HUMAN-ROBOT AND MULTI-AUTONOMOUS AGENT COLLABORATIONS IN CYBER-PHYSICAL ENVIRONMENTS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Jovan Clive Menezes

December 2023

ABSTRACT

The focus of robotics, mechanical engineering, and computer science research on human-robot teams requires evaluating software and algorithms in complex real-world scenarios that are challenging to replicate in laboratory experiments/environments. Conversely, conducting field experiments in natural settings lacks the necessary level of detailed information for comparing and validating performance. Additionally, using pre-recorded real-world datasets has limitations in assessing the effectiveness of perception, control, and decision strategies. Moreover, the cost and logistical challenges of involving a large number of humans or robots in experiments make it impractical, especially when factors such as environmental conditions disrupt testing. To address these issues, this work presents a cyber-physical framework that serves as a testbed for conducting such research. The presented framework combines humans alongside virtual and real robots in simulated photo-realistic environments using motion capture technology, virtual reality (VR), wearable sensors, and physics-based simulations for the robot platforms. This creates an extended reality (XR) testbed where humans and real robots can experience virtual worlds with real-time visual feedback and interaction. The movements and actions made by the real human/robot agents are transferred from the physical world or laboratory setting to a synthetic virtual environment using VR coupled with 3D body tracking and motion capture systems. This process generates avatars that replicate the behavior of real agents in real-time and enable them to receive feedback from the virtual world. Synthetic environments created such that they narrow the gap between reality and simulation, allowing the inclusion of autonomous agents

with multi-modal sensor suites. The potential of the framework is demonstrated through three experiments which showcase interactions between agents in different domains, leveraging the advantages of both real-world and simulation experimentation to complement and enhance each other.

BIOGRAPHICAL SKETCH

Jovan Menezes is currently pursuing a Master of Science degree majoring in Mechanical Engineering with minors in Computer Science and Electrical and Computer Engineering in the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University. In 2019, he obtained his Bachelor of Engineering degree in Mechanical Engineering from Mumbai University (MU). Prior to commencing his graduate studies at Cornell, Jovan gained valuable industry experience as a Design Engineer at Petrofac Engineering India Pvt. Ltd. His research focuses on areas including deep learning-based perception algorithms, nonlinear control theory, and the utilization of augmented reality for the development and evaluation of intelligent and autonomous robots.

This document is dedicated to my family and friends for their unconditional support and encouragement.

ACKNOWLEDGEMENTS

I would like to extend my heartfelt appreciation to Dr. Silvia Ferrari, my advisor, for her exceptional guidance and unwavering support throughout my tenure as a Master's student at Cornell University. It has been a tremendous honor to conduct my thesis research under her expert mentorship. I am also deeply grateful to Dr. Nils Napp and Dr. Bharath Hariharan, my esteemed committee members, for their invaluable insights and advice on my research endeavors.

I would like to express my profound gratitude to Andre Paradise and Sushrut Surve, PhD students at (LISC), for their invaluable mentorship and constructive feedback that greatly contributed to the success of my research. I am immensely thankful to every member of our esteemed LISC team, whose unwavering support and camaraderie created a welcoming and supportive laboratory environment that felt like a true community.

Furthermore, I would like to extend my thanks to my friends at Cornell University and beyond for their constant encouragement and support. I am eternally grateful to my family, especially my parents, for their unconditional love, emotional support, and constant encouragement throughout this journey. I also want to express my appreciation to my lively brother, whose vibrant spirit has been a constant source of inspiration. My family's guidance and support has been instrumental in my achievements and the completion of this transformative journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

Simulation systems have played a crucial role in the advancement of robotic vehicles for a considerable period. These systems not only enable researchers to swiftly prototype and showcase their concepts, but also provide engineers with the ability to detect and address errors in the early stages of development. Despite the effectiveness of these simulators in expediting the development process, a level of skepticism continues to be maintained by some researchers towards outcomes produced within these simulation systems. This skepticism arises due to the inherent nature of simulation systems, which inevitably deviates from reality to some extent, as they are abstractions of the real world. The skepticism surrounding results solely derived from simulation studies is encapsulated in Rodney Brooks' famous quote from 1993 which essentially states that experiments conducted in simulations are doomed to succeed primarily since they can never be made sufficiently realistic [4].

In spite of the skepticism surrounding simulation results, the research community has witnessed the emergence of several trends in recent years that have compelled researchers to enhance simulation systems for a variety of reasons. One significant trend driving the development of more realistic simulators is the rise of data-driven based algorithms in the domain of robotics, particularly the approaches that are based on methods of machine learning (ML) which necessitate vast amounts of data. Simulation systems, while generating abundant data, also label this data which is essential for training these ML-based algo-

rithms. They serve as a valuable resource by providing a safe and controlled environment for training reinforcement learning methods as demonstrated in [38]. The growing prominence of data-driven algorithmic methods has created a pressing need for the advancement of simulation systems.

Researchers recognize that in order to generate accurate and reliable results, simulations must closely mimic the intricacies of the real world. As a result, there is a concerted effort to develop simulation systems that accurately capture the complexities of the physical environment, encompassing factors such as sensor noise, environmental variability, and complex interactions between agents and their surroundings. This driving trend has propelled the research community to explore innovative approaches and techniques to improve the realism and fidelity of simulation systems. The focus is on developing better models, incorporating more accurate physics-based simulations, and refining the representation of objects and agents in the virtual environment. By aligning simulations more closely with reality, researchers can generate data that is not only extensive but also more representative of real-world scenarios, thereby enhancing the effectiveness of data-driven algorithms.

1.2 Literature Review

Recent years have witnessed the emergence of several enabling trends that have facilitated the development of more realistic and enhanced simulation systems. The primary trend responsible for this growth is the advancement in computing resources that enable realistic rendering. The rapid progress specifically in 3D graphics rendering engines and in game engine technology as a whole has un-

locked access to advanced features like real-time reflections, improved material characteristics, sophisticated illumination, and volumetric lighting through deferred rendering pipelines. Notably, commercially available software packages such as Unity [39] and Unreal Engine (UE™) [14] have matured to the point where they can be used for rendering high-fidelity environments in applications over and above video games, including robotics simulation. Additionally, through optimized configurations for real-time ray tracing along with greater transistor density, the latest generation of graphics processors have significantly improved capabilities. Also, the computation cores incorporated in these processors leverage machine learning techniques, such as training with real environment images to generate realistic renderings [7]. This trend presents an opportunity to leverage improved hardware and software resources that achieve more realistic sensor simulations.

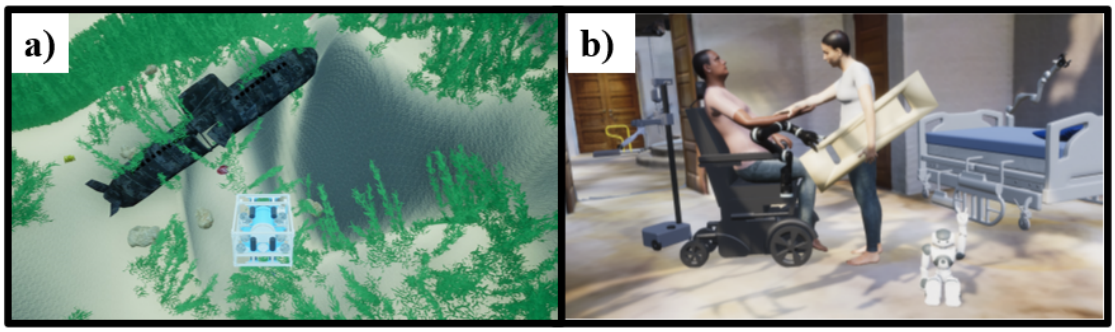

Figure 1.1: Examples of simulators developed using 3D graphics rendering game engines. a) HoloOcean [28] developed using UE™ for underwater robotics. b) RCareWorld [45] developed using Unity™ for caregiving robotics.

The widespread adoption of motion capture facilities in robotics research can be credited as the secondary enabling trend for the growth of these advanced simulation systems. These facilities utilize various technologies, including laser tracking, infrared cameras, and ultra-wideband radio, to enable precise tracking

of both agents—humans as well as robots. This trend offers the opportunity to combine the safety, efficiency, and flexibility of simulation with the physical dynamics and agent behavior observed in the real world by integrating real-time behavior and motion of robots and humans into the test environemnt. This integration of motion capture technologies empowers researchers to simulate the actions and responses of robotic vehicles and humans with greater authenticity and realism, resulting in more accurate and representative simulation outcomes.

Traditionally, simulation systems have utilized physics-based "models" using ordinary or partial differential equations to replicate the behavior of the various agents and their environments, encompassing their sensory inputs, movement, and environmental adaptations. This research employs the concept of VR system to incorporate a variety of features ranging from human behavior or agent dynamics to inertial sensors in a realistic fashion into the simulation environment; thus leveraging data to enhance the realism of simulations. Instead of modeling these effects, the robot agents are placed within motion-capture facilities. By capturing the real-time poses and configurations of these agents, corresponding avatars are created within the simulation environment. The proprioceptive measurements for each autonomous robot is acquired through the use of on-board sensors such as odometers and inertial measurement units, while the exteroceptive sensors are rendered photorealistically in real-time. Additionally, the human behavior observed by the vehicles is generated by real humans reacting to the simulation through the use of VR. In essence, the vehicles operating within the cyber-physical testbed experience human behavior, genuine dynamics, photorealistic synthetic exteroceptive sensor measurements, and inertial sensing.

Conventional systems simulate various agents and their environment typically through a physics engine that uses a model. While the behavior of a general vehicle or actor may be accurately represented by these models, they are inadequate for ensuring simulation results are effectively translated to the real world. Complex factors such as human behavior and vehicle dynamics (e.g., vibrations and unsteady aerodynamics) can have a significant impact on results, but are difficult to be captured accurately within a physics-based model. Popular physics engines, commonly implemented in recent research, are described in [10]. Robotics simulators utilize a rendering engine for graphics in conjunction with a physics engine to generate exteroceptive sensor data. Gazebo, often combined with the Robot Operating System (ROS) to facilitate hardware-in-the-loop simulation, is one of the most commonly used simulation platforms which allows users to choose from various underlying engines [20], [23]. However, the Gazebo simulator generally lacks the ability to render photorealistic scenes. For unmanned aerial vehicle simulation, two popular simulators built on the Gazebo platform are the RotorS [12] and the Hector Quadrotor package [24]. While these simulators include vehicle dynamics and exteroceptive sensor models, they do not have the potential to render photorealistic camera streams. On the other hand, AI2-Thors [21] and Habitat AI [31] generate environments that are highly photorealistic by using photogrammetry but primarily focus on indoor mobile robotic platforms or agents with simpler dynamics models. AirSim allows for rendering photorealistic camera streams from autonomous robots and is designed specifically on the Unreal rendering game engine [33]. However, when it comes to robot dynamics and inertial measurements, it still faces limitations inherent to typical physics engines [22]. Flightmare [36] addresses these limitations by enabling users the to flexibly utilize custom physics engines

or real-world flight data to overcome these shortcomings.

The emergence of data-driven algorithms in autonomous robotics has created a demand for extensive labeled datasets. Simulation presents an alternative to gathering experimental data, offering numerous advantages such as cost efficiency, safety, repeatability, and the ability to generate an essentially unlimited diversity as well as quantity of data. In recent years, various synthetic or virtual datasets have been introduced in the literature. For example, datasets like Virtual KITTI [13] and Synthia [30] utilize Unity for photorealistic rendering of urban environments, while others [42] leverage the UE™ platform to generate data for visual inertial odometry. Similar techniques using rendering engines have been employed in different visual SLAM datasets, where the quality of rendered environments is enhanced by the use of point cloud data or 3D images [42], [37]. A large-scale point cloud is used to generate a fine-grained dataset in the Stanford BuildingParser [2] while Matterport3D [6], for instance, utilizes the Matterport 3D camera sensor [32]. ICL-NUIM [18] provides synthetic renderings of indoor environments based on pre-recorded handheld trajectories. Real-world ground truth and inertial measurements of a quadcopter captured by motion capture technology are incorporated in the Blackbird Dataset [1], along with photorealistic camera imagery rendered in FlightGoggles (shown in figure 1.2). The availability of FlightGoggles [17] as an open-source platform, along with its photorealistic assets, allows users to not only easily generate additional data but also includes real-time photorealistic renders based on real-world vehicles and actors.

Figure 1.2: FlightGoggles is a high-fidelity simulator developed primarily for UAVs (highlighted in the above images) with limited capabilities for human-robot interaction.

1.3 Motivation and Contribution

Human autonomy teams (HATs) can harness the strengths and mitigate the deficiencies of different agents. Robots or autonomous agents posses the ability to process data at frequencies of the magnitude of gigahertz (GHz), integrate a vast variety of sensor modalities, operate in complex/dangerous environment that would be hazardous for humans to venture in, and they have a distinct, directional and bounded field-of-view. On the other hand, while humans have limited data processing capacity, they possess exhaustive field experience and domain knowledge and have a comprehensive interpretation of complex mission objectives that is extremely difficult to be embodied within an autonomous agent. With the increasing integration of intelligent machines and autonomous robots in collaborative human teams, extensive research has been conducted to explore efficient methods of task allocation, human-robot communication, and multi-robot coordination [26], [15]. In this context, FlightGoggles has emerged as a powerful tool capable of integrating not only simulated and real robots but also humans within a desired computing and rendering framework. How-

ever, FlightGoggles has faced limitations when it comes to creating and testing intricate interactions and perception between robot agents and humans across different realities. This means that the platform has not been fully equipped to handle complex scenarios where human and robot agents exist in various virtual or physical environments. Addressing this gap in capabilities is crucial for advancing the development of autonomous systems that seamlessly interact and collaborate with humans in diverse contexts. By enabling the creation and subsequent evaluation of intricate interactions and perception, researchers can gain deeper insights into the dynamics and challenges of human-robot collaboration, paving the way for more efficient and effective integration of autonomous robots into human teams.

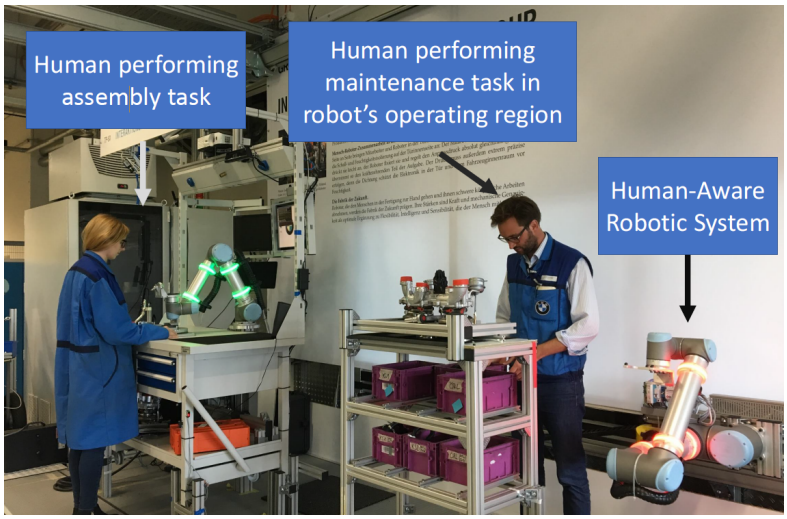

Figure 1.3: Example of an HAT where humans and robots collaborate in an engine assembly task [40].

At its core, the aim of the presented research is to provide a cutting-edge framework/testbed for simulation that enables the study of complex, multi-modal human-robot and multi-robot interactions in real-time. The proposed framework lies at the intersection of three fundamental techniques used in modern robotics research: i) Active perception which studies modeling and control

strategies for perception along with the interaction between them for a given purpose, ii) Mobile sensing that revolves around the idea of active/passive sensing modalities which can move around in the workspace, and iii) VR which simulates experience that immerses user in a synthetic virtual world. The cyber-physical facility developed in this work, unlike traditional simulation systems, offers a unique blend of data-driven exteroceptive sensor simulation and real physics. A combination of various proprioceptive and exteroceptive sensors are installed on both real-world agents as well as their virtual counterparts in order to facilitate complex feedback and further analysis. Building upon previous advancements in simulation platforms for robotics, this work introduces a unique cyber-physical environment. In this environment, customized physical laboratory environments are seamlessly merged with intricate virtual worlds with their agents in it. The primary objective of this research is to showcase a testbed for perception-driven intelligent interactions among simulated and real human-robot teams. The main contribution of the research presented lies in the development of a novel framework, a framework that integrates embedded systems design with VR technology and graphics tools to facilitate photo-realistic, real-time, and safe multi-agent interactions. In order to support these interactions while simultaneously enabling effective perception, planning, and control of various agents, innovative hardware and software integration tools are designed and developed. Moreover, the concept presented in this thesis demonstrates the ability to support not only perception-based control but also communication between virtual and real agents. The work also showcases its capability to test interactions as well as collaboration within HATs in challenging and realistic test environments. By providing this advanced framework, this thesis contributes to the advancement of research in decision-making and human-robot

collaboration, particularly in the context of cyber-physical environments.

1.4 Thesis Outline

The organization of the thesis is as follows. After providing an introduction about the thesis alongside related work and motivation in Chapter 1, Chapter 2 describes the mathematical formulation of the framework developed in this work. Chapter 3 forms the core of this thesis as it provides a detailed description about the system setup and explains the simulation environment, human and robot interfaces, and perception, planning, and control. In Chapter 4, we discuss the various experiments carried out in the system to demonstrate its capabilities and show their results. Chapter 5 summarizes the work carried out by providing a conclusion based on the experiments carried out in chapter 4 and also gives a glimpse into the future work.

CHAPTER 2

PROBLEM FORMULATION AND MATHEMATICAL PRELIMINARIES

The research presented in this thesis focuses on the development of a unique cyber-physical framework for conducting research on HATs that include a number of collaborative agents, operating and interacting within simulated virtual worlds as well as across physical/laboratory workspace. The physical laboratory where the human and autonomous agents operate is denoted by the real workspace $\mathcal{W} \subset \mathbb{R}^3$ (shown in figure 2.1). Accordingly, the simulated environment or virtual workspace created using UE™ is denoted as $\mathcal{U} \subset \mathbb{R}^3$. The various agents present in the testbed are categorized into the following four types depending on their fundamental agent-environment interactions: *real agents*, *virtual agents*, *avatars*, and *real agents with XR*. From this categorization, virtual agents and real agents operate and sense solely in $\mathcal{U}$ and $\mathcal{W}$ respectively. On the other hand, avatars sense or perceive the environment in $\mathcal{U}$ while simultaneously transmitting this sensory information to their physical world counterparts in $\mathcal{W}$, namely real agents with XR. The virutal avatars of the real agents with XR are kinematically to them and are thus teleoperated by the real agents with XR. Hence, the real agents in XR see what their avatars see, and accordingly react to their perception and by flow of control their resulting states are relayed, in real-time, to their avatars in the virtual world. The perception and the corresponding visual interactions of the agent avatars and virtual agents are restricted to the virtual workspace $\mathcal{U}$. An example of the types of agents in HATs is shown in figure 2.2.

We assume that the index sets of robots and humans operating in $\mathcal{W}$ is denoted by R and H, where each robot is associated with an index $i \in R$ while each

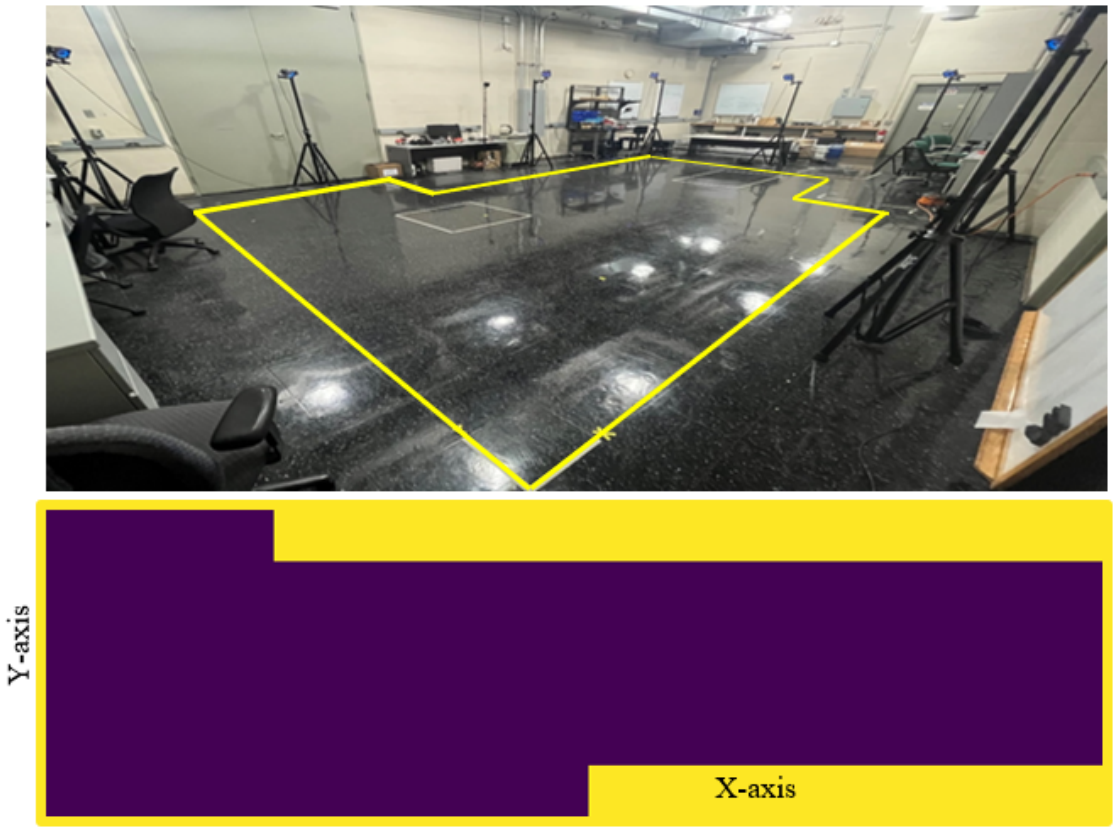

Figure 2.1: Physical workspace in which the real-world agents (humans and robots) coexist.

human is associated with an index $j \in H$. Similarly, for the robots and humans operating in $\mathcal{U}$, the index set is denoted by P and Q respectively and $i \in P$ and $j \in Q$ associates an index for each robot and human respectively. For brevity in notation, it is assumed that the avatars of the real agents with XR assume the same indices in P and Q as their real-world counterparts in R and H respectively. Thus, it can be confirmed that the indices of real humans with XR belong to $H \cap Q$ while the indices of real robots with XR belong to $R \cap P$. This further leads to the property that $R \cup P$ denotes the index set of all robots in the HAT while $H \cup Q$ denotes the same for all humans present in the cyber-physical framework.

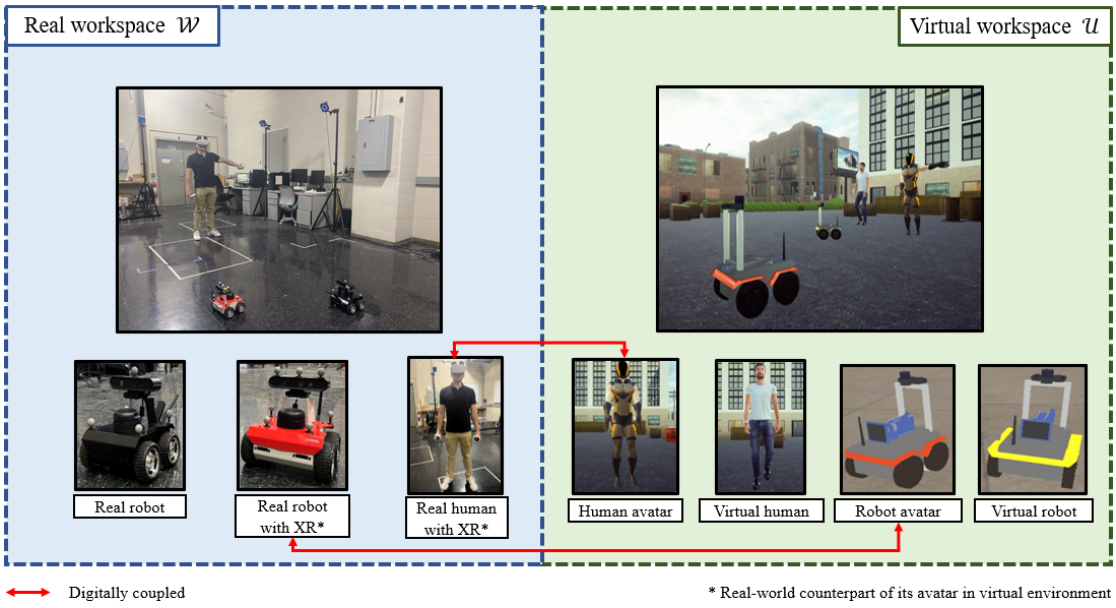

Figure 2.2: Real and virtual agents in the facility designed based on the proposed framework, where real agents with XR are digitally coupled with virtual avatars in UE™.

The field of view (FOV) of robot $i \in R$ with rigid-body geometry $\mathcal{A}_i \subset \mathcal{W}$ and of human $j \in H$ with deformable geometry $\mathcal{H}_j \subset \mathcal{W}$ are denoted as $\mathcal{S}_i$ and $\mathcal{S}_j$ which are defined with respect to the body fixed reference frames, $\mathcal{F}_{\mathcal{A}}$ and $\mathcal{F}_{\mathcal{H}}$, attached to their respective geometries, defined relative to the inertial frame $\mathcal{F}_{\mathcal{W}}$ embedded in $\mathcal{W}$. The geometries of agents in $\mathcal{U}$ and their corresponding FOVs and body-fixed frame of references can be defined accordingly. There exists a transformation, denoted by $\mathbf{T}$, from $\mathcal{W}$ to $\mathcal{U}$, whose inverse, represented by $\mathbf{T}^{-1}$, is assumed to exist and known *a priori*. In this research, the robot agents used to showcase the abilities of the proposed framework are unmanned ground vehicles (UGVs), however, the presented architecture can be extended easily to incorporate other agents such as autonomous aerial vehicles (AAVs) and unmanned underwater vehicles (UUVs). It is also assumed that the robot agents and humans present in $\mathcal{W}$ and $\mathcal{U}$ move on the ground plane, aligned with the

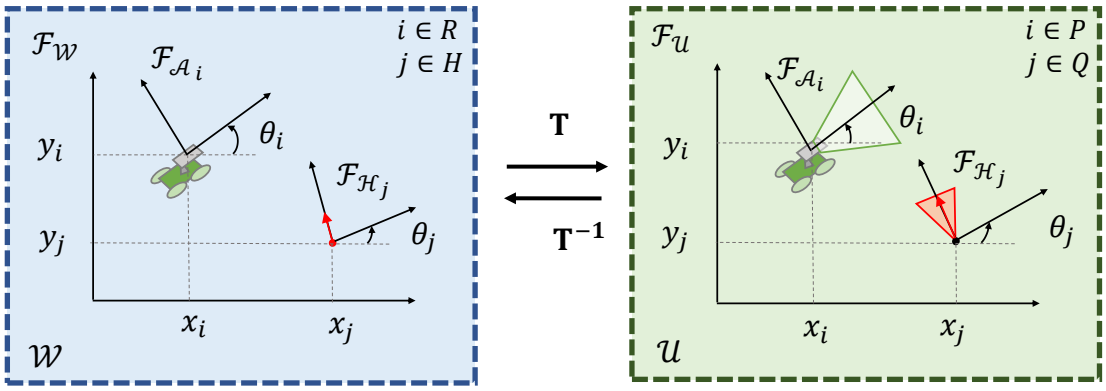

Figure 2.3: Configurations of a human and robot agent in the real workspace (left) and their corresponding avatars in the virtual workspace (right).

co-ordinate plane of the inertial frames $\mathcal{F}_{\mathcal{W}}$ and $\mathcal{F}_{\mathcal{U}}$ and are embedded in $\mathcal{W}$ and $\mathcal{U}$ respectively. The configurations of robot and human in $\mathcal{W}$ are denoted by $\mathbf{q}_i$, $i \in R$ and $\mathbf{s}_j$, $j \in H$ while those operating in the $\mathcal{U}$ are denoted by $\mathbf{q}_i$, $i \in P$ and $\mathbf{s}_j$, $j \in Q$ respectively. In detail, the state vector $\mathbf{q}_i$ denotes the configuration of a robot $i \in R \cup P$ and is defined as $\mathbf{q}_i = [x_i, y_i, \theta_i]$. In a similar manner, the pose of human j is defined by the facing direction and position which can be represented by $\mathbf{s}_j = [x_j, y_j, \theta_j]$. Using a motion capture system, the configuration of robot agents that are operating in $\mathcal{W}$ can be estimated while the configuration of humans in $\mathcal{W}$ is obtained through VR tracking. On the other hand, the configurations of agents operating in $\mathcal{U}$ are known accurately inside the simulation environment in UE™. For every robot $i \in R \cap P$ with configuration $\mathbf{q}_i$, it is assumed that the dynamics are given by the unicycle motion model [11].

$$\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix} = \mathbf{f}(\mathbf{q}_i, \mathbf{u}_i), \tag{2.1}$$

where the control vector for the robot, $\mathbf{u}_i = [v_i, w_i]^{\mathrm{T}} \in \mathbb{R}^2$, includes the linear velocity v_i and the angular velocity w_i. The aforementioned notations can be clearly visualized in figure 2.3.

CHAPTER 3

SYSTEM SETUP AND METHODOLOGY

Within the cyber-physical framework presented in this research, there exists communication and various interactions between real humans and physical robots with simulated agents as well as avatars in photorealistic virtual environments (explained in detail under section 3.1). Studying different sensor modalities in pressure-induced environments can help to seamlessly interface robot capabilities with humans for both heterogeneous and homogeneous teams in decentralized planning. For the various agents present in the testbed, the senses are collocated in these virtual environments (shown in figure 3.1), while the framework supports their individual operation in completely distant or different workspaces altogether. In order to achieve this functionality, we leverage the integrative cyber-physical interfaces proposed in this research (refer to sections 3.2-3.4) which in turn facilitate collaborative decision-making. These cyber-physical environments, through the aforementioned sophisticated integration, act as a medium for safe as well as realistic inter-agent and agent-environment interactions. It allows for modelling, analyzing, and leveraging these different interactions for a variety of applications while, at the same time, avoiding the risks associated with testing robots in the real world like collisions with obstacles and safety around humans. The proposed framework implements the 3D graphics development software, UE™, to develop virtual environments. This gives researchers the pliability to test algorithms and collect data in a wide variety of environments like cities, subways, oceans, and offices under varying weather and lighting conditions and with a diverse set of user-defined static and dynamic obstacles which can be easily incorporated. The virtual environment can be populated with completely virtual humans with pre-

defined actions or behaviors to simulate various social settings with varying crowd densities to perform experiments for social robotics. The architecture designed based on the proposed framework allows for multiple user-controlled and autonomous agents parallelly, thus allowing the facility to be utilized for online multi-agent control and coordination experiments. Section 3.5 describes in detail the different policies implemented for planning and control of both real and virtual robots.

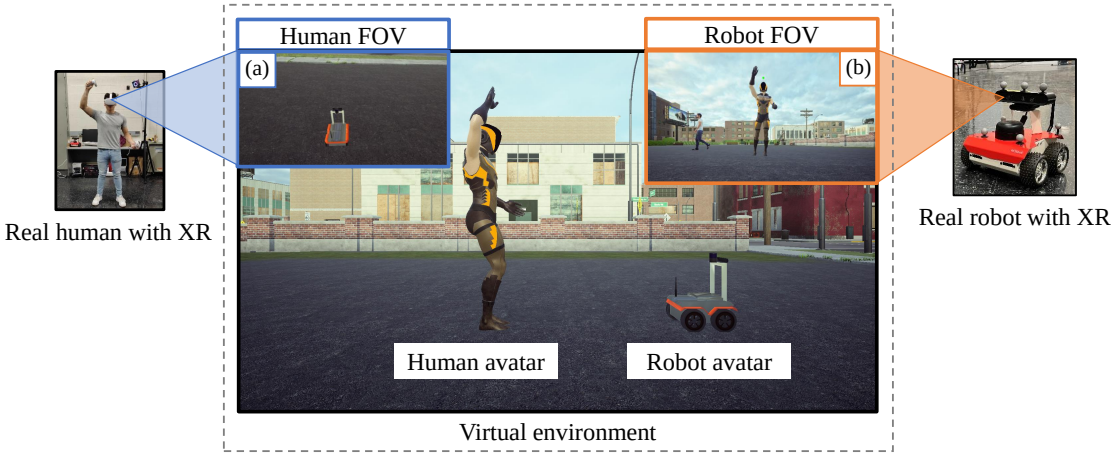

Figure 3.1: Visual perception of UE™ industrial city environment via avatar by (a) human and (b) robot agents who may then interact in real time inside the simulated world.

The system architecture for the developed testbed which is used to achieve human-robot collaboration between humans and real-world robots with their avatars in the virtual environment defined in UE™ is shown below in figure 3.2. The communication or transfer of the kinematics from real-world agents to their respective avatars in the virtual workspace are denoted with blue arrows, while the transfer of perception of these virtual avatars to their real-world counterparts is denoted by orange arrows in the figure. The human avatars in the workspace $\mathcal{U}$ are controlled by the human operators present in the workspace

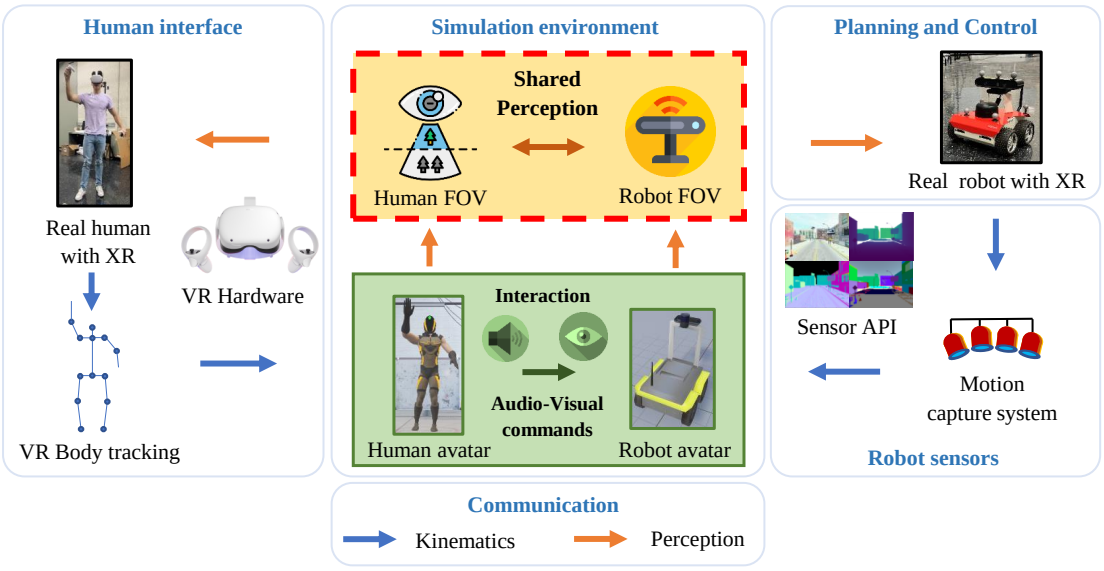

Figure 3.2: Proposed framework that enables research on human-autonomy interactions and collaborations through virtual augmentation in photo-realistic simulated UE™ worlds.

$\mathcal{W}$ bia the VR headsets and cameras that enable real-time VR body tracking. The motion capture system installed in the workspace $\mathcal{W}$ streams the real robot state to their respective robot avatars by detecting the reflective markers adhered to the robots operating in this workspace. In order to simulate the FOV of the robot agents, virtual sensors are implemented through sensor APIs (described in section 3.3), while the VR headset allows the humans present in the facility to observe the workspace. The FOVs of not only robots but also humans in the facility can be monitored constantly and in real time to facilitate shared perception in agents as the FOVs are defined inside the virtual environment. This feature of tracking the FOVs of the human and robot agents simultaneously allows for direct sharing of visual cues observed by any agent with any other agent, human or robot, and thus provides a unique advantage for the proposed framework over existing facilities which only consider sharing cues inside the

robot FOV [25]. In addition, section 4.1 and 4.2 describes the multi-modal form of communication (audio or visual) between the humans and real and virtual robots which supports research and testing on human-robot interactions. Table 3.1 gives a detailed summary of the various software and hardware components essential in developing the proposed architecture.

Component	Application
Software	
UE™	Game engine that renders and hosts the virtual environments alongside virtual robot and human actors.
Motive 3.0	Skeletal solver used for creating rigid bodies of robots in the real-world through tracked markers.
DeepMotion SDK	Three-point VR tracker that transfers human movement to UE4™ actor.
NatNet SDK	Transfers localization information from Motive to UE4™ and ROSbot.
Hardware	
OptiTrack Prime^x 22 Camera (10 units)	Camera testbed used for tracking and localizing the physical robots through placed markers.
ROSbot 2.0 (02 units)	Ground robots used in the physical experiments.
Meta Quest 2	VR equipment put on by the human user that allows the human to see, hear, and interact with the virtual world.
Dell Alienware Aurora R13	Primary desktop computer that simultaneously runs all software and perception algorithms.
Alienware x15 R2 Laptop (02 units)	Control stations that receive waypoints (x, y, θ) from the primary desktop and communicate control commands to the ROSbot.

Table 3.1: Description of software and hardware components of the facility.

3.1 UE™ Simulation Environment

The environment simulated in UE™ forms the core of the entire architecture presented in this research and acts as the interface between the virtual and real agents. The UE™ software is the platform in this work as it is widely considered the most visually realistic tool, thus bridging the simulation-to-reality gap in perception-related tasks while also providing uninterrupted access to all agents operating within. The framework shown in figure 2.2 leverages UE™ for supporting real-time rendering and manipulation of multiple photorealistic and programmable environments. In this work, the industrial city simulated environment (figure 3.3(a)), obtained from UE™ Marketplace, serves as the base environment for conducting the experiments presented in chapter 4. In order to include digital avatars and virtual agents, stream data amongst various agents of the HATs, and support agent associated sensing modalities (section 3.4), the base environment is further modified. The facility provides the user the flexibility to define, programmatically manipulate, and test a broad range of environmental conditions such as fog, time of day, and luminosity. Most of these conditions would be difficult to replicate in real-world or laboratory physical experiments and could influence visual perception in both robots as well as humans. Digital avatars are created to resemble their real-world counterparts in aesthetic, as required by the application, and in function by establishing a kinematic and sensing coupling between them as described section 3.2 and section 3.3.

Based on the requirements of the test or experimental scenario, as well as functionally by establishing a kinematic and sensing coupling with their real-world counterparts, digital avatars can be created that aesthetically resemble

their rel-world counterpartsas explained in section 3.2 and Section 3.3. This integration enables the development of a real-time hardware-in-loop simulation environment for unified control and perception of collaborative agents in the real and virtual world. Actors such as pedestrians, mobile vehicles, and virtual robots are created that are programmed using C++ and can be controlled offline via predefined trajectories, or are equipped with simulated dynamics, perception, and control algorithms. These algorithms run online and allow the testing of collaboration and autonomy with not only hardware but also software-in-the-loop, as explained in section 3.3 and section 3.5 respectively. figure 3.3 shows an example of such a programmed actor with with feedback controller-in-the-loop, comprised of an AAV, while sensing their environment by the means of onboard RGBD camera.

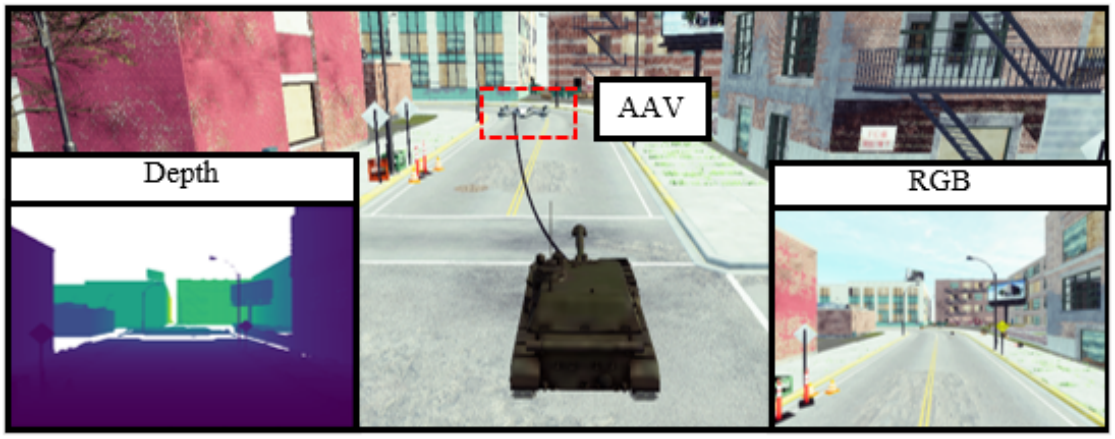

Figure 3.3: Virtual world, inclusive of autonomous robots, sensors, and artificial intelligence algorithms, developed exclusively for this facility using UE4™ is an industrial city monitored by an AAV equipped with onboard RGBD camera.

3.2 Human Interface

The goal of the architecture developed in this research is to provide a facility for a variety of applications in human-robot collaboration and in order to achieve achieve this, the human operators need to perceive and interact with the simulation environment and the different autonomous agents in it while simultaneously synchronizing their body movements with their avatars. As an example, humans in the testbed may need to react to robot motions and behaviors, while also providing commands to their robot teammates in the HAT by means of hand gestures or semantics. In addition, in order to test, study and analyze different types of collaborative tasks performed by larger HATs, human and robot avatars present in the facility may be required to interact with virtual humans or other human avatars in the UE™ simulated environment. To ensure this, the architecture created in this research provides the capability to simulate these human avatars using the Meta Quest 2 hardware with a Steam VR backend. The real humans with XR are granted perception of the virtual environment and control of their avatars in the virtual workspace through the Meta Quest 2 hardware. The real human with XR is able to view the simulated environment as rendered frames and also listen to audio, as sensed by the human avatar, through the VR headset. The VR headset allows continuous video and audio streaming from the first-person perspective of the virtual human avatar and thus provides the user with an immersive first-person experience of the environment. This headset and the accompanying handheld controllers communicate their positions to the system running Steam VR connected with UE™ over Wi-Fi using the trackers installed in them. In the developed testbed, to transform this data from the VR hardware into joint motions of a pre-defined skeleton of a

human avatar, the DeepMotion SDK [9] is used which performs the three-point VR body tracking while implementing an inverse kinematic solver. The interface of human in the virtual environment, taking control of the virtual avatar and interact with the simulation environment is shown in figure 3.4. This approach facilitates real-time seamless kinematic coupling and interaction, with a mean latency of 10 ms, by providing the human operators inside the testbed or the facility with the ability to control their virtual avatars without attaching any extra markers to the body. When testing machine learning-based algorithms trained on real-world datasets that are obtained from application-driven environments, such as industrial workshops and offices, or when using simulation worlds based in UE™ to generate synthetic datasets, the appearance of human avatars is of great importance. In order to reduce the sim-to-real gap and aid in solving the problem of reality-to-simulation and simulation-to-reality transfer, a user interface is built through the UE™ Blueprint to facilitate modification and selection of avatars of interest with ease, depending on the choice of test, domain, or HAT application. In the research presented in this thesis, the primary focus is on the visual and audio interaction between the human and robot agents in the testbed, as explored in section 4.1 and 4.2, however, it is worthy to note that the developed architecture can also be extended to other forms of interaction such as touch with the integration of other human-worn sensors, for example haptic feedback gloves.

3.3 Robot Sensing

The development of autonomous mobile robots equipped with multi-sensing capability has led to advances in information-driven planning and control [11]

Figure 3.4: UE™ Environment with DeepMotion SDK (avatar) and virtual robot along with the human with XR headset and handle controllers.

and modeling of robot sensor measurements is one of the most crucial components for HAT simulations. The robot platforms present within the designed cyber-physcial environment need to gather, process, and relay the knowledge about the environment in which they are operating as well as the agents within it while simultaneously augmenting their interaction capabilities. To achieve these functionalities successfully for robust operation, the sensing modalities available on not only the real but also the virtual robots plays a pivotal role in determining the types of HAT collaborations that can be designed and simulated within the facility. The facility, designed based on the framework proposed in this thesis, hosts a unique array of exteroceptive sensors, which measure the state of the operating environment, and proprioceptive sensors that measure the ego state of the robot. In the workspace $\mathcal{W}$, the Husarion ROSbots that driven by ARM processors, operate and act as the UGV robot agents for the experiments that are carried out in this work. Using rotary encoders and IMU, these real robots are equipped to localize with dead reckoning [19]. These

sensors may be suitable to simulate navigation in GPS-denied scenarios as they are heavily subjected to drift. In addition to localization using the embodied sensors in the real robots, the architecture, through the OptiTrack motion capture system, provides robot localization (as shown in figure 3.5) within 10 mm accuracy using the reflective markers mounted on the robots. This localization data is streamed in real time to the base stations controlling the robots as shown in figure 3.8, which can either use the dead-reckoning localization or this motion-capture-based localization. While running experiments, it is possible that the FOV of the motion capture cameras observing the reflective markers can sometimes be blocked by the multiple robots, humans, obstacles, etc. populating the real workspace, resulting in loss of localization. For such scenarios, the localization strategy is designed such that it can switch autonomously to dead-reckoning-based localization, with initialization at the robot's last know localization and revert back to the motion-capture-based localization once the cameras trace the markers again.

On the perception front, different modalities have also been developed in the facility by integrating recent advances in sensor modeling, traditional computer vision algorithms, and sensor APIs from UnrealCV [29]. This integration provides the simulation of 4-channel 8-bit data streams like RGBA cameras, surface normal estimation, and online panoptic segmentation. The facility also provides single channel (16-bit) images for ground truth depth which can be acquired in real-time using pre-defined depth cameras or stereo RGB cameras though UnrealCV. To process the RGB data stream obtained from UnrealCV and generate 16-bit (2 8-bit channel) images of dense optical flow using 8-bit (1 channel) grayscale images and the Farneback Estimation algorithm, an online processing pipeline is created using OpenCV [3]. The facility also supports novel modular

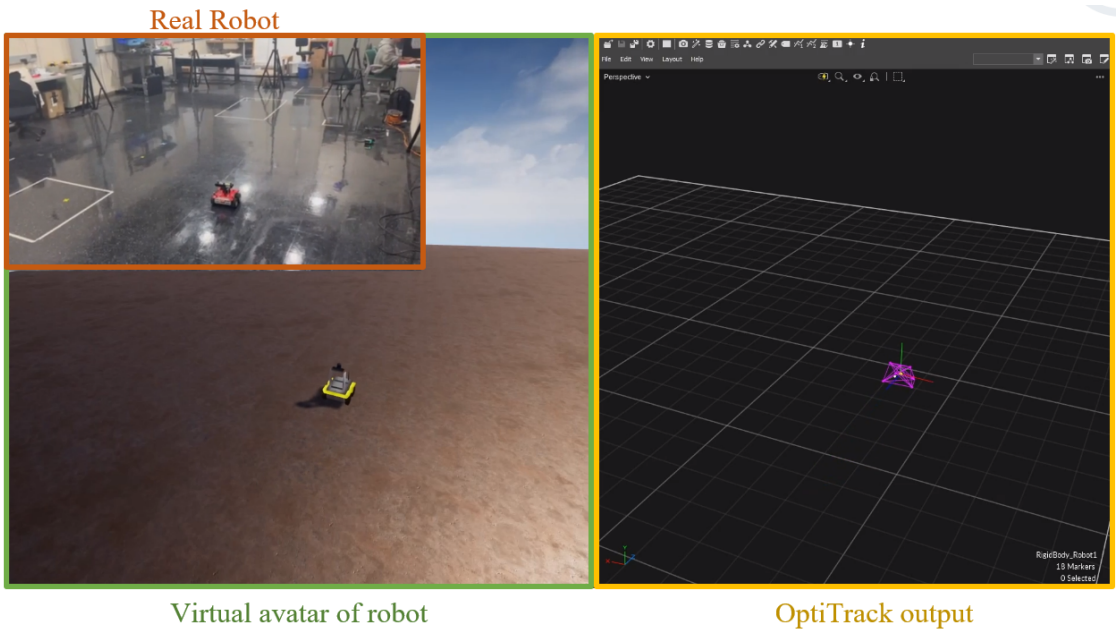

Figure 3.5: Robot localization and motion transfer to the virtual world using OptiTrack Motion Capture System.

blueprint controllable and/or C++ programmable robot agents that are developed in UE™ with the aforementioned sensors interfaced with the robot avatars or virtual robots to enable them perception while moving in the simulation environment. In order to enable the use of these sensors as static sensors, that monitor the virtual environment in the simulation, or to be programmed to move on defined trajectories for collecting/generating dataset, a separate Python script is created. In short, all these sensors, summarized in figure 3.6, defined in the virtual environment with user-defined noise characteristics can be either used as static sensors placed in the environment or dynamic sensors mounted on the agents in the environments or be programmed to move on trajectories that are defined offline.

All the real robots (without avatars) sense the real workspace and are equipped with Orbecc Astra RGBD camera, RP LIDAR A2 (laser range scan-

ner), and Time-of-flight (TOF) sensors (shown in figure 3.7). By implementing a ROS architecture, the output of these sensors mounted on the virtual and real robots is communicated to robot planners (explained in detail in section 3.5) that run onboard for active perception tasks facilitating inter-agent interaction.

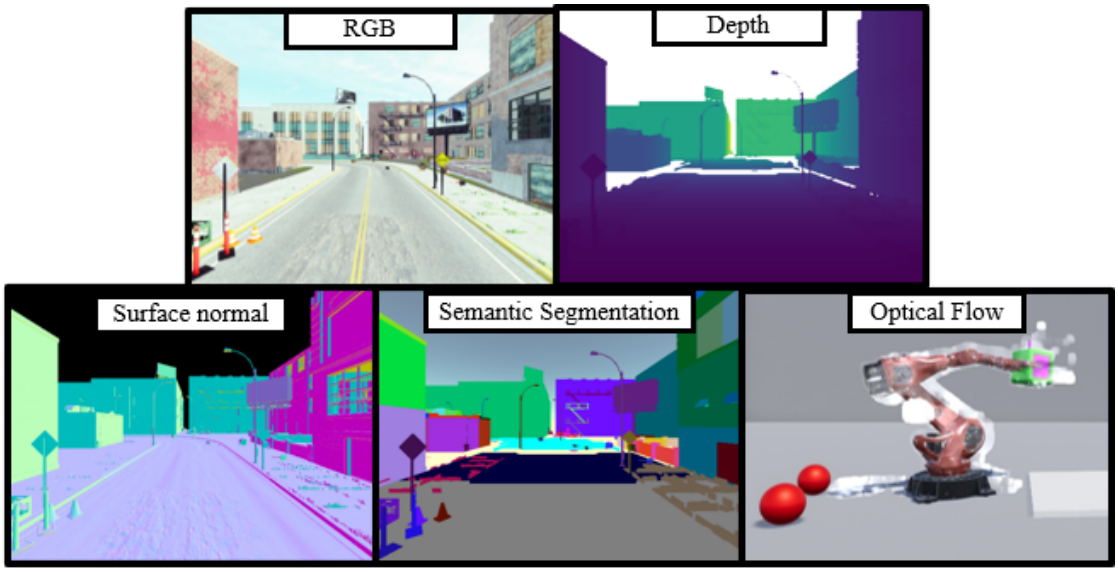

Figure 3.6: Examples of simulated sensing modalities and computer vision algorithms in the developed facility.

3.4 Communication Between the Real and Virtual Workspaces

The testbed/facility developed in this work passes message in order to facilitate inter-agent communication in and across the virtual and real workspaces by hosting different communication. Figure 3.8 provides an overview of the overall message-passing framework. The communications/data from the motion caption system are harbored on an Alienware Aurora R13 system, that acts as a head substation, while simultaneously hosting the virtual environment in UE™, and supporting the VR tracking and associated hardware. Based on the

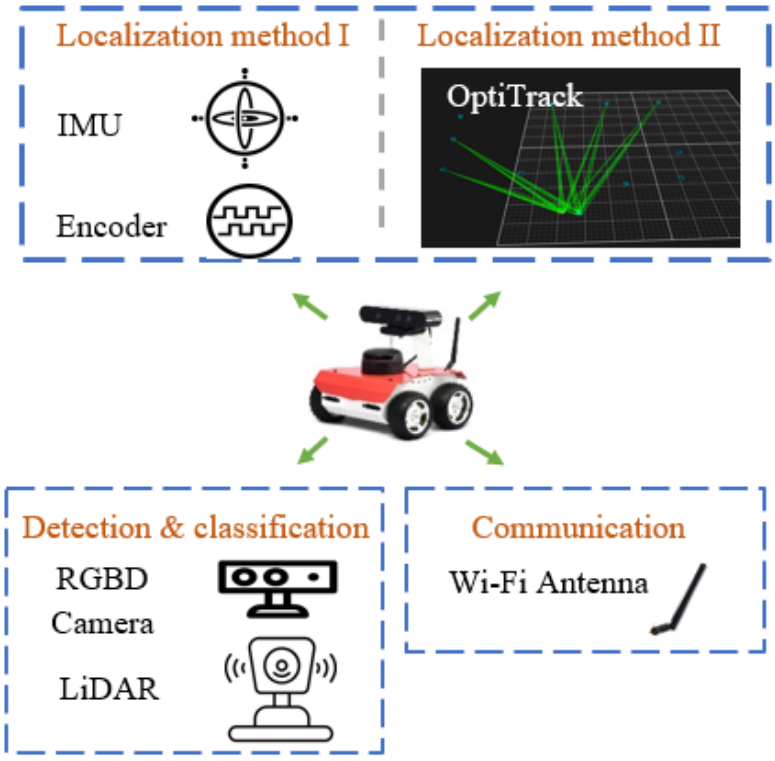

Figure 3.7: Sensing modalities available on the UGV robots used while conducting experiments.

cooperation strategy or control policy employed, and depending on the result of the interactions between the different agents in the virtual environment, the head substation generates desired wyapoints orgoal state for the robots. For the robots operating in $\mathcal{W}$, the intended waypoints are communicated to their base control stations over LAN. These waypoints are then converted into control commands that are transmitted to the real tobots over the 2.4 GHz Wi-Fi networks, based on the planning and control framework running on the stations (described in detail in section 3.5), set up on distinct custom channels to avoid aliasing. Communication between the base control stations is established over a LAN connection through UDP protocols which allows the simulation of real-time inter-robot communication with low latency. The OptiTrack motion-capture cameras stream data packets, as the robots move in the workspace $\mathcal{W}$,

comprising the marker data, via a bridge connection running at a frequency of 120 Hz, to the head substation. The facility employs a proprietary data processing software for OptiTrack called Motive which uses the marker data to infer the localization of the robots that are defined through a collection of markers as user-specified rigid bodies. The inferred robot localization is then streamed to the real robots over UDP channels via the base control stations for the purpose of planning and control followed by transmission to the robot avatars in the virtual environment $\mathcal{U}$. This transfer establishes a kinematic coupling, using the proprietary NatNet SDK clocking at 120 Hz and with a latency of 10 ms, between the robot avatars and their real-world counterparts. The minimum overall system time delay, from sending a desired waypoint from the head station to detecting the corresponding effects in the robot state in the virtual environment, is approximately 0.051 s. This is seen in the response of the system to a step input in figure 3.9, where the input command is the waypoint provided and the response/output is the real-time position of the robot in the virtual world.

3.5 Planning and Control

This subsection introduces the waypoint-following policy implemented on real and virtual robots in this testbed. For a robot $i \in R$ operating in $\mathcal{W}$ (real robots or real robots with XR), its base control station receives a waypoint denoted by $\mathbf{q}_i^*$, $i \in R$ in $\mathcal{W}$ directly or receives a waypoint in $\mathcal{U}$ and maps it to a desired waypoint in $\mathcal{W}$ using the transformation $\mathbf{T}^{-1}$, before executing the waypoint-following policy to determine the controls. For virtual robots, however, the waypoint $\mathbf{q}_i^*$, $i \in P$ is directly used by the policy. For brevity in notation, agent indices are omitted in the rest of this section. Assuming $\mathbf{q}^*(k) = [\mathbf{p}^*, \theta^*]^\mathrm{T}$ is the

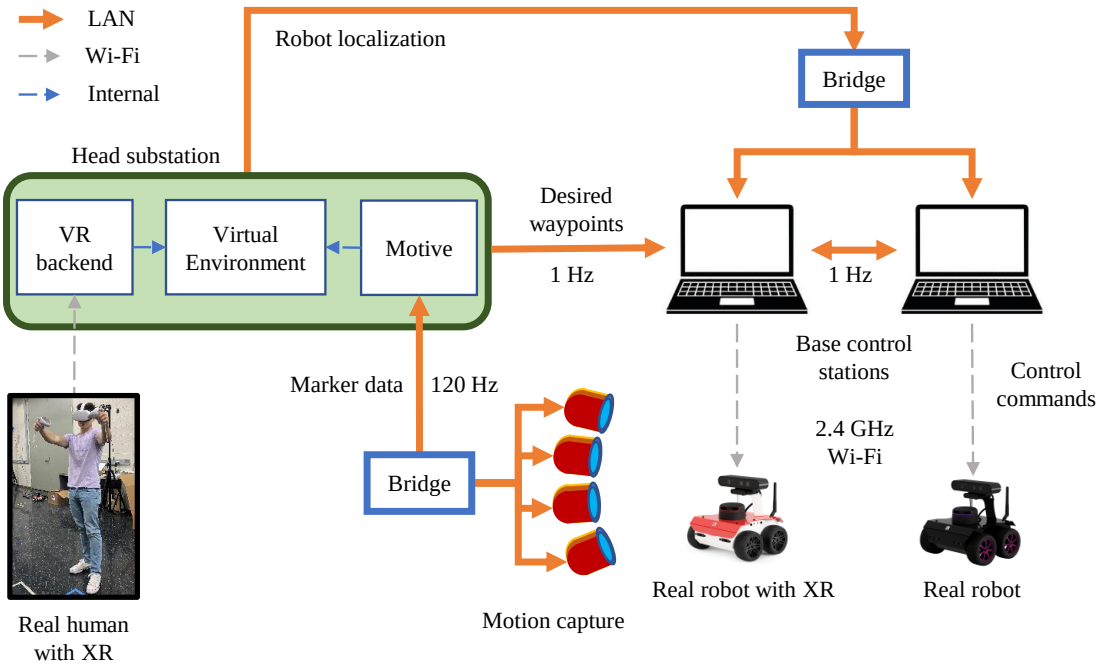

Figure 3.8: The cyber-physical facility interfaces and communication pipeline which enable seamless integration of real agents with XR as avatars in the virtual workspace.

desired waypoint for a robot in $\mathcal{W}$ at time-step k, where the desired position and desired orientation is $\mathbf{p} = [x^*, y^*]^{\mathrm{T}}$ and θ^* respectively. The algorithm for waypoint-following implemented on the robot is a move-then-turn policy. At state $\mathbf{q} = [x, y, \theta]^{\mathrm{T}}$, the robot first turns to point towards the desired waypoint position $\mathbf{p}^*$, moves towards it, and then rotates to reach the desired orientation θ^*. This policy outputs the control command $\mathbf{u} = [v, w]^T$ comprised of linear velocity v and angular velocity w, which has been summarized in the order of execution as follows:

$$v = 0 \qquad w = k_\theta \left(\tan^{-1}\left(\frac{y^* - y}{x^* - x}\right) - \theta\right) \tag{3.1}$$

$$v = k_x \left(x^* - x\right) + k_y \left(y^* - y\right) \qquad w = 0 \tag{3.2}$$

$$v = 0 \qquad w = k_\theta \left(\theta^* - \theta\right) \tag{3.3}$$

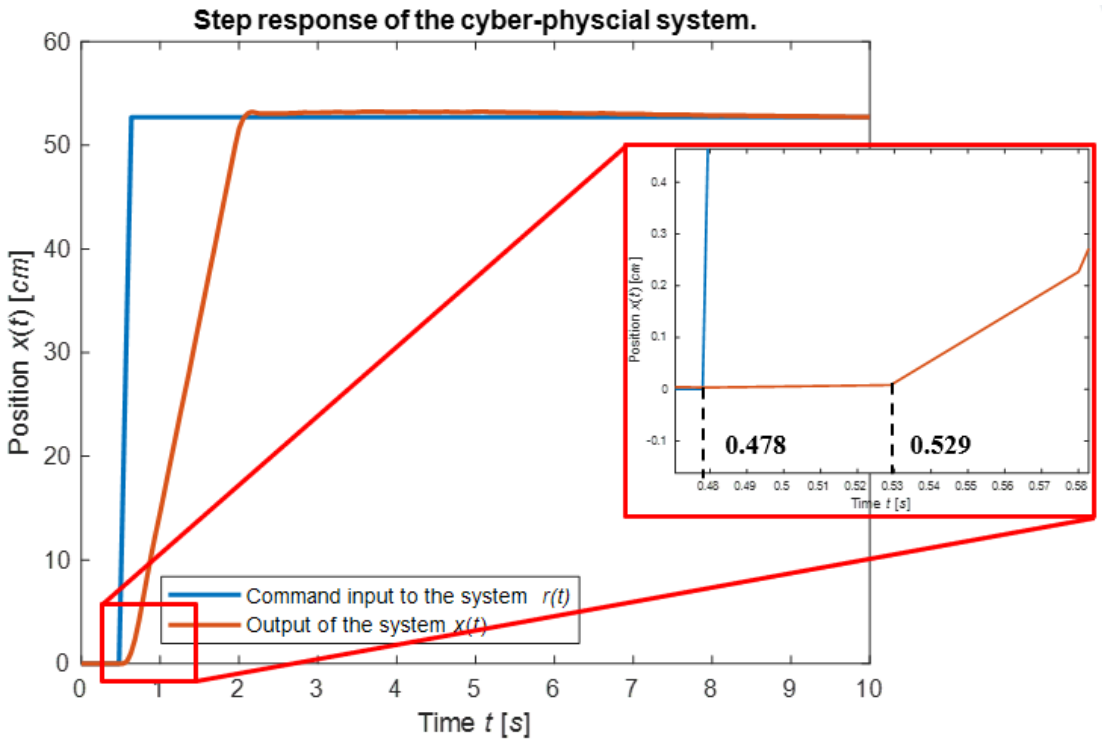

Figure 3.9: Response of the cyber-physical system to a step input to demonstrate the overall system time delay.

where $k_x,\ k_y,\ k_\theta \in \mathbb{R}^+$ are user-defined parameters with larger values representing a faster response to the error in desired and current pose of the robot. At all times throughout this process, the motion capture systems track the motion of these robots in $\mathcal{W}$ and record their states. As mentioned in Section 3.4, this localization information $\hat{\mathbf{q}}_i,\ i \in R$, serving as an estimate of $\mathbf{q}_i$, is then streamed to the robot $i \in R$ base control station and to its robot avatar $i \in P$ in UE™ if it exists, which is moved to $\hat{\mathbf{q}}_i,\ i \in P$ obtained using the transformation $\mathbf{T}$ as shown in figure 3.2. This control loop, to couple the real robot with its avatar, runs at a frequency of 120 Hz in real time. The planner continuously streams waypoints according to different tasks as demonstrated in Section 4 with the desired run-rate frequency and the control algorithm outputs controls according to the latest desired waypoint.

CHAPTER 4

EXPERIMENTS AND RESULTS

To convey the functionalities and capabilities of the developed facility, based on the proposed framework, three experiments are conducted to highlight different types of agent interactions across real and virtual workspaces. The first experiment will focus on human-robot interaction using vision-based modality of interaction, the second experiment will test the same but instead implement an audio-based modality of interaction, and the third will showcase multi-robot teaming across the real and virtual workspace.

4.1 Human-Robot Vision-based Interaction and Control

The first experiment is designed to showcase interaction-based control of virtual robots and robot avatars through the mode of vision using gesture commands from a human teammate as summarized in figure 4.2. This demonstration takes place in the industrial city environment, built in UE™, hosting the following actors: a human avatar, a robot avatar, and a virtual robot. As shown in figure 3.1, all agents can perceive the virtual workspace using simulated RGB cameras. Human avatars communicate with the robot agents using gestures as shown in figure 4.1 to command the next waypoint. These gestures are detected by a real-time human-pose detection algorithm, OpenPose [5], [43], implemented on each of the robot agents. OpenPose utilizes deep learning techniques and employs a convolutional neural network (CNN) to estimate the keypoints. It processes the input image and generates a heatmap representation where each body part corresponds to a peak which are then connected to form the pose estimation. Three distinct pose commands are defined for commanding the robot to move

in three different directions: left, forward, and right.

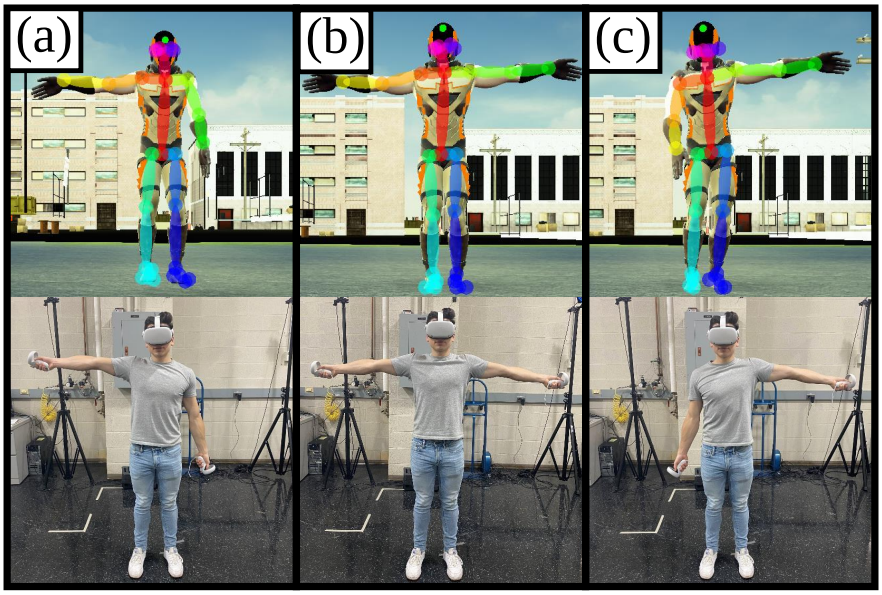

Figure 4.1: Through the designed architecture, a virtual autonomous agent (UGV) inside the industrial city, equipped with a virtual RGB camera and implementing OpenPose for keypoint detection (top row), is able to recognize and interpret the manual commands provided by a real human in $\mathcal{W}$, namely (a) right, (b) forward, and (c) left, by virtue of the human avatar created in real time using VR body tracking.

Based on the pose commands received as visual cues, the planner generates desired waypoints in the commanded direction. These waypoints are then streamed to the base control stations of the real robots with XR over LAN and to the virtual robots in the environment as described in section 3.4. The desired orientation at each of these waypoints is selected to make the robots face the human avatar to perceive the next gesture command. The waypoint-following policy as described in Section 3.5 is then used to calculate the control commands on each of the robot agents to reach the desired waypoints. This experiment is summarized using the schematic in figure 4.2 for a virtual robot $\imath$, $\imath \in P$ and a real robot with XR $\jmath$, $\jmath \in P \cap R$.

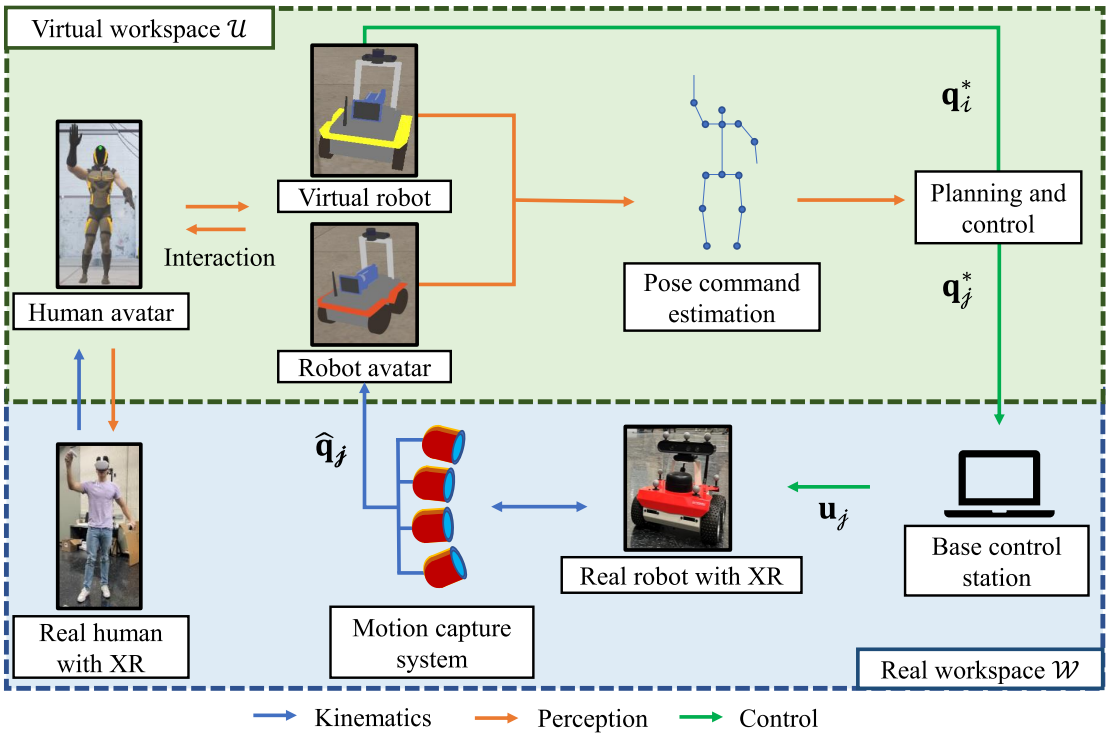

Figure 4.2: Human-robot collaboration is achieved by a human avatar, teleoperated by a real human with XR, commanding heading directions to the virtual robot $\imath$ and robot avatar $\jmath$ operating in $\mathcal{U}$, using pose commands generated by the keypoint detection as shown in Figure. 4.1

In this experiment, the virtual robot and the robot avatar are placed alongside each other in the simulation environment at a fixed distance from the human avatar and facing it. With each pose command, the robot agents move a distance of 0.5 meters in the commanded direction. Figure 4.3 shows an instance of the human operator providing command to the robots to go right while conducting the experiment in the cyber-physical environment. A total of ten pose commands were presented to each agent and their position trajectory is plotted in figure 4.4. The plotted results show that both the virtual robot and the robot avatar were able to correctly identify and react to all the pre-defined gesture commands. The figure also provides the error in the position of the real robot

for the experiment. From the plot, it is observed that the maximum error in the position of the real robot is 0.08 m. Both the virtual robot and the robot avatar successfully traverse the distance with proper heading directions as shown in figure 4.4. Comparing the trajectories of the robot avatar and the virtual robot, it can be clearly observed that the robot avatar successfully incorporates the dynamics of its real-world counterpart and hence moves in a more realistic way as compared to the virtual robot. This experiment demonstrates how the facility is able to successfully simulate proximate visual interactions, embodied agents, and incorporate real-world dynamics while providing a safe medium for human-robot collaboration.

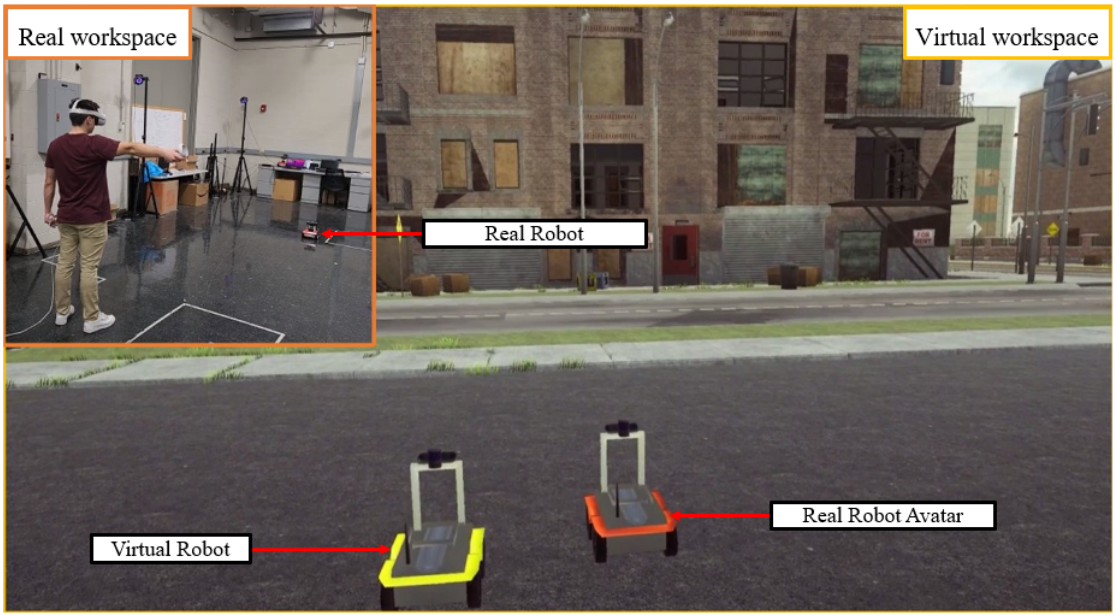

Figure 4.3: Human-robot collaboration using vision-based control. The robot agents are seen in the virtual workspace from the human avatar's perspective. The real workspace is seen in the upper-right corner with the human and the real robot.

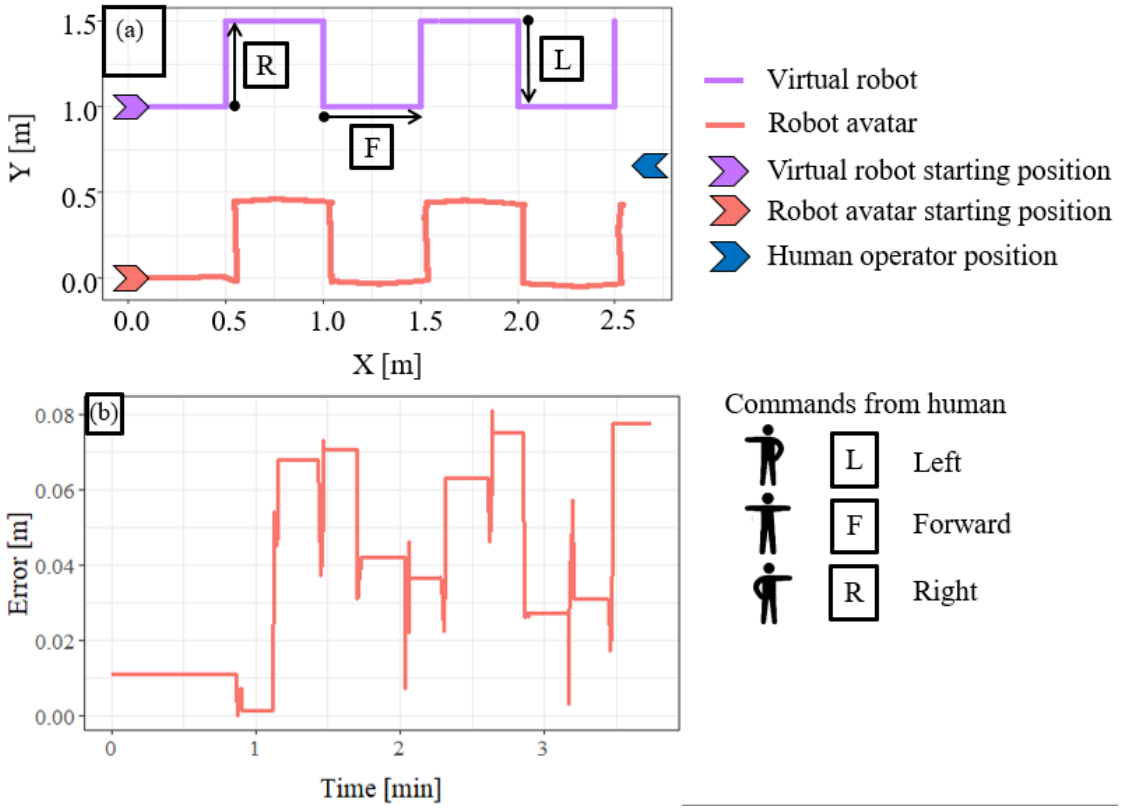

Figure 4.4: (a) Trajectory results of the pose estimation experiment for human-robot interaction. (b) Plot for error in the position of the real robot over the duration of the experiment.

4.2 Human-Robot Audio-based Interaction and Control

In section 4.1 the pre-defined commands can also be communicated to the robots as audio cues by using the Google Audio speech-to-text interface [16] running on the head substation. The real human with XR can speak any of the three pre-defined commands into the internal microphone of the VR headset which is then transcribed to text. This section aims to showcase the multi-modal interaction capabilities of the facility between humans, virtual robots, and real robot avatars at a more robust level. This is achieved primarily through audio mode of perception, while using Iterative Visual Question Answering (iVQA) from in-

puts given by human operator as summarized in figure 4.6. This demonstration also takes place in the industrial city environment in UE™, hosts the same actors as in section 4.1, and all agents perceive the virtual workspace. The setup devel-

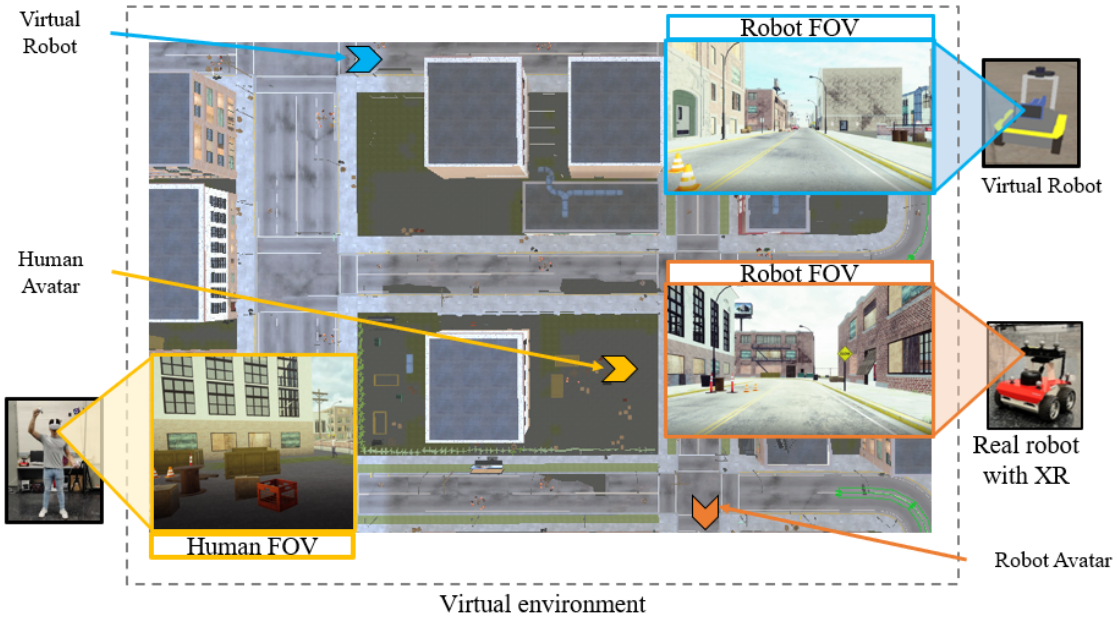

Figure 4.5: Setup for the audio-based human-robot collaboration experiment in the virtual environment resembling a multi-agent, multi-target detection scenario.

oped for the experiment resembles a multi-target detection scenario using multiple agents where each agent k (including humans represented by i and robots represented by j) operating in $\mathcal{U}$ searches a sub-region A_k of the workspace, i.e., $A_k \in U$ with $k \in i \cup j$. Figure 4.5 shows a birds-eye view of the workspace in the simulation environment and the location of the agents at the start of the experiment. In this case, human avatars communicate with the robot agents using audio-based inputs. These inputs are transformed into a set of questions using the Large Language Model (LLM) Generative Pretrained Transformer (GPT-3.5 turbo) [27] to search different targets (t) present in the simulation environment. Each robot then poses these set of questions to the RGB images obtained us-

ing the cameras onboard the agents as iVQA by implementing the Generative Image-to-text Transformer (GiT) [41] to determine if the given target is present in the scene or not. The positive detections are provided as results to the human operator as feedback from the robot agents.

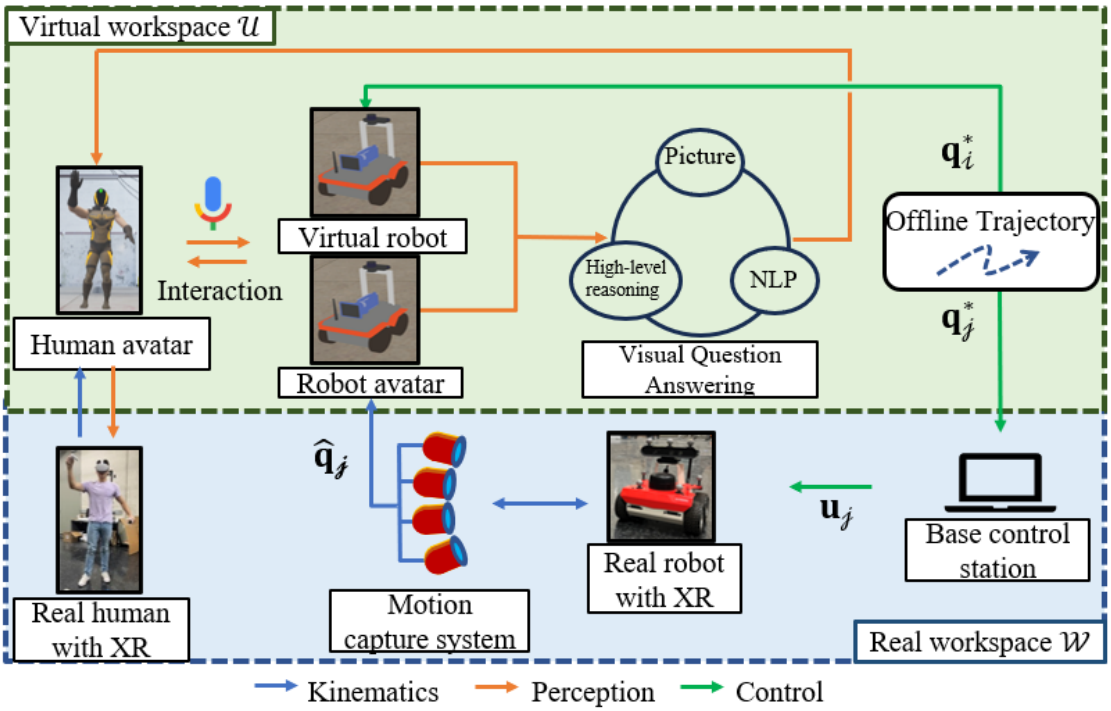

Figure 4.6: Human-robot collaboration is achieved by a human avatar, teleoperated by a real human with XR, providing instructions to the virtual robot $\imath$ and robot avatar $\jmath$ operating in $\mathcal{U}$ for target detection in a centralized framework.

As mentioned in section 1.3, human operators collaborating with autonomous can help integrate expertise, domain knowledge, and situational awareness, thus making these systems more robust, adaptive, and efficient. The pipeline developed in this experiment helps integrate and demonstrate these advantages of human expertise into the overall architecture of the facility and furthers the applicability of the developed facility. In this setup, the human operator provides a description of the targets that are present in the simulated

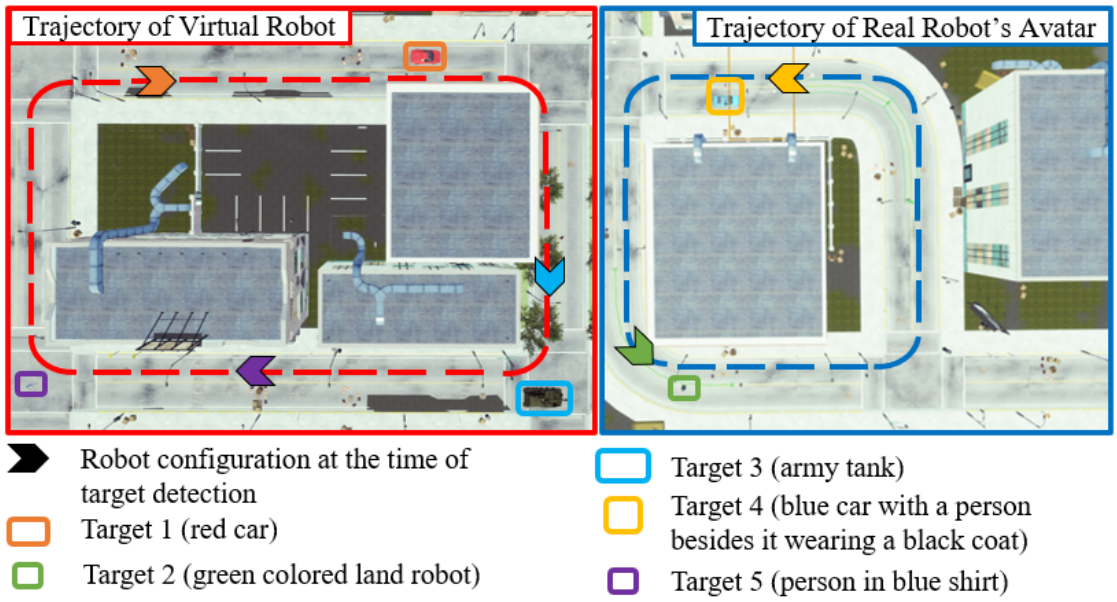

Figure 4.7: Trajectory of the virtual robot and real robot with XR indicating the configuration of the robot at the instance of target detection.

environment via speech and have to be detected by the different autonomous agents operating in the virtual world. This speech description is then converted into text using the Google Audio interface which is passed as input to the LLM that processes this raw description of the various targets and outputs a set of questions with each question corresponding to a target [8], [44]. The questions are framed in a particular manner to check if the described target is present in the given scene or not and are employed for iVQA on RGB images obtained from cameras located on each of the robot platforms [35], [34]. GiT is a state-of-the-art transformer network developed for unifying vision-language operations such as image/video question answering and captioning with a simple architecture of single image encoder and single text decoder. The image encoder in the network is based on the contrastive pre-trained model while the text decoder is a transformer module that is used to predict the text description, both of which are pre-trained by using a generation task. By using GiT, the human

operator has an opportunity to provide a detailed description of the target, such as color, scene text, shape, etc., instead of simply providing the object names for detection, such as car, building, etc. If the output of the iVQA task is positive, the pipeline then uses the depth map generated at the same timestep to provide an approximate location of the detected target from the robot. The schematic presented in figure 4.6 summarizes the experiment consisting of a virtual robot ι, $\iota \in P$ and a real robot with XR $\jmath$, $\jmath \in P \cap R$ alongside a human operator.

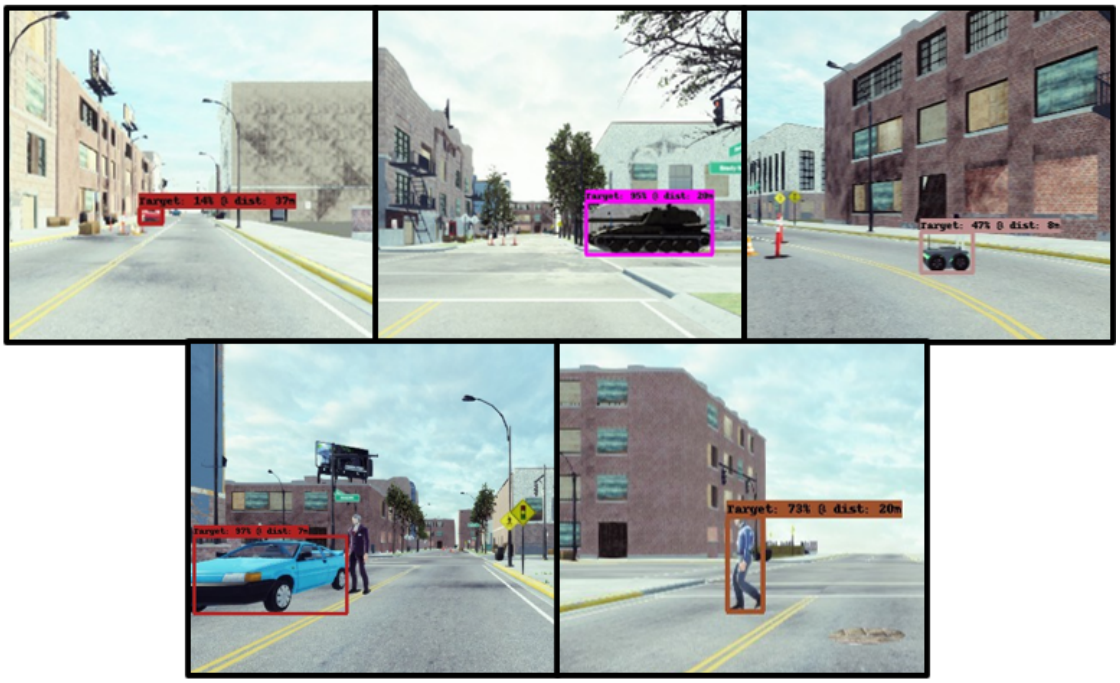

Figure 4.8: Target detections by the robot agents while also providing the distance of the target from the robot.

In this setup, the virtual robot, the robot avatar, and the human avatar are placed in separate quadrants of the simulation environment such that they are distant and do not see each other. The robot agents move along trajectories that are defined offline and cover separate sectors of the workspace. Three dynamic targets (a red sports car, an army tank, and a pedestrian) and two static targets

(a green colored land robot and a blue car with a person besides it) are placed at random locations. The human operator provides the input to the robots at the start of the experiment by saying "Please let me know if there is a red car in the scene or a green colored land robot in the scene or an army tank in the scene or a person in a blue shirt in the scene or a blue car with a person besides it wearing a black coat in the scene." Figure 4.7 shows the trajectory followed by the virtual robot and the location of the robot at the moment it detected each target. Figure 4.8 shows the output from the perception pipeline, i.e., the detections post VQA along with the average planar distance of the target from the robot, which is provided as feedback to the human. The results show that the robots are able to correctly identify the desired dynamic targets in real-world environments based on the description provided as input by the human. While there were errors in certain detections, they were eliminated by increasing the confidence estimates of the network to 0.2 which indicates that the pipeline has potential to be used in even more cluttered environments. This experiment demonstrates how the facility is able to successfully simulate proximate audio interactions coupled with iVQA techniques and incorporate real-world dynamics while providing a safe medium for human-robot collaboration.

4.3 Multi-Robot Interaction for Formation Control

The third experiment is designed to illustrate closed-loop interaction and control between multiple robot agents in the facility as shown in figure 2.2, namely a virtual robot, a robot avatar sensing in $\mathcal{U}$ and, a real robot sensing in $\mathcal{W}$. The purpose of this experiment is to demonstrate the ability of this facility to bridge the gap of data transfer between agents existing in simulation and

the real world. In this experiment, the robot team is tasked with a leader-follower-based formation control objective. A virtual robot and a robot avatar are placed into the industrial city environment created in UE™ and a real robot and the real robot with XR corresponding to the robot avatar are operating in the physical lab workspace. The virtual robot is designated as the leader robot which independently moves along a pre-specified path designed offline. The real robot with XR determines its waypoints using the localization information of the virtual robot, as communicated to its virtual-world counterpart (robot avatar), while the real robot does so, in turn, by using the localization of the real robot with XR. A formation control policy is implemented onboard all the robot agents to calculate these waypoints which ensures that the robot team maintains a desired formation.

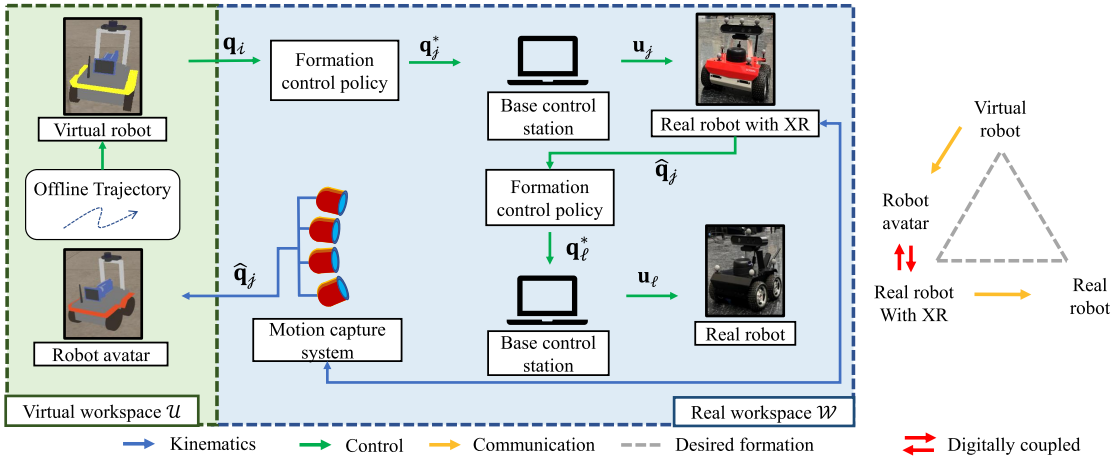

Figure 4.9: Leveraging the communication pipeline of the designed architecture, a virtual robot $\imath$, real robot with XR $\jmath$ with its avatar and a real robot ℓ co-ordinate amongst themselves to maintain the isosceles triangle formation.

This experiment is illustrated in figure 4.9 which features a multi-robot team comprising of a virtual robot $\imath$, $\imath \in P$, a real robot with XR $\jmath$, $\jmath \in P \cap R$, and a real

robot ℓ, $\ell \in P$. The virtual robot, in the role of a leader, moves along an elliptical trajectory using a spline path. The state of the leader is streamed via socket connection to the base control stations of the real robot with XR in $\mathcal{W}$. This base control station calculates the desired waypoint based on the state of the leader in real time to maintain an isosceles triangle formation. Simultaneously, the state of this real robot with XR is also streamed to the base control station of the real robot via socket programming through the inter-robot LAN connection, which calculates the desired waypoint for this robot to maintain the formation. This allows for a decentralized approach to formation control of a multi-robot team regardless of whether the agent of the team exists in $\mathcal{W}$ or $\mathcal{U}$. All the robot agents use the policy defined in Section 3.5 to reach the desired waypoints obtained online. It is important to note that in this experiment, only the state of real robot with XR is streamed back to the virtual environment since it is the only robot with a virtual avatar.

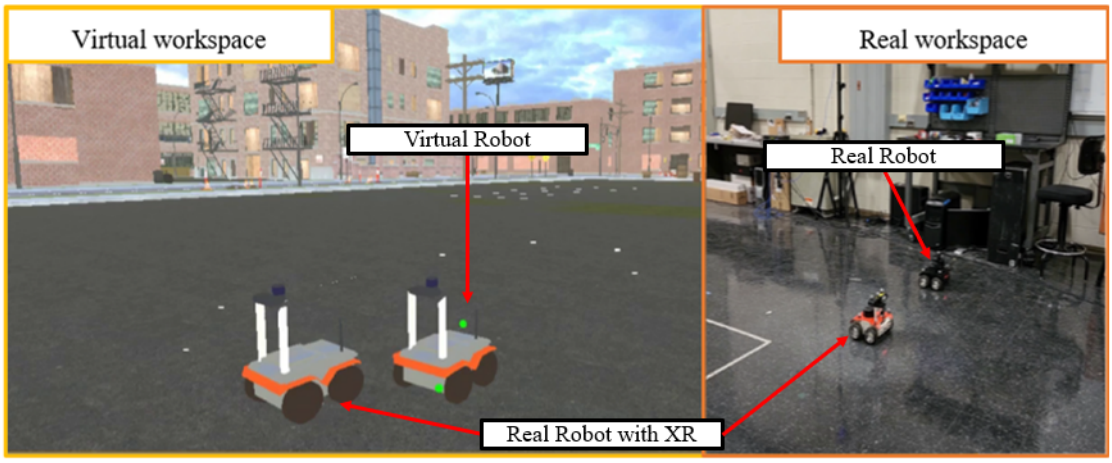

Figure 4.10: Demonstration of multi-robot interaction for consensus control in the cyber-physical environment for maintaining triangular formation.

The trajectories of the agents in the robot team are plotted in figure 4.11(a).

The virtual robot in UE4™ follows the elliptical trajectory as designed, and the successful coupling between the real robot with XR and its virtual avatar can be clearly observed. The robot team maintains the desired isosceles triangle formation throughout the experiment as illustrated in different instances in figure 4.11(a). Since the path of the leader and the desired formation were pre-defined, the desired trajectories for all the robots are determined offline and the error between their positions during the experiment as compared with these trajectories are recorded. High positional accuracy was achieved from both agents as the largest positional error was within 0.10 m as shown in figure 4.11 (b). This performance plot also shows that the second follower (real robot) consistently experiences lower positional accuracy when compared to the first follower (a real robot with XR). However, this can be attributed to the aggregation of errors due to the decentralized nature of coordination amongst the agents. Figure 4.10 shows an instance of the multi-robot experiment being conducted in the facility in real-time with the leader and avatar of the real robot with XR being shown in the simulation environment while the real robot and the real robot with XR operate in the physcial workspace. These results successfully demonstrate the capability of this testbed to establish communication between various agents existing in the real and virtual workspace and, as a result, enable real-time interaction between real and simulated agents. This unlocks the potential to do large-scale multi-robot experiments without space and hardware constraints while maintaining real-world dynamics in selected agents.

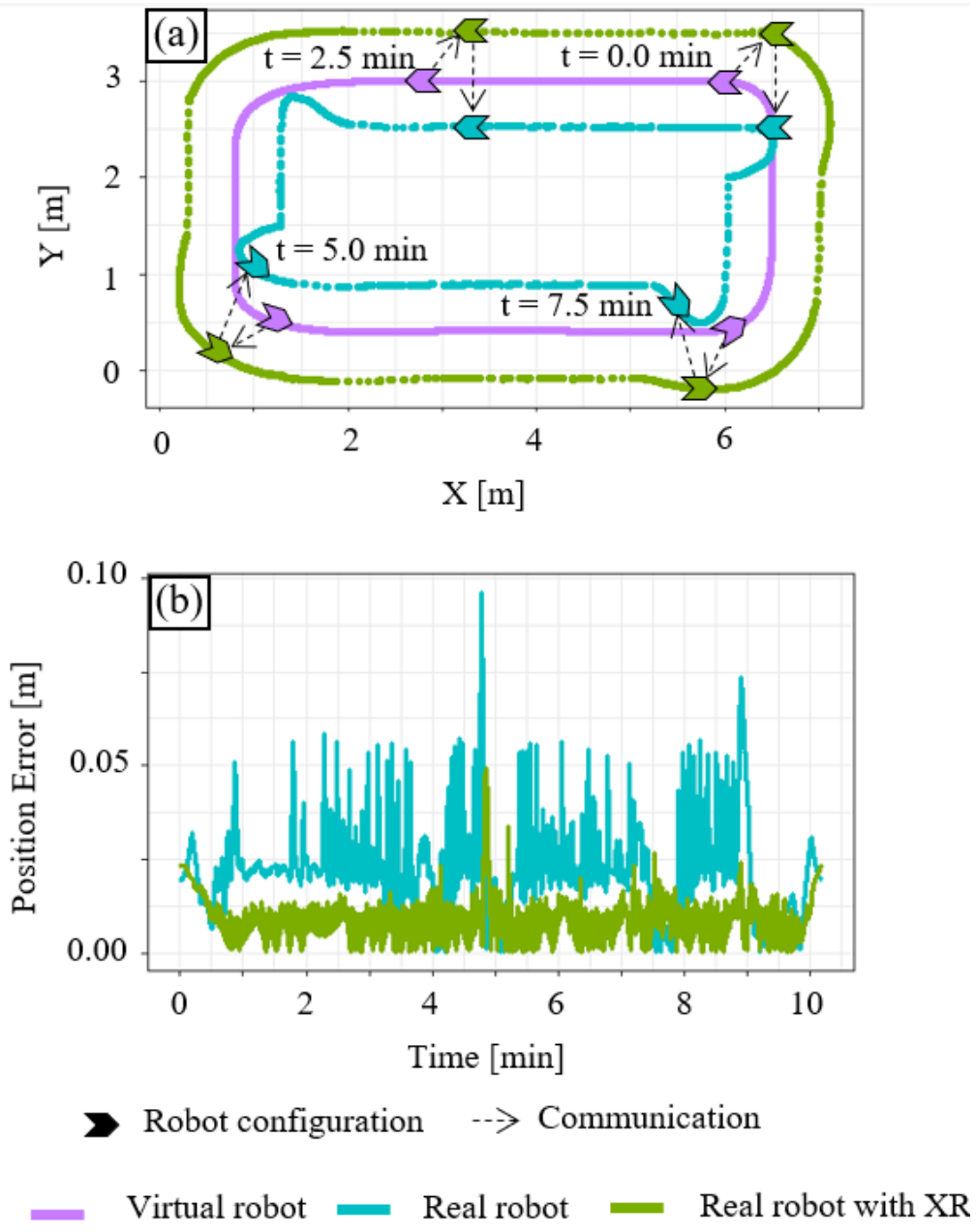

Figure 4.11: (a) Trajectory results of the leader-follower formation control experiment. The virtual robot is the leader, the real robot with XR is the first follower, and the real robot is the second follower. (b) The position error of each robot follower over the duration of the experiment.

CHAPTER 5

DISCUSSION AND FUTURE WORK

5.1 Conclusion

This research presents a framework for developing a multi-modal cyber-physical XR facility that leverages state-of-the-art robotics, visualization tools, motion capture, and VR technology to enable novel experimental testbed interfacing physical and simulated worlds. UE™ is used to create photorealistic simulated environments which facilitate interactions amongst HATs comprising of real agents, virtual agents, and agent avatars, tasked with achieving various objectives. These agent avatars operating in the simulation environment are teleoperated by the real agents with XR operating in a physical environment, thus sharing real-world dynamics while their perception from the avatars is shared for planning and decision-making. Communication pipelines enable seamless interfacing of the real and virtual workspaces to enable real-time collaboration amongst various agents in the HATs. The results of the three experiments demonstrate the capability of this system's framework to effectively host highly flexible environments with interactive agents spanning a combination of both the real and virtual worlds. The first experiment focuses on establishing human-robot perception and effectively demonstrated closed-loop control of both virtual robots and real robots with coupled virtual avatars. Using only body gestures, the human operators effectively communicate commands with robot agents and control the trajectory of each agent in real time. The second experiment focuses on demonstrating multi-modal human-robot collaboration by fusing LLMs and Vision Transformers (ViTs). With perception and control suc-

cessfully established between the robots and humans in this testbed, the third experiment demonstrates the ability to establish decentralized communication between varying robot agents. By implementing a leader-follower and formation control scenario on robot teams, this experiment effectively conveys the modality of RealTHASC to host real-time communication between the simulated and physical worlds and extends its reach to be used for multi-robot experiments. Apart from such pre-deployment testing of collaboration algorithms, this testbed can also be used as a closed-loop interaction interface to facilitate downstream tasks like online learning and data collection for safety-critical applications like social navigation. Finally, the research presented in this thesis led to the development of the RealTHASC (Real-Time Human Autonomous Systems Collaborations) facility.

5.2 Future Work

Future work will include extending the capabilities of the facility to include new interfaces for human operators like haptic feedback devices and leveraging this facility to study (1) AI-supported teamwork via collaborative virtual environments, (2) decentralized AI-supported multi-agent planning and perception, (3) integration of emerging neuromorphic and insect-scale technologies, and (4) distributed sensing and control for very-large networks of agents. Human-robot collaboration is an essential tool to develop safe and reliable autonomy in challenging scenarios. However, testing and experimentation in such scenarios is also difficult and potentially entail serious safety concerns to humans and robots. The work presented here is currently being extended to such settings, namely underwater environments. The goal of this research is to develop robot

agents (also called robot buddy) to assist scuba divers in underwater exploration experiments. The hope is that the ReaTHASC facility will be deployed for conducting a variety of human-robot collaboration experiments involving researchers from different countries.

BIBLIOGRAPHY

[1] Amado Antonini, Winter Guerra, Varun Murali, Thomas Sayre-McCord, and Sertac Karaman. The blackbird dataset: A large-scale dataset for uav perception in aggressive flight, 2018.

[2] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016.

[3] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[4] Rodney A. Brooks and Maja J. Mataric. *Real Robots, Real Learning Problems*, pages 193–213. Springer US, Boston, MA, 1993.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, 2017.

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments, 2017.

[7] Nvidia Corporation. NVIDIA Turing GPU architecture. Technical report, Nvidia Corporation, 2018.

[8] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] DeepMotion. DeepMotion SDK - Virtual Reality Tracking. `https://www.deepmotion.com/virtual-reality-tracking`, 2023. [Accessed 07-Jul-2023].

[10] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404, 2015.

[11] Silvia Ferrari and Thomas A Wettergren. *Information-Driven Planning and Control*. MIT Press, 2021.

[12] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. *RotorS—A Modular Gazebo MAV Simulator Framework*, pages 595–625. Springer International Publishing, Cham, 2016.

[13] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis, 2016.

[14] Epic Games. Unreal Engine — The most powerful real-time 3D creation tool — unrealengine.com. `https://www.unrealengine.com/`, 2023. [Accessed 07-Jul-2023].

[15] Jake Gemerek, Silvia Ferrari, Brian H Wang, and Mark E Campbell. Video-guided camera control for target tracking and following. *IFAC-PapersOnLine*, 51(34):176–183, 2019.

[16] Google LLC. Google Cloud Speech API. `https://cloud.google.com/speech-to-text/docs/`, 2022. [Accessed 07-Jul-2023].

[17] Winter Guerra, Ezra Tal, Varun Murali, Gilhyun Ryou, and Sertac Karaman. FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019.

[18] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014.

[19] Wei-Wen Kao. Integration of gps and dead-reckoning navigation systems. In *Proc. of the Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 635–643. IEEE, 1991.

[20] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.

[21] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs,

Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2022.

[22] Ratnesh Madaan, Nicholas Gyde, Sai Vemprala, Matthew Brown, Keiko Nagami, Tim Taubner, Eric Cristofalo, Davide Scaramuzza, Mac Schwager, and Ashish Kapoor. Airsim drone racing lab, 2020.

[23] Jovan Menezes, Shubhankar Das, Bhavik Panchal, Nitesh P. Yelve, and Praseed Kumar. Mapping, trajectory planning, and navigation for hexapod robots using ros. In Kannan Govindan, Harish Kumar, and Sanjay Yadav, editors, *Advances in Mechanical and Materials Technology*, pages 851–866, Singapore, 2022. Springer Nature Singapore.

[24] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In Itsuki Noda, Noriaki Ando, Davide Brugali, and James J. Kuffner, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[25] Illah R Nourbakhsh, Katia Sycara, Mary Koes, Mark Yong, Michael Lewis, and Steve Burion. Human-robot teaming for search and rescue. *IEEE Pervasive Computing*, 4(1):72–79, 2005.

[26] Dimitri Ognibene, Tom Foulsham, Letizia Marchegiani, and Giovanni Maria Farinella. Active vision and perception in human-robot collaboration. *Frontiers in Neurorobotics*, 16:7, 2022.

[27] OpenAI. OpenAI Platform — platform.openai.com. `https://platform.openai.com/docs/models/gpt-3-5`, 2022. [Accessed 07-Jul-2023].

[28] Easton Potokar, Spencer Ashford, Michael Kaess, and Joshua G. Mangelson. Holoocean: An underwater robotics simulator. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3040–3046, 2022.

[29] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *Proc. of the Computer Vision – ECCV 2016 Workshops*, pages 909–916. Springer International Publishing, 2016.

[30] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images

for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[31] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research, 2019.

[32] Matterport sensor. Capture, share, and collaborate the built world in immersive 3D — matterport.com. `https://matterport.com/`. [Accessed 07-Jul-2023].

[33] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles, 2017.

[34] Vikram Shree, Beatriz Asfora, Rachel Zheng, Samantha Hong, Jacopo Banfi, and Mark Campbell. Exploiting natural language for efficient risk-aware multi-robot sar planning. *IEEE Robotics and Automation Letters*, 6(2):3152–3159, 2021.

[35] Vikram Shree, Wei-Lun Chao, and Mark Campbell. Interactive natural language-based person search. *IEEE Robotics and Automation Letters*, 5(2):1851–1858, 2020.

[36] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A flexible quadrotor simulator, 2021.

[37] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces, 2019.

[38] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *CoRR*, abs/1804.10332, 2018.

[39] Unity Technologies. Unity Real-Time Development Platform — 3D, 2D,

VR and AR Engine — unity3d.com. https://unity3d.com/, 2023. [Accessed 07-Jul-2023].

[40] Vaibhav V. Unhelkar, Przemyslaw A. Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A. Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3(3):2394–2401, 2018.

[41] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022.

[42] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam, 2020.

[43] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[44] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] Ruolin Ye, Wenqiang Xu, Haoyuan Fu, Rajat Kumar Jenamani, Vy Nguyen, Cewu Lu, Katherine Dimitropoulou, and Tapomayukh Bhattacharjee. Rcareworld: A human-centric simulation world for caregiving robots, 2022.