interface is a web-based GUI, implemented using web programming languages, such as HTML, CSS and JavaScript. A lightweight JavaScript client library, `ros.js`, has been included to facilitate the implementation of data communication between the web-based GUI and ALSTER. The GUI is as described in Section 4.9.2.

Audio-Video Conferencing Link via WebRTC

Besides providing a data communication link for user to monitor and control the robot, a video conferencing link has been implemented within ALSTER to facilitate human-to-human communication. The Web Real-Time Communication (WebRTC) [202] technology has been used to implement the video conferencing communication link. It is an open project that provides web browsers and mobile applications with Real-Time Communications (RTC) capabilities.

The greatest advantage is that WebRTC enables real-time tele-communication via voice, video and data without installing any special plug-ins or applications, and without having to pay for these services. It is also cross platform and so the user is able to control the robotic avatar through any device that has a web browser, such as Chrome, Firefox and Opera.

The crucial component of the WebRTC engine is the provision of a server for the signaling process. It is possible to create a simple WebRTC application without any server component for signaling, however that can only work well when there is no NATs (Network Address Translation) or firewalls. WebRTC uses the Interactive Connectivity Establishment (ICE) through the STUN and TURN servers. The STUN (Session Traversal Utilities for NAT) server is used to obtain an external network address while the TURN (Traversal Using Relay around NAT) server helps to relay traffic when the peer-to-peer connection fails. ICE is a blanket term to describe the coordination between STUN and TURN, while establishing a connection.

A WebRTC API, as shown in Figure 4.24 has been utilized to generate and exchange Session Description Protocol (SDP) and send along new ICE candidates whenever they are discovered. The description includes information about which codecs to use, whether this will be an audio or video session and also consists of a list of ICE candidates. An ICE candidate is a pair of IP addresses and port numbers that is available for the robot to connect to the user with.

After the user and robot have exchanged SDPs, there will be a series of connectivity checks on all IP-and-port pairs, and one valid pair will be selected and data transmission commences between the user's browser and robot's endpoint. This entire process typically takes milliseconds.

4.9.2 Graphical User Interface

The graphical user interface (GUI) serves to process and present the received data from the WebSocket communication link as well as the video conferencing media from the WebRTC communication link. It is a web-based interface, uploaded to a web server, and can be run on any devices that have a web browser. Web programming languages, such as HTML, CSS

and JavaScript, have been utilized to implement and style the interface. The GUI comprises two webpages — the login PAGE and the main GUI CONTROL webpage.

First, the user is required to enter the robot name and password at the LOGIN page for authentication. After performing the authentication process, the main GUI CONTROL webpage will be displayed. The interface will first determine the type of device on which it is loaded. If it is a hand-held device, such as a tablet or mobile smart phone, a mobile version of the CONTROL webpage will be presented to the user. Some features of the GUI will not be loaded fully on the mobile version of CONTROL page, due to the limited screen size of the device.

At the CONTROL webpage, the user will first be prompted to allow the GUI to access the device's camera and microphone for WebRTC. Once the user allows the access, the video conferencing session via WebRTC will begin.

Figure 4.25: Graphical User Interface

Camera Views

The main central view of the GUI is the view of the video stream from the robot's front camera, as shown in Figure 4.25. This view can be displayed in fullscreen by clicking on the fullscreen button, which is located at the top right hand corner of the GUI. When in fullscreen mode, the control panels at the bottom will be hidden, and the front camera view will be extended to fit the size of the screen. Beside the main front camera view, the user may display up to two more camera views on the interface. The additional camera views can be added by clicking on the add-camera button. This is useful when the robot carries

more than one camera on-board, for instance a rear-facing camera for docking, and that the user is interested to look at the view from that camera.

User Camera View

The user camera view is located at the bottom-left corner of the GUI, as shown in Figure 4.25. The buttons below the camera view are common video-conferencing related controls, such as the end-call button, the video and audio buttons as well as the volume-control and camera-setting buttons. The end-call button terminates the entire telepresence session, while the video and audio buttons can be used to switch on or off the user's webcam and microphone respectively. The volume-control button determines the loudness of sound that is played on the robot's speaker. The camera-setting button can be used to adjust the user's webcam setting, such as brightness and aspect ratio.

Map View

The map of the robot's environment is displayed on the map view panel as shown in Figure 4.25. Although the dimensions of the map view panel is fixed, a scroll bar has been provided so that maps of different size can be displayed on the map panel. A zoom-in button is also provided to enable the user to enlarge the view of a certain area on the map. During the session, the green square on the map indicates the location of the robot. The robot's pose information is displayed as text on the robot pose button, in the format of (x, y), bearing angle. The user can reset the position of the robot by clicking on the robot-pose button. Lastly, a show-waypoint toggle button is provided to allow the user to show or hide any defined waypoints on the map.

Controls for Navigation

The navigation control panel has a number of buttons, which allows the user to either control the robot to navigate around manually or select some navigational tasks for the robot to perform autonomously. These push buttons also act as indicators, which would light up to indicate the action or task that the robot is performing now.

Manual Controls. The user can manually control the robot to navigate around in three different ways, using either push buttons, keyboard, or gamepad control. The default way of control is by pressing the six graphical arrow-shaped push buttons on the GUI. This enables the robot to move in its six basic motions (refer to Section 4.2.1) — translate forward or backward, strafe left or right and rotate clockwise or anti-clockwise. The advantage of this control method is that it requires no additional hardware, as the user can use a mouse on a computer or touch these buttons directly on a touchscreen phone or tablet device.

A keyboard or a gamepad can also be used to control robot navigation, if the user is using a computer, and not a phone or tablet device, for the telepresence session. The user can activate the keyboard-control mode, by clicking on the keyboard button. In this mode, the user can move the robot by using the numpad on the keyboard. On the other hand, if the user wants to use a gamepad or joystick device for navigation control, the joystick-control mode has to be activated by clicking on the gamepad button.

Autonomous Controls. The user may change the mode of control from manual control mode to other higher-level navigation control modes, such as following a person (refer to Section 4.5.4), walking side-by-side with a person (refer to Section 4.5.4) and waypoint navigation (refer to Section 4.5.2). When the person-following button (or the side-by-side-following button) is pressed, the user is prompted to click on the main video view to indicate the person he or she would like the robot to follow. The button would light up to indicate that the robot is performing the following task, and by pressing the button again would disengage the robot from the following task.

When the waypoint-navigation button is pressed, the user is prompted to click on the map to indicate the location that he or she would like the robot to go to. The user can also load a text file, which contains a list of waypoints for the robot to navigate to. All waypoints can be displayed on the map by pressing the show-waypoints (flag) button on the map view panel.

Lastly, the user can command the robot to perform auto-docking by clicking the docking button. The docking button also serves a secondary function of displaying the robot's battery level. When the battery level reaches a critical low value of 20 volts, the user will be notified as the robot will stop its current task to perform auto-docking autonomously.

Controls for Humanoid Gesticulation

The GUI is also expandable to control a telepresence humanoid by using the gesticulation control panel. It consists of a number of buttons, which allows the user to either control the head and arms of the humanoid manually or select some gestures for the robot to perform autonomously.

Manual Controls. The user can manually control the head and arms of the humanoid robot by using a Kinect sensor (see Section 4.6.2). To do so, the user needs to stand in front of a Kinect sensor, which is connected to the GUI's computer. The user can then click on the Kinect icon button to execute the manual gesticulation routine, such that the arms and head motion that the user performs would be mapped from Kinect skeletal data to humanoid joint angles. The humanoid joint angles are then sent to the robot, and converted to motor commands to move the head and arms joints. The captured Kinect skeletal data is also being displayed on the GUI, as shown in Figure 4.25, for real-time visualization.

Autonomous Controls. The user can also get the humanoid robotic avatar to perform scripted gesticulations autonomously by clicking on the provided gesture buttons accordingly. In Figure 4.25, there are eight predefined gesture buttons. The top four gesture buttons (look left, look right, nod head and shake head) are predefined head gestures while the next four buttons (point to me, point to you, wave hand and clap hands) are arm gestures. Additionally, the user can load a file that contains a script, which enables the humanoid to perform a scripted animation (refer to Section 4.6.3), by clicking on the scripted-animation button.

Lastly, there is an autonomous gesticulation button, which is located below the Kinect button. The user may click on this button to activate the reactive gesticulation behavior (refer to Section 4.6.4) within the humanoid robot. This enables the robotic avatar to be aware of the people around it and give eye contact to the speaking person autonomously.

4.9.3 Text-to-Speech Synthesizer

The text-to-speech synthesizer is a module that provides a process to take in text input and generate human speech output. This has been implemented in ALSTER to allow social robots to “speak” and interact with the interactants. The Microsoft Speech API has been utilized to implement the text-to-speech synthesizer, and various language packs can be installed to allow the module to produce speech in any available languages.

Figure 4.26: Text-to-Speech Module

Currently, two speech synthesizers have been developed in ALSTER, to generate human speech in two different languages — English (EN) and Mandarin (ZH), as shown in Figure 4.26. A network server has also been implemented, on each synthesizer, to receive any text input from any connected clients. Lastly, the text-to-speech module can be extended to include a new synthesizer of a different language or accent.

4.9.4 Automated Speech Recognizer

The automated speech recognition (ASR) module is a process that takes in human speech input and generates textual representation within the framework. This is to allow social robots to “hear” and interact with the interactants. The recognized text is subsequently used for reasoning in the automated dialog engine (4.7.4), where the text is further broken down into semantic representation through natural language understanding.

No speech recognition engine or API has been implemented in ALSTER. Instead, the ASR module is made up of (1) the functionality to acquire audible signal from microphones, (2) the functionality to utilize various third-party speech recognition APIs or existing ASR engines to recognize the acquired signals, and (3) disseminating the recognized text within the framework, so that other modules can use the ASR result without the need to adapt individually. Currently, the third-party ASRs used in ALSTER include the Google ASR [203] and the Windows Speech Recognition System [204] for English speech recognition, as well as the research ASR engine [205] developed in NTU, for Singaporean-accented English speech recognition.

4.10 Chapter Summary

The implementation of the Autonomy Levels for Social and Telepresence Enabled Robot (ALSTER) framework has been described in this chapter by presenting the various modules and processes within the framework.

The framework has been implemented with the use of two robotic avatars: MAVEN a telepresence robot, and EDGAR a telepresence humanoid. The subsystems (labeled as S) and software modules within the framework are organized within six layers:

- Layer 1: The physical layer, which represents the physical components (S1) of the two robot platforms.
- Layer 2,3 and 4 are the reactive-executive-deliberative layers, which have been combined to express collectively in the implementation of the controllers (S2), sensing (S3), the service elements (S4), and the perception (S5) subsystems.
- Layer 5: The supervisory layer, which is a pivotal layer that encapsulates the subsystems for memory (S6), learning (S7) and session-supervision (S8). These three subsystems enable the robot to perform a complex task autonomously and accurately.
- Layer 6. The social layer, which is primarily about the human-robot interface (S9) required for deploying the social and telepresence robot. It also handles the two-way communication between human participants in the deployment session.

The implementation of the behavioral modules, which span multiple autonomy levels in the navigation, gesticulation and verbal-communication service elements, can be considered as the focus in terms of the practical portion of this effort. In the paragraphs that follow, a short description of the software modules implemented for each service element is offered. 12 service-element modules have been implemented in all.

The navigation service elements pertain to modules that get the robotic avatar to move around its environment. They include:

- Manual teleoperation control. This is a module that requires a continuous stream of data for sensors and controls from the inhabitator. It is of the lowest level of autonomy.
- Waypoint navigation. This is a module that allows the robot to plan and move autonomously from one point to another.
- Auto-docking. A function to enable the robot to move and dock at the docking station when its battery level runs low.
- Person-following. This module is focused on navigation with respect to the interactant which is at the robot's vicinity. Therefore, it allows the robot to follow, walk side-by-side with and also guide the interactant in the robot's environment.

The gesticulation service elements allow the robot to display gestures similar to those that humans use during verbal communication. The modules within include:

- Video conferencing. This is the ability to display live-streaming video texture on the telepresence humanoid's face which acts as a screen. This is a low autonomy-level feature.
- Kinect avateering. This is the method to teleoperate the humanoid's body for gesturing, via the Microsoft Kinect sensor. This is also a low autonomy-level feature.
- Scripted gesturing movement. This feature equips the robot to select from a database of gestural movements to perform in order to accompany scripted verbal content, i.e. speeches. This is a higher-level autonomy feature.
- Reactive gesticulations. This feature, also related to gesturing but is mean to accompany reactive speech in response to stimuli at the robot's location. This is also a higher-level autonomy feature.

The conversation service elements allow the robot to interact verbally with interactants. The modules within include:

- Audio tele-conferencing. This is essentially the same as telephone technology, facilitating the transfer of sounds and speech between the inhabitator and the robot's location. This is low in terms of autonomy.
- Manual text input to speech. This function allows the inhabitator to type text at the control station, which the robot will speak when instruction to say the string of text is given. This is low in autonomy.

- Scripted speech content. This is a module for the robot to refer to a text script and then recite the script. This speech is typically recited with accompanying gestures from the “scripted gesturing movement” module. It is a higher autonomy-level function.
- Automated dialog module. This module has been implemented to afford the greatest autonomy level during conversation.

In addition to the software modules described above, 5 perception modules (human detection, sound localization, QR code detection, localization and map server), as well as 3 controller modules (mobile base controller, humanoid joint controller, virtual human controller) have been implemented within the framework to facilitate functionality of the framework and robot system.

Next, to ensure that the robot is able to perform the service element autonomously, safely and for a longer duration, 3 supervisory modules were implemented. The *session supervision module*, which uses state machines to create and keep track the progress of the service elements, as well as to provide for failure recovery. The *memory module*, which allows the user to create or modify the task autonomy, world description or robot configuration of the service element. Lastly, with the aim for long-term autonomous deployment, a *learning module* is included to enable the robot to record, learn and adapt to new task sessions.

Finally, to enable human-robot interaction, human-in-loop control, as well as to allow users to adjust the autonomy of the robot, 4 interfacing modules were implemented at the social layer. This includes the *network module* between the remote human operator and robot, the *GUI* for the remote inhabitator to control the robot, as well the interactant-robot social interface, such as the *text-to-speech synthesizer* and the *automated speech recognizer*.

In sum, a total of 9 subsystems, which contain 27 software modules have been implemented within these six layers for the framework. Although these modules, subsystems and layers appear to have been implemented in a piecewise manner, work has been performed to integrate them within a single framework on the testbed robots, such that these elements function cohesively to facilitate robot deployment with autonomy demonstrated across the navigational, gesticulation and verbal communication aspects. The following chapter will describe the case studies conducted to validate the framework.

CHAPTER 5

EVALUATION STUDIES, RESULTS AND DISCUSSION

The framework described in Chapters 3 and 4 have been implemented on to four individual robot platforms, two of which are humanoid and two are non-anthropomorphic. However, because of constraints relating to time and funding expiration, it was not feasible for the work to be installed within a platform capable of simultaneously demonstrating adjustable autonomy across all three aspects of navigation, gesticulation and verbal communication. Nevertheless, as the case studies will show, the framework functions on the robot testbeds, demonstrating the various levels of autonomous behaviors when deployed within the study scenarios. This chapter presents the evaluation studies used to validate the framework. The chapter also presents as well as discusses the results.

The evaluation has been performed by installing the developed framework on four different robots, which are deployed at four separate occasions, as shown in Table 5.1.

No.	Robot	Robotic Application	Evaluation Study
1	MAVEN-Gray	Non-anthropomorphic robotic virtual tour guide	User experience study
2	MAVEN-1	Telepresence non-anthropomorphic robotic avatar	Workload study
3	EDGAR-1	Telepresence humanoid avatar	Usability study
4	EDGAR-2	Social humanoid assistant	Acceptance study

Table 5.1: Overview of the evaluation studies

Figure 5.1: General robotic software architecture used in all evaluation studies

All robots have an identical software architecture (see Figure 5.1) which has been implemented to allow different applications to be executed. It has the ALSTER framework, where multiple algorithms, modules and information are encapsulated and provided as

reusable interfaces for implementing applications. This enhances software portability, reduces development effort and simplifies subsequent modification of the robotic system.

5.1 Study Objectives

The objective of presenting these evaluation studies can be divided into two categories: 1) to demonstrate the implemented framework as well as 2) to evaluate the developed framework for various social and telepresence scenarios.

5.1.1 Demonstrating the implemented framework

Firstly, these evaluation studies would serve to show that the proposed solution has been successfully implemented. This means that the ALSTER framework, based on the proposed solution, can be used to control a host of robots, including an anthropomorphic robot. The evaluation studies would also verify if the behaviors, implemented for navigation, gesticulation and conversation, can be adjusted to various autonomy levels for different applications.

Secondly, these studies would provide a proof-of-concept demonstration that the implemented ALSTER framework can perform what it has been designed for. This means that the ALSTER framework would have to show that it is capable of deploying a humanoid avatar or a non-anthropomorphic robotic avatar to interact with one or many interactants in various social and telepresence scenarios.

Thirdly, the evaluation studies would demonstrate the feasibility of deploying the developed ALSTER framework on different robotic platforms.

5.1.2 Evaluating the developed framework

Subjective measures would be conducted at each scenario to evaluate the performance of the robots with their designated roles. The aim is to provide an evaluation on the performance of the ALSTER framework on different robots at different scenarios.

During each scenario, standardized questionnaires from HRI disciplines or self-developed questionnaire will be utilized to investigate at least one of the following research questions:

- What is the user experience during interaction with the robot?
- What is the user perception of the robot in terms of usability?
- What is the workload of using such a robot?

5.2 Scenario 1 — A Robotic virtual tour guide and user experience study

This evaluation study illustrates the application of the ALSTER framework to a robotic virtual tour guide. The main role of this robot tour guide is to lead visitors around a center and provides information regarding displayed exhibitions.

For this study, a self-designed user experience evaluation has been made to investigate the usability as well as the experience of using such a robotic tour guide.

5.2.1 The Robot

In this study, the ALSTER framework has been realized and evaluated on a mobile non-anthropomorphic robotic avatar named MAVEN-gray, as shown in Figure 4.2(c). It has a holonomic mobile platform, and its maximum forward and lateral speed has been limited to 0.6 m/s while the rotational speed has been limited to 0.9 rad/s.

A flat 2D screen has been installed on the robotic platform and a virtual human or any media content can be projected on the screen. The Virtual Human Toolkit [173] is utilized to generate a virtual character for the robotic avatar to guide application. This virtual character is able to perform text to speech and basic nonverbal gesticulations while speaking. The virtual human can be customized to provide information of the exhibits at each station during a tour and perform ad-hoc chat with the visitors. Currently, the virtual human toolkit has been implemented on an additional laptop computer that runs the Windows operating system.

A Kinect RGB-D camera has been mounted on the robot, as seen in Figure 4.2(c). This sensor is used to track the visitors and it is connected to the tour module, which runs a person detection and following algorithm. The module is responsible for acquiring skeleton data from the Kinect sensor, running the algorithm, and sending velocity and control commands to the navigation module which in turn controls the robot's movement.

5.2.2 Implementation of the Tour Guiding Application

The ALSTER framework has been applied to develop MAVEN-gray into a robotic museum guide. The modules used in the tour guide application are as highlighted in black in Figure 5.2.

The state diagram for the tour guiding application is as shown in Figure 5.3. The states of the diagram correspond to the modes of operation that the robot would perform. There are five states in total and they have been implemented in the tour guide module.

Figure 5.2: ALSTER Framework applied to a robotic tour guide application. Modules highlighted in black are the modules used for the application.

The process begins in the *STANDBY* mode, where the robot is installed at the entrance foyer, docked to a charging station. When in this mode, the robot executes the human detection routine by utilizing a Kinect RGB-D sensor. Once the robot detects people entering the foyer, it switches into the *GREET* mode and tries to gain their attention by greeting them promptly. If the visitors show interest and move towards the robot, it would introduce itself and offer them a guided tour, as illustrated in Figure 5.4(a).

The visitors are provided with a digital menu which can be installed on a touch screen tablet or similar device, as shown in Figure 5.4(b). If the visitors decide to engage the robot for a guided tour, they can use the menu to select their desired stations to visit during the tour. After making their selections, the visitors may click the *START-TOUR* button and activate the robot to begin the tour.

During the tour, the robotic avatar is first in the *NAVIGATION* mode where it plans the shortest path based on the selected stations and guides its visitors to each station efficiently. The robot is able to avoid both static and dynamic obstacles while navigating and adjusts its

Figure 5.3: State diagram of the robotic tour guide

velocity accordingly to the visitors' pace [192][193]. This is depicted in Figure 5.4(c).

When the robot reaches the goal position on the selected station, it stops navigating and turns around to face the visitors, as seen in Figure 5.4(d). This is because the robotic avatar assumes that the visitors have been following from behind. When the robot detects a person, it stops turning and goes into the NARRATE mode. In this mode, the robot will present the station's information. This information would have been digitally prepared and loaded onto the robot. A virtual character, which has been projected on the screen, will gesture while reciting the prepared commentary.

During the narration, the robot will also try to emulate human-to-human interaction by giving attention to each visitor who has been following the tour. First, the robot performs human detection and calculates a centroid position, based on the number and position of visitors that it detects. It then turns and faces this centroid position. Subsequently, the robot tracks the visitors and randomly selects to face individuals during its narration. This provides an impression that the virtual character is able to give attention to and gaze at each visitor while speaking, much like human guides often do.

After the robot finishes its narration, it prompts the visitors to follow it to the next station. The robot continues to move and repeat the routine until it reaches the final selected station. After narrating at the final station, the robot will ask the visitors to continue their own activities while it navigates back to the entrance and docks for charging. It will then switch back to the STANDBY mode.

Figure 5.4: The sequence of interactions between the visitors and robot during a guided tour session — (a) Greeting mode (b) Usage of digital menu (c) Navigation mode (d) Narration mode

5.2.3 Result

Figure 5.4 illustrates the deployment of MAVEN as a museum guide in the Robotics Research Centre. The developed framework has been configured to enable the robot to guide visitors around six different research areas in the centre.

Six visitors went on the tour, which was led by the robotic tour guide. After the tour, each participant returned a questionnaire, with questions as shown in Table 5.2. Each question was scored on a 7-point Likert scale, with 1 being total disagreement and 7 being total agreement with the statement.

Figure 5.5 shows a summary graph for the questionnaire result of the six visitors. The result is also as tabulated in Table 5.3. Overall, the visitors gave a high median rating of 6, 4.5 and 5 (out of 7) to questions pertaining to ease-of-use, robot's performance and tour satisfaction respectively. This means that the visitors found the robotic museum guide easy to use ($mean = 5.72$, $SD = 1.27$). They were confident that MAVEN has performed its part as a museum guide well ($mean = 4.44$, $SD = 1.34$) and were satisfied with the tour ($mean = 4.83$, $SD = 1.38$).

In terms of ease-of-usage, all visitors found the robotic tour guide easy to use ($mean = 5.83$, $SD = 0.41$). Specifically, the touch screen digital menu for selecting stations is simple to use

Category	Questions
Ease of Use	Q1. The instructions given by the robotic tour guide are clear. Q2. The menu on the tablet is easy to use. Q3. It is easy to use the robotic tour guide.
Robot's Performance	Q4. The robotic tour guide delivers the speech well. Q5. You are confident of the robot's navigation capabilities. Q6. The robot maintains eye contact with you when it talks to you.
Tour Satisfaction	Q7. You learn something new on this tour. Q8. You enjoyed the tour more because of the robot. Q9. You will recommend other people to use the robotic tour guide.

Table 5.2: Questions used to evaluate the robotic tour guide application

Figure 5.5: Box-and-whiskers plot for the questionnaire responses of the visitors. Questions are available in Table 5.2. Responses were given on an anchored 7-point Likert scale, where 1 is total disagreement and 7 is total agreement of the statement. The result can also be seen in Table 5.3.

($mean = 6.33$, $SD = 0.52$). All visitors were able to hear and follow the instructions given by the robot ($mean = 5$, $SD = 2$), although two visitors mentioned that the robot's voice was lifeless and not as expressive as a human's voice. In fact, the visitors reported that the speech of the robot is sub-optimal ($mean = 4$, $SD = 1.41$).

It has been observed that the MAVEN robot was able to navigate from one station to another with no collisions. There were however, two occasions where the robot moved close to obstacles and came near to collisions, causing some alarm to the visitors. Therefore, as a whole, the visitors were mildly satisfied with the robot's navigation capabilities ($mean = 4.17$, $SD = 0.98$).

At each station, the robot was generally able to face and engage the visitors. There was one

Participant	Ease of Use			Robot's Performance			Tour Satisfaction		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
1	4	6	6	2	5	4	4	4	4
2	4	6	6	4	3	3	3	6	3
3	2	6	5	3	3	5	5	6	5
4	7	7	6	5	5	7	4	7	6
5	6	7	6	4	5	6	3	6	3
6	7	6	6	6	4	6	5	7	6
Median ($\tilde{x}$)	5	6	6	4	4.5	5.5	4	6	4.5
Mean ($\bar{x}$)	5	6.33	5.83	4	4.17	5.17	4	6	4.5
SD (σ_x)	2	0.52	0.41	1.41	0.98	1.47	0.89	1.09	1.38
SUMMARY									
$\tilde{x}$	6			4.5			5		
$\bar{x}$	5.72			4.44			4.83		
σ_x	1.27			1.34			1.38		

Table 5.3: Questionnaire result of the visitors

occasion where the robot did not properly face the visitors because a passerby walked past the robot while it was visually searching for its users. The passerby caused the robot to assume that the visitors have been detected and it began its narration prematurely. Despite this, most users felt that the robot was reacting to them and establishing eye contact with them ($mean = 5.17$, $SD = 1.47$). As such, the visitors perceived that the robot had acted intelligently and according to their expectations.

Most visitors stated that they had enjoyed the tour because of the robot ($mean = 6$, $SD = 1.09$), and would recommend others to use MAVEN as the robotic tour guide ($mean = 4.5$, $SD = 1.38$). However, some have stated that they did not learn something new during this tour ($mean = 4$, $SD = 0.89$). It is unclear if this was due to the content of the narration or MAVEN's aptitude in delivering the content.

When asked if they would be willing to pay for the use of a robotic tour guide, most visitors said that they would be willing to pay up to 7.00 Singapore dollars (or approximately 5.30 US dollars at the time of this experiment).

5.3 Scenario 2 — A Telepresence Robot and Workload Study

This evaluation study illustrates the application of the ALSTER framework to a mobile non-anthropomorphic robotic avatar so that it can function as a telepresence robot. A remote user is expected to use the telepresence robotic avatar to perform a walk-and-talk scenario, which simulates a scenario of an office worker having ad-hoc conversation with a co-worker, while walking along the office corridor.

In this study, a human-robot interaction study is conducted to evaluate the workload of using such a telepresence robotic avatar. It seeks to address the objectives of equipping the robotic avatar with autonomy in navigation to reduce the cognitive workload of the remote human user.

5.3.1 The Robot

A telepresence non-anthropomorphic robotic avatar called MAVEN [24][25], which stands for Mobile Avatar for Virtual Engagement by NTU, has been used for this study. Specifically, MAVEN-1 (see Figure 4.2(a)) has been used in the human-robot interaction experiment.

The robot is a holonomic robot with 4 mecanum wheels. It has an on-board computer for controlling the drive motors. For this study, the maximum forward and lateral speed of the robot has been limited to 0.6 m/s while the rotational speed has been limited to 0.9 rad/s.

5.3.2 The Telepresence Robot Application

The ALSTER framework has been applied to implement a telepresence non-anthropomorphic robotic avatar application. The modules used in this study are as highlighted in black in Figure 5.6.

In this study, the telepresence robotic avatar is able to function in both the teleoperation and autonomous mode. When in teleoperation mode, the remote inhabitator can use the graphical interface to move the telepresence robot around. The cameras, which are mounted on the robot, are used for the two-way AV teleconferencing. The RGB imagery of the inhabitator would be projected on a flat 2D screen that is installed on the robotic platform.

When in autonomous mode, the Kinect RGB-D camera is used to track the interactant, so that the robot is able to perform autonomous person following. The Hokuyo LIDAR is also being used when the robot is in autonomous mode, to facilitate obstacle avoidance.

Figure 5.6: ALSTER Framework applied to the telepresence robot application. Modules highlighted in black are the modules used for the application.

5.3.3 The HRI Experimental Study

One objective of this thesis is to design and develop a robotic framework that allows the robotic avatar to be controlled with various autonomous behaviors at different degrees of autonomy. Although autonomous behaviors for navigation, gesticulation and conversation have been implemented within the framework, autonomous navigational behavior would be evaluated in an experiment in this study. The aim of the experiment is to fill the gap of knowledge on whether imbuing a telepresence robot with fully-autonomous navigation behaviors can help to enhance the experience of telepresence, by reducing the user's cognitive workload.

It will ascertain if employing the conventional manual-controlled telepresence robotic avatar will induce more workload and result in degraded navigation performance. The experiment will also validate whether using a robotic avatar with full-autonomous behaviors is able to improve the overall performance efficiency during a walk-and-talk scenario. The aim is to determine if (a) the manual-controlled robotic avatar's susceptibility to collisions and

(b) user's provision of more attention to the robot-driving task to ensure safety during a walk-and-talk scenario can negatively affect the overall performance and workload.

Hypotheses

In this study, the experiment was designed and carried out to evaluate the following hypotheses:

Hypothesis-1 The addition of fully-autonomous navigation behaviors to a robotic avatar helps to ensure safety and better performance (less time taken) during the navigation of the robotic avatar in a remote environment.

Hypothesis-2 The addition of fully-autonomous navigation behaviors to robotic avatar helps to reduce the user's cognitive workload.

Participants

A total of twelve volunteers were sourced to become unpaid participants for the experiment. All were men and the average age of the participants was 30.2 years ($SD = 4.9$) with a range of 24 to 35. Participants were asked to provide their demographic information including age and occupation. The information about their prior robot experience, prior video-conferencing experience, driving proficiency as well as their computer and video game usage in the last 12 months was also obtained.

All participants had extensive experience with computers and had prior experience in using video-conferencing tools, such as Skype. Eight participants had a driving license but did not drive often. Five of the participants had prior experience in using or operating a robot (such as Roomba vacuum cleaner robot, Lego Mindstorm and other robotic research platforms).

Experiment Procedure

The experiment was conducted in an indoor laboratory environment. The schematic layout of the test environment is as shown in Figure 5.7.

A familiarization session has been conducted to acquaint the participants with the laboratory and the inhabitator station. The inhabitator station was placed in an enclosed room and it comprised a laptop, which was connected to the Internet, as well as a gamepad that was used to control the robot manually.

The participant would control the robotic avatar from the GUI webpage loaded in a Web browser. Live video and audio feeds from the robotic avatar's webcam were displayed on

Figure 5.7: Schematic Layout of the test environment

the GUI. All participants were tested individually. Each participant would complete a 2-min warm-up drive while the research assistant was in the room. The research assistant would guide the participants regarding the use of the GUI interface during the warm-up drive and answer any questions about the tasks. While the training time taken for each participant during the warm-up drive was not tracked, it was estimated that each familiarization session was typically under ten minutes.

After the familiarization session, the robotic avatar was moved to the start location. The start location was approximately 20 m away, from the room where the inhabitor station was set up, as shown in Figure 5.7.

During each run of the test, the participant was left alone in an enclosed room. The research assistant adopted the role of the interactant. As such, the research assistant would be able to see and hear the participant via the robotic avatar.

The participant was instructed to control the robotic avatar and to follow the interactant. The interactant would move along a predefined path. Markers were affixed on the floor to ensure that the interactant followed the predefined path in Figure 5.7. The interactant will try to have a conversation with the participant during the experiment, simulating a walk-and-talk scenario using a non-anthropomorphic robotic avatar.

The run began when the participant commanded the robotic avatar to move away from the starting point. The run terminated when the robot returned to its start position. At the end of each run, the participant was asked to fill up a post-test questionnaire. The entire process took 15 minutes on average.

There were two tests for each participant.

Test-1 The participant would manually control the robotic avatar with a gamepad. This test served to evaluate the participant's performance while operating the robotic avatar in manual-control mode.

Test-2 The participant would control the robotic avatar by issuing a "person-following" command with the GUI, and the robot would move to follow the selected person autonomously during the run. The second test served to evaluate the performance of the participant while the robot operated in autonomous configuration.

Experimental Variables

The design of the experiment was a simple comparative experiment design. The selected participants would perform two test treatments, and to avoid confounding due to order effects, these participants will be assigned into 2 groups randomly.

The first group would perform Test-1 run first, take a 15-min break, and perform the Test-2 run subsequently. The second group received the same treatments, but in the order of Test-2 run first. Since this was a randomized design, the experiment variables of the two samples could be compared using the within-subjects Mann-Whitney U test as well as a Kruskal–Wallis one-way analysis.

A number of performance measures were recorded during the experiment. These measures were the dependent variables and they were continuous variables.

- **Number of collisions:** The number of collisions was counted as the number of interventions made by the research assistant during the experiment. Instead of allowing real collisions to occur during the experiment, the research assistant would intervene and press the wireless emergency stop button to immobilize the robot if the robot behaved in an unexpected and dangerous manner. The research assistant would also do so when the robot was about to collide into objects. This measure provides a way to "penalize" the participants for failing to maintain adequate situational awareness of the robot's surroundings.
- **Time taken to complete the task:** The time taken to complete the task was measured as the time in seconds required for controlling the robotic avatar to follow the interactant. This would include the time taken to move the robot from the start location to end location, while trying to keep a safe distance from the interactant as well as having conversation with the interactant.
- **Participant workload:** The cognitive workload of the participant was measured using the six scales of the NASA-task load index (TLX) [9]. It is a questionnaire that was administered at the end of the test sessions and rated workload according to six

dimensions: mental workload, physical demand, temporal demand, as well as the perceived level of task performance by the participant, effort and frustration. The definitions for each dimension are described in Appendix B. The participant would rate each of these dimensions along a scale with 20 divisions. Each division represents a 5-point increment, effectively allowing the participant to rate the dimension with a score ranging from 0 to 100. The participant's overall workload can be derived with the mean score from all six dimensions [206].

5.3.4 Result

The result of the experiment has been tabulated in Table 5.4. A comparison of the average number of collisions and time to complete the task, as well as the average workload, using the different control modes is also presented.

Participant Number	Teleoperation			Autonomous		
	Time Taken (sec)	Number of Collisions	Workload (Mean NASA-TLX)	Time Taken (sec)	Number of Collisions	Workload (Mean NASA-TLX)
1	215.0	6	51.25	188.5	0	52.92
2	208.5	5	72.92	175.5	0	65.42
3	234.5	3	67.92	186.0	0	62.92
4	180.5	3	54.17	171.5	0	46.67
5	177.0	2	77.83	171.0	0	75.42
6	188.0	5	57.82	178.0	0	46.09
7	145.0	2	54.58	143.5	0	46.67
8	143.5	1	55.49	120.5	0	36.53
9	238.0	1	53.81	147.5	0	36.66
10	134.5	0	60.00	119.0	0	44.67
11	141.0	1	49.50	104.5	0	37.54
12	194.0	1	51.12	136.0	0	42.82
Mean	183.3	2	58.87	153.5	0	49.53
SD	36.5	1.9	9.17	28.8	0	12.39

Table 5.4: Experimental Results

Completion Time. To assess whether there is a difference in performance due to robot autonomy, Mann-Whitney U test ($U = 37.5$, $p < 0.05$) and Kruskal-Wallis test ($H = 3.9675$, $p < 0.05$, $df = 1$) have been performed using the completion time data. The difference in task performance was found to be statistically significant and it was concluded with 95% confidence that this difference is affected by robot autonomy. By making direct observation from the table, all participants took less time to complete the task when they were operating the robotic avatar in autonomous configuration. The time taken to complete the task is statistically lower (16.3% lower) when the robot is in autonomous mode ($mean = 153.5$)

than when the robot is in manual mode ($mean = 183.3$). This supports Hypothesis-1, which states that the addition of autonomous navigation behaviors to a robotic avatar helps to ensure better performance (less time taken) during the navigation of the robotic avatar in the local environment.

Number of Collisions. The autonomous navigation mode has also helped the inhabitant to complete the task with statistically fewer collisions. The statistical result from the Mann-Whitney U test is $U = 6, p < 0.05$ and the Kruskal-Wallis test result is $H = 14.52, p < 0.05, df = 1$. From observation, when the robotic avatar was controlled in the manual mode, a total of 27 collisions were recorded. On the other hand, no collision occurred when the robotic avatar was operated with the autonomous configuration. This finding supports Hypothesis-1, which states that the addition of autonomous navigation behaviors to the robotic avatar helps to ensure safety when the remote inhabitant uses the robot to navigate in the environment.

Cognitive Workload. A within-subject Mann-Whitney U test ($U = 32, p < 0.05$) and a Kruskal-Wallis test ($H = 5.333, p < 0.05, df = 1$) have been performed on the mean NASA-TLX workload data to assess whether there is a difference in cognitive workload due to different modes of autonomy. The difference in workload was found to be statistically significant and it was concluded with 95% confidence that there is a difference in workload when different robot autonomy is selected. From observation, the mean workload has been found to be statistically higher when the robotic avatar was operated to follow an interactant in the manual mode ($mean = 58.87$), compared to when performing the same task in the autonomous configuration ($mean = 49.53$). This result provides significant evidence in support of Hypothesis-2, which states that the addition of autonomous navigation behaviors help to reduce the cognitive workload of the inhabitant.

The mathematics to calculate the Mann-Whitney U test and the Kruskal-Wallis test for each variable can be found in Appendix C and Appendix D respectively.

5.4 Scenario 3 — A Telepresence Humanoid and Usability Study

This evaluation study illustrates the application of the ALSTER framework to a robot so that it can function as a telepresence humanoid avatar. It is an anthropomorphic robot that has an upper torso, such that the user's face and arm gestures can be projected on the humanoid avatar.

The telepresence humanoid has been deployed numerous times for the purposes of telecommunication, and has been extensively demonstrated in the news media and TV shows. No proper study can be conducted during these deployments, due to the limited time of show production. Hence, snippets of dialogue, from one of the TV shows, have been recorded as observable user experiences.

An usability study is carried out by the research assistants, who use the telepresence humanoid during these deployments to communicate with the interactants remotely. This helps to gather opinions on the perceived usability of such a telepresence humanoid, for communication purposes.

5.4.1 The Robot

The robot that is used in this study is EDGAR-1, a novel telepresence humanoid avatar, as shown in Figure 4.3(a). It is an anthropomorphic upper torso humanoid that can mimic human movements from waist up.

One unusual feature of EDGAR-1 is that the head of the humanoid is a rear-projection system [170] such that a person's face can be projected on the robot for telecommunication. Detailed descriptions of EDGAR-1 can be found at Section 4.1.2.

5.4.2 The Telepresence Humanoid Application

The ALSTER framework has been applied to implement the telepresence humanoid avatar. The modules used for the application are as highlighted in black, as shown in Figure 5.8.

As seen in Figure 5.9, a computer has been set up at the remote site, to acquire information about the remote user's physical pose and facial features using a Kinect sensor. This information is encoded and relayed over the Internet and received by EDGAR-1. The humanoid then decodes the information to mimic the user's movements. EDGAR's rear-projection head will also display the user's head based on the received information. Lastly, the inhabitator can wear a data glove if he or she would like to manipulate the fingers on the humanoid avatar.

Figure 5.8: ALSTER Framework applied to the telepresence humanoid application. Modules highlighted in black are the modules used for the application.

On the robot side, video imagery and sounds are acquired by a camera attached on EDGAR. These data are then transmitted back to the inhabitator, to facilitate communication and interaction with the interactants.

5.4.3 Report of Local User Experience

The National Geographic Channel was the first documentary channel to cover EDGAR, the telepresence humanoid system. It has been featured on one of the Machine Impossible series. Although both the inhabitator and the interactants were in the same location, the inhabitator station and EDGAR, the telepresence humanoid, were about 10 meters apart.

The hosts of the Machine Impossible show, Mischa Pollack and Max McMurdo, conducted an interview about the telepresence humanoid system with one of the researchers, as shown in Figure 5.10(a). Subsequently, the two hosts played a game of charade using the robot, as shown in Figure 5.10(b).

Figure 5.9: Direct teleoperation of EDGAR to use it as a telepresence robotic humanoid. Left image depicts the control station set-up and the right image depicts the robot mimicking the remote user's actions while interacting with the local interactants.

During the charade game, Max was the inhabitator, as shown in Figure 5.10(c). He is the remote user of the telepresence robot, who would “act out” some movie titles.

On the other side, Max's face was being projected on the robot, as shown in Figure 5.10(d). The robot would mimic Max's arms movement, and Mischa, who was standing beside the robot, was to guess the movie titles that were acted out on EDGAR.

While describing his experience when interacting with the robot, Mischa stated,

"The humourous interaction and the human sense I got from it (the robot), came from you (Max). And other androids and other robots, you don't get that sense, even when another person is controlling it. I saw you in there, and that was incredible."

At the end of the show, Mischa concluded and said that,

"It can be done, we can have social interaction and fun with the person who is not physically in front of us. I think this thing can be great when it comes to teaching people or social interaction. And this is by far the most humorous experience I ever have with a robot."

5.4.4 The Usability Study and Result

The telepresence humanoid has been deployed on various news media and TV shows, similar to the one described in Section 5.4.3. This section discusses an usability study that

Figure 5.10: Snapshots taken from the Machine Impossible show on the National Geographic Channel

is performed to investigate the remote user's perception of using the telepresence humanoid EDGAR as a telecommunication device during these deployments.

For this study, two research assistants would act as remote users. This is because during deployments, either one of the researchers would demonstrate the usage of EDGAR as a remote user, and perform the conversations with all other interactants. The research assistant would use the control station, as depicted in Figure 5.9, which was usually set up at a distance from the telepresence robot, or at a separate room from the robot. The local interactants could see the researcher via the EDGAR telepresence robot. This provided the same situation that would exist if the people were making the teleconference call.

It was important to find out if the telepresence humanoid is a usable device so as to continue using it for telecommunication purposes. Hence, a questionnaire has been designed to investigate how the remote users perceived the usability of EDGAR the telepresence humanoid. The questionnaire is presented in Appendix E: Usability Questionnaire for remote user, and the completed questionnaire is as shown in Appendix F: Usability Questionnaire for remote user, completed by two research assistants.

Category	Definition	Mean	Standard Deviation
Perceived Ease of Use	The degree to which one believes that the system is easy to use	4.125	0.719
Self Efficacy	The belief in self's own ability to succeed in using the system	3.8	0.632
Reliability	The belief that the system performs with reliability	4.375	1.188
Perceived Usefulness	The belief that the system is useful and can provide benefits	3.875	0.641
Perceived Sociability	The perceived ability of the system to perform sociable behavior	4.0	0.756
Feel of Presence	The experience of sensing a social entity when interacting with the system	3.917	0.515
Overall Satisfaction	The overall opinion of the user about the system	4.25	0.5

Table 5.5: Result of the Usability Questionnaire for Remote User

Result

The two research assistants scored the questionnaire based on their usage of the robot and the communications they had with the interactants. The result is as shown in Table 5.5. On the whole, the two users gave a high overall satisfaction rating ($mean=4.25$, $SD=0.50$) for using the EDGAR humanoid for telecommunication.

When teleoperating EDGAR to communicate with gestures, the remote users found the telepresence humanoid easy to use, and they gave a high average score of 4.13 ($SD=0.72$) in terms of perceived ease of use. At each telepresence session, there was no notable error. The remote users also scored the reliability of the telepresence humanoid application with a high rating of 4.38 ($SD=1.19$).

Although the remote users did not find the telepresence system unnecessarily complex, they felt that many people may not have the confidence to learn or use the telepresence humanoid, without the support of a technical person. Hence, they gave the self-efficacy category a middle average score of 3.8 ($SD=0.63$). The remote users perceived EDGAR the telepresence humanoid to be useful for telecommunication ($mean=3.88$, $SD=0.64$), despite taking consideration that the robot is still a prototype.

The telepresence humanoid was found to be able to perform social interactions during the deployments, and the average perceived sociability rating is 4 ($SD=0.76$). The remote users were able to “feel” being there with the interactants ($mean=3.92$, $SD=0.51$) when they used the telepresence humanoid avatar for telecommunication.

5.5 Scenario 4 — A Social Humanoid Assistant and Acceptance Study

This evaluation study demonstrates the application of the ALSTER framework to a humanoid assistant, which is capable of having social interaction with the people around it autonomously. The humanoid assistant can hear and understand verbal inquiries from its interactants and provide the requested information accordingly. Additionally, it can also perform a scripted content to promote something or to draw a crowd to itself.

The implemented social humanoid assistant has been deployed at a number of venues, including a museum, a shopping mall, several roadshows and exhibitions, as shown in Figure 5.11. To evaluate the perception of social interactions with the humanoid assistant, an acceptance study has been carried out using the Godspeed questionnaire [141] on the general public during an exhibition roadshow. This helps to gather opinions on the public impressions and acceptance of such a humanoid assistant, for social interaction purposes.

Figure 5.11: Various Edgar deployments: (a) Co-hosting at 2017 Singapore National Day Parade (b) Giving directions and recommendations at Westgate-CapitaLand Shopping Mall (c) Exhibiting at the ArtScience Museum, Singapore (d) Meet-and-greet Prime Minister of India, Narendra Modi

5.5.1 The Robot

In general, a social humanoid assistant is a social robot, which is capable of having social interaction with the people around it autonomously. The main goal of a social humanoid assistant is to assist humans in the context of a social interaction, and this can range

from simple supportive tasks, such as answering questions or making recommendations to complex collaborative tasks, such as assistive healthcare. Hence, a social humanoid assistant can be deployed in various human collaborative settings like healthcare, education, retailing, hospitality and other businesses.

The robot that is discussed in this study is the second version of EDGAR, as shown in Figure 4.3(b). It is an anthropomorphic upper torso humanoid avatar. It has been programmed to produce speech in English and Mandarin, so as to perform simple supportive tasks in education and business settings. Unlike the first version, it does not have a rear-projection head unit.

5.5.2 The Implementation of a Social Humanoid Assistant

To develop a social humanoid assistant that is capable of performing simple supportive tasks, the basic requirement is the communication skills. The ALSTER framework has been used to implement the social humanoid assistant application on EDGAR-2. Conversational service elements of higher autonomy have been applied to enable the robot to interact and communicate with its interactants autonomously. As seen in Figure 5.12, the modules that are used to implement the social humanoid assistant are as highlighted in black.

The developed social humanoid assistant, EDGAR-2 is capable of performing two main functions: 1) It is able to “act out” a scripted content, with speech and gesticulations. 2) It is able to listen to what the interactant says and provide a verbal response accordingly.

The implemented humanoid assistant has been deployed in many occasions, and these two functionalities are demonstrated in most deployments. Depending on the requirements of the deployment, the scripted content will be programmed accordingly in advance. Typically, it is some self-introduction and promotional content that can be played repeatedly during the deployment.

However, there are some deployments that require the social humanoid assistant to act or host an event collaboratively with a human on stage. For instance, the robot had been deployed to perform as one of the hosts on the Singapore National Day Parade in 2017 [207], as shown in Figure 5.11(a). In such a stage performance or presentation, the scripted contents are triggered sequentially in a particular timing.

After the scripted narration, the humanoid will be set to the interactive question-and-answer mode, which allows its audience to interact with it in natural language. This functionality requires three subsystems, which are the automated dialog system (as described in Section 4.7.4), the speech synthesizer (in Section 4.9.3) and the reactive gesticulations (in Section 4.6.4).

First, audio data is acquired by the Kinect sensor and transmitted to a speech recognizer,

Figure 5.12: ALSTER Framework applied to the social humanoid assistant application. Modules highlighted in black are the modules used for the application.

which converts spoken words to textual strings. The strings are then processed in the automated dialog system, which generates an appropriate response for the robot to vocalize. Finally, the humanoid assistant vocalizes the generated reply via the text synthesizer module, while its arms concurrently perform some human-like gestures to accompany the speech.

During the interaction, the reactive gesticulations module is running in the background to enable the humanoid assistant to maintain “liveliness”. The reactive gesticulations module allows the humanoid to be aware of the stimuli coming from its surrounding, especially the presence of humans.

When there is no human interacting with it, the social humanoid assistant will perform some random “looking around” motions. When there is a loud sound, the robot will turn its head toward the direction of the sounds, giving an impression that it is interested in the sound. While turning and checking out the source of sound, the robot will verify if a human is in its field-of-view (FOV). If there is one person in the view, it will turn its torso to face that person.

When there is a group of people within the FOV, the robot first turns its torso to face the centroid of the cluster of people. If no one is speaking, the humanoid moves its head to face each individual in the group randomly. However, when someone in the group speaks, the humanoid assistant gets this “sound” stimulus and rotates its head to face the speaking individual.

5.5.3 Result

Report of Experience

Figure 5.13 shows the deployment of EDGAR-2 the social humanoid assistant at the 2016 Children’s Season event, held at the Chinese Heritage Centre.

Figure 5.13: Children’s Season 2016 at the Chinese Heritage Centre; (a) storytelling performance, (b) game interaction, and (c) question-and-answer session.

The ALSTER framework has been used to implement a scripted storytelling content for EDGAR-2 to assist a human storyteller to tell a story on stage. The purpose of this deployment is to boost interest of young children for their heritage language (or mother

tongue) through fun activities such as storytelling, games and songs. This would help to dispel the bias in Singapore that learning a mother tongue is “uncool” or boring.

The intention of using a social humanoid assistant is to inculcate curiosity in children, hoping to strengthen their engagement in the storytelling sessions as well as to motivate them to love their mother tongue. Storytelling can help to enhance learning language (in this case, the Chinese language) because children involved in storytelling programs exhibit better listening skills and increase language appreciation [208].

In total, two storytelling sessions were conducted during the Children’s Season event with approximately one hundred parents and children at each session. The children were mainly preschoolers and students from primary schools, age ranging approximately from four to eight years old.

Storytelling was done by EDGAR-2 in collaboration with a human storyteller in Mandarin. This is shown in Figure 5.13(a). After each storytelling performance, EDGAR-2 interacted with the children through a simple game, as seen in Figure 5.13(b), followed by photo-taking. The story and game are based on Chinese idioms. Due to the programs scheduled during the event, only the crowd from the second session had the opportunity to chat and ask EDGAR questions, as seen in Figure 5.13(c).

The robotic performance and interactions were well received by the children and their parents. Most of the audience, including the Chinese and non-Chinese speaking children, as well as their parents, were willing to participate in interacting with EDGAR during the storytelling session (see Figure 5.13(a)).

It has been observed that the children approached the humanoid on their own and initiated conversation with EDGAR-2, without their parents’ supervision. The same did not occur with the human storyteller.

Several parents commented that they brought their children to the event because they were curious about the humanoid and wanted to see EDGAR.

During the game interaction, the children were enthusiastic to answer the questions that EDGAR posted, even when they did not know the answer. The parents were raising their hands and showing interest during the game interaction as well. Some parents took the opportunity to explain the idioms to their children.

These observations have demonstrated the potential of using a social humanoid to promote the heritage language and cultural learning in young children.

User Evaluation

To evaluate the perception of social interactions with the social humanoid assistant, an acceptance study has been carried out using the Godspeed questionnaire [141]. The

intention is to gather opinions on the impressions and acceptance of such a humanoid assistant, for social interaction purposes.

33 participants (17 male) took part in this evaluation. The mean age of the participants was 38.3 years ($SD = 11.5$) with a range of 19 to 59, and only 12 of them had experience using a robot, such as a robotic vacuum cleaner.

Each participant was asked to interact with EDGAR-2 the social humanoid assistant, by chatting with it or asking it questions using natural language, for fifteen minutes. Subsequently, they were asked to rate the robot and their experience on the Godspeed questionnaire, which is a popular subjective measurement tool for a (social) human-robot interaction.

The questionnaire has five scales — Anthropomorphism, Animacy, Likeability, Perceived Intelligence and Perceived Safety. Each scale was rated on a 5-point Likert semantic differential scale, with the lower score corresponding to a more negative assessment of the humanoid. The questionnaire can be found in Appendix G.

Category	Cronbach's Alpha	Mean	Standard Deviation
Anthropomorphism	0.821	2.648	0.819
Animacy	0.851	3.005	0.896
Likeability	0.862	3.745	0.513
Perceived Intelligence	0.720	3.455	0.462
Perceived Safety	0.740	3.677	0.838

Table 5.6: Result of the Godspeed questionnaire on EDGAR version 2

The result from the Godspeed questionnaire is as shown in Table 5.6, and it summarizes the interactant responses to each category of the Godspeed questionnaire.

The Cronbach's alpha has been calculated for each category to ensure internal consistency reliability. Nunnally [209] recommended a minimum Cronbach's alpha of 0.7 and as seen in the first column of the table, the consistency of the measurements for all categories are in the acceptable range, where all $\alpha > 0.70$. The mean response for anthropomorphism was near the middle of the 5-point semantic differential scale, while the rating scores for the remaining categories were above the middle.

The participants scored anthropomorphism with an average rating of 2.65 ($SD=0.83$). This is relatively low largely due to the machinery appearance of the humanoid. EDGAR-2 achieved an average score of 3.01 ($SD=0.90$) in terms of animacy. This is because the participants only interacted with EDGAR-2 via its question-and-answer mode, in which the robot only performed random speaking gestures while it talked. EDGAR-2 was also rated more mechanical and artificial as opposed to organic and lifelike in the animacy category. Among all Godspeed category, the likeability category and perceived safety category have

been rated a high average score of 3.75 ($SD=0.51$) and 3.68 ($SD=0.84$) respectively, indicating that EDGAR-2 is well-liked and is acceptable by the participants. Lastly, EDGAR-2 is perceived to be intelligent, with an average score of 3.46 ($SD=0.46$).

5.6 Discussion

This chapter described four evaluation studies, which have been implemented using the ALSTER framework. For each study, the application of ALSTER framework to the robot has been illustrated, and the modules that are used for the robot have been highlighted accordingly.

In the first study, a social tour guide robot has been implemented. It is an autonomous robot that is able to lead visitors around a centre and provide information regarding the displayed exhibitions verbally. The robot used in this study is non-anthropomorphic.

The second study described the implementation of a telepresence robot. This depicted a typical telepresence application, which has a low autonomy level of control, teleoperated by a remote user to interact with an interactant. The robot used in this evaluation is non-anthropomorphic.

The third evaluation study also described a telepresence application. However, unlike the second study, the robot used in this application is an anthropomorphic humanoid avatar. The humanoid is fully teleoperated by a remote user during a telepresence session.

Lastly, the fourth study demonstrates the application of the ALSTER framework to a social humanoid assistant. It is an autonomous robot that is able to perform a scripted content and to have a conversation with its interactants.

5.6.1 Demonstration of ALSTER framework

These evaluation studies have successfully demonstrated the feasibility of deploying the implemented ALSTER framework on four different robotic platforms — two non-anthropomorphic robots, in Study 1 and 2, as well as two anthropomorphic humanoids in Study 3 and 4.

The evaluation studies have also proved that the ALSTER framework can be deployed on a humanoid avatar or a non-anthropomorphic robotic avatar for various social and telepresence scenarios. Both the tour guide robot in Study 1, as well as the social humanoid assistant in Study 4 are examples of robots being used for a social scenario. While Study 2 and Study 3 demonstrated a telepresence non-anthropomorphic robotic avatar and a telepresence humanoid application respectively.

Finally, the four evaluation studies have verified that a combination of navigation, gesticulation and conversational service elements can be used to achieve different autonomy levels for different applications.

For instance, Study 1 described an autonomous tour guide robot, which used modules of higher autonomy from the conversation, gesticulation and navigation service elements. On the other hand, Study 2 described a fully teleoperated robot, which utilized modules of the lowest autonomy from the conversation, gesticulation and navigation service elements. Furthermore, in Study 2, an experimental effort was carried to investigate whether adding autonomous navigation behaviors is able to improve the overall performance and the user's cognitive workload when using a telepresence robot. Hence, a high autonomy module (person following module) from the navigation service element has been included to facilitate the experimental study.

Similarly, modules of the lowest autonomy from the conversation and gesticulation service elements had been utilized for the full manual control of a telepresence humanoid in Study 3. And in Study 4, modules of the higher autonomy from conversation and gesticulation service elements have been applied to achieve a social humanoid assistant. In both Study 3 and 4, the humanoid is not equipped with a mobile platform, hence no module from the navigation service element was used.

5.6.2 Evaluation of ALSTER framework

Subjective studies have been carried out to evaluate the robots when they are performing their designated roles. The aim is to provide an evaluation on the performance of the ALSTER framework on different robots in different scenarios.

In Study 1, a self-designed questionnaire had been used to evaluate the user experience of using the tour guide robot. On a 7-point Likert scale, the participants rated the autonomous navigation service element of ALSTER with a middle average score of 4.17 ($SD=0.98$). The autonomous gesticulation service element (reactive gesticulations) achieved a high average score of 5.17 ($SD=1.47$). The conversation service element (scripted content) received a middle rating, as some participants stated they didn't learn something new during this tour ($mean = 4$, $SD = 0.89$) and that the speech of the robot is lifeless and sub-optimal ($mean = 4$, $SD = 1.41$). This implied that the performance of the conversation service element can be affected by an uninteresting content of the script, or the robot's aptitude in delivering the speech.

Nevertheless, people still enjoyed the tour because of the robot ($mean = 6$, $SD = 1.09$), and were willing to pay for the usage of a robotic tour guide. As a whole, the robot has functioned the role of a robotic tour guide successfully well ($mean = 5.72$, $SD = 1.27$), when using the ALSTER framework.

Study 2 described a full HRI experimental study to evaluate whether adding an autonomous navigation behavior to a telepresence can ensure safety and better performance, as well as to investigate whether an autonomous navigation behavior can help to reduce the user's cognitive workload. The result supported the hypotheses of the study:

1. The addition of autonomous navigation behavior to a telepresence robotic avatar ensures a better performance (less time taken) in navigating the robot from one point to another.
2. The addition of autonomous navigation behavior to a telepresence robotic avatar ensures safety when the remote inhabitator use the robot to navigate in the local environment.
3. The addition of autonomous navigation behavior helps to reduce the cognitive workload of the remote inhabitator when using the telepresence robot for interaction.

Next in Study 3, an usability questionnaire had been used to investigate the remote users' perception of using a telepresence humanoid for telecommunication. Predominantly, the users gave a high overall satisfaction rating of 4.25, on a 5-point Likert scale, when using the humanoid for telepresence application. The remote users found the robot easy to use and gave a high rating of 4.13 in terms of perceived ease of use. In particular, the gesticulation service element (Kinect Avateering) of the ALSTER framework, which was implemented on the robot for this study, had been rated an average of 4 in terms of perceived ease of use. The teleconferencing (audio and video) service elements were also rated quite highly because the remote users were able to see and hear the interactants well enough to have a sense of "being there" with the interactants ($mean=3.92$, $SD=0.51$). Lastly, the telepresence humanoid application worked reliably ($mean=4.38$, $SD=1.19$), with no notable errors during deployments.

Finally in Study 4, the result suggested that the social humanoid assistant, which was implemented using the ALSTER framework, was likeable and can be useful when used to promote learning in young children. The Godspeed questionnaire had been used to gather opinions on the impressions and acceptance of interacting with the social humanoid assistant. The questionnaire has five categories on a 5-point semantic differential scale, and the social humanoid assistant was rated with an overall mean score of 3.31, implying that the participants had a positive impression of the robot and hence were able to accept using the humanoid for social interaction. In particular, the likeability, perceived safety and perceived intelligence were rated relatively high in this study.

In conclusion, the evaluation studies have proven the feasibility of deploying ALSTER framework on different robots for various social and telepresence applications. Additionally, the subjective studies have provided an insight to the performance of the ALSTER framework, when it is applied to different robots for different scenarios.

CHAPTER 6

CONCLUSION

This thesis has presented a framework for the software architecture of social robots. This framework accommodates the software needs of both non-anthropomorphic as well as humanoid social robots for social and telepresence applications.

The motivation for this work stems from recognizing the massive potential for social and telepresence robots while also appreciating the elevated cognitive workload associated with controlling a humanoid robot. The framework's goal is thus to point out the software modules necessary for remote deployment of such robots, as well as to structure them so that these modules can function cohesively to bring about an optimal user-experience for both the robot's inhabitant and interactant.

Furthermore, the feature of adjustable autonomy allows the appropriate autonomy level to be applied at various moments during robot deployment, so that the inhabitant avoids being out-of-the-loop if the preset autonomy level is too high, but also not overloaded with cognitive tasks if it is too low. The inhabitant can for instance, engage a high-level autonomous-navigation behavior during a typical walk-and-talk portion of robot deployment but call upon lower-level navigation-related autonomy like obstacle avoidance when piloting the robot within a cluttered office space.

6.1 Thesis Summary

Humans communicate, via face-to-face interaction, for personal and business activities. However, there is a need to provide alternate ways to communicate without having the interacting parties to travel to one another and be physically co-located. Numerous societal conditions are aligning to create this need: the unprecedented but pervasive restrictions on travel due to the pandemics, natural disasters or political unrest, the need to reduce environmental footprint as well as the demand to reduce costs and fatigue.

It has been recognized that, in addition to words, nonverbal cues and physical interaction are used to enhance interpersonal communication. Therefore, although video-conferencing tools are popular means of telecommunication now, they lack the ability to facilitate physical interactions. Telepresence robot, on the other hand, is a new type of telecommunication tool that embodies gestures within a medium for distance interpersonal communication. These telepresence robots are available commercially. They are designed with the aim to address the limitation, by integrating a mobile robotic platform with a screen for video-conferencing.

Such robots, although not resembling human form (or non-anthropomorphic), allow the user to drive the platform from point to point, and to interact with another moving person.

The core theme in this thesis has been the development of a framework for structuring software modules within a social or telepresence robot. The process of developing this framework began with examining the evolution of telecommunication tools, starting with technology for relaying only sound and then to the present with know-how to communicate physical actions using telepresence robots. These robots can be non-anthropomorphic or humanoid. A trend towards anthropomorphic or humanoid robots is emerging. Through humanoid robots, a human user can additionally project his physical presence over a distance to the robot's locality by navigating the robot, conversing with people at the robot's locality, using the robot for gestures and even manipulating objects. This thesis has thus identified navigating, gesturing and conversing as the three key aspects of deploying such robots.

While such robots are more capable, the mental workload associated with controlling such robots can also be very high. A typical episode of use can for instance, require the user to safely drive the robot, manipulate the robot's arms while also attempting a meaningful conversation. This can lead to mental overload, and consequently result in problems like collisions or ineffective communication. This thesis has noted that developers have typically addressed such concerns by equipping robots with autonomy. However, autonomy has to be applied judiciously as overuse, just as unavailability or underused, can cause frustration, diminished situation awareness and unintended outcomes. This thesis also acknowledges that various degrees of autonomy may have to be relied upon in a single deployment episode because autonomous behaviors like obstacle avoidance may hinder when attempting to direct the robot to closely approach an object like a table. Hence, while software for such robotic systems should have modules for autonomous behaviors pertaining to the three key aspects or navigation, gesticulation and conversation, the software framework should also contain modules catering for autonomous behaviors of different autonomy levels.

Past frameworks for similar purposes exist but are limited by two reasons: 1) a focus only on the navigation aspect, neglecting the gesticulation and conversation aspects, and 2) being demonstrated only on one robot or multiple iterations of the same robot. Thus, this work has identified a need to assist software developers by developing these modules of various autonomy levels according to the three key aspects within a framework while also ensuring that the framework is applicable to non-anthropomorphic and anthropomorphic robots.

With this requirement in mind, the ALSTER (Autonomy Levels for Social and Telepresence Enabled Robots) framework has been developed. It is a set of core software modules categorized according to the three key aspects of telepresence robot deployment. This array of software modules has also been cohesively arranged in a six-layered structure

to facilitate the adjustable activation of autonomous behaviors so that the user is neither cognitively overloaded nor confused by over-automation. The six layers include the physical layer (which comprises the physical hardware components and their respective software interfaces), the reactive, executive and deliberative layers (which correspond to the popular three-tiers architecture), as well as the supervisory layer (which supervise other layers to ensure safe robot autonomy) and social layer (which deal with the various human-robot interactions).

To implement the ALSTER software framework, nine subsystems have been identified and implemented. They are the hardware components, the controllers, the sensing and perception subsystems, as well as the service elements, the human-robot interfaces, the memory, learning and session supervision subsystems.

One important feature of the framework is that the collection of software modules is characterized by three main classes of service elements, that is navigational, gestural and conversational. An assortment of social and telepresence robotic operations can be set up based on the framework. They can range from fully manual to fully autonomous operations, from a mobile robot that is capable of navigation only to a complex humanoid system that is capable of navigation, gesticulation, and having conversation with humans. The navigational service elements include the direct control of the robotic avatar using an input device, as well as autonomous behaviors, such as waypoint navigation, auto-docking and person-following. The gestural service elements include the direct control of the robot's arm using direct human-to-robot motion mapping, as well as reactive gesticulations which enables the robot to perform non-verbal gestures and facial expressions autonomously. Lastly, the conversational service elements include mechanisms to allow 2 human users to have a dialog with each other as well as an automated dialog system enabling the robot to have a dialog with another person autonomously.

Four telepresence robot platforms have been used to implement and test the framework: 2 non-anthropomorphic and 2 humanoid avatars. Additionally, four case studies that simulate possible deployment scenarios were conducted to validate the framework and implementation. In addition, the case studies include the experiment to evaluate the implemented autonomy in robot navigation. The results suggest that such autonomy can reduce workload while promoting safety.

6.2 Contributions

The focus of this dissertation has been the proposal and design of the ALSTER (Autonomy Levels for Social and Telepresence Enabled Robot) software framework for telepresence and autonomously-social robots. This is based on the motivation for creating a reference for developers to base their software architecture of their social-telepresence robot creations on.

A key contribution of this solution lies within the holistic approach in its design such that consideration has been made for both non-anthropomorphic as well as humanoid robots. Furthermore, the framework has been validated via implementation and demonstration on both these two types of robots in four case-study scenarios. These achievements form the contributions of this dissertation and are described in this section.

6.2.1 Proposal and design of a software framework for telepresence and autonomously-social robots

The study envisaged a framework of software modules for autonomous behaviors to address the workload-associated issues of deploying a telepresence robot, while at the same time retaining the robots' core function as a telepresence platform. More specifically, the developed framework addresses the automation needs of the inhabitant when controlling the robot during navigation, gesticulation and conversation, via the service elements associated with each of these aspects. The framework has thus been structured according to six aspects for social and telepresence robot capabilities. Six architectural components pertaining to state-of-art social and telepresence robot capabilities have been considered. These components are manifested as the six main software layers within the ALSTER framework: physical, reactive, executive, deliberative, supervisory, and social.

A further contribution is the manner in which the software framework is not only a collection of software modules but rather, a structure of modules organized into three classes: the navigational, the gestural and the conversational classes. This arrangement has been done as a strategy to specifically target the key tasks within remote deployment of a telepresence robot for social interaction.

In addition to servicing the needs in terms of workload alleviation, the proposed framework has been designed to uphold the quality of robot-facilitated human-to-human interaction so that users are not distracted by unreliability associated with the technology. To achieve this, the framework provides a high-level ability for the robot's software to supervise the performance of the lower layers along the navigational, gesticulation and conversational elements. For instance, this supervisory layer reacts with recovery actions in the presence of problems like unexpected obstacles to facilitate this goal.

Furthermore, this study has acknowledged that varying levels of autonomy may be required during robot-deployment, in order that inhabitant input or engagement to robot control can be adjusted according to the needs of the situation. Therefore, the framework spans not only a variety of autonomous behaviors but also behaviors of different degrees of autonomy; each of which can be applied when necessary during deployment.

6.2.2 Implementation and demonstration of the proposed framework on non-anthropomorphic and humanoid robots

The review of the literature has indicated that prior studies regarding software structures for social robots usually limit their consideration only either to non-anthropomorphic robots or to humanoid robots. This is despite the many examples of non-anthropomorphic as well as humanoid robots described as being social robots. Therefore, unlike previous work, this framework in this effort accommodates functionalities along the spectrum of social-robot anthropomorphism levels. A demonstration of the proposed framework has been performed via implementation on these two types of robotic avatar platforms: two non-anthropomorphic and two humanoid robots.

The implementation of the framework has required the development of the software associated with autonomous robot behaviors, the construction of a physical control station capable of facilitating remote human-robot interaction, as well as the development of the graphical user interface. Additionally, four robot avatars have been fitted with the required ability for sensing, communication and communication to facilitate a demonstration of the framework in different situations. These four scenarios are:

1. A non-anthropomorphic social robotic,
2. A non-anthropomorphic telepresence robot,
3. An anthropomorphic telepresence humanoid and,
4. An anthropomorphic social humanoid assistant.

The feasibility and validity of the framework for use in social-robot and telepresence scenarios has been shown in the four use-case scenarios described in Chapter 5. The use-case scenarios have provided a proof-of-concept demonstration of the framework, showing that robots programmed using the framework are capable of performing the telepresence and social-robotics tasks. The scenarios also verified that the navigational, gesticulation and conversational service elements can be seamlessly combined with different levels of autonomy to suit various applications.

This work has additionally shown the ALSTER framework's versatility by describing the framework's implementation on both Windows and Linux operating systems. Furthermore, the framework has been implemented using cross compile-able languages such as C++ and Python.

Summary

In Figure 6.1, the contributions are pictorially represented. The framework's evolution is described in the left side of the figure, beginning from the system overview of the envisioned system of structuring modules for adjustable autonomy in navigational, gesticulation and

Figure 6.1: A pictorial representation of the contributions.

conversational subtasks within a framework. The subsequent framework to fulfill this key goal is also shown on the left side. The right side of the figure shows the robots used in implementing and demonstrating the validity of the framework within each of the four use-case scenarios. Table 6.1 further highlights the contributions of the work by comparing the ALSTER framework with the existing robotic frameworks examined in Chapter 2, Section 2.5.4. These frameworks are the following: 1. ROTI 2. ERICA and Geminoid 3. Robovie 4. Robonaut 5. Grace and 6. Kismet and Huggable.

Reviewed Existing Frameworks	ALSTER Framework
Other frameworks usually limit their considerations only either to non-anthropomorphic robots or to humanoid robots.	The ALSTER framework accommodates functionalities along the spectrum of robot anthropomorphism levels.
Usually, other frameworks were demonstrated on one robot or multiple iterations of the same robot.	The ALSTER framework was implemented and demonstrated on 4 different robots.

Most frameworks focus only on one or two of the 3 aspects (navigation, gesticulation, conversation) but not three.	The ALSTER framework verified that modules can be structured into navigational, gestural and conversational service elements.
Most frameworks have limited varying autonomy and hence can only demonstrate telepresence or social application but not both.	The ALSTER framework verified that the navigational, gestural and conversational service elements can be seamlessly combined with different levels of autonomy to suit various telepresence and social applications.

Table 6.1: Comparison of ALSTER framework with the reviewed existing frameworks

6.3 Future Work

This section presents related limitations in this thesis. It also points out additional work that can be done to enhance social-robot abilities.

6.3.1 Limitations regarding this study

Firstly, the ALSTER framework has been applied only to robots built within the Robotics Research Centre in the NTU campus. While the framework has already been implemented on four robots, its effectiveness as well as any needs for adjustment may be better understood if it has been implemented on more robotic platforms. A suitable robot to test the framework with would be the Pepper robot from SoftBank Robotics. The Robotics Research Centre where this work was conducted, did have access to a loaned Pepper robot for some time, and the Kinect Avateering module (as described in Section 4.6.2) had been utilized to move the arms and head of Pepper for gesticulations successfully. However, it was unfortunately returned before there was an opportunity to further develop and test the ALSTER framework with it.

Secondly, it is noted that while the evaluation studies that have been conducted in this work have helped to demonstrate the framework's feasibility, the derived metrics were based on subjective questionnaires. However, such questionnaires are likely still one of the most effective ways of ascertaining perceived workloads. It is also noted that larger sample sizes for the studies conducted would have been more desirable for observations to be more conclusive.

Thirdly, there is a need to acknowledge the participants were not experts in using robots. Their lack of exposure as well as the broader public attitude of mystery regarding such technology may have limited acceptance levels. The willingness and aptitude towards using robot-mediated human-to-human interaction may improve once such robots become more commonplace. Still, this research has performed the initial steps contributing to accepting social robots in our work and living spaces.

6.3.2 Further improvements

While the design of the framework for achieving the goals of this work has been accomplished, the following additions are desired. The framework's structure would not be altered by these additions.

Firstly, it has been noted in Study 1 that the performance of the conversation service element may be improved if it is able to produce intonations and patterns more suited for the Singaporean ear. While this is obviously because the participants were either Singaporeans or people living in Singapore, various other cultures would likely exhibit a preference for locally-accented English in text-to-speech synthesizers deployed in their environments. However, the vast majority of such synthesizers are presently able to speak only US or UK English. If locally-accented English can be synthesized and made widely available, it could significantly improve the quality and perception of robot-mediated interactions.

Secondly, it is also desired for more autonomous navigation behaviors to be made available within the framework. The navigational capabilities which have been implemented presently allow the robot to move within a space such as a University campus or laboratory. It is probable that there would be some research and commercial interests to equip social robots with similar capabilities for public spaces which are more unpredictable and have more human traffic. Such capabilities would include more sophisticated locomotion behaviors beyond the "path-navigation plus obstacle-avoidance" routine. This could bring about new considerations regarding the attitude on space-sharing between humans and robots when in a crowd.

Thirdly, the inclusion of advanced vision modules is advocated to support behaviors linked to higher autonomy. For example, capabilities such as face and object recognition supported by off-line machine learning can help social robots be perceived as more sociable and intelligent.

The future work may also include the study regarding the dynamics of having multiple people interacting simultaneously with the same robotic avatar or even multiple people and multiple avatars interacting at the same time. Such studies may reveal nuances or challenges that are not present between single-person interaction with a single avatar. For

instance, user satisfaction may be decreased when some interactants are unable to clearly view the inhabitant through a flat-panel screen.

Lastly, the use of 5G communication technology is suggested. Such technology would greatly support the transmission of data between remote users and social-telepresence robots, enriching the experience of robot-mediated human-to-human interaction while also enhancing its reliability.

6.4 Closing Remarks

In closing, this work has proposed and performed the implementation of a software framework for social and telepresence robots of various capability-types; notably those which are non-anthropomorphic as well as those that are humanoid. The framework seeks to facilitate the application of autonomy to key attributes that make the robots social: navigational, gestural and conversational. This effort has also provided practical evaluations of the ALSTER framework using case studies.

As a result of the current COVID-19 pandemic, there is increased attention to measures and technologies such as that presented in this thesis, which can telecommunicate one's social and physical presence in place of actually being there. This technology can support long distance verbal communication as well as convey the rich nuances unique to human physical movement or action. Therefore, the knowledge from this research effort is expected to be valuable to developers of telepresence and social robots.

PUBLICATIONS

The following is a list of original publications and technical disclosures that are directly related to the work presented in this thesis.

Journal publications published

W. C. Pang, C. Y. Wong and G. Seet. “Exploring the Use of Robots for Museum Settings and for Learning Heritage Languages and Cultures at the Chinese Heritage Centre.” *Presence: Teleoperators and Virtual Environments*, vol. 26, no. 4, pp. 420–435, 2018. DOI: https://doi.org/10.1162/PRES_a_00306

W. C. Pang, G. Seet and X. Yao. “A Study on High-level Autonomous Navigational Behaviors for Telepresence Applications.” *Presence: Teleoperators and Virtual Environments*, vol. 23, no. 2, pp. 155–171, 2014. DOI: https://doi.org/10.1162/PRES_a_00178.

Conference publications published

W. C. Pang, C. Y. Wong, and G. Seet, “Design and development of EDGAR - A Telepresence Humanoid for Robot-Mediated Communication and Social Applications,” in *Proceedings of the International Conference on Control and Robotics Engineering (ICCRE)*, pp. 1–4, IEEE, 2016. DOI: <https://doi.org/10.1109/ICCRE.2016.7476147>.

W.C. Pang, G. Seet, and X. Yao, “A Multimodal Person-following System for Telepresence Applications,” in *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology (VRST)*, pp. 157–164, ACM, 2013. DOI: <https://doi.org/10.1145/2503713.2503722>

W. C. Pang, Burhan, and G. Seet, “Design Considerations of a Robotic Head for Telepresence Applications,” in *Intelligent Robotics and Applications (Lecture Notes in Computer Science)*, vol. 7508, pp. 131–140, Springer Berlin Heidelberg, 2012. DOI: https://doi.org/10.1007/978-3-642-33503-7_14

G. Seet, **W. C. Pang**, Burhan, I.-M. Chen, I. Viatcheslav, G. William, and C. Y. Wong, “A Design for a Mobile Robotic Avatar - Modular Framework,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, Zurich, Switzerland, 2012. DOI: <https://doi.org/10.1109/3DTV.2012.6365444>.

Workshop publication

G. Seet, **W. C. Pang**, and Burhan, “Towards the Realization of MAVEN - Mobile Robotic Avatar,” presented in *The 25th International Conference on Computer Animation and Social Agents (CASA)*, Singapore, 2012.

Technology Disclosures

C. Y. Wong, **W. C. Pang**, and G. Seet, “EDGAR-2 Social Robot Hardware Design,” April 2019. TD Ref No.: 2019-103.

W. C. Pang, C. Y. Wong, and G. Seet, “EDGAR-2 Social Robot Software Design,” April 2019. TD Ref No.: 2019-104.

G. Seet, Burhan, **W. C. Pang**, C. Y. Wong, Y. B. Ang and V. Iastrebov , “A Compact Modular Mobile Robotic Platform,” July 2016. TD Ref No.: TD/083/16.

W.C. Pang, G. Seet, and X. Yao, “A Multimodal Person Following System For Telepresence Applications,” February 2015. TD Ref No.: TD/033/15.

C. Y. Wong, **W. C. Pang**, Burhan and G. Seet, “Facial Display and Upper Body Gesturing - Humanoid Robot,” October 2014.

BIBLIOGRAPHY

- [1] D. Kortenkamp and R. Simmons, "Robotic systems architectures and programming," in *Springer Handbook of Robotics*, pp. 187–206, Springer, 2008.
- [2] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [3] O.-H. Kwon, S.-Y. Koo, Y.-G. Kim, and D.-S. Kwon, "Telepresence robot system for english tutoring," in *2010 IEEE Workshop on Advanced Robotics and its Social Impacts*, pp. 152–155, IEEE, 2010.
- [4] D. F. Glas, T. Minato, C. T. Ishi, T. Kawahara, and H. Ishiguro, "ERICA: The ERATO Intelligent Conversational Android," in *Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 22–29, IEEE, 2016.
- [5] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "A communication robot in a shopping mall," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 897–913, 2010.
- [6] M. A. Diftler, C. Culbert, R. O. Ambrose, R. Platt, and W. Bluethmann, "Evolution of the NASA/DARPA Robonaut control system," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 2543–2548, IEEE, 2003.
- [7] D. Kortenkamp and R. Simmons, "Robotic systems architectures and programming," in *Springer Handbook of Robotics*, pp. 187–206, Springer, 2008.
- [8] C. L. Breazeal, *Designing Sociable Robots*. MIT press, 2004.
- [9] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in *International Journal of Advances in Psychology*, vol. 52, (Tempe, USA), pp. 139–183, Elsevier, 1988.
- [10] M. Minsky, "Telepresence," *Omni*, vol. 2, pp. 45–52, 1980.
- [11] R. L. Daft and R. H. Lengel, "Organizational information requirements, media richness and structural design," *Management science*, vol. 32, no. 5, pp. 554–571, 1986.
- [12] J. Short, E. Williams, and B. Christie, *The social psychology of telecommunications*. John Wiley and Sons Ltd, 1976.
- [13] G. R. Walker and P. Sheppard, "Telepresence - the future of telephony," *BT technology journal*, vol. 15, pp. 11–18, 1997.
- [14] T. Szigeti, K. McMenemy, R. Saville, and A. Glowacki, *Cisco TelePresence Fundamentals*. Indianapolis, Indiana, USA: Cisco Press, 2009.
- [15] H. Fuchs, A. State, and J.-C. Bazin, "Immersive 3d telepresence," *Computer*, vol. 47, no. 7, pp. 46–52, 2014.

- [16] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, *et al.*, “Holoportation: Virtual 3d teleportation in real-time,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 741–754, 2016.
- [17] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, and M. Billinghamurst, “Mini-me: An adaptive avatar for mixed reality remote collaboration,” in *Proceedings of the 2018 CHI conference on human factors in computing systems*, pp. 1–13, 2018.
- [18] E. A. Isaacs and J. C. Tang, “What video can and cannot do for collaboration: a case study,” *Multimedia Systems*, vol. 2, no. 2, pp. 63–73, 1994.
- [19] Willow Garage, “Texai Remote Presence System.” <http://www.willowgarage.com/pages/texai/overview>, June 2011. [Online; accessed on 2013-05-15].
- [20] InTouch Technologies, Inc., “RP Endpoint Devices.” <http://www.intouchhealth.com/>, 2011. [Online; accessed on 2013-06-24].
- [21] Anybots, Inc., “Anybots | ANYBOTS Virtual Presence Systems. It’s you, anywhere....” <https://www.anybots.com/>, 2010. [Online; accessed on 2013-05-27].
- [22] G. Venolia, J. Tang, R. Cervantes, S. Bly, G. Robertson, B. Lee, and K. Inkpen, “Embodied social proxy: mediating interpersonal connection in hub-and-satellite teams,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, (Atlanta, GA, USA), pp. 1049–1058, ACM, 2010.
- [23] P. Lincoln, G. Welch, A. Nashel, A. Ilie, and H. Fuchs, “Animatronic shader lamps avatars,” in *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, (Orlando, FL, USA), pp. 27–33, IEEE, 2009.
- [24] G. Seet, W. C. Pang, and Burhan, “Towards the Realization of MAVEN - Mobile Robotic Avatar,” in *The 25th International Conference on Computer Animation and Social Agents*, (Singapore), 2012.
- [25] G. Seet, W. C. Pang, Burhan, I.-M. Chen, I. Viatcheslav, G. William, and C. Y. Wong, “A Design for a Mobile Robotic Avatar - Modular Framework,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2012.
- [26] P. D. Thacker, “Physician-robot makes the rounds,” *JAMA The Journal of the American Medical Association*, vol. 293, p. 150, 2005 2005.
- [27] L. M. Ellison, P. A. Pinto, F. Kim, A. M. Ong, A. Patriciu, D. Stoianovici, H. Rubin, T. Jarrett, and L. R. Kavoussi, “Telerounding and patient satisfaction after surgery,” *Journal of the American College of Surgeons*, vol. 199, pp. 523–530, 2004.
- [28] M. K. Lee and L. Takayama, “Now, i have a body: Uses and social norms for mobile remote presence in the workplace,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, (Vancouver, BC, Canada), pp. 33–42, ACM, 2011.
- [29] National Museum of Australia, “Robot tours | National Museum of Australia.” <http://www.nma.gov.au/engage-learn/school-programs/digital-outreach/robot-tours>, March 2015. [Online; accessed on 2016-06-29].

- [30] C. Neustaedter, G. Venolia, J. Procyk, and D. Hawkins, "To beam or not to beam: A study of remote telepresence attendance at an academic conference," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pp. 418–431, ACM, 2016.
- [31] K. Shiarlis, J. Messias, M. van Someren, S. Whiteson, J. Kim, J. Vroon, G. Englebienne, K. Truong, V. Evers, N. Pérez-Higueras, *et al.*, "Teresa: A socially intelligent semi-autonomous telepresence system," *Workshop on Machine Learning for Social Robotics at ICRA-2015*, 2015.
- [32] K. M. Tsui, M. Desai, H. A. Yanco, and C. Uhlik, "Exploring use cases for telepresence robots," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 11–18, IEEE, March 2011.
- [33] I. Rae, L. Takayama, and B. Mutlu, "In-body experiences: embodiment, control, and trust in robot-mediated communication," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1921–1930, ACM, 2013.
- [34] E. T. Hall, *The Hidden Dimension*. New York, New York, USA: Anchor Books New York, 1990.
- [35] R. L. Birdwhistell, *Kinesics and context: Essays on body motion communication*. University of Pennsylvania press, 2010.
- [36] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita, "Android as a telecommunication medium with a human-like presence," in *Proceedings of on the 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 193–200, IEEE, 2007.
- [37] N. M. Thalmann, L. Tian, and F. Yao, "Nadine: A social robot that can localize objects and grasp them in a human way," in *Frontiers in Electronic Technologies*, pp. 1–23, Springer, 2017.
- [38] S. O. Adalgeirsson and C. Breazeal, "Mebot: a robotic platform for socially embodied presence," in *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pp. 15–22, IEEE Press, 2010.
- [39] K. Hasegawa and Y. Nakauchi, "Telepresence robot conveying pre-motions for avoiding speech collisions in teleconference," *2013 IEEE RO-MAN*, pp. 350–351, 2013.
- [40] E. Paulos and J. Canny, "Designing personal tele-embodiment," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, pp. 3173–3178, IEEE, 1998.
- [41] Frank Tobe, "The Beginning of Something Important: InTouch Health and iRobot's RP-VITA." <http://www.everything-robotic.com/2012/07/the-beginning-of-something-important.html>, 2012. [Online; accessed on 2016-07-24].
- [42] B. R. Duffy, "Anthropomorphism and the social robot," *Robotics and autonomous systems*, vol. 42, no. 3, pp. 177–190, 2003.

- [43] G. Hoffman and W. Ju, "Designing robots with movement in mind," *Journal of Human-Robot Interaction*, vol. 3, no. 1, pp. 89–122, 2014.
- [44] A. Nagendran, A. Steed, B. Kelly, and Y. Pan, "Symmetric telepresence using robotic humanoid surrogates," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 271–280, 2015.
- [45] F. Tanaka, A. Cicourel, and J. R. Movellan, "Socialization between toddlers and robots at an early childhood education center," *Proceedings of the National Academy of Sciences*, vol. 104, no. 46, pp. 17954–17958, 2007.
- [46] J.-J. Cabibihan, W.-C. So, S. Saj, and Z. Zhang, "Telerobotic pointing gestures shape human spatial cognition," *International Journal of Social Robotics*, vol. 4, no. 3, pp. 263–272, 2012.
- [47] W. C. Pang, C. Y. Wong, and G. Seet, "Design and development of EDGAR - A Telepresence Humanoid for Robot-Mediated Communication and Social Applications," in *Proceedings of the International Conference on Control and Robotics Engineering (ICCRE)*, pp. 1–4, IEEE, 2016.
- [48] T.-C. Tsai, Y.-L. Hsu, A.-I. Ma, T. King, and C.-H. Wu, "Developing a telepresence robot for interpersonal communication with the elderly in a home environment," *Telemedicine and e-Health*, vol. 13, no. 4, pp. 407–424, 2007.
- [49] E. de Visser and R. Parasuraman, "Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload," *Journal of Cognitive Engineering and Decision Making*, vol. 5, no. 2, pp. 209–231, 2011.
- [50] M. S. Prewett, R. C. Johnson, K. N. Saboe, L. R. Elliott, and M. D. Coovert, "Managing workload in human-robot interaction: A review of empirical studies," *Computers in Human Behavior*, vol. 26, no. 5, pp. 840–856, 2010.
- [51] T. B. Sheridan, "Musings on telepresence and virtual presence," *MIT Press Journal - Presence: Teleoperators and Virtual Environments*, vol. 1, pp. 120–126, 1992 1992.
- [52] T. B. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report," *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995.
- [53] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, *et al.*, "Mars Science Laboratory mission and science investigation," *Space science reviews*, vol. 170, no. 1-4, pp. 5–56, 2012.
- [54] A. Bowen, D. Yoerger, C. Taylor, R. McCabe, J. Howland, D. Gomez-Ibanez, J. Kinsey, M. Heintz, G. McDonald, D. Peters, *et al.*, "The Nereus hybrid underwater robotic vehicle," *Underwater Technology*, vol. 28, no. 3, pp. 79–89, 2009.
- [55] B. M. Yamauchi, "Packbot: a versatile platform for military robotics," in *Defense and Security*, pp. 228–237, International Society for Optics and Photonics, 2004.
- [56] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the KUKA lightweight robot," in *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, 2010.

- [57] J. V. Draper and L. M. Blair, "Workload, flow, and telepresence during teleoperation," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 1030–1035, IEEE, 1996.
- [58] Suitable Technologies, Inc., "Suitable Technologies Beam Smart Presence System." <https://www.suitabletech.com/beampro/>, 2016. [Online; accessed on 2016-07-17].
- [59] Double Robotics, Inc., "Double Robotics - Telepresence Robot for Telecommuters." <http://www.doublerobotics.com/>, 2016. [Online; accessed on 2016-07-17].
- [60] VGo Communications, Inc., "VGo robotic telepresence for healthcare, education and business." <http://www.vgocom.com/>, 2013. [Online; accessed on 2016-07-17].
- [61] iRobot Corporation, "About iRobot | iRobot." <http://www.irobot.com/About-iRobot.aspx>, January 2017. [Online; accessed on 2017-01-03].
- [62] M. Desai, K. M. Tsui, H. A. Yanco, and C. Uhlik, "Essential features of telepresence robots," in *2011 IEEE Conference on Technologies for Practical Robot Applications*, pp. 15–20, IEEE, 2011.
- [63] iRobot Corporation, "iRobot Ava 500 - A new dimension in telepresence for the enterprise." <https://www.irobot.com/~media/Files/Robots/Commercial%20Applications/Ava20500/iRobot-Ava-500-Solution-Overview.ashx>, January 2013. [Online; accessed on 2016-05-19].
- [64] A. Kristoffersson, S. Coradeschi, and A. Loutfi, "A review of mobile robotic telepresence," *Advances in Human-Computer Interaction*, vol. 2013, p. 3, 2013.
- [65] W. A. IJsselsteijn, H. de Ridder, J. Freeman, and S. E. Avons, "Presence: concept, determinants, and measurement," in *Electronic Imaging*, pp. 520–529, International Society for Optics and Photonics, 2000.
- [66] G. Riva, F. Davide, and W. IJsselsteijn, "Being there: The experience of presence in mediated environments," *Being there: Concepts, effects and measurements of user presence in synthetic environments*, pp. 3–16, 2003.
- [67] M. K. Lee and L. Takayama, "Now, i have a body: Uses and social norms for mobile remote presence in the workplace," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 33–42, ACM, 2011.
- [68] A. Gandsas, M. Parekh, M. M. Bleech, and D. A. Tong, "Robotic telepresence: profit analysis in reducing length of stay after laparoscopic gastric bypass," *Journal of the American College of Surgeons*, vol. 205, no. 1, pp. 72–77, 2007.
- [69] P. M. Vespa, C. Miller, X. Hu, V. Nenov, F. Buxey, and N. A. Martin, "Intensive care unit robotic telepresence facilitates rapid physician response to unstable patients and decreased cost in neurointensive care," *Surgical neurology*, vol. 67, no. 4, pp. 331–337, 2007.
- [70] Jay Fitzgerald, "After surgery, a robot may be at your side." <http://www.vgocom.com/after-surgery-robot-may-be-your-side>, December 2011. [Online; accessed on 2016-08-16].

- [71] S. Coradeschi, A. Cesta, G. Cortellessa, L. Coraci, J. Gonzalez, L. Karlsson, F. Furfari, A. Loutfi, A. Orlandini, F. Palumbo, *et al.*, “Giraffplus: Combining social interaction and long term monitoring for promoting independent living,” in *2013 6th International Conference on Human System Interactions (HSI)*, pp. 578–585, IEEE, 2013.
- [72] F. Michaud, P. Boissy, D. Labonte, H. Corriveau, A. Grant, M. Lauria, R. Cloutier, M.-A. Roux, D. Iannuzzi, and M.-P. Royer, “Telepresence robot for home care assistance,” in *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pp. 50–55, California, USA, 2007.
- [73] D. Sampsel, G. Bharwani, D. Mehling, and S. Smith, “Robots as faculty: student and faculty perceptions,” *Clinical Simulation in Nursing*, vol. 7, no. 6, pp. e209–e218, 2011.
- [74] R. Agarwal, A. W. Levinson, M. Allaf, D. V. Makarov, A. Nason, and L.-M. Su, “The roboconsultant: telementoring and remote presence in the operating room during minimally invasive urologic surgeries using a novel mobile robotic interface,” *Urology*, vol. 70, no. 5, pp. 970–974, 2007.
- [75] K. M. Lee, “Presence, explicated,” *Communication theory*, vol. 14, no. 1, pp. 27–50, 2004.
- [76] M. Lombard and T. Ditton, “At the heart of it all: The concept of presence,” *Journal of Computer-Mediated Communication*, vol. 3, no. 2, pp. 0–0, 1997.
- [77] J. Steuer, “Defining virtual reality: Dimensions determining telepresence,” *Journal of communication*, vol. 42, no. 4, pp. 73–93, 1992.
- [78] W. A. IJsselsteijn, “History of telepresence,” *3D Communication: Algorithms, concepts and real-time systems in human centred communication*, John Wiley & Sons: Chichester, 2005.
- [79] D. W. Schloerb, “A quantitative measure of telepresence,” *Presence: Teleoperators & Virtual Environments*, vol. 4, no. 1, pp. 64–80, 1995.
- [80] S. Tachi, H. Arai, I. Morimoto, and G. Seet, “Feasibility experiments on a mobile tele-existence system,” in *Proceedings of the International Symposium and Exposition on Robots*, pp. 625–636, IEEE, 1988.
- [81] J. V. Draper, D. B. Kaber, and J. M. Usher, “Telepresence,” *Human Factors: The journal of the human factors and ergonomics society*, vol. 40, no. 3, pp. 354–375, 1998.
- [82] W. Wirth, T. Hartmann, S. Böcking, P. Vorderer, C. Klimmt, H. Schramm, T. Saari, J. Laarni, N. Ravaja, F. R. Gouveia, *et al.*, “A process model of the formation of spatial presence experiences,” *Media psychology*, vol. 9, no. 3, pp. 493–525, 2007.
- [83] L. Muhlbach, M. Bocker, and A. Prussog, “Telepresence in videocommunications: A study on stereoscopy and individual eye contact,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 2, pp. 290–305, 1995.
- [84] F. Biocca, “The cyborg’s dilemma: Progressive embodiment in virtual environments [1],” *Journal of Computer-Mediated Communication*, vol. 3, no. 2, pp. 0–0, 1997.

- [85] Digital Video Enterprises, Inc., "DVE Immersion Room." http://dvetelepresence.com/room/tele_immersion.htm, 2010. [Online; accessed on 2013-07-17].
- [86] Digital Video Enterprises, Inc., "Immersive Podium by DVE Telepresence." http://www.dvetelepresence.com/immersive_podium.php, 2011. [Online; accessed on 2015-05-17].
- [87] F. Biocca, C. Harms, and J. K. Burgoon, "Toward a more robust theory and measure of social presence: Review and suggested criteria," *Presence*, vol. 12, pp. 456–480, Oct 2003.
- [88] C. De Gournay and Z. Smoreda, "Communication technology and sociability: Between local ties and 'global ghetto'," *Machines that become us: The social context of personal communication technology*, pp. 57–70, 2003.
- [89] K. Kreijns, P. A. Kirschner, and W. Jochems, "The sociability of computer-supported collaborative learning environments," *Educational Technology & Society*, vol. 5, no. 1, pp. 8–22, 2002.
- [90] R. E. Rice, "Media appropriateness," *Human Communication Research*, vol. 19, no. 4, pp. 451–484, 1993.
- [91] C. N. Gunawardena and F. J. Zittle, "Social presence as a predictor of satisfaction within a computer-mediated conferencing environment," *American journal of distance education*, vol. 11, no. 3, pp. 8–26, 1997.
- [92] C. Heeter, "Being there: The subjective experience of presence," *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 2, pp. 262–271, 1992.
- [93] F. Biocca and C. Harms, "Defining and measuring social presence: Contribution to the networked minds theory and measure," *Proceedings of PRESENCE*, vol. 2002, pp. 1–36, 2002.
- [94] E. Goffman, *Behavior in public places*. Simon and Schuster, 2008.
- [95] M. R. Endsley, "Design and evaluation for situation awareness enhancement," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 32, pp. 97–101, SAGE Publications, 1988.
- [96] A. Cesta, G. Cortellessa, A. Orlandini, and L. Tiberio, "Evaluating telepresence robots in the field," in *International Conference on Agents and Artificial Intelligence*, pp. 433–448, Springer, 2012.
- [97] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," tech. rep., DTIC Document, 1978.
- [98] M. R. Endsley, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics*, vol. 42, no. 3, pp. 462–492, 1999.
- [99] J. Beer, A. D. Fisk, and W. A. Rogers, "Toward a framework for levels of robot autonomy in human-robot interaction," *Journal of Human-Robot Interaction*, vol. 3, no. 2, p. 74, 2014.
- [100] M. Tambe, P. Scerri, and D. V. Pynadath, "Adjustable autonomy for the real world," *Journal of Artificial Intelligence Research*, vol. 17, no. 1, pp. 171–228, 2002.

- [101] D. A. Norman, "How might people interact with agents," *Communications of the ACM*, vol. 37, no. 7, pp. 68–71, 1994.
- [102] R. Kirby, R. Simmons, and J. Forlizzi, "Companion: a constraint-optimizing method for person-acceptable navigation," in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, (Toyama, Japan), pp. 607–612, IEEE, 2009.
- [103] E. A. Sisbot, R. Alami, T. Siméon, K. Dautenhahn, M. Walters, and S. Woods, "Navigation in the presence of humans," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pp. 181–188, IEEE, 2005.
- [104] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5495–5502, IEEE, 2009.
- [105] L. Takayama, E. Marder-Eppstein, H. Harris, and J. M. Beer, "Assisted driving of a mobile remote presence system: System design and controlled user evaluation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1883–1889, IEEE, 2011.
- [106] J. M. Riley, D. B. Kaber, and J. V. Draper, "Situation awareness and attention allocation measures for quantifying telepresence experiences in teleoperation," *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 14, no. 1, pp. 51–67, 2004.
- [107] J. K. Lee, R. L. Toscano, W. D. Stiehl, and C. Breazeal, "The design of a semi-autonomous robot avatar for family communication and education," in *The 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 166–173, IEEE, 2008.
- [108] Engineered Arts, "RoboThespian." <https://www.engineeredarts.co.uk/robot/robothespian/>, 2021. [Online; accessed on 2021-07-15].
- [109] J. Lee and S. Marsella, "Nonverbal behavior generator for embodied conversational agents," in *International Workshop on Intelligent Virtual Agents*, pp. 243–255, Springer, 2006.
- [110] Mantaro Product Development Services, Inc., "MantaroBot TelePresence Robots." https://www.mantarobot.com/index.php?option=com_content&Itemid=304&id=108&lang=en&layout=edit&view=article, 2021. [Online; accessed on 2021-07-24].
- [111] L. Li, B. Cox, M. Diftler, S. Shelton, and B. Rogers, "Development of a telepresence controlled ambidextrous robot for space applications," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 58–63, IEEE, 1996.
- [112] K. Hertkorn, *Shared grasping: A combination of telepresence and grasp planning*. KIT Scientific Publishing, 2016.
- [113] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic

- environments,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–6, IEEE, 2009.
- [114] D. Kent, M. Behrooz, and S. Chernova, “Crowdsourcing the construction of a 3d object recognition database for robotic grasping,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4526–4531, IEEE, 2014.
- [115] M. Argyle and M. Cook, *Gaze and mutual gaze*. Cambridge U Press, 1976.
- [116] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross, “Gaze correction for home video conferencing,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 174, 2012.
- [117] C. A. Twigg, “Models for online learning,” *Educause review*, vol. 38, pp. 28–38, 2003.
- [118] S. Yelon, “Face-to-face or online? choosing the medium in blended training,” *Performance Improvement*, vol. 45, no. 3, pp. 22–26, 2006.
- [119] L. Allen, A. Abella, T. Alonso, and H. Jeremy, “Automated natural spoken dialog,” *IEEE Computer Magazine*, vol. 35, no. 4, pp. 51–56, 2002.
- [120] S. Young, M. Gašić, B. Thomson, and J. D. Williams, “Pomdp-based statistical spoken dialog systems: A review,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
- [121] W. Ward, R. Cole, D. Bolaños, C. Buchenroth-Martin, E. Svirsky, S. V. Vuuren, T. Weston, J. Zheng, and L. Becker, “My science tutor: A conversational multimedia virtual tutor for elementary school science,” *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 7, no. 4, p. 18, 2011.
- [122] T. Noma, N. I. Badler, and L. Zhao, “Design of a virtual human presenter,” 2000.
- [123] R. Simmons, D. Goldberg, A. Goode, M. Montemerlo, N. Roy, A. C. Schultz, M. Abramson, I. Horswill, D. Kortenkamp, and B. Maxwell, “Grace: An Autonomous Robot for the AAI Robot Challenge,” tech. rep., DTIC Document, 2003.
- [124] J. Scholtz, J. Young, J. L. Drury, and H. A. Yanco, “Evaluation of human-robot interaction awareness in search and rescue,” in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 3, pp. 2327–2332, IEEE, 2004.
- [125] H. A. Yanco and J. Drury, ““Where Am I?” Acquiring situation awareness using a remote robot platform,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 3, pp. 2835–2840, IEEE, 2004.
- [126] D. Labonte, P. Boissy, and F. Michaud, “Comparative analysis of 3-d robot teleoperation interfaces with novice users,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 5, pp. 1331–1342, 2010.
- [127] T. Ito, “Hand gesture-based manipulation of a personalized avatar robot in remote communication,” in *Symposium on Human Interface*, pp. 425–434, Springer, 2011.
- [128] V. Venkatesh and F. D. Davis, “A theoretical extension of the technology acceptance model: Four longitudinal field studies,” *Management science*, vol. 46, no. 2, pp. 186–204, 2000.

- [129] A. M. Lund, "Measuring Usability with the USE Questionnaire," *Usability Interface*, vol. 8, no. 2, pp. 3–6, 2001.
- [130] J. R. Lewis, "Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use," *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.
- [131] D. R. Olsen and M. A. Goodrich, "Metrics for evaluating human-robot interactions," in *Proceedings of PERMIS*, vol. 2003, p. 4, 2003.
- [132] B. Cain, "A review of the mental workload literature," tech. rep., Defence Research And Development Toronto (Canada), 2007.
- [133] G. E. Cooper and R. P. Harper Jr, "The use of pilot rating in the evaluation of aircraft handling qualities," tech. rep., No. AGARD-567. Advisory Group for Aerospace Research and Development Neuilly-sur-Seine (France), 1969.
- [134] P. S. Tsang and V. L. Velazquez, "Diagnosticity and multidimensional subjective workload ratings," *Ergonomics*, vol. 39, no. 3, pp. 358–381, 1996.
- [135] G. B. Reid and T. E. Nygren, "The subjective workload assessment technique: A scaling procedure for measuring mental workload," *Advances in psychology*, vol. 52, pp. 185–218, 1988.
- [136] S. Rubio, E. Díaz, J. Martín, and J. M. Puente, "Evaluation of subjective mental workload: A comparison of swat, nasa-tlx, and workload profile methods," *Applied Psychology*, vol. 53, no. 1, pp. 61–86, 2004.
- [137] T. Nomura, T. Suzuki, T. Kanda, and K. Kato, "Measurement of negative attitudes toward robots," *Interaction Studies*, vol. 7, no. 3, pp. 437–454, 2006.
- [138] D. S. Syrdal, K. Dautenhahn, K. L. Koay, and M. L. Walters, "The negative attitudes towards robots scale and reactions to robot behaviour in a live human-robot interaction study," *Adaptive and Emergent Behaviour and Complex Systems*, 2009.
- [139] K. M. Tsui, M. Desai, H. A. Yanco, H. Cramer, and N. Kemper, "Measuring attitudes towards telepresence robots," *International journal of intelligent Control and Systems*, vol. 16, 2011.
- [140] E. Broadbent, R. Stafford, and B. MacDonald, "Acceptance of healthcare robots for the older population: review and future directions," *International Journal of Social Robotics*, vol. 1, no. 4, pp. 319–330, 2009.
- [141] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International journal of social robotics*, vol. 1, no. 1, pp. 71–81, 2009.
- [142] K. M. Tsui, M. Desai, and H. A. Yanco, "Towards measuring the quality of interaction: communication through telepresence robots," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pp. 101–108, ACM, 2012.
- [143] International Telecommunication Union, "ITU-T Recommendation P.805: Subjective Evaluation of Conversational Quality," *Series P: Telephone Transmission Quality, Telephone Installations, Local Line Networks - Methods for objective and subjective assessment of quality*, 2007.

- [144] International Telecommunication Union, "ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications," *Series P: Telephone Transmission Quality, Telephone Installations, Local Line Networks - Audiovisual quality in multimedia services*, 2008.
- [145] B. G. Witmer, C. J. Jerome, and M. J. Singer, "The factor structure of the presence questionnaire," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 3, pp. 298–312, 2005.
- [146] S. Yarosh and P. Markopoulos, "Design of an instrument for the evaluation of communication technologies with children," in *Proceedings of the 9th International Conference on Interaction Design and Children*, pp. 266–269, ACM, 2010.
- [147] M. Lombard, T. B. Ditton, and L. Weinstein, "Measuring presence: the temple presence inventory," in *Proceedings of the 12th Annual International Workshop on Presence*, pp. 1–15, 2009.
- [148] A. Kristoffersson, K. S. Eklundh, and A. Loutfi, "Measuring the quality of interaction in mobile robotic telepresence: A pilot's perspective," *International Journal of Social Robotics*, vol. 5, no. 1, pp. 89–101, 2013.
- [149] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research*, vol. 10, no. 2, pp. 141–160, 2009.
- [150] H.-Q. Chong, A.-H. Tan, and G.-W. Ng, "Integrated cognitive architectures: a survey," *Artificial Intelligence Review*, vol. 28, no. 2, pp. 103–130, 2007.
- [151] R. C. Arkin, "Motor schema—based mobile robot navigation," *The International journal of robotics research*, vol. 8, no. 4, pp. 92–112, 1989.
- [152] R. Peter Bonasso, R. James Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, "Experiences with an architecture for intelligent, reactive agents," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 237–256, 1997.
- [153] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots," in *AAAI*, vol. 1992, pp. 809–815, 1992.
- [154] A. Newell, *Unified theories of cognition*. Harvard University Press, 1994.
- [155] D. E. Meyer and D. E. Kieras, "A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms.," *Psychological review*, vol. 104, no. 1, p. 3, 1997.
- [156] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: An architecture for general intelligence," *Artificial intelligence*, vol. 33, no. 1, pp. 1–64, 1987.
- [157] A. Guerra-Hernández, A. El Fallah-Seghrouchni, and H. Soldano, "Learning in BDI multi-agent systems," in *International Workshop on Computational Logic in Multi-Agent Systems*, pp. 218–233, Springer, 2004.
- [158] J. R. Anderson, M. Matessa, and C. Lebiere, "ACT-R: A theory of higher level cognition and its relation to visual attention," *Human-Computer Interaction*, vol. 12, no. 4, pp. 439–462, 1997.

- [159] G. Trafton, L. Hiatt, A. Harrison, F. Tamborello, S. Khemlani, and A. Schultz, "ACT-R/E: An Embodied Cognitive Architecture for Human-Robot Interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 30–55, 2013.
- [160] R. Simmons and J. Dale, "Inter Process Communication (IPC)." <http://www.cs.cmu.edu/~./IPC/>, November 1996. [Online; accessed on 2016-11-29].
- [161] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.
- [162] M. Mori, K. F. MacDorman, and N. Kageki, "The Uncanny Valley [From the Field]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 98–100, 2012.
- [163] C. Bartneck, T. Kanda, H. Ishiguro, and N. Hagita, "My robotic doppelgänger—a critical look at the uncanny valley," in *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 269–276, IEEE, 2009.
- [164] J. Kätsyri, K. Förger, M. Mäkräinen, and T. Takala, "A review of empirical evidence on different uncanny valley hypotheses: support for perceptual mismatch as one road to the valley of eeriness," *Frontiers in psychology*, vol. 6, p. 390, 2015.
- [165] S. M. Goza, R. O. Ambrose, M. A. Diftler, and I. M. Spain, "Telepresence control of the nasa/darpa robonaut on a mobility platform," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 623–629, 2004.
- [166] E. Short, K. Swift-Spong, J. Greczek, A. Ramachandran, A. Litoiu, E. C. Grigore, D. Feil-Seifer, S. Shuster, J. J. Lee, S. Huang, *et al.*, "How to train your dragonbot: Socially assistive robots for teaching children about nutrition through play," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 924–929, IEEE, 2014.
- [167] D. E. Logan, C. Breazeal, M. S. Goodwin, S. Jeong, B. O'Connell, D. Smith-Freedman, J. Heathers, and P. Weinstock, "Social robots for hospitalized children," *Pediatrics*, vol. 144, no. 1, p. e20181511, 2019.
- [168] I. Kotseruba, O. J. A. Gonzalez, and J. K. Tsotsos, "A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications," *arXiv preprint arXiv:1610.08602*, pp. 1–74, 2016.
- [169] P. Singh, "An architecture for cognitive diversity," in *Visions of mind: architectures for cognition and affect*, pp. 312–331, IGI Global, 2005.
- [170] W. C. Pang, Burhan, and G. Seet, "Design Considerations of a Robotic Head for Telepresence Applications," in *Intelligent Robotics and Applications (Lecture Notes in Computer Science)*, vol. 7508, pp. 131–140, Springer Berlin Heidelberg, 2012.
- [171] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, ACM, October 2011.

- [172] Aldebaran,
“Pepper Links Aldebaran 2.0.6.8 documentation.” http://doc.aldebaran.com/2-0/family/juliette_technical/links_juliette.html, November 2018. [Online; accessed on 2018-11-29].
- [173] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch, “All Together Now: Introducing the Virtual Human Toolkit,” in *Proceedings of the International Conference on Intelligent Virtual Agents*, pp. 368–381, Springer Berlin Heidelberg, 2013.
- [174] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. Sebastopol, California, USA: O’Reilly Media, Incorporated, 2008.
- [175] T. J. Soon, “Qr code,” *Synthesis Journal*, vol. 2008, pp. 59–78, 2008.
- [176] H. Zhang, C. Zhang, W. Yang, and C.-Y. Chen, “Localization and navigation using qr code for mobile robot in indoor environment,” in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2501–2506, IEEE, 2015.
- [177] R. Katsuki, J. Ota, Y. Tamura, T. Mizuta, T. Kito, T. Arai, T. Ueyama, and T. Nishiyama, “Handling of objects with marks by a robot,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 1, pp. 130–135, IEEE, 2003.
- [178] G. Michalos, P. Karagiannis, S. Makris, Ö. Tokçalar, and G. Chryssolouris, “Augmented reality (ar) applications for supporting human-robot interactive cooperation,” *Procedia CIRP*, vol. 41, pp. 370–375, 2016.
- [179] OpenCV, “OpenCV: Detection of ArUco Markers.” https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html, June 2018. [Online; accessed on 2019-06-28].
- [180] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [181] H. Kato, K. Tachibana, M. Billinghurst, and M. Grafe, “A registration method based on texture tracking using ARToolkit,” in *2003 IEEE International Augmented Reality Toolkit Workshop*, pp. 77–85, IEEE, 2003.
- [182] Ivan, Dryanovsk and Bill, Morris, “ar_pose - ROS Wiki.” http://wiki.ros.org/ar_pose, April 2016. [Online; accessed on 2019-07-23].
- [183] Jeff, Brown, “ZBar bar code reader.” <http://zbar.sourceforge.net/>, July 2011. [Online; accessed on 2017-07-12].
- [184] S. J. Julier and J. K. Uhlmann, “New extension of the kalman filter to nonlinear systems,” in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–194, International Society for Optics and Photonics, 1997.
- [185] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo Localization: Efficient Position Estimation for Mobile Robots,” in *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI’99)*, pp. 343–349, July 1999.

- [186] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [187] D. Fox, “Kld-sampling: Adaptive particle filters,” in *Advances in Neural Information Processing Systems*, vol. 14, pp. 713–720, 2002.
- [188] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. MIT press, 1992.
- [189] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2432–2437, IEEE, 2005.
- [190] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, “Efficient trajectory optimization using a sparse model,” in *2013 European Conference on Mobile Robots (ECMR)*, pp. 138–143, IEEE, 2013.
- [191] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.
- [192] W. C. Pang, G. Seet, and X. Yao, “A Multimodal Person-following System for Telepresence Applications,” in *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology (VRST)*, (Singapore), pp. 157–164, ACM, 2013.
- [193] W. C. Pang, G. Seet, and X. Yao, “A Study on High-level Autonomous Navigational Behaviors for Telepresence Applications,” *PRESENCE: Teleoperators and Virtual Environments*, vol. 23, no. 2, pp. 155–171, 2014.
- [194] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face Alignment at 3000 FPS via Regressing Local Binary Features,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1685–1692, 2014.
- [195] Blaise, Gassend, “audio_common - ROS Wiki.” http://wiki.ros.org/audio_common, January 2016. [Online; accessed on 2019-01-21].
- [196] A.L.I.C.E. AI Foundation, Inc., “AIML - The Artificial Intelligence Markup Language.” <http://www.alicebot.org/aiml.html>, January 2007. [Online; accessed on 2016-09-24].
- [197] R. Wallace, “The elements of AIML style,” *Alice AI Foundation*, 2003.
- [198] Ioan, Sucan and Jackie, Kay, “urdf - ROS Wiki.” <http://wiki.ros.org/urdf>, January 2019. [Online; accessed on 2019-09-09].
- [199] J. Bohren and S. Cousins, “The SMACH High-Level Executive [ROS News],” *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.
- [200] B. Sankaran, B. Pitzer, and S. Osentoski, “Failure recovery with shared autonomy,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 349–355, IEEE, 2012.
- [201] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, “Rosbridge: ROS for non-ros users,” in *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

- [202] A. B. Johnston and D. C. Burnett, *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012.
- [203] Google, “Speech-to-Text: Automatic Speech Recognition | Google Cloud.” <https://cloud.google.com/speech-to-text>, June 2019. [Online; accessed on 2019-06-19].
- [204] Microsoft, “System.Speech.Recognition Namespace | Microsoft Docs.” <https://docs.microsoft.com/en-us/dotnet/api/system.speech.recognition?view=netframework-4.8>, June 2019. [Online; accessed on 2019-06-18].
- [205] N. T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, “A first speech recognition system for mandarin-english code-switch conversational speech,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4889–4892, IEEE, 2012.
- [206] M. S. Young and N. A. Stanton, “Taking the load off: investigations of how adaptive cruise control affects mental workload,” *Ergonomics*, vol. 47, pp. 1014–1035, 2004.
- [207] Audrey, Tan, “Robot to host this year’s NDP.” <https://www.straitstimes.com/singapore/heres-your-ndp-host-edgar-the-robot>, 11 July 2017. *The Straits Times*. [Online; accessed on 2019-11-21].
- [208] K. M. Speaker, “The art of storytelling: A collegiate connection to professional development schools,” *Education*, vol. 121, no. 1, pp. 184–184, 2000.
- [209] J. C. Nunnally, *Psychometric Theory 2nd ed.* Mcgraw hill book company, 1978.
- [210] D. Pagliari and L. Pinto, “Calibration of Kinect for XBox One and comparison between the two generations of Microsoft sensors,” *Sensors*, vol. 15, no. 11, pp. 27569–27589, 2015.

APPENDIX A

KINECT FOR WINDOWS SENSOR

A Kinect RGB-D camera, also known as the Kinect for Windows Sensor, is a marker-less human tracking device that has been used in many modules of the ICS as well as on the robots. The sensor comprises a RGB (red-green-blue) camera as well as an infrared (IR) emitter and camera that measure depth in millimeter resolution. Therefore, it can be used to capture color and depth images as well as to detect and track human bodies, within a "sweet spot" of 0.8m to 3.5m from the sensor.

Lastly, the Kinect sensor has an array of 4 microphones and the SDK has functions for noise cancellation as well as sound localization. It is able to detect audio input from + to - 50 degrees in front of the sensor, as shown in Figure. By using the SDK, the microphone array can be used to point at a specific direction of 5-degree increments within the 180-degree range. The microphone array cannot remove other ambient noise completely, but it is able to cancel 20 decibels of ambient noise, which improves audio fidelity. The audio specifications of the Kinect sensor are as summarized in Figure A.1.

A.1 Comparison of two generations of Kinect

There are two versions of Kinect Sensor, as shown in Figure A.2 and we have used both versions in the implementation of the framework. The first version is Kinect for Xbox 360 (Kinect v1) sensor, which measures distance based on structured light, while the second version, Kinect Xbox One (Kinect v2) sensor, is based on time-of-flight principle. Kinect v2

Figure A.1: Kinect Audio Specifications

has a wider field-of-view (FOV) and is able to produce RGB images of a higher resolution. Table A.1 tabulates some of the basic characteristics of the two Kinect sensors. A detailed comparison of the two sensors can be found in [210].

(a) Kinect Xbox 360 sensor (Kinect v1)

(b) Kinect Xbox One sensor (Kinect v2)

Figure A.2: Two Generations of Kinect sensor

	Kinect v1	Kinect v2
Operating System	Windows 7/8/8.1	Windows 8/8.1/10
Processor Type	32-bit (x86)/64-bit (x64)	64-bit (x64)
Min Processor Speed	Dual-core 2.66-GHz	Dual-core 3.1 GHz
Min RAM Size	2G	4G
Minimum Hard Drive Space	230 MB	280 MB
Software Requirement	Kinect for Windows SDK v1.8	Kinect for Windows SDK 2.0
Connectivity Technology	Wired	Wired, including a power hub
Interface	USB 2.0	USB 3.0
Resolution of RGB image	1280×1024 pixels	1920×1080 pixels
Horizontal FOV	57°	70°
Vertical FOV	43°	60°
Skeleton joints define	20	26
Full skeleton tracking	2	6

Table A.1: Characteristics of Kinect v1 and Kinect v2 Sensors

A.2 Skeletal Data of Kinect v1 and Kinect v2

Figure A.3 illustrates the joint tracking map of Kinect v1 and Kinect v2, while Table A.2 shows the skeletal joints names and index of Kinect v1 and Kinect v2.

Figure A.3: Body joints that are tracked by Kinect v1 and Kinect v2

Joint ID	Kinect v1	Kinect v2	Joint ID	Kinect v1	Kinect v2
0	HipCenter	SpineBase	13	KneeLeft	KneeLeft
1	Spine	SpineMid	14	AnkleLeft	AnkleLeft
2	ShoulderCenter	Neck	15	FootLeft	FootLeft
3	Head	Head	16	HipRight	HipRight
4	ShoulderLeft	ShoulderLeft	17	KneeRight	KneeRight
5	ElbowLeft	ElbowLeft	18	AnkleRight	AnkleRight
6	WristLeft	WristLeft	19	FootRight	FootRight
7	HandLeft	HandLeft	20		SpineShoulder
8	ShoulderRight	ShoulderRight	21		HandTipLeft
9	ElbowRight	ElbowRight	22		ThumbLeft
10	WristRight	WristRight	23		HandTipRight
11	HandRight	HandRight	24		ThumbRight
12	HipLeft	HipLeft			

Table A.2: Skeletal Data of Kinect v1 and Kinect v2 Sensors

APPENDIX B

NASA-TASK LOAD INDEX (TLX)

Dimension	Meaning of the dimension
Mental Demand	How much mental and perceptual activity (e.g. thinking, deciding, searching, etc.) was required? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand	How much physical activity was required ? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand	How much time pressure did you feel? Was the pace at which the task occurred slow and leisurely or rapid and frantic?
Performance	How successful do you think you were in accomplishing the goals of the task? How satisfied were you with your performance in accomplishing these goals?
Effort	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Level	How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

Table B.1: Definitions of the NASA-TLX dimensions [9]

APPENDIX C

MANN-WHITNEY U TEST CALCULATIONS

An autonomous navigation system has been implemented for a mobile robotic avatar. This study seeks to address the objectives of equipping the robotic avatar with autonomy in navigation. In Chapter 5, experiments have been carried out to evaluate the following hypotheses:

Hypothesis-1 The addition of fully-autonomous navigational behaviors to robotic avatar helps to ensure safety and better performance (less time taken) during the navigation of the robotic avatar in a remote environment.

Hypothesis-2 The addition of fully-autonomous navigational behaviors to robotic avatar helps to reduce the inhabitant's cognitive workload.

The analysis of experiment will ascertain if employing the conventional manual-controlled robotic avatar will induce more workload and result in degraded navigation performance.

As the sample sizes are small, a Mann-Whitney U test is used to determine if there is a statistically significant difference in performance between the two groups. We want to know if robot autonomy leads to a change in the performance of the robotic avatar. Given that n_1 and n_2 are the sample size for sample 1 and sample 2 respectively and R_1 is the sum of the ranks in sample 1, the Mann-Whitney U formula (U) is the $\min(U_1, U_2)$ where

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \text{ and } U_2 = n_1 n_2 - U_1$$

C.1 Measurement of navigation performance

Mann-Whitney U test has been used to determine if there is a difference in time taken to complete the task of navigating and also collision between the two modes of robot control. Assuming critical level of significance at $\alpha = 0.05$, the null hypothesis (H_0) states that robot autonomy does not perform better than manual control, as both have the same distribution

of scores. the alternative hypothesis (H_a) states that robot autonomy performs better than manual control. H_0 is rejected if the calculated Mann-Whitney U-value is less than the critical U value.

C.1.1 Mann-Whitney U Test for Completion Time

Taken from Table 5.4,

Manual Mode	Autonomous Mode	Manual		Autonomous	
		Sorted Values	Ranks	Sorted Values	Ranks
215	188.5	134.5	4	104.5	1
208.5	175.5	141	6	119	2
234.5	186	143.5	7.5	120.5	3
180.5	171.5	145	9	136	5
177	171	177	14	143.5	7.5
188	178	180.5	16	147.5	10
145	143.5	188	18	171	11
143.5	120.5	194	20	171.5	12
238	147.5	208.5	21	175.5	13
134.5	119	215	22	178	15
141	104.5	234.5	23	186	17
194	136	238	24	188.5	19

SUMMARY

Groups	Count	Median	Sum of ranks	Mean of ranks	U-value	U
Manual	12	184.3	184.5	15.38	37.5	37.5
Autonomous	12	159.3	115.5	9.62	106.5	

Conclusion

The obtained Mann-Whitney U -value is 37.5. The critical value of U at ($p < .05, n_1 = n_2 = 12$, one-tailed) is 42. Since the obtained U -value is smaller than the critical U -value, the null hypothesis is rejected. Therefore, at 5% level of significance, the data provide sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving faster completion time. This supports our hypothesis-1.

C.1.2 Mann-Whitney U Test for Number of Collisions

Taken from Table 5.4,

Manual Mode	Autonomous Mode	Manual		Autonomous	
		Sorted Values	Ranks	Sorted Values	Ranks
6	0	0	7	0	7
5	0	1	15.5	0	7
3	0	1	15.5	0	7
3	0	1	15.5	0	7
2	0	1	15.5	0	7
5	0	2	18.5	0	7
2	0	2	18.5	0	7
1	0	3	20.5	0	7
1	0	3	20.5	0	7
0	0	5	22.5	0	7
1	0	5	22.5	0	7
1	0	6	24	0	7

SUMMARY

Groups	Count	Median	Sum of ranks	Mean of ranks	U-value	U
Manual	12	2.0	216	18	6	6
Autonomous	12	0.0	84	7	138	

Conclusion

The obtained Mann–Whitney U -value is 6. The critical value of U at ($p < .05, n_1 = n_2 = 12$, one-tailed) is 42. Since the obtained U -value is smaller than the critical U -value, the null hypothesis is rejected. Therefore, at 5% level of significance, the data provide sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving lesser number of collisions. This supports our hypothesis-1.

C.2 Measurement of Workload (NASA-TLX Index)

Taken from Table 5.4,

Manual Mode	Autonomous Mode	Manual		Autonomous	
		Sorted Values	Ranks	Sorted Values	Ranks
51.25	52.92	50	9	36.53	1
72.92	65.42	51	10	36.66	2
67.92	62.92	51	11	37.54	3
54.17	46.67	54	13	42.82	4
77.83	75.42	54	14	44.67	5
57.82	46.09	55	15	46.09	6
54.58	46.67	55	16	46.67	7.5
55.49	36.53	58	17	46.67	7.5
53.81	36.66	60	18	52.92	12
60	44.67	68	21	62.92	19
49.5	37.54	73	22	65.42	20
51.12	42.82	78	24	75.42	23

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Median</i>	<i>Sum of ranks</i>	<i>Mean of ranks</i>	<i>U-value</i>	<i>U</i>
Manual	12	55.0	190	15.8333	32	32
Autonomous	12	46.4	110	9.1667	112	

Conclusion

The obtained Mann–Whitney U -value is 32. The critical value of U at ($p < .05, n_1 = n_2 = 12$, one-tailed) is 42. Since the obtained U -value is smaller than the critical U -value, the null hypothesis is rejected. Therefore, there is sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving lower workload. This supports our hypothesis-2.

Critical Values of the Mann-Whitney U (One-tailed Testing)

n ₂	α	n ₁																	
		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	.05	0	0	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10	11
	.01	--	0	0	0	0	0	1	1	1	2	2	2	3	3	4	4	4	5
4	.05	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18
	.01	--	--	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	10
5	.05	1	2	4	5	6	8	9	11	12	13	15	16	18	19	20	22	23	25
	.01	--	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
6	.05	2	3	5	7	8	10	12	14	16	17	19	21	23	25	26	28	30	32
	.01	--	1	2	3	4	6	7	8	9	11	12	13	15	16	18	19	20	22
7	.05	2	4	6	8	11	13	15	17	19	21	24	26	28	30	33	35	37	39
	.01	0	1	3	4	6	7	9	11	12	14	16	17	19	21	23	24	26	28
8	.05	3	5	8	10	13	15	18	20	23	26	28	31	33	36	39	41	44	47
	.01	0	2	4	6	7	9	11	13	15	17	20	22	24	26	28	30	32	34
9	.05	4	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54
	.01	1	3	5	7	9	11	14	16	18	21	23	26	28	31	33	36	38	40
10	.05	4	7	11	14	17	20	24	27	31	34	37	41	44	48	51	55	58	62
	.01	1	3	6	8	11	13	16	19	22	24	27	30	33	36	38	41	44	47
11	.05	5	8	12	16	19	23	27	31	34	38	42	46	50	54	57	61	65	69
	.01	1	4	7	9	12	15	18	22	25	28	31	34	37	41	44	47	50	53
12	.05	5	9	13	17	21	26	30	34	38	42	47	51	55	60	64	68	72	77
	.01	2	5	8	11	14	17	21	24	28	31	35	38	42	46	49	53	56	60
13	.05	6	10	15	19	24	28	33	37	42	47	51	56	61	65	70	75	80	84
	.01	2	5	9	12	16	20	23	27	31	35	39	43	47	51	55	59	63	67
14	.05	7	11	16	21	26	31	36	41	46	51	56	61	66	71	77	82	87	92
	.01	2	6	10	13	17	22	26	30	34	38	43	47	51	56	60	65	69	73
15	.05	7	12	18	23	28	33	39	44	50	55	61	66	72	77	83	88	94	100
	.01	3	7	11	15	19	24	28	33	37	42	47	51	56	61	66	70	75	80
16	.05	8	14	19	25	30	36	42	48	54	60	65	71	77	83	89	95	101	107
	.01	3	7	12	16	21	26	31	36	41	46	51	56	61	66	71	76	82	87
17	.05	9	15	20	26	33	39	45	51	57	64	70	77	83	89	96	102	109	115
	.01	4	8	13	18	23	28	33	38	44	49	55	60	66	71	77	82	88	93
18	.05	9	16	22	28	35	41	48	55	61	68	75	82	88	95	102	109	116	123
	.01	4	9	14	19	24	30	36	41	47	53	59	65	70	76	82	88	94	100
19	.05	10	17	23	30	37	44	51	58	65	72	80	87	94	101	109	116	123	130
	.01	4	9	15	20	26	32	38	44	50	56	63	69	75	82	88	94	101	107
20	.05	11	18	25	32	39	47	54	62	69	77	84	92	100	107	115	123	130	138
	.01	5	10	16	22	28	34	40	47	53	60	67	73	80	87	93	100	107	114

APPENDIX D

KRUSKAL-WALLIS TEST CALCULATIONS

Kruskal-Wallis non parametric hypothesis test has been used to determine whether the medians of the two modes of robot control (manual and robot autonomy) are different. Assuming critical level of significance at $\alpha = 0.05$, the null hypothesis (H_0) states that robot autonomy does not perform better than manual control, as both have the same distribution of scores. the alternative hypothesis (H_a) states that robot autonomy performs better than manual control. H_0 is rejected if the calculated Kruskal-Wallis test statistic is greater than the critical U value.

D.1 Kruskal-Wallis Test (Completion Time)

Taken from [C.1.1](#),

SUMMARY				
<i>Groups</i>	<i>Count (n)</i>	<i>Median</i>	<i>Sum of Ranks (R)</i>	<i>R²/n</i>
Manual	12	184.3	184.5	2836.688
Autonomous	12	159.3	115.5	1111.688

At $N = 24$, the Kruskal-Wallis test statistic is

$$H = \frac{12}{N(N+1)} \sum \frac{R_i^2}{n_i} - 3(N+1)$$

$$H = 0.02 \times 3948.375 - 75 = 3.9675$$

The calculated H -value is 3.9675, while for a right tailed chi-square test with 95% confidence level, and $df = 1$, critical χ^2 value is 3.841. Since the Kruskal-Wallis test statistic is greater than the critical value of χ^2 value, the null hypothesis is rejected. Therefore, at 5% level of significance, the data provide sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving faster completion time.

D.2 Kruskal-Wallis Test (Number of Collision)

Taken from C.1.2,

SUMMARY

<i>Groups</i>	<i>Count (n)</i>	<i>Median</i>	<i>Sum of Ranks (R)</i>	<i>R²/n</i>
Manual	12	2.0	216	3888
Autonomous	12	0.0	84	588

At $N = 24$, the Kruskal-Wallis test statistic is

$$H = \frac{12}{N(N+1)} \sum \frac{R_i^2}{n_i} - 3(N+1)$$

$$H = 0.02 \times 4476 - 75 = 14.52$$

The calculated H -value is 14.52, while for a right tailed chi-square test with 95% confidence level, and $df = 1$, critical χ^2 value is 3.841. Since the Kruskal-Wallis test statistic is greater than the critical value of χ^2 value, the null hypothesis is rejected. Therefore, at 5% level of significance, the data provide sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving lesser number of collisions.

D.3 Kruskal-Wallis Test (Workload)

Taken from C.2,

SUMMARY

<i>Groups</i>	<i>Count (n)</i>	<i>Median</i>	<i>Sum of Ranks (R)</i>	<i>R²/n</i>
Manual	12	55.0	190	3008.333
Autonomous	12	46.4	110	1008.333

At $N = 24$, the Kruskal-Wallis test statistic is

$$H = \frac{12}{N(N+1)} \sum \frac{R_i^2}{n_i} - 3(N+1)$$

$$H = 0.02 \times 4016.667 - 75 = 5.333$$

The calculated H -value is 5.333, while for a right tailed chi-square test with 95% confidence level, and $df = 1$, critical χ^2 value is 3.841. Since the Kruskal-Wallis test statistic is greater than the critical value of χ^2 value, the null hypothesis is rejected. Therefore, at 5% level of significance, the data provide sufficient evidence to show that the autonomous control mode performs better than manual control mode in achieving lower workload.

Chi-Square Distribution Table

Degrees of Freedom	Area to the Right of Critical Value									
	0.995	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01	0.005
1	—	—	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.071	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.299
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.042	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.194	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.257	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.954	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

APPENDIX E

USABILITY QUESTIONNAIRE FOR REMOTE USER

Perceived Ease of Use		Rating
1	I think it was easy to start the EDGAR Telepresence Application	
2	I think it was easy to control the robot	
3	I think it was easy to hang up after making a call	
4	I think it was easy to hear what the person I called said	
5	I think it was easy to see the person I called	
6	I think it was easy to hold a conversation with the person while controlling the robot to gesture	
7	I think it was easy to understand the application and the instruction to use it	
8	Overall, I found the EDGAR Telepresence Application easy to use for the purpose of communicating with another person	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Self Efficacy		Rating
1	I found the application unnecessarily complex	
2	I can learn to use the application very quickly	
3	I don't need to learn a lot of things before I could get going with the application	
4	I think that i would need the support of a technical person to use EDGAR robot	
5	I felt very confident using the EDGAR Telepresence Application	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Perceived Sociability		Rating
1	The EDGAR robot is well designed for social interaction	
2	I found that the audio/video lag in connection did not affect the quality of communication	
3	The size of EDGAR robot is suitable for telepresence and making social interaction with the interactants and other applications	
4	The noise when moving is negligible	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Perceived Usefulness		Rating
1	Using the EDGAR Telepresence Application improves the quality of communication	
2	Using the EDGAR Telepresence Application gives me greater control over my communication	
3	The EDGAR Telepresence Application increases the effectiveness of communication	
4	Overall, I found the EDGAR Telepresence Application useful in communicating with others	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Feel of Presence		Rating
1	I could see the changes in the facial expressions of the interactants	
2	I could hear the changes in the tone of voice in the interactants	
3	I could see the body language of the interactants	
4	I could see the eye movements of the interactants	
5	I have a greater sense of closeness when having conversation with the interactants	
6	Overall, I felt being there with the interactants	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Affordable Cost		Rating
	Please indicate the price range that you would be willing to pay for this device	
1	Less than \$3000	
2	\$3000 to less than \$6000	
3	\$6000 to less than \$9000	
4	\$9000 or more	

Reliability		Rating
1	I found the various functions (call, movement, video and audio system) in the EDGAR Telepresence Application were well integrated	
2	The battery duration is adequate for my usage with the EDGAR robot	
3	Overall, I found the EDGAR Telepresence Application free of errors.	

4	How many errors did you encounter when using the EDGAR Telepresence Application?	
	If you did experience errors, please describe in detail.	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

Overall satisfaction		Rating
1	I think that I would use the EDGAR Telepresence Application frequently	
2	Overall, I am satisfied with using the EDGAR Telepresence Application	
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)		

APPENDIX F

USABILITY QUESTIONNAIRE FOR REMOTE USER (COMPLETED BY TWO USERS)

Perceived Ease of Use		P1	P2
1	I think it was easy to start the EDGAR Telepresence Application	4	2
2	I think it was easy to control the robot	4	4
3	I think it was easy to hang up after making a call	4	5
4	I think it was easy to hear what the person I called said	4	5
5	I think it was easy to see the person I called	4	5
6	I think it was easy to hold a conversation with the person while controlling the robot to gesture	4	4
7	I think it was easy to understand the application and the instruction to use it	4	4
8	Overall, I found the EDGAR Telepresence Application easy to use for the purpose of communicating with another person	4	5
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Self Efficacy		P1	P2
1	I found the application unnecessarily complex	2	1
2	I can learn to use the application very quickly	3	4
3	I don't need to learn a lot of things before I could get going with the application	5	4
4	I think that i would need the support of a technical person to use EDGAR robot	4	3
5	I felt very confident using the EDGAR Telepresence Application	4	4
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Perceived Sociability		P1	P2
1	The EDGAR robot is well designed for social interaction	4	5
2	I found that the audio/video lag in connection did not affect the quality of communication	3	3
3	The size of EDGAR robot is suitable for telepresence and making social interaction with the interactants and other applications	4	4
4	The noise when moving is negligible	4	5
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Perceived Usefulness		P1	P2
1	Using the EDGAR Telepresence Application improves the quality of communication	3	5
2	Using the EDGAR Telepresence Application gives me greater control over my communication	4	4
3	The EDGAR Telepresence Application increases the effectiveness of communication	3	4
4	Overall, I found the EDGAR Telepresence Application useful in communicating with others	4	4
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Feel of Presence		P1	P2
1	I could see the changes in the facial expressions of the interactants	4	4
2	I could hear the changes in the tone of voice in the interactants	4	5
3	I could see the body language of the interactants	4	4
4	I could see the eye movements of the interactants	4	3
5	I have a greater sense of closeness when having conversation with the interactants	3	4
6	Overall, I felt being there with the interactants	4	4
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Affordable Cost		P1	P2
	Please indicate the price range that you would be willing to pay for this device		
1	Less than \$3000		
2	\$3000 to less than \$6000		
3	\$6000 to less than \$9000		
4	\$9000 or more	×	×

Reliability		P1	P2
1	I found the various functions (call, movement, video and audio system) in the EDGAR Telepresence Application were well integrated	3	2
2	The battery duration is adequate for my usage with the EDGAR robot	5	5
3	Overall, I found the EDGAR Telepresence Application free of errors.	5	5

4	How many errors did you encounter when using the EDGAR Telepresence Application?	0	0
	If you did experience errors, please describe in detail.		
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

Overall satisfaction		P1	P2
1	I think that I would use the EDGAR Telepresence Application frequently	4	4
2	Overall, I am satisfied with using the EDGAR Telepresence Application	4	5
RATING from 1 to 5 : strongly agree (5) — agree (4) — neutral (3) — disagree (2) — strongly disagree (1)			

APPENDIX G

GODSPEED QUESTIONNAIRE

Anthropomorphism						
Please rate your impression of Edgar on these scales:						
Fake	1	2	3	4	5	Natural
Machinelike	1	2	3	4	5	Humanlike
Unconscious	1	2	3	4	5	Conscious
Artificial	1	2	3	4	5	Lifelike
Moving rigidly	1	2	3	4	5	Moving elegantly
Animacy						
Please rate your impression of Edgar on these scales:						
Dead	1	2	3	4	5	Alive
Stagnant	1	2	3	4	5	Lively
Mechanical	1	2	3	4	5	Organic
Artificial	1	2	3	4	5	Lifelike
Inert	1	2	3	4	5	Interactive
Apathetic	1	2	3	4	5	Responsive
Likeability						
Please rate your impression of Edgar on these scales:						
Dislike	1	2	3	4	5	Like
Unfriendly	1	2	3	4	5	Friendly
Unkind	1	2	3	4	5	Kind
Unpleasant	1	2	3	4	5	Pleasant
Aweful	1	2	3	4	5	Nice
Perceived Intelligence						
Please rate your impression of Edgar on these scales:						
Incompetent	1	2	3	4	5	Competent
Ignorant	1	2	3	4	5	Knowledgeable
Irresponsible	1	2	3	4	5	Responsible
Unintelligent	1	2	3	4	5	Intelligent
Foolish	1	2	3	4	5	Sensible
Perceived Safety						
Please rate how you felt when using the robot on these scales at the beginning:						
Anxious	1	2	3	4	5	Relaxed
Agitated	1	2	3	4	5	Calm

Quiescent	1	2	3	4	5	Surprised
Please rate on these scales how you felt towards the end:						
Anxious	1	2	3	4	5	Relaxed
Agitated	1	2	3	4	5	Calm
Quiescent	1	2	3	4	5	Surprised
Please indicate what Edgar was for you						
(Put a tick in the table beside the word that fits best):						
Brother				Friend		
Classmate				Parent		
Stranger				Teacher		
Relative				Neighbor		
Demographic Questions						
Gender : F / M	Age:		Have you used a robot before? Y / N			