DEVELOPMENT OF A NOVEL TELEMANIPULATION FRAMEWORK FOR HUMAN-ROBOT COLLABORATION USING PTC THINGWORX AND VUFORIA STUDIO

by

Preet Parag Modi

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Engineering

at

The University of Wisconsin-Milwaukee

May 2022

ABSTRACT

DEVELOPMENT OF A NOVEL TELEMANIPULATION FRAMEWORK FOR HUMAN-ROBOT COLLABORATION USING PTC THINGWORX AND VUFORIA STUDIO

by

Preet Parag Modi

The University of Wisconsin-Milwaukee, 2022
Under the Supervision of Professor Mohammad Habibur Rahman

Significant advancements in contemporary telehealth care applications are enforcing the demand for effective and intuitive telerehabilitation tools. The current techniques for observing live process parameters of robots frequently require complex and inefficient methods, which fundamentally limits the human administrator's ability to settle on the most educated decisions possible. Telemanipulation can minimize the distance and costs in varieties of robot applications, including industry for object manipulation and robot-aided rehabilitation. This research aims to develop a novel telemanipulation framework to deliver robot-assisted rehabilitation using PTC ThingWorx's Industrial Internet of Things (IIoT), and Vuforia Studio's Augmented Reality (AR) platforms. This communication architecture is stable and has low latency for teleoperation. Our research shows that a remote operator (e.g., a remote therapist) can provide a variety of upper-limb rehab exercises using the developed AR-based graphical user interface (GUI), which transmits bidirectional data between the AR platform and the robot through the ThingWorx IIOT platform. For the kinematic modeling of the robotic manipulator (xArm-5 robot) used in this study, modified Denavit-Hartenberg (DH) parameters were used. The developed AR platform displays the robot data in real-time, such as joint angles, velocity, motor current, etc., thus allowing an operator (e.g., a caregiver/therapist) to monitor the robot motion representing various upper-limb rehabilitation

therapies. This research also developed a telemanipulation approach using a physical joystick to teleoperate the xArm-5 robot. The experimental results revealed that the xArm-5 can be teleoperated successfully and effectively using the developed AR platform and a physical joystick to provide upper-limb telerehabilitation therapy.

Keywords: Augmented Reality, Telemanipulation, Upper-limb Rehabilitation, Telerehabilitation, xArm-5 robot, Rehabilitation Robot, IIoT, ThingWorx, Vuforia Studio.

To

my parents & grandparents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I should express my gratitude to God (Becharaji and Sathya Sai) that it would not have been possible without his blessings. I want to express my love, respect, and prayer for my parents and grandparents because of their sacrifice for me. I would like to acknowledge my advisor and mentor Dr. Mohammad Habibur Rahman, for his immense support and guidance. I am glad to work with Dr. Rahman for my research in BioRobotics Lab. It is always a pleasure to work under the supervision of someone who gives research-related support and gives positive counseling in life's difficult situations, and I have found that kind of personality in him. I also want to thank Dr. Ilya Avdeev, Dr. Jacob R Rammer, and Dr. Md Rasedul Islam for accepting my request to be on the master's comprehensive exam committee. My heartiest Thanks to Raga Madhuri Podugu, for all her love and support. I am also grateful to all my lab colleagues for their support, and I always found them well-disposed toward me even with their busy schedules.

CHAPTER 1

INTRODUCTION

Stroke affects over 750,000 individuals each year [1], and most stroke survivors have impairments in varying degrees of their motor function [2]. After a stroke, the most frequent problem is upper extremity (UE) dysfunction, including muscle weakness, spasms, and multi-joint coordination problems [3]. Approximately 85% of stroke survivors will experience UE dysfunction (UED) in the early stages [4]. Patients with motor dysfunction lose their self-care and employment capacities, decrease their life quality, and bring a heavy economic burden on their families [5]. It is critical to focus on the rehabilitation of the UE to enhance the quality of life that is compromised after a stroke. A therapist is usually dedicated to a stroke patient during hospitalization to provide one-on-one therapy. A therapist can only train one patient at a time, leading to the high demand for therapists [6]. Recent studies have shown that robots can help recover from neurological diseases such as stroke. Robots are reliable and consistent at repeating tasks [7]. A consistent and successful rehabilitation process can be achieved with robot-assisted treatment [8-10]. Furthermore, a rehabilitation robot can provide reliable information about functional assessment of patient recovery [11].

The modern era has seen a surge in the development of first-generation robot-assisted rehabilitation equipment and methods, which began in the 1960s [12]. Different set-up configurations are developed according to the purpose of exoskeleton robots: rehabilitation, haptics, assistive device, teleoperations, and power augmentation [13]. Existing exoskeleton-type therapeutic robots possess some risk since it is directly attached to the human body [14]. Compared to exoskeleton-type robots, the end-effector therapeutic robots are safer and suitable for many subjects who face difficulties in wearing an exoskeleton-type robot [15].

Telerehabilitation uses information technology (IT) to provide individuals with physical and/or cognitive disabilities with remote assistance and evaluation [16]. The remote operating system for rehab in medical applications can minimize the distance between patients and therapists, allowing for better and more frequent communication. Further, the data collected (such as types of exercise and patient-specific recovery trends) during the rehabilitation treatment process will increase the understanding of therapists on the recovery of patients and will increase the range of possible applications. The importance of using telerehabilitation appears to improve economic barriers by reducing the cost and time of traveling [17]. Another benefit of telerehabilitation is that a single therapist could potentially administer more patients per day [18].

Therefore, to contribute to this area, this research aims to develop a novel telemanipulation framework for human-robot collaboration using PTC ThingWorx and Vuforia studio. This novel framework helps to provide upper-limb telerehabilitation using an interactive five-degree of freedom end-effector robot, xArm-5. This system is a cross-platform, collaborative, and comprehensive telerehabilitation system that leverages the power and reliability of the ThingWorx cloud platform and Vuforia Studio's AR platform to deliver a cutting-edge experience to both patients and providers [19]. The system uses the secure internet Hypertext Transfer Protocol Secure (HTTPS) protocol provided by the ThingWorx server. Using Vuforia Studio, we have designed a user-friendly graphical user interface with virtual buttons to control and visualize the rehab robot, xArm-5, remotely [20]. This interactive approach will allow patients to take rehabilitation therapy at home even after the COVID-19 pandemic abates. Furthermore, the developed system provides a key platform for storing rehabilitation-related data and exercise results to analyze patients' recovery.

1.1 Research Goals

The specific aims of this research project are as follows:

Aim 1: *Develop a novel, low latency, and intuitive telemanipulation framework to enhance human-robot collaboration.*

Aim 2: *Develop a digital twin of a 5DoF robotic manipulator and an Augmented Reality user interface for telemanipulation of a rehab robot (xArm-5).*

Aim 3: *Perform human subject experiments to evaluate the developed telerehabilitation system to provide various robot assistive rehabilitation exercises.*

This thesis is organized in chapters as described below.

CHAPTER 2: Background and Literature Review

This chapter represents the related work on robot-assisted telerehabilitation.

CHAPTER 3: Robot-Assisted Telerehabilitation

The purpose of this chapter is to discuss rehabilitation, telerehabilitation, neuroplasticity, neurorehabilitation techniques, robot-assisted rehabilitation, and robot-assisted telerehabilitation.

CHAPTER 4: xArm-5 Robot Specification, Workspace, Features, Kinematics, and Control

This chapter introduces the design, specification, workspace, features, kinematics, and control of the xArm-5 robot. The modified Devavit-Hartenberg (DH) notations are used to develop the kinematic model of the xArm-5 robot.

CHAPTER 5: Digital Twin of An Assistive Robot

This chapter describes the digital twins, digital threads, and a detailed procedure for developing a digital twin of the xArm-5 robot using PTC Vuforia Studio.

CHAPTER 6: Telemanipulation Framework for Human-Robot Collaboration

This chapter describes the development of a telemanipulation framework component for human-robot collaboration using PTC ThingWorx and a detailed configuration of ThingWorx composer to communicate with the xArm-5 robot and Vuforia Studio.

CHAPTER 7: Experiments and Results

This chapter presents the telemanipulation framework experimental setup and graphical representation of the experiment results.

CHAPTER 8: Conclusions and Future Work

In the conclusion section, the research findings are summarized, and some further research directions are presented.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

This chapter discusses state-of-the-art technologies and methods used in telerehabilitation.

2.1 Related Work on Telerehabilitation

Telehealth interventions employ telecommunications and/or virtual environments, including virtual reality (VR), augmented reality (AR), or mixed reality (MR), to provide healthcare at a distance. Telerehabilitation is a subset of telehealth that enables a clinician/therapist/caregiver to deliver rehabilitation services to patients at a distance [21]. To enable pulmonary rehabilitation treatments to be supplied to a satellite healthcare center or directly to the patient's home, remote contact between the patient and healthcare practitioner may include telephone, internet, or webcam technology [22]. Individuals who are regionally or socially isolated, patients who work or study full-time, or those who find traveling difficult owing to illness severity or complications may benefit from telerehabilitation. There is evidence that some Chronic obstructive pulmonary disease (COPD) patients who attend pulmonary rehabilitation are enthusiastic about using telerehabilitation services [23]. Telerehabilitation models may include additional aspects of center-based pulmonary rehabilitation, such as self-management education and illness management education, and exercise instruction. Telerehabilitation methods for cardiac rehabilitation offer the potential to improve pulmonary rehabilitation service uptake and accessibility for all patients with chronic respiratory problems.

Rehabilitation is required to regain people's lost functionality or maximize their functioning and quality of life. While rehabilitation is a multidisciplinary, multicomponent, and comprehensive intervention, the particular health condition and other predictors of the healthcare system or

available resources identify the bare minimum required components, which frequently incorporate physical therapy. The consequences of a disability can be felt across the neighborhood, culture, and economy.

Telerehabilitation, also known as teletherapy, involves the use of information and telecommunication technologies such as the telephone, video conferencing, and virtual reality to assist patients in receiving medical services and therapies remotely [24-26]. Telerehabilitation's technological capabilities are typically achieved through video conferencing and include the capacity for therapists to see patients' movements while performing rehabilitation tasks. The therapists in charge of telerehabilitation programs typically use telephone technology at the beginning of implementation [27]. With the invention of the Internet in the 20th century, the telerehabilitation implementation project has become more realized. The usefulness and efficiency of telerehabilitation implementation for stroke patients are expected to improve when the Internet is upgraded with high-speed wireless [17].

The year 1997 marks the release of the first scientific paper on telerehabilitation [28]. Lewis and Griffin (1997) have explored the VR technology in multiple rehabilitation areas and concluded that this technology has great potential in healthcare [28]. The study has analyzed that some users experienced adverse effects during and after experiencing VR environments, including disorientation, balance disturbances, nausea, and ocular problems [28]. Figure 2.1 shows the number of patients treated from 1998 to 2008 through telerehabilitation techniques [29]. Kenyon et al. (2004) in [30] examined whether or not virtual environments are beneficial for therapeutic therapies that require adaptation to sophisticated, multimodal contexts.

Carey et al. (2007) conducted a literature review in the rehabilitation area of physiotherapy, particularly cortical reorganization after stroke [31]. They found that patients suffering from

chronic stroke may benefit from telerehabilitation [31]. Parmanto and Saptono (2007) researched the multiple Telerehabilitation areas from an informatics perspective. They concluded that healthcare practitioners were not prepared to manage a telerehabilitation strategy that is used on a daily basis [32]. Zampolini et al. (2008) performed a literature review on multiple rehabilitation areas from the standpoint of telerehabilitation practices in the future. They found technological limitations, and physicians were not ready yet to adopt this practice [33]. The same conclusion has been drawn by the study conducted in [29].

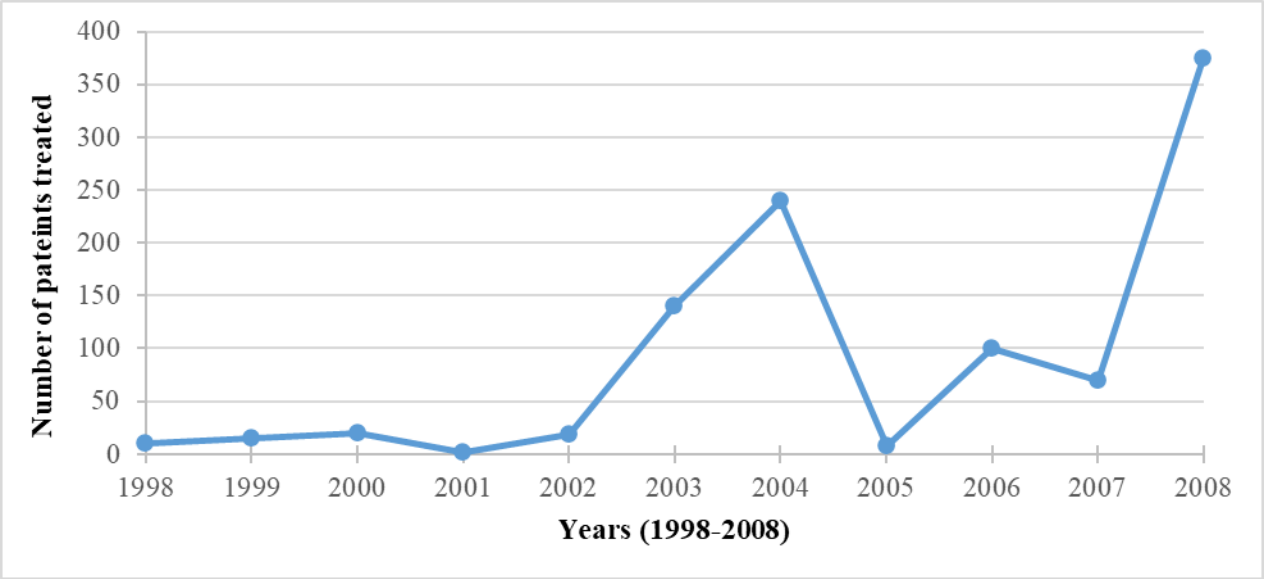


Figure 2.1 Number of patients treated from 1998 to 2008 through telerehabilitation techniques [29]

Ackerman et al. (2010) analyzed the implications of VR technology in broader rehabilitation areas by considering it as the next-generation telehealth tool at that time [34]. The associated persons were not ready to adopt this technology because of incompatibility in telerehabilitation and dataflow [34]. Larson et al. (2014) analyzed multiple rehabilitation research using VR technology and found telerehabilitation using VR is a promising approach [35]. Weiss et al. [36] proposed a telerehabilitation system for passive and assisted wrist and finger training using an end-effector device. This system includes a data gathering and communication module for remote monitoring,

visual biofeedback, and gaming applications for the patient. The system contains rehabilitation games as well as patient feedback on therapeutic progress.

A few research groups have been developing sophisticated robotic devices that can be operated remotely and can be worn by a person to manage the remote slave robot (that can also act as a master robot, i.e., exoskeletons) [37, 38]. Zhang et al. (2015) [39] developed a telerehabilitation system based on an exoskeleton device that can be controlled remotely as a slave device by a therapist through a master device, but the therapist can only monitor the patient's status visually via a web camera.

Kairy et al. (2016) performed research on 104 patients to provide upper limb rehabilitation through VR. They concluded that once patients are discharged from rehabilitation, this strategy has the potential to improve continuity of care [40].

Tchero et al. (2018) performed an empirical research study and meta-analysis to determine whether or not telerehabilitation is effective for those who have had a stroke [91][41]. Their findings show that telerehabilitation may be a viable alternative to traditional rehabilitation in post-stroke patients, particularly in rural or underserved regions of the country.

Appleby et al. (2019) conducted a systematic review to analyze telerehabilitation's effectiveness in managing adults having stroke problems [42]. Their study also finds telerehabilitation as an alternative rehabilitation method for post-stroke patients. The investigation concluded that an unequivocal suggestion could not be made because of methodological and practical difficulties. Using the findings of this research, future policies and practices surrounding the use of telerehabilitation for stroke patients may be developed and implemented.

Camden et al. (2019) investigated the range of telerehabilitation methods for children with disabilities and the characteristics of effective interventions [43]. This study focused on randomized control trials in which at least one rehabilitation professional delivered services remotely [43]. There is a wide range of teleintervention features and populations. Effective therapies tended to be directed to the parents, focused on an exercise program (8+ weeks long), aimed to improve children's behavioral functioning, and were delivered at least once a week [43].

Seron et al. (2021) studied the effectiveness of telerehabilitation in physical therapy. Systematic reviews of various ailments, groups, and situations were included, with the intervention being telerehabilitation by physical therapy [44]. Clinical efficacy was determined by the functionality, particular condition, quality of life, adherence, safety, and satisfaction of the participants. Despite the contradictory results, the researchers concluded that telerehabilitation in physical therapy could be equivalent to in-person rehabilitation or preferable to no rehabilitation for circumstances like low back pain, osteoarthritis, knee and hip replacement, multiple sclerosis, and cardiac and pulmonary rehabilitation. The study implies that equipping professionals, primarily physical therapists, with the best available information on the usefulness of telerehabilitation would influence decision-making and result in improved clinical results for patients, both now and in the future, during the COVID-19 pandemic.

In the fourth Industrial Revolution, using modern smart technology and IIoT are interconnected to boost automation, connectivity, self-monitoring, and the development of intelligent machines that can interpret and diagnose problems through teleoperation by using human interaction [45]. Robotic devices use computing technologies to create a virtual environment that mimics/digital twins of the real environment. Virtual technologies have the potential to create a controlled and safe virtual environment that can later be applied to real-life healthcare situations. *Therefore, in*

this research, we develop a novel telemanipulation framework using IIoT and AR cloud-based platforms to facilitate human-robot collaboration. The developed framework was tested to simulate providing robot-assisted telerehabilitation.

CHAPTER 3

ROBOT-ASSISTED TELEREHABILITATION

This chapter discusses rehabilitation, telerehabilitation, neuroplasticity, neurorehabilitation techniques, robot-assisted rehabilitation, and robot-assisted telerehabilitation.

3.1 Rehabilitation

According to the World Health Organization (WHO) [46], 'rehabilitation' is defined as "a set of interventions designed to optimize functioning and reduce disability in individuals with health conditions in interaction with their environment."

3.2 Telerehabilitation

Telerehabilitation is a method for delivering rehabilitation services over the internet and telecommunications networks [47]. Rehab/clinical services can be provided remotely to patients through telerehabilitation, which can be used to assess patients and provide therapy [47].

3.3 Different Types of Rehabilitation

Rehabilitation can be divided into two major categories: Sensory Rehabilitation and Motor Rehabilitation. A sensory rehabilitation program restores the basic functions of the five traditional senses of the human body. Senses such as sight, hearing, smell, and taste can be restored with augmentation or via sensory substitution systems. The objective of motor rehabilitation is to assist patients in overcoming movement impairments that have a negative impact on their quality of life. Motor rehabilitation aims to improve patients' functional health and everyday activities, including relearning previously learned movements and skills lost due to disease or sensory, motor, or cognitive impairment.

3.4 Neuroplasticity

A network of neural cells in the brain can change, modify, and adapt both structure and function throughout life growth and reorganization, a phenomenon known as neuroplasticity and is also known as neural plasticity or brain plasticity [48, 49]. There are two types of neuroplasticity: Functional plasticity is the movement of brain functions from a damaged area of the brain to an undamaged area of brain, where the brain can change its physical structure due to its Structural plasticity as it learns. Kleim and Jones have reviewed ten principles of experience-dependent neural plasticity and considerations in applying them to the damaged brain [50], and those principles are explained below from the source material [50],

1. **Use It or Lose It:** Degradation of brain function can result from failure to drive specific brain functions.
2. **Use it and improve it:** Enhancing a specific brain function with training can improve abilities.
3. **Specificity:** Training experiences affect the degree of brain plasticity (change in brain structure).
4. **Repetition matters:** A sufficient amount of repetition is necessary for change (plasticity).
5. **Intensity matters:** In order to change (plasticity), intensive training is needed.
6. **Time matters:** Different forms of change (plasticity) in the brain happen at different times during training.
7. **Salience matters:** During training, the brain undergoes different kinds of change (plasticity).
8. **Age matters:** Younger human brains have a greater capacity for training-induced change (plasticity).

9. **Transference:** One training experience can influence the functioning of another and even lead to the gain of other similar skills.
10. **Interference:** Bad habits can change the brain plasticity, which can obstruct learning good habits.

3.5 Principles of Neurorehabilitation

Maier et al. [51] outlined principles of neurorehabilitation after stroke based on motor learning and brain plasticity mechanisms are as follows:

1. **Massed practice/repetitive practice:**

   The massed practice comprises work episodes in which individuals repeatedly practice a task. Several animal studies have indicated that repetitive motor behavior can cause major changes in the area responsible for motor planning. Evidence showed that repetitive practice depends on the duration or number of sessions. The repetitions should be quantified during sessions or therapies to determine the effects of massed practice [51].

2. **Spaced Practice**

   Spaced practice, also called distributed practice, suggests that work episodes must consist of rest periods. The exact mechanism of action behind the distributed practice is still unknown [51]. Various studies imply that final test performance can be improved by breaking the training process into smaller sessions. This can help facilitate long-term memory formation [51].

3. **Dosage**

   In rehabilitation, the dosage can be defined as the total number of hours spent in therapy, duration of a training session, frequency of training sessions, or amount of practice needed

to stimulate learning. Some evidence suggested that increasing the number of therapy hours would be helpful to speed up the recovery process. Although, some findings revealed no benefit of increased dosage early after stroke. Therefore, the accurate dose-response still needs to be determined [51].

4. **Specific Practice**

Task-specific practice suggests that to change the conditions of a task, one must need a change in the abilities required to perform it. Based on this principle, rehabilitation allows the patients to execute activities of daily living (ADL) without any help. This helps the patients to obtain fundamental movement skills [52]. Several studies suggested that task-specific practice improves motor functionality, learning, and retention. While some studies also found decreased activity in the contralesionally cortex. Generally, motor skill performance can be increased if more attention is given to the effect of movement [51].

5. **Goal-Oriented Practice**

Goal-oriented practice does not focus only on movement patterns but also helps the patients look at couplings that are convenient to complete the task [53]. Generally, motor skill performance can be enhanced if more attention is given to the effect of movement [54]. Goal-oriented activities provide good performance compared to the same exercises without any goal [55].

6. **Variable Practice**

There are two ways to achieve variable practice: (1) variability of practice and (2) random practice. Both courses are highly effective in untrained tasks or movements [51]. Functional magnetic resonance imaging (fMRI) and transcranial magnetic stimulation (TMS) studies determined that variable practice enhances neuronal activity, especially in

motor learning areas [56]. The effectiveness of variable practice might be associated with neuromodulatory systems that maintain neuronal plasticity, such as the dopaminergic system [57].

7. Increasing Difficulty

Task difficulty is typically associated with many problems or tasks that explore motor learning. Task difficulty can be easily determined by the requirements for training and other conditions related to the task or simply how difficult the activity is compared to an individual's skill. Prediction errors can be minimized by more practice [58]. If individuals can manage the task difficulty, their motor performance will be significantly better [59]. But if difficulty exceeds an individual's abilities to achieve success, it may negatively affect their overall performance [60].

8. Multisensory Stimulation

The sensorimotor control and perception of various senses are significant brain abilities. A study conducted on a cat indicated that a single neuron could be responsible for various sensory modalities [61]. These pieces of training benefits patients recovering from visual, unimodal, or auditory deficits [62].

9. Rhythmic Cueing

Neuroentrainment includes the study of the temporal association between the movement of bodies and rhythmic stimulation generated from the surrounding environment. Stimulus or sensory modality includes visual, hearing vestibular that can be utilized for entrainment. The auditory-motor synchronization mainly controls the internal rhythmic movement [63]. Therefore, auditory cues play a crucial role in synchronizing movements to rhythmic patterns [64]. A study suggested that neuronal connections exist between the auditory and

motor systems [65]. Several meta-analyses found that rhythmic cueing helps improve upper-limb dysfunctions, especially after stroke [66].

10. Explicit Feedback/Knowledge of Results

Explicit Feedback can be described as feedback that can be verbal, terminal, or augmented about certain goals. This feedback can be more significant than just extrinsic rewards [67]. This type of feedback uses cognitive processing and is provided through explicit feedback [68]. It is given on quantitative or qualitative process results, for example, success or failure [51]. A study was conducted to analyze different feedback types; it was reported that explicit feedback has positive effects on motor function [69].

11. Implicit Feedback/Knowledge of Performance

It is defined as the type of feedback in verbal explanations or demonstrations about moment execution. With the latest technology, implicit feedback can be provided online in an implicit manner. VR representations are also a technique to visualize arm movements [51].

12. Modulate Effector Selection

Severe pain and weakness can typically reduce affected limb use after a stroke. As a result, the under-used impaired limb can experience a loss of neuronal activity. Constraint-induced movement therapy (CIMT) is the most successful and popular therapeutic approach to tackle this issue. An fMRI study indicated that the CIMT program is beneficial in increasing the strength and usage of the affected limb [51].

13. Action Observation/Embodied Practice

A study conducted on humans observed that individuals who performed the specific task after observing the performance of other individuals performed better [51]. A meta-

analysis indicated that human parietal and promotor areas are responsible for movement execution [51]. Therefore, action observation can be significant for stroke patients due to its brain activation abilities.

14. Mental Practice/Motor Imagery

Motor Imagery and Mental Practice is a process in which an individual imagines a psychological rehearsal of future movements. The motor imagery provides learning effects and clinical efficacy, but the physical practice has more desirable results on learning. Motor imagery can also benefit patients undergoing stroke treatment. A meta-analysis also supported mental practice's positive findings [51].

15. Social interaction:

Social interaction can be described as a social exchange between two individuals. A failure to perform it can produce many undesirable effects. A study indicated that animal retrieving from an artery occlusion showed positive results with social interaction [51].

3.6 Robot-Assisted Upper Limb Rehabilitation

The robot-assisted rehabilitation program should incorporate the activities outlined in sections 3.4 and 3.5 and provide a variety of upper-limb rehabilitation exercises, including passive, active-assist, resistive, and active exercises. Passive rehabilitation exercises do not take any movement from the patient. In this exercise, the patient receives assistive or directed movement from an external source, such as a physical therapist or a rehab device. This exercise aims to increase patients' range of motion (ROM). The rehab device will move the patient’s arm in 2D/3D space in passive rehabilitation mode, mainly to reduce spasticity and increase the ROM. Active-assistive rehabilitation is defined as exercises where the patient actively uses the muscles to complete a specified task. For instance, the subject will be given a task to trace a straight line. If the patient

fails to complete the task, the rehab device will help the patient to complete the task. The goal of this exercise is to increase muscle strength. In resistive rehab exercises, the patient also actively uses the muscles to achieve a specified task. For instance, the subject will be assigned to trace a square box on screen/paper. In this case, the rehab device will moderately provide resistance to motion to make it difficult for the patient to complete the task. The goal of this exercise is to increase muscle strength. In active exercise, the patient mainly performs stretching exercises (in 2D/3D space) without external assistance/resistance.

Therefore, to provide these various types of rehabilitation therapy, we have developed a telemanipulation framework for human-robot collaboration, and through this system, patients can get motor rehabilitation therapy remotely.

CHAPTER 4

xArm-5 ROBOT SPECIFICATION, WORKSPACE, FEATURES, KINEMATICS AND CONTROL

4.1 Background

This chapter outlines the design, specification, workspace, features, and kinematics of the xArm-5 robot.

4.2 Specifications of xArm-5 robot

The xArm-5 robot developed by UFACTORY is a high-precision multi-function robotic manipulator designed explicitly for desktop use. To test and validate the proposed telerehabilitation framework to provide robot-assisted upper-limb rehabilitation, we have used the xArm-5 robot (see Fig. 4.1). The xArm-5 is a 5 DoFs robot with blushless servos, harmonic reducers, and a 6-axis force-torque sensor at the end-effector. Its position repeatability is ± 0.1 mm [70]. It uses Modbus-TCP and Modbus-RTU communication protocols [70]. Table 4.1 summarizes the specification of the xArm-5 Robot.

Table 4.1 Specification of xArm-5 Robot [71]

Specifications	
Number of Axes	5
Payload	3kg
Max. Reach	700mm
Weight (robotic arm only)	11.2kg
Position Repeatability	± 0.1mm
Max. Joint Speed	180°/s
Max. Speed of End-Effector	1m/s
Working Temperature Range	0-50 °C
Robotic Arm Communication Protocol	Modbus-TCP
End-effector Communication Protocol	Modbus-RTU
Input Power Supply	24 V DC, 16.5 A
Power Consumption	Min 8.4 W, Typical 200W, Max 500W

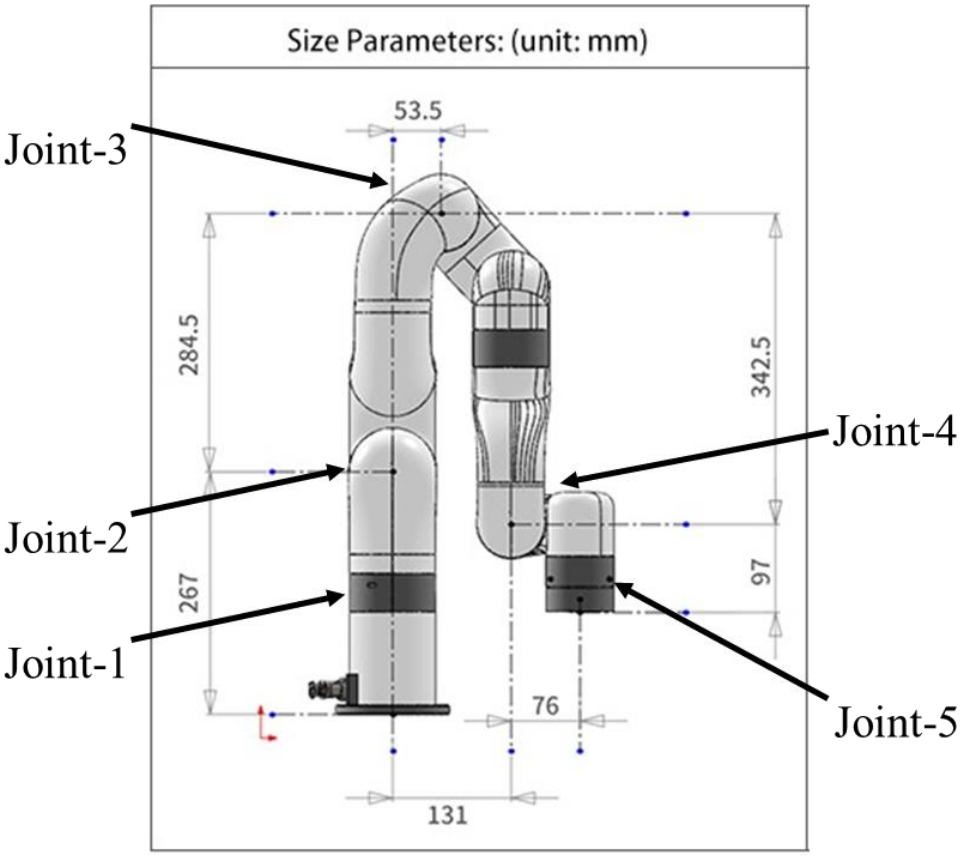


Figure 4.1 Size Parameters [72]

4.3 Physical Properties of the xArm-5 robot

The xArm-5 is made of Carbon fiber and Aluminum, and it weighs around 11.2kg (arms, brushless servo motors, and end-effector tool head), this robot system comes with a DC control box and configuration of DC control box is described in table 4.2.

Table 4.2 DC Control Box of xArm-5 Robot [71]

DC Control Box	
Input	24VDC
Output	24VDC 16.5A
Control Box Communication Protocol	Modbus TCP
Control Box Communication Model	Ethernet
Control Box I/O Interface	8*CI (Digital In), 8*CO (Digital Out) 2*AI (Analog In), 2*AO (Analog Out)
Weight	1.6kg
Dimension(L*W*H)	180*145*68mm

4.4 Workspace of xArm-5 robot

The maximum reach of xArm-5 is 700 mm. In this maximum reach of xArm-5, joint 1, 2, 3, 4, and 5 (Fig. 4.1) correspond to working range -360º to +360º, -118º to +120º, -225º to +11º, -97º to +180º, and -360º to +360º, respectively [70]. The front view and top view of workspace is depicted in Fig. 4.2.

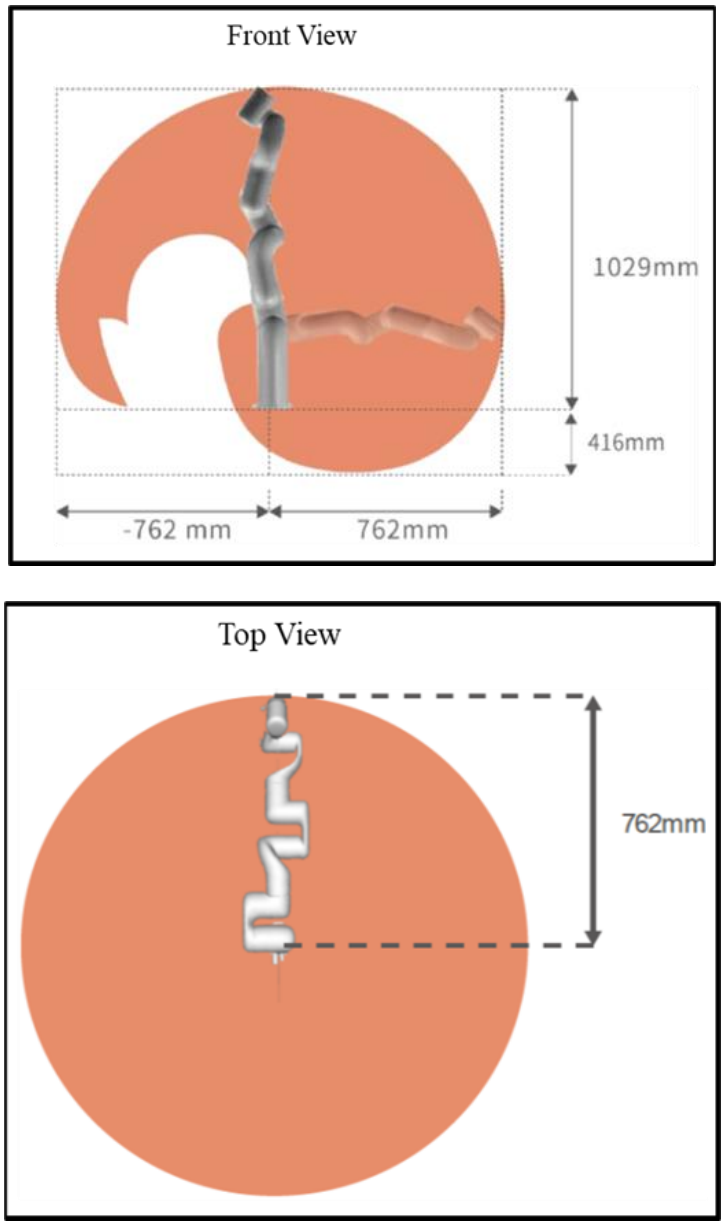


Figure 4.2 Workspace [71]

4.5 Application Software and SDK

The graphical user interface (GUI) for operating the robot is called xArm Studio is developed by Ufactory and it is represented in Fig. 4.3. XArm Studio allows users to plan the robot's motion trajectory without knowing how to program it. By dragging and dropping Blocky code blocks, users may define parameters, move the robot in real-time, and design a motion trajectory with this software [71]. The manufacturer also offers Python, ROS, and C++ SDK for more user integration flexibility. In our study, we have used Python SDK to control the xArm-5 for advanced functionality.

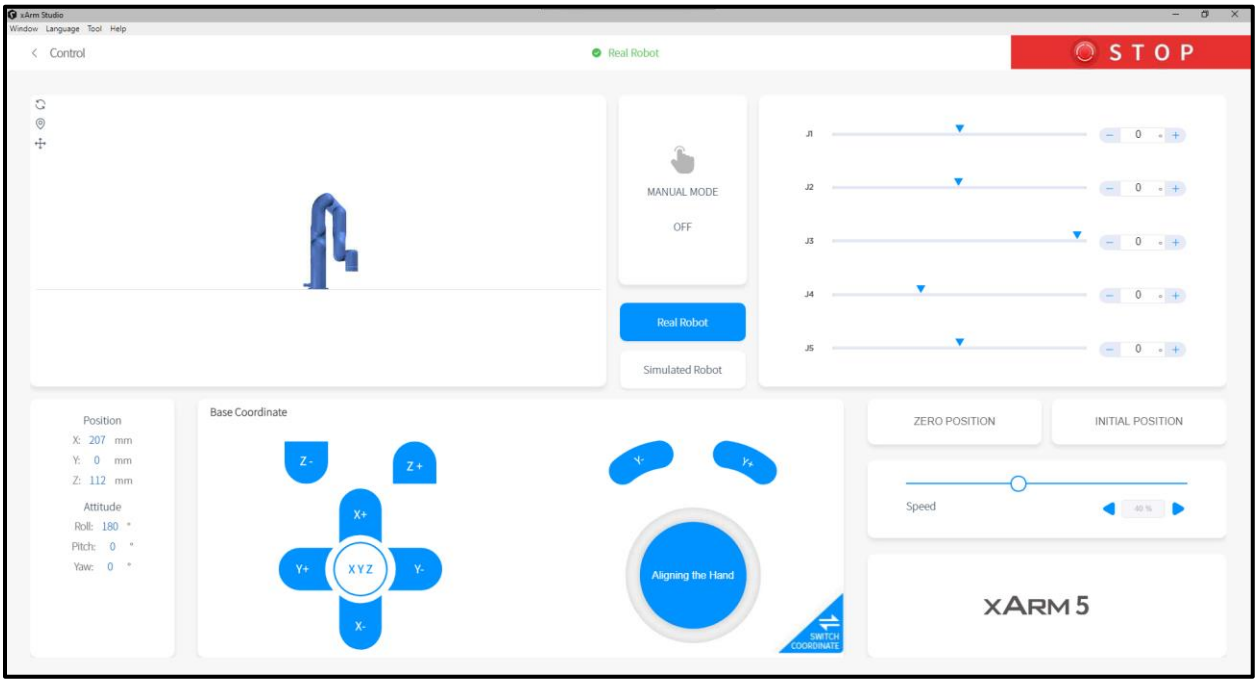


Figure 4.3 xArm Studio Software Interface [71]

4.6 Robotic Arm Kinematics

The xArm-5 robot's kinematic model was created using modified Denavit-Hartenberg (DH) notations. Section 4.6.1 summarizes the coordinate frame assignment technique and the definition of modified DH parameters.

4.6.1 Coordinate Frame Assignment Procedure

There are several ways to assign coordinate frames to the manipulator links. For the xArm-5, we have followed the modified Denavit-Hartenberg method [73, 74]. The steps are as follows [75]:

- Assume each joint is 1 DoF (Degree of Freedom) revolute joint (Fig. 4.4);
- Identify and locate the axes of rotation;

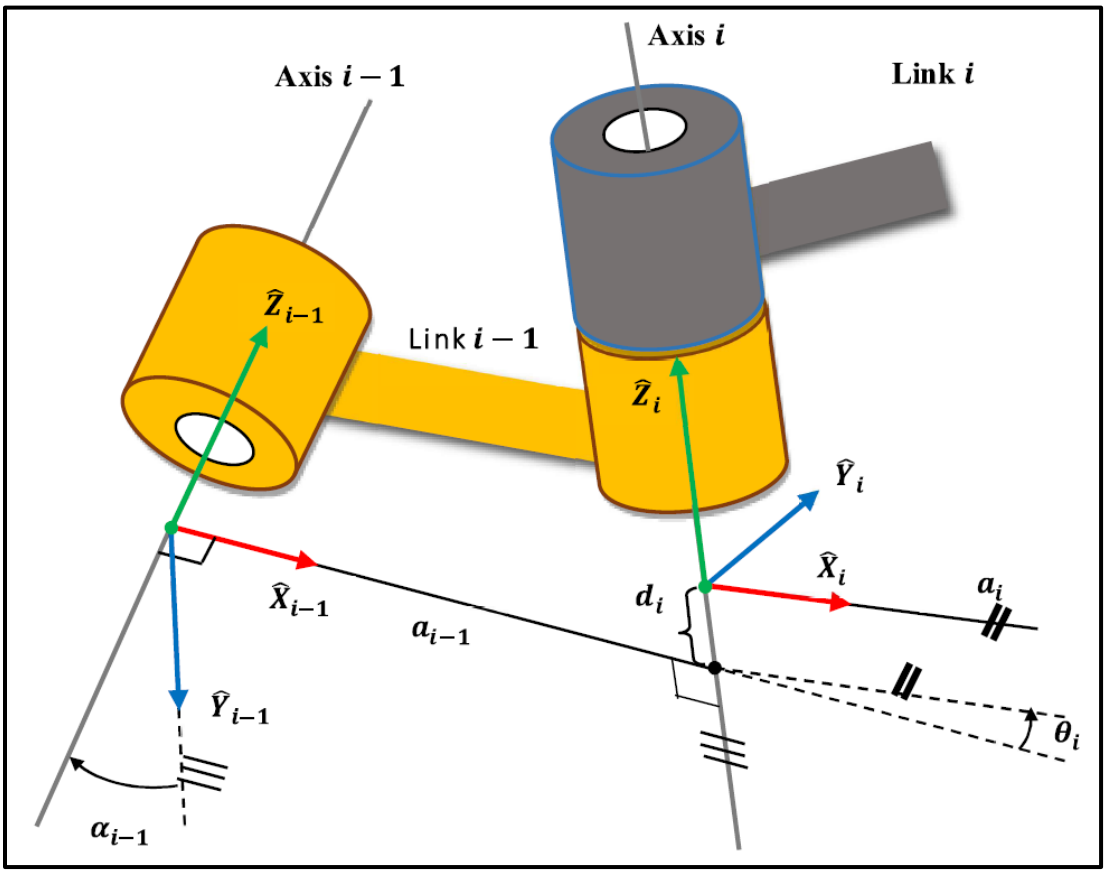


Figure 4.4 Coordinate frame assignment [73, 74]

- Label the joint axes $Z_0,\ldots, Z_n$;

- Locating the origin of each link-frame (O_i) where the common perpendicular line between the successive joint axes (i.e., Z_{i-1} and Z_i) intersects. If the joint axes are not parallel, it is required to locate the link-frame origin at the point of intersection between the axes;
- locating the X_i axis (at link frame origin O_i) as pointing along the common normal line between the axes Z_{i-1} and Z_i. If the joint axes intersect, it is required to establish X_i in a direction normal to the plane containing both axes (Z_{i-1} and Z_i);
- establishing the Y_i axis through the origin O_i to complete a right-hand coordinate system.

4.6.2 Definition of Modified D-H Parameters

A serial link of a robot can be described by four parameters (two parameters for describing the link itself and the other two for describing the link's relation to a neighboring link) if we assign the co-ordinate frames as described above [73, 74]. These parameters are known as Denavit-Hartenberg (DH) parameters. The definitions of the DH parameters are given below [75]:

- Link Length (a_i): the length measured along X_i, from axis Z_i to axis Z_{i+1};
- Link Twist (α_i): the angle measured about X_i, from axis Z_i to axis Z_{i+1};
- Link Offset (d_i): the distance measured along the axis Z_i; from X_{i-1} to X_i, and
- Joint Angle (θ_i): the angle measured about Z_i, from X_{i-1} to X_i

The goal of the forward kinematics is to compute the end-effector position as a function of joint angles. To do this, we first attach reference frames for each link in the xArm-5 robot. Figure 4.5 presents a front view of the xArm-5 robot with link-specific reference frames shown. The intersections between the dotted black lines represent the actual origin positions of the respective reference frames.

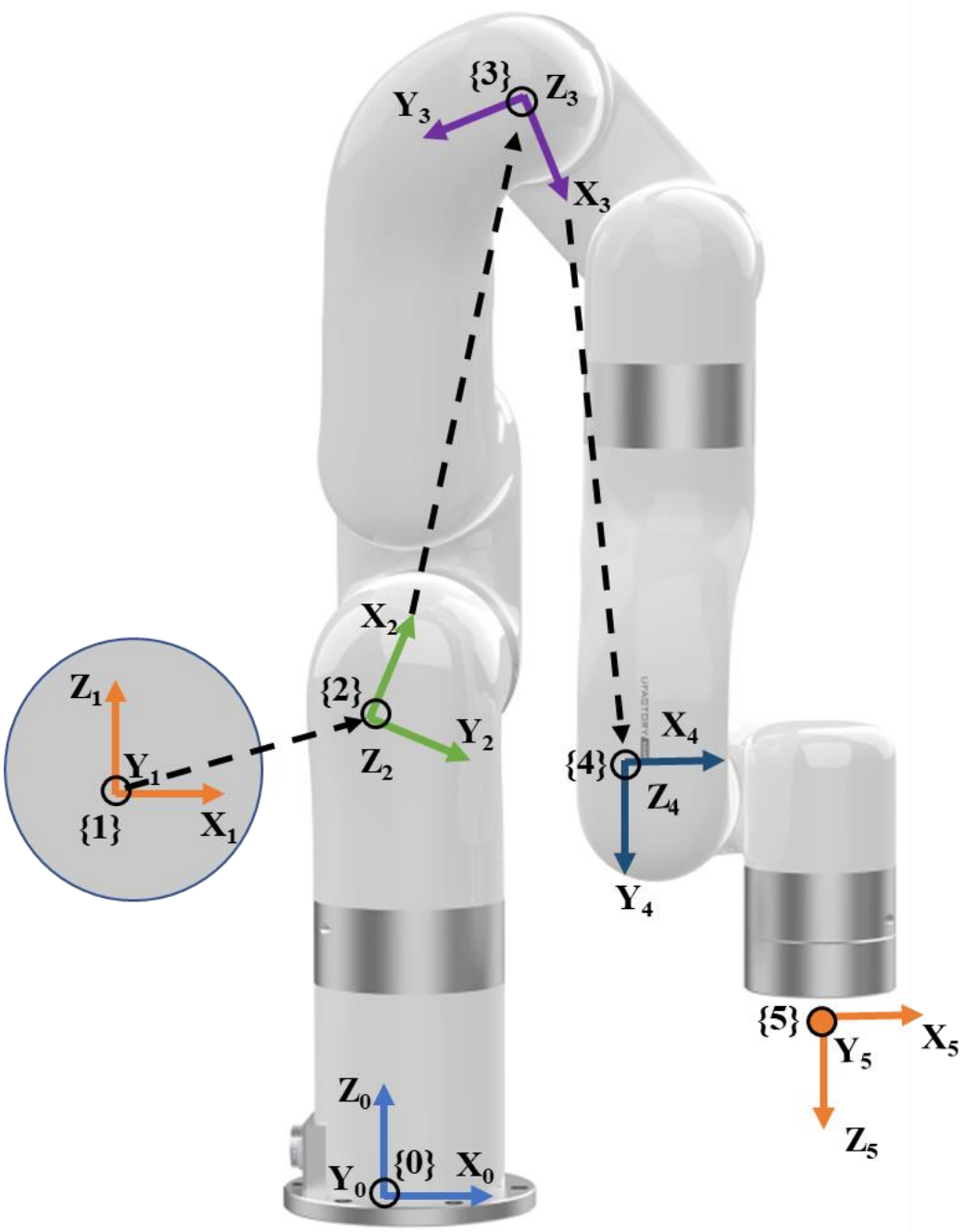


Figure 4.5 Frame Assignment of xArm-5 [72]

To obtain the DH parameters, it is assumed that the coordinate frames (i.e., the link-frames that map one axes of rotation to the successive one) coincide with the corresponding joint axes of rotation, i.e., frame {1} coincides with joint 1, frame {2} with joint 2, and so on. The frame {0} defines the base frame (world frame) of the robot.

As shown in figure 4.5, the joint axes of rotation of the xArm-5 corresponding to joint 1, joint 2, joint 3, joint 4, and joint 5 are indicated with different color arrows with a solid color in the middle where the rotations are around Z axes (i.e., Z_1, Z_2, Z_3, Z_4, *and* Z_5). In this model, frame {1},{2}{3},{4}, and {5} correspond to joint 1, joint 2, joint 3, joint 4 and joint 5 respectively. Joint 1 and joint 5 deliver vertical rotation, and joint 2, joint 3, and joint 4 deliver horizontal rotation.

The origin of frame {1} is moved to frame {2} so that frame {1} and {2} are at a same distance L_1 from base frame {0}. The distance between frame {2} to frame {3} is L_2, the distance between frame {3} to frame {4} is L_3, and the height and the distance between frame {4} to frame {5} is a_4 and L_4 respectively. The end-effector attachment (grippers) aligns with the frame {5}.

Table 4.5 summarizes the modified DH parameters corresponding to the location of the link frames (in figure 4.5). The homogeneous transformation matrix, which describes the locations and orientations of the reference frame with relation to the fixed reference frame, is obtained using these modified DH parameters.

Table 4.3 Modified DH Parameters [72]

Joint (i)	α_{i-1}	a_{i-1}	d_i	θ_i
Joint 1	0	0	l_1	θ_1
Joint 2	$-\pi/2$	0	0	$\theta_2 + {}^0\theta_2$
Joint 3	0	l_2	0	$\theta_3 + {}^0\theta_3$
Joint 4	0	l_3	0	$\theta_4 + {}^0\theta_4$
Joint 5	$-\pi/2$	l_4	l_5	θ_5

Here, i represents the joint, α_{i-1} is the link twist, a_{i-1} corresponds to link length, d_i stands for link offset, θ_i is the joint angle(radian), ${}^0\theta_i$ represents the offset of the joint angle θ_i, and L_i represents the length of i of xArm-5. These ${}^0\theta_i$ and L_i dimensional parameters are presented in Table 4.5 [72].

Table 4.5 Dimensional Parameters of xArm-5

L_1	L_2	L_3	L_4	L_5	${}^0\theta_2$	${}^0\theta_3$	${}^0\theta_4$
267 mm	289 mm	351 mm	76 mm	97 mm	-1.3849 rad	2.7331 rad	-1.3483 rad

The general form of a link transformation that relates frame $\{i\}$ relative to the frame $\{i-1\}$ [73, 74] is:

$$ {}^{i-1}_{i}T = \begin{bmatrix} {}^{i-1}_{i}R^{3\times3} & {}^{i-1}_{i}P^{3\times1} \\ 0^{1\times3} & 1 \end{bmatrix} \tag{4.1} $$

where, ${}^{i-1}_{i}R$ is the rotation matrix that describes frame $\{i\}$ relative to frame $\{i-1\}$ and can be expressed as the following:

$$ {}^{i-1}_{i}R = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} \end{bmatrix} \tag{4.2} $$

and ${}^{i-1}_{i}P$ is the vector that locates the origin of the frame $\{i\}$ relative to frame $\{i-1\}$ and can be expressed as the following:

$$ {}^{i-1}_{i}P = [a_{i-1} \quad -sin\,\alpha_{i-1}\,d_i \quad cos\,\alpha_{i-1}\,d_i]^T \tag{4.3} $$

Using Equations (4.1), (4.2), and (4.3), the individual homogeneous transfer matrices that relate two successive frames of xArm-5 (Fig. 4.5) are as follows:

$$
{}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}T = \begin{bmatrix} \cos(\theta_2 - 1.38) & -\sin(\theta_2 - 1.38) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2 - 1.38) & -\cos(\theta_2 - 1.38) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}T = \begin{bmatrix} \cos(\theta_3 + 2.73) & -\sin(\theta_3 + 2.73) & 0 & L_2 \\ \sin(\theta_3 + 2.73) & \cos(\theta_3 + 2.73) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{4}T = \begin{bmatrix} cos(\theta_4 - 1.35) & -sin(\theta_4 - 1.35) & 0 & L_3 \\ sin(\theta_4 - 1.35) & cos(\theta_4 - 1.35) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}T = \begin{bmatrix} cos\,\theta_5 & -sin\,\theta_5 & 0 & a_4 \\ 0 & 0 & 1 & L_4 \\ -sin\,\theta_5 & cos\,\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}
$$

The homogenous transformation matrix that relates frame {5} to frame {0} can be obtained by multiplying individual transformation matrices that result in the generic form Equation (4.5).

$$
{}^{0}_{5}T = [{}^{0}_{1}T.\,{}^{1}_{2}T.\,{}^{2}_{3}T.\,{}^{3}_{4}T.\,{}^{4}_{5}T] \tag{4.5}
$$

The single transformation matrix found from Equation (4.5) represents the reference frame's positions and orientations attached to the end-effector {5} with respect to the base reference frame {0}.

$$
{}^{0}_{5}T = \begin{bmatrix} u_x & v_x & w_x & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}
$$

where,

$\boldsymbol{u_x} = \sin\theta_1 * \sin\theta_5 + \cos\theta_5 * (0.5 * \cos(\theta_2 - \theta_1 + \theta_3 + \theta_4) + 0.5 * \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4))$

$\boldsymbol{u_y} = -\cos\theta_1 * \sin\theta_5 - \cos\theta_5 * (0.5 * \sin(\theta_2 - \theta_1 + \theta_3 + \theta_4) - 0.5 * \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4))$

$\boldsymbol{u_z} = -0.5 * \sin(\theta_2 + \theta_3 + \theta_4 - \theta_5) - 0.5 * \sin(\theta_2 + \theta_3 + \theta_4 + \theta_5)$

$\boldsymbol{v_x} = -\cos\theta_5 * \sin\theta_1 - \sin\theta_5 * (0.5 * \cos(\theta_2 - \theta_1 + \theta_3 + \theta_4) + 0.5 * \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4))$

$\boldsymbol{v_y} = \cos\theta_1 * \cos\theta_5 + 0.5 * \sin\theta_5 * (\sin(\theta_2 - \theta_1 + \theta_3 + \theta_4) - \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4))$

$\boldsymbol{v_z} = 0.5 * \cos(\theta_2 + \theta_3 + \theta_4 - \theta_5) - 0.5 * \cos(\theta_2 + \theta_3 + \theta_4 + \theta_5)$

$\boldsymbol{w_x} = -0.5 * \sin(\theta_2 - \theta_1 + \theta_3 + \theta_4) - 0.5 * \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4)$

$\boldsymbol{w_y} = 0.5 * \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) - 0.5 * \cos(\theta_2 - \theta_1 + \theta_3 + \theta_4)$

$\boldsymbol{w_z} = -\cos(\theta_2 + \theta_3 + \theta_4)$

The vector that gives the position of the end-effector with respect to frame {0} (Fig. 4.5) is denoted by the following Equation (4.7),

$$ {}^{0}_{5}P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{4.7} $$

where,

$\boldsymbol{p_x} = 0.5 * L_3 * \cos(\theta_1 + \theta_2 + \theta_3 + 1.348) - 0.5 * L_4 * \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) + 0.5 * a_4 * \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) + 0.5 * L_2 * \cos(\theta_1 - \theta_2 + 1.385) + 0.5 * L_2 * \cos(\theta_1 + \theta_2 - 1.385) + 0.5 * L_3 * \cos(\theta_2 - \theta_1 + \theta_3 + 1.348) - 0.5 * L_4 * \sin(\theta_2 - \theta_1 + \theta_3 + \theta_4) + 0.5 * a_4 * \cos(\theta_2 - \theta_1 + \theta_3 + \theta_4)$

$\boldsymbol{p_y} = 0.5 * L_3 * \sin(\theta_1 + \theta_2 + \theta_3 + 1.348) + 0.5 * L_4 * \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) + 0.5 * a_4 * \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) + 0.5 * L_2 * \sin(\theta_1 - \theta_2 + 1.385) + 0.5 * L_2 * \sin(\theta_1 + \theta_2 - 1.385) - 0.5 * L_3 * \sin(\theta_2 - \theta_1 + \theta_3 + 1.348) - 0.5 * L_4 * \cos(\theta_2 - \theta_1 + \theta_3 + \theta_4) - 0.5 * a_4 * \sin(\theta_2 - \theta_1 + \theta_3 + \theta_4)$

$\boldsymbol{p_z} = L_1 - L_2 * \sin(\theta_2 - 1.385) - L_3 * \sin(\theta_2 + \theta_3 + 1.348) - L_4 * \cos(\theta_2 + \theta_3 + \theta_4) - a_4 * \sin(\theta_2 + \theta_3 + \theta_4$

4.6.3 Inverse Kinematics

We have used Gradient Descent method [76] to find inverse kinematics of the xArm-5 robot. The following is a summary of the Gradient Descent approach.

Definitions:

- The vector ($\vec{\theta}$) that represents the joint angles of the xArm-5;

  $\vec{\theta} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5]^T$;

- The vector ($\vec{p}$) that gives the position of the end-effector with respect to frame {0} from the forward kinematics (Equation 4.7) is:

  $\vec{p} = [p_x \quad p_y \quad p_z]^T$;

- FK($\vec{\theta}$) computes the forward kinematics of the xArm-5. In other words, $FK(\vec{\theta})$ computes end-effector position vector ($\vec{p}$) as a function of given the joint angles ($\vec{\theta}$).

- J($\vec{\theta}, \vec{p}$) computes Euclidean distance between FK($\vec{\theta}$) and ($\vec{p}$). $\Delta\overrightarrow{\theta_i}$ is 5 × 1 matrix of all zeros except row *i*, which is equal to the sampling interval (SI). For example, $\Delta\overrightarrow{\theta_i} = [0 \quad 0 \quad SI \quad 0 \quad 0]^T$.

- SI = sampling interval. This is the step size used to approximate the gradient numerically.

- α = Learning Rate. This proportional constant determines the relative size of the step taken during each iteration of gradient descent.

- At each iteration of gradient descent, the following update rule is applied simultaneously to each joint angle;

$$\theta_j = \theta_j - \alpha \frac{\delta J(\vec{\theta}, \vec{p})}{\delta \theta_j}$$

  Where the following equation is used to approximate the gradient

$$\frac{\delta J(\vec{\theta}, \vec{p})}{\delta \theta_j} \approx \frac{J(\vec{\theta} + \Delta\overrightarrow{\theta_j}, \vec{p})}{\left\|\Delta\overrightarrow{\theta_j}\right\|}$$

Pseudo-code:

while $J(\theta, p)$ > threshold do:

gradient = $\left[\frac{\partial J(\theta,p)}{\partial \mathrm{q}_1} \ldots \frac{\partial J(\theta,p)}{\partial \mathrm{q}_n}\right]$

for j = 0…4:

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta,p)}{\partial \mathrm{q}_j}$$

end for

end while

4.7 Control of xArm-5 robot

Three different kinds of control loops are used to implement this control architecture: a position loop, a speed control loop, and a current control loop (see Figure 4.6). The xArm-5 controller sends torque commands (sampling time 4ms) to the joints. The torque commands are translated to motor currents and then to reference voltage as voltage value is the driving signal for the motor drivers. A low-level PI controller (sampling time 0.1ms) is used in the current loop to achieve real-time control of the system and ensure that the proper control torque commands are sent to the joints and the reference voltage commands to the drivers. Sample rates of 0.1 ms are used to measure current signals measured from the current monitor output of motor drivers.

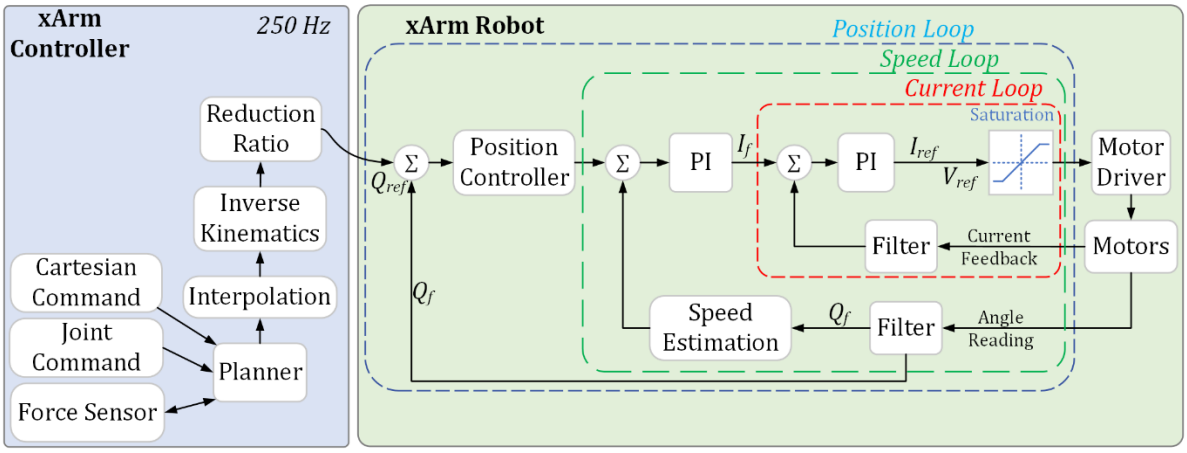


Figure 4.6 Control architecture of the system

The main aim of the current loop is to regulate the torque, which affects speed and, therefore, position. As seen in Fig. 4.6, the current loop (CL) is nested within the speed loop, making the CL the innermost loop, with the speed loop (SL) in the center and the position loop on the outermost loop of the chain of events. The speed loop also uses a PI controller. It is important to note that although the quantity of proportional gain is directly related to how much error is there, the amount of integral gain grows with time and is utilized at the conclusion of each motion to "push" the motor to zero error. The position loop in this cascaded system only used a proportional gain. The PI control approach used for preliminary testing and the joint toque command of the xArm-5 can be expressed by Equation (4.8) [73, 74].

$$\tau = K_P(\theta_d - \theta) + K_I \int (\theta_d - \theta)\, dt \tag{4.8}$$

Where,

$\theta_d, \theta \in \mathbb{R}^5$ are the vectors of desired and measured joint angles, respectively,

K_P, K_I are the diagonal positive definite gain matrices, and

$\tau \in \mathbb{R}^5$ is the generalized torque vector.

Error vector E and it can be expressed by the Equation (4.9):

$$E = \theta_d - \theta \tag{4.9}$$

Therefore, Equation (4.8) can be re-formulated as an error equation:

$$\tau = K_P E + K_I \int E \, dt \tag{4.10}$$

The relation Equation (4.10) is decoupled; therefore, individual torque commands for each joint can be expressed by the Equation (4.11)

$$\tau_i = K_{P_i} e_i + K_{I_i} \int e_i \, dt \tag{4.11}$$

CHAPTER 5

DIGITAL TWIN OF AN ASSISTIVE ROBOT

Digital twins, digital threads, and features of digital threads are discussed at the beginning of this chapter. Next, the step-wise procedure of developing a digital twin using PTC Vuforia Studio is described. The end of the chapter presents the development of an assistive robot's augmented reality (AR) digital twin.

5.1 Digital Twin [77]

A digital twin is defined as a digital model that helps to represent a physical product, task, or primary process virtually. A digital model is more than just a digital copy of anything. It contains real-time data, and without data, it is not a true digital twin. Digital twins work uniquely for every product, process, or task. With the help of augmented reality (AR), it is easy to take a digital representation of a physical product or task and put it into the real world for connecting clients and servers. A scanning code is required to access the digital twin, which may be available in various forms, on a smartphone or any other device which helps to load and make the whole experience for observing in AR. Furthermore, augmented reality can enable multiple digital twin experiences under a single AR experience. For instance, a single AR experience could have one experience of reading a single set of data for any product. Another could be about doing maintenance on that product. [77]

5.2 Digital Thread [78]

The digital thread can be defined as a network of communication that allows the synchronization of real-time data. In addition to this, it works as a connection between the physical and digital world. This expandable set of data that is accessible to the general user allows continuity among

objects, processes, and individuals. In simple words, it can also be described as a combination of multiple data obtained throughout a product's life. Digital threads help obtain information about how customers use specific products and what more they expect from the product. Data obtained with the help of digital threads is beneficial for the organization to improve its products. This data can be cost-effective and helpful in improving the overall quality of customer compliance. Last but not least, digital threads can boost any organization. [78]

5.3 Digital Thread Features [79]

The digital thread is also defined as a crucial piece of making a digital twin. Both digital thread and digital twin are interlinked. Digital twins obtain data from a physical thing's digital thread and generate a digital representation of the data obtained. A digital twin can be used to access the data from the digital thread. [79]

We have highlighted below some of the things that digital twins may enable [79]:

- It makes use of digital thread data of a real product in the digital twin.
- It stores product history relevant to continuance and instructions of work.
- It provides multiple configurations of the same model.
- It gives suggestions related to intelligent design on the basis of service history.
- It also gives maintenance alerts and remote access.

5.4 PTC Vuforia Studio

Vuforia Studio is considered a web-native tool and was developed by PTC Inc [80]. This tool can be used easily for various task-specific *Experience Services* [77]. This platform provides a development environment for developing industrial 2D/3D AR applications. According to Fig. 5.1. (Vuforia Studio architecture and process flow), Vuforia Studio can optimize CAD models,

bind those optimized models to IIoT platforms (ThingWorx), and generate AR Studio Experience Services.

Figure 5.1 Vuforia Studio architecture and process flow [81]

AR applications can be designed for smartphones, tablets, and wearable devices like Microsoft HoloLens 2 [82]. Studio Experience Service can be accessible in the Vuforia View app. The studio interface is a drag-and-drop functionality that enables users to develop innovative applications without prior programming experience [82]. Integrating various PTC products allows transferring data straightforwardly and incorporating the data across multiple environments [82]. Vuforia Studio also enables optimization of equipment performance and analyzes operating conditions using real-time data from the Internet of Things (ThingWorx) [81]. Therefore, considering the above benefits, we have decided to incorporate Vuforia Studio in this research.

5.5 Digital Twin Development Process

Assembling the 3D CAD model of the robot was accomplished using PTC Creo while converting the CAD model to a digital twin was performed using PTC Vuforia Studio. Figure 5.2 shows the steps taken to create a digital twin of a robot using Creo and Vuforia Studio and divides it into three main steps: Nest the assembly in Creo, Create the Model Hierarchy in Vuforia Studio, and Vuforia Studio application development.

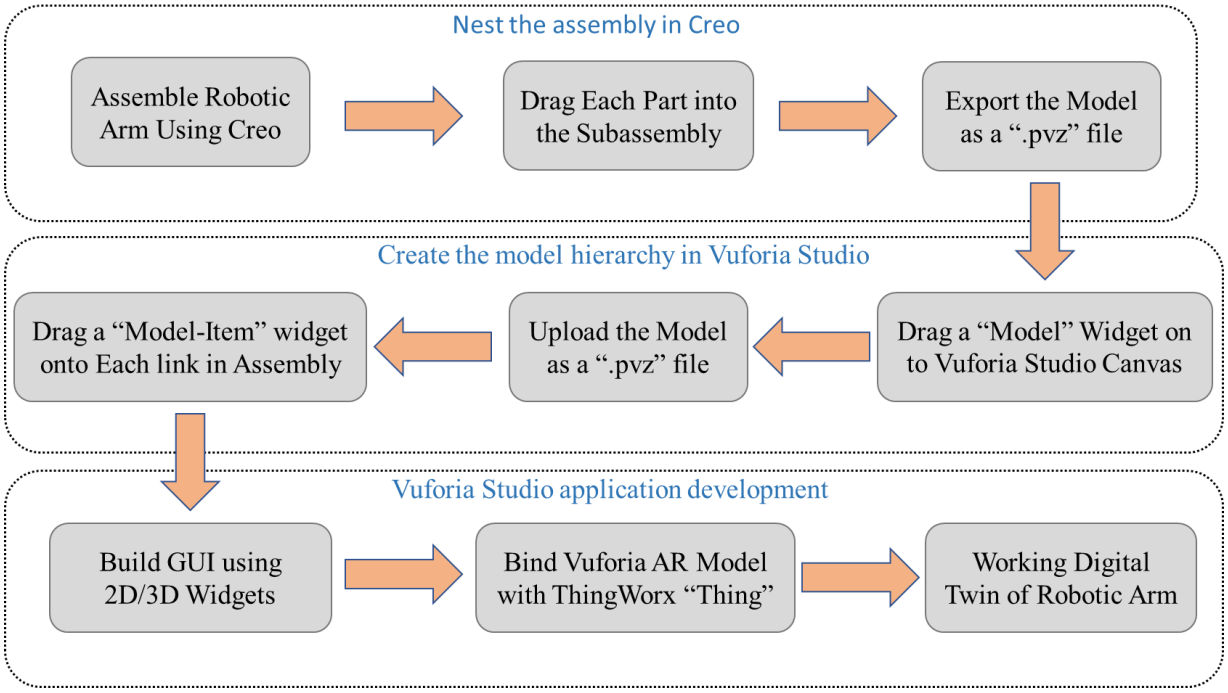
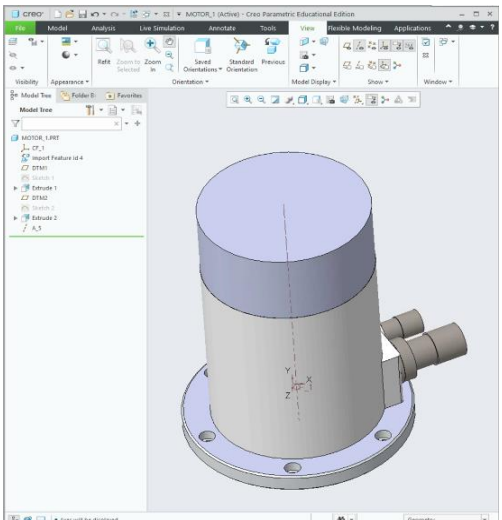


Figure 5.2 Flowchart for building Digital Twin of the robot using Creo and Vuforia Studio

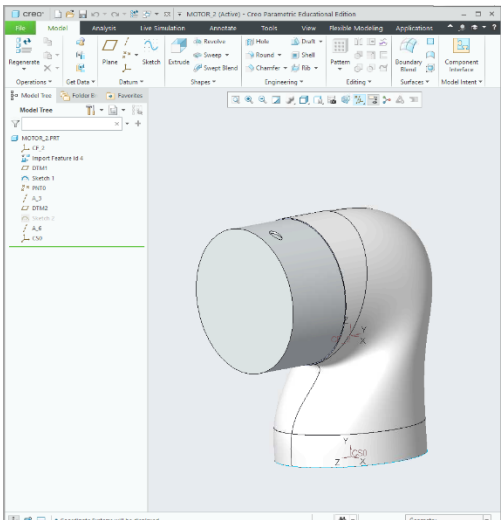
5.5.1 Nest the Assembly in Creo

The development of a digital twin began with assembling joint 1 to joint 5 of the xArm-5 robot. All the robot joints are assembled using PTC Creo (CAD modeling software). Individual joints of the robot are shown in Fig. 5.3, and the assembled robot is depicted in Fig. 5.4. Figure 5.5 shows a way to create subassemblies for each joint. It can be seen from Fig. 5.6 that each joint is dragged into the subassembly under the hierarchy, and the model is ready to export for Vuforia Studio as a ". pvz" file.

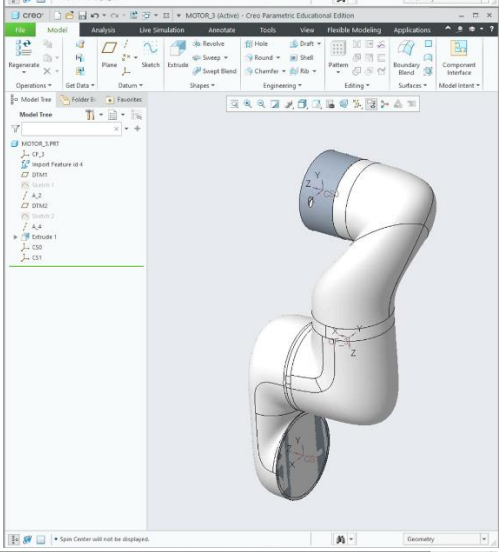
Joint 1



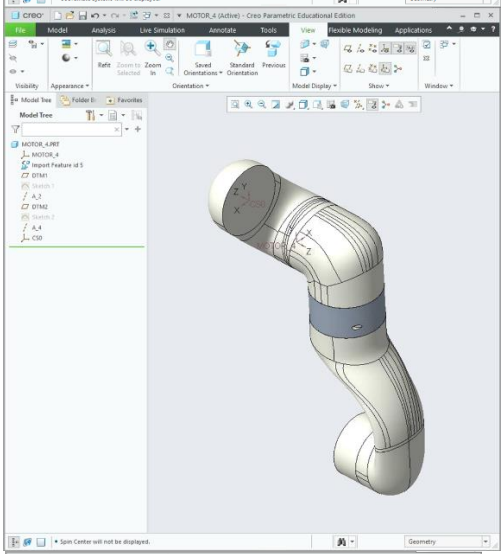
Joint 2



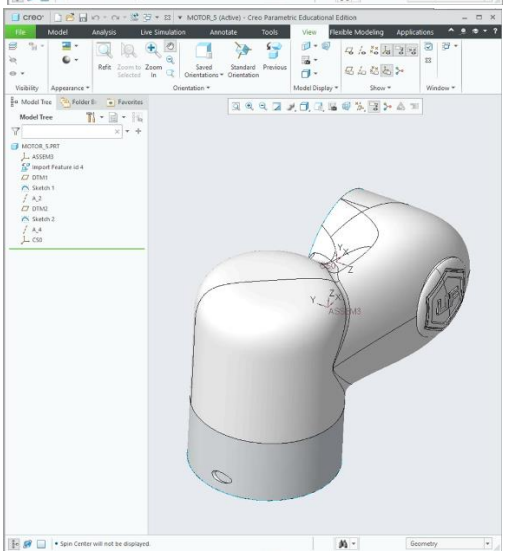
Joint 3



Joint 4



Joint 5



End-effector handle

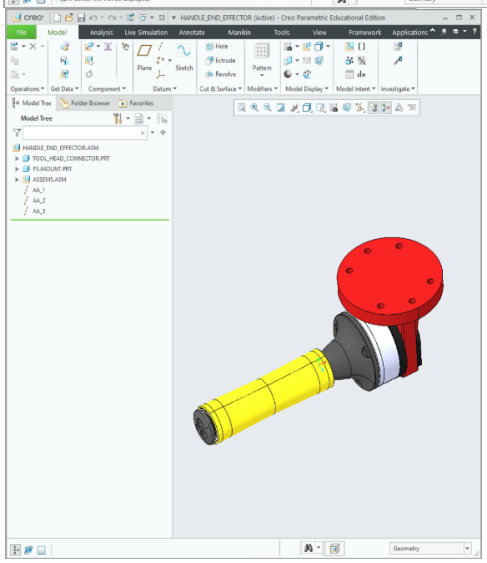


Figure 5.3 Individual joints of the robot

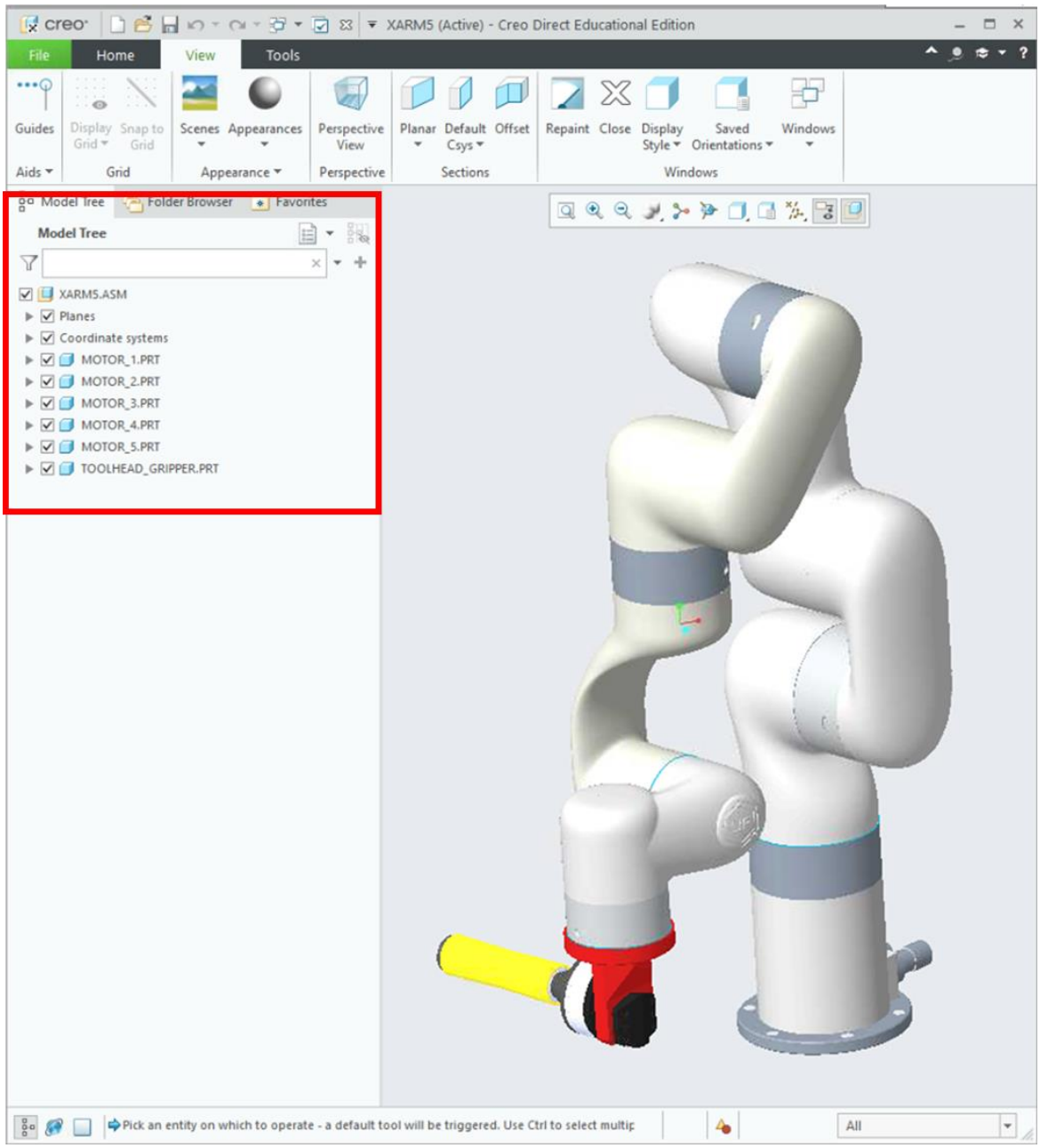


Figure 5.4 Assemble robot using Creo

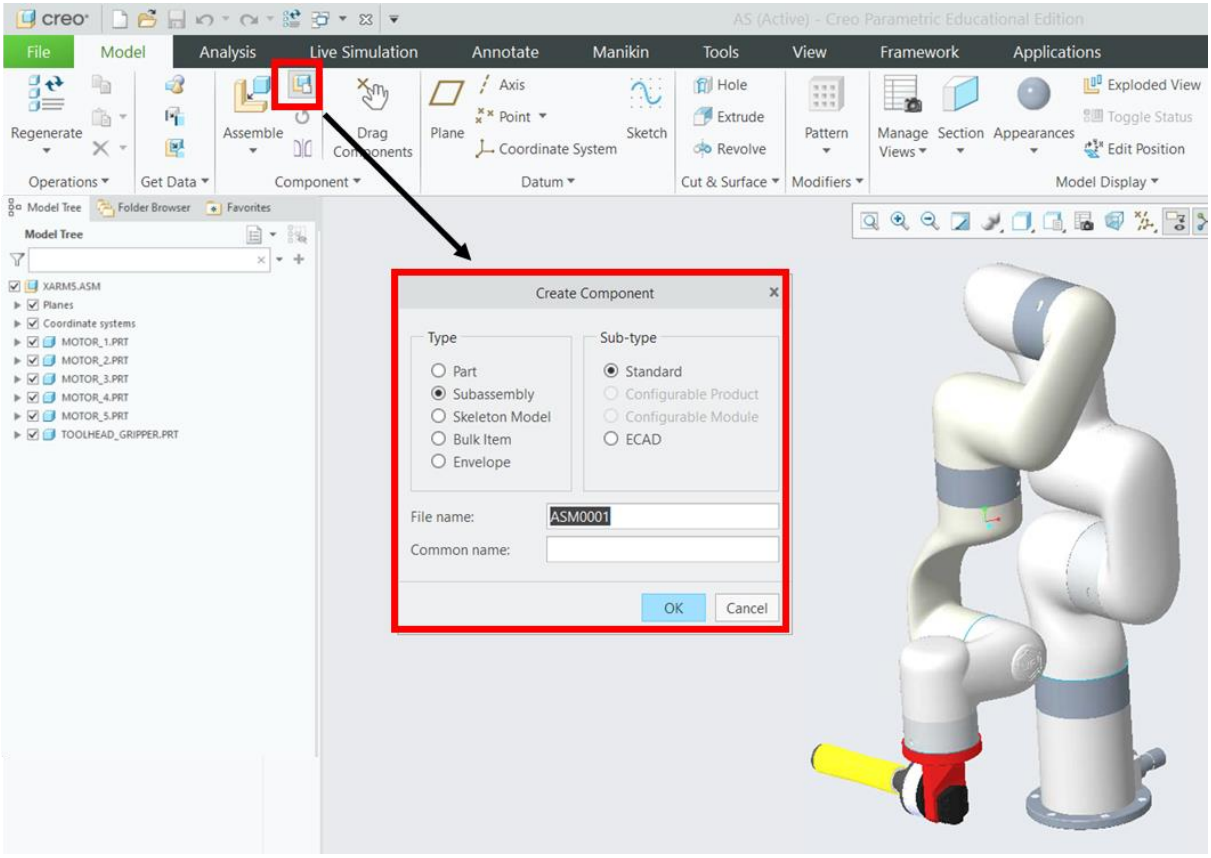


Figure 5.5 Creating subassemblies for each joints

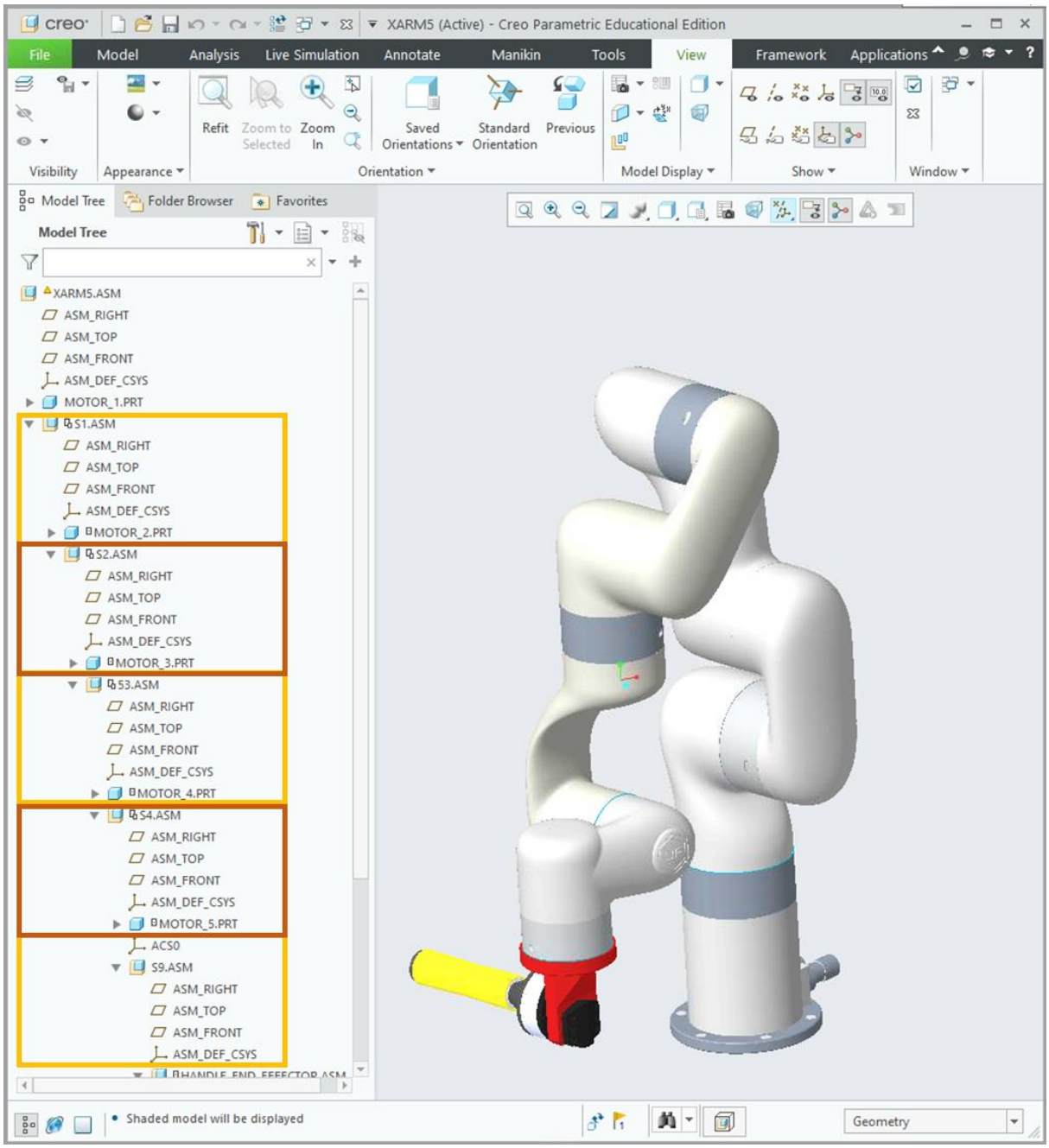


Figure 5.6 Final robot assembly for Vuforia Studio

5.5.2 Create the Model Hierarchy in Vuforia Studio

The process of creating a new project in Vuforia studio starts with a unique project name and experience service URL. Since ThingWorx server data is required for integration of the AR application, the experience service URL is equivalent to that of the ThingWorx server, as displayed in Fig. 5.7.

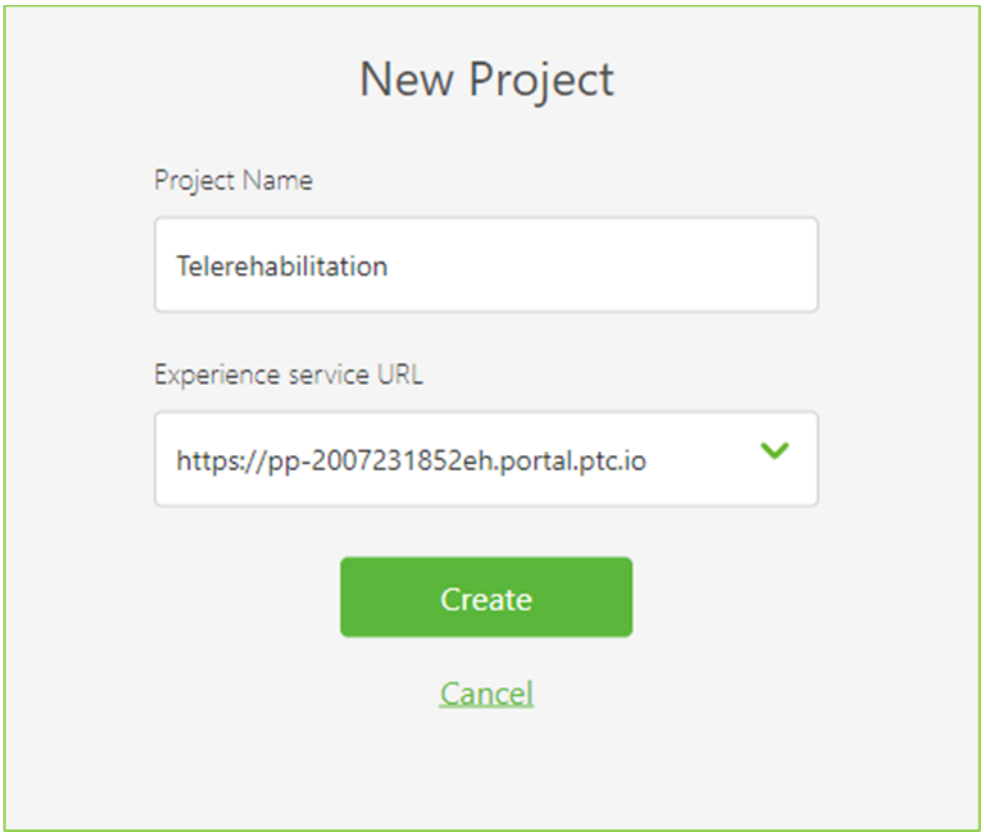


Figure 5.7 Project name and experience service URL for Vuforia Studio

The development environment is now displayed after creating a new project. It is necessary to import the CAD model of the robot into the development environment of Vuforia Studio in order to develop the robot (model) hierarchy. “Model” widget can be added to a virtual space by dragging it onto canvas under the *tab Augmentations* as shown in Fig. 5.8. CAD model (exported at the end of section 5.5.1) can be selected under the *Properties tab* by clicking on the *Resource tab*. A picture of the robot model on a canvas of Vuforia Studio is displayed in Fig. 5.8.

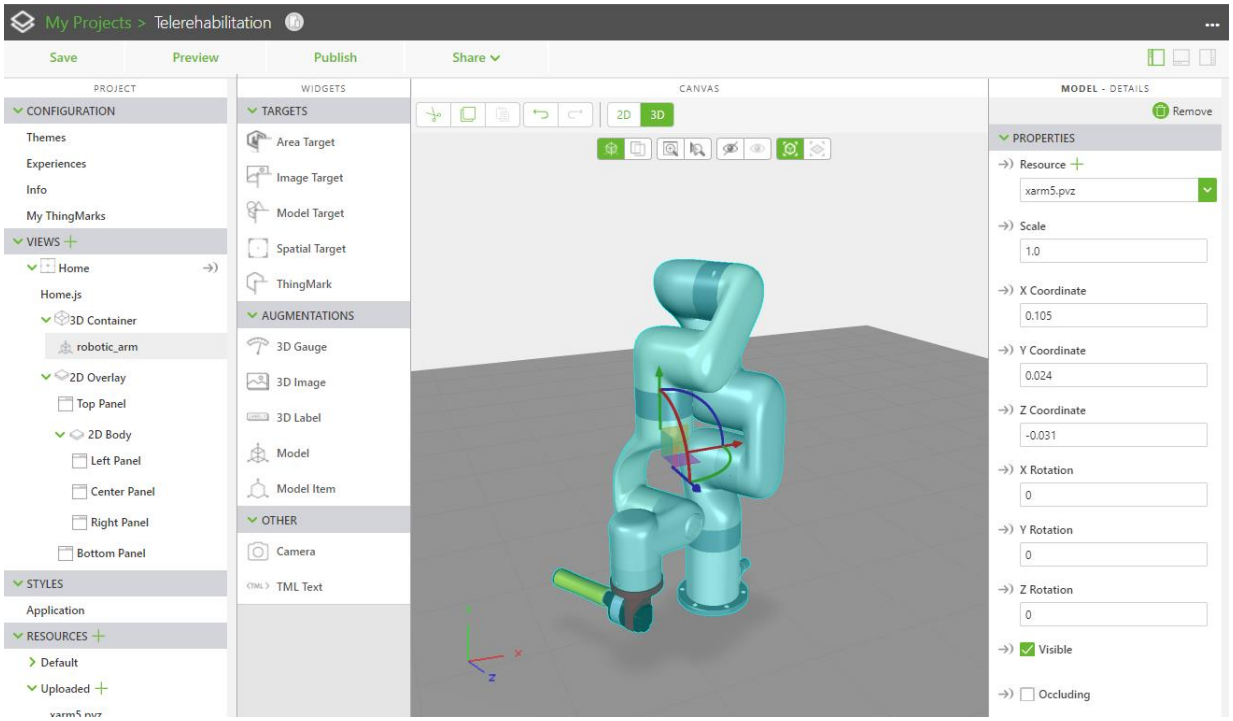


Figure 5.8 Imported robot model on Vuforia Studio canvas

Once the CAD model has been displayed, the coordinates and rotations can be modified. The "Model Item" widget can be dragged onto each link in the assembly of the robot model to convert it into a hierarchical model. It can be started with the component that is the bottom-most parent model in the hierarchy of the robot. Therefore, we began by dragging "Model Item" onto joint 1 and edited the "Component Occurrence" by deleting the last values, as shown in Fig. 5.9. It can be seen that joint 1 is the child model of the robot parent model. The process must be repeated for all joints while working up the hierarchy sequentially, beginning with those at the bottom of the robot. It can be seen the child model items automatically move to be nested under their parent model item, as shown in Fig. 5.10. Now, the "Model" is ready for developing the Vuforia Studio application.

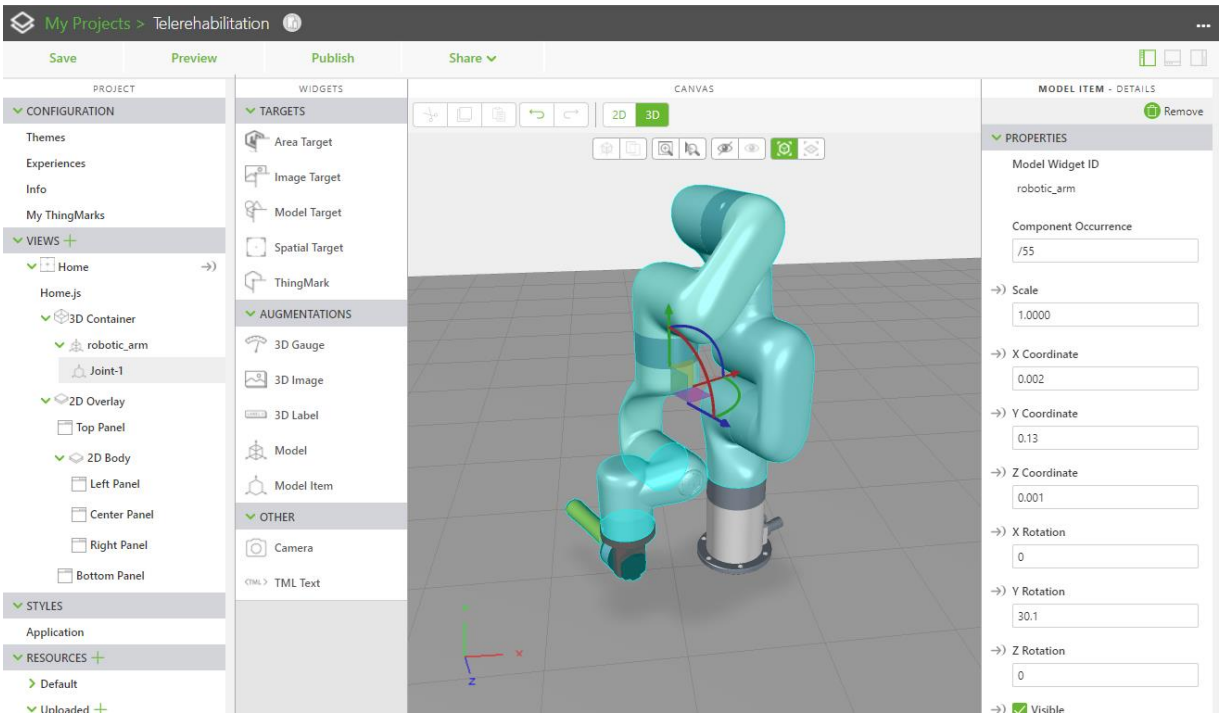


Figure 5.9 Joint-1 child model of the robot parent model

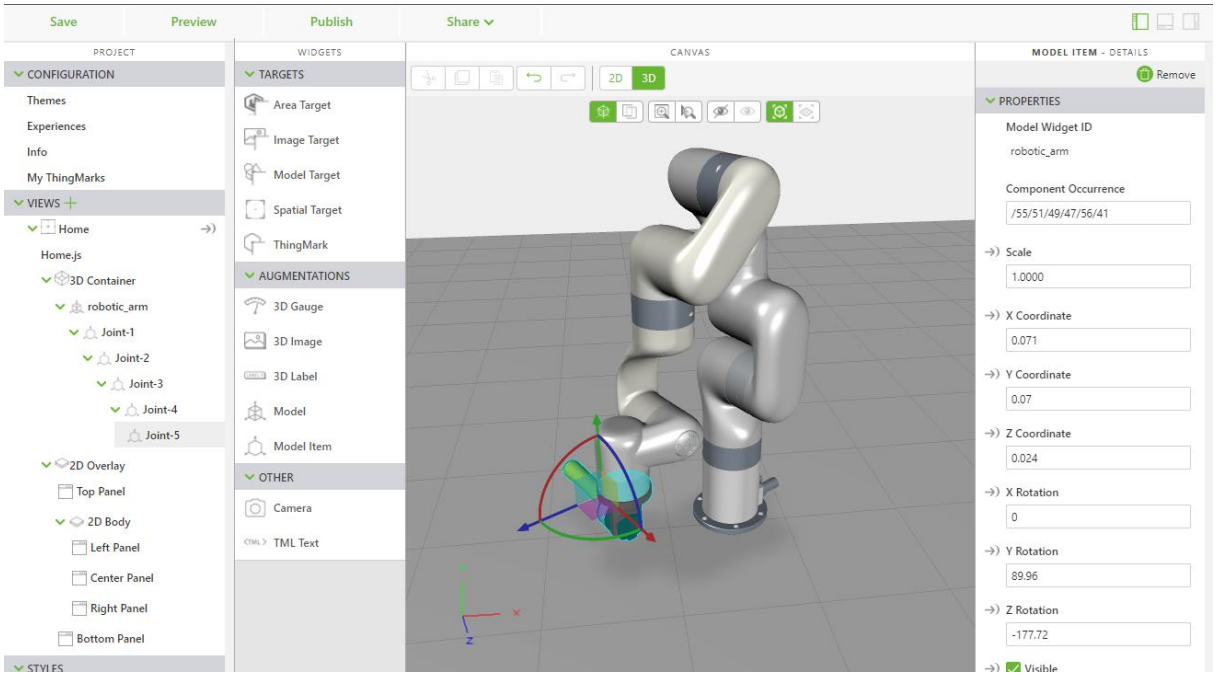


Figure 5.10 Joint-5 child model item of the joint-4 parent model

5.5.3 Vuforia Studio Application Development

To develop the application for Vuforia Studio, 2D/3D widgets can be used to develop GUIs for interacting with the AR robot model on canvas. The 3D label Augmentation is used to display data (e.g., joint angles, currents, end-effector coordinates, etc.). All 3D widgets follow the same process of dragging and dropping on the virtual environment canvas. Figure 5.11 shows different 3D labels, 3D gauges, and 3D images around the robot model.

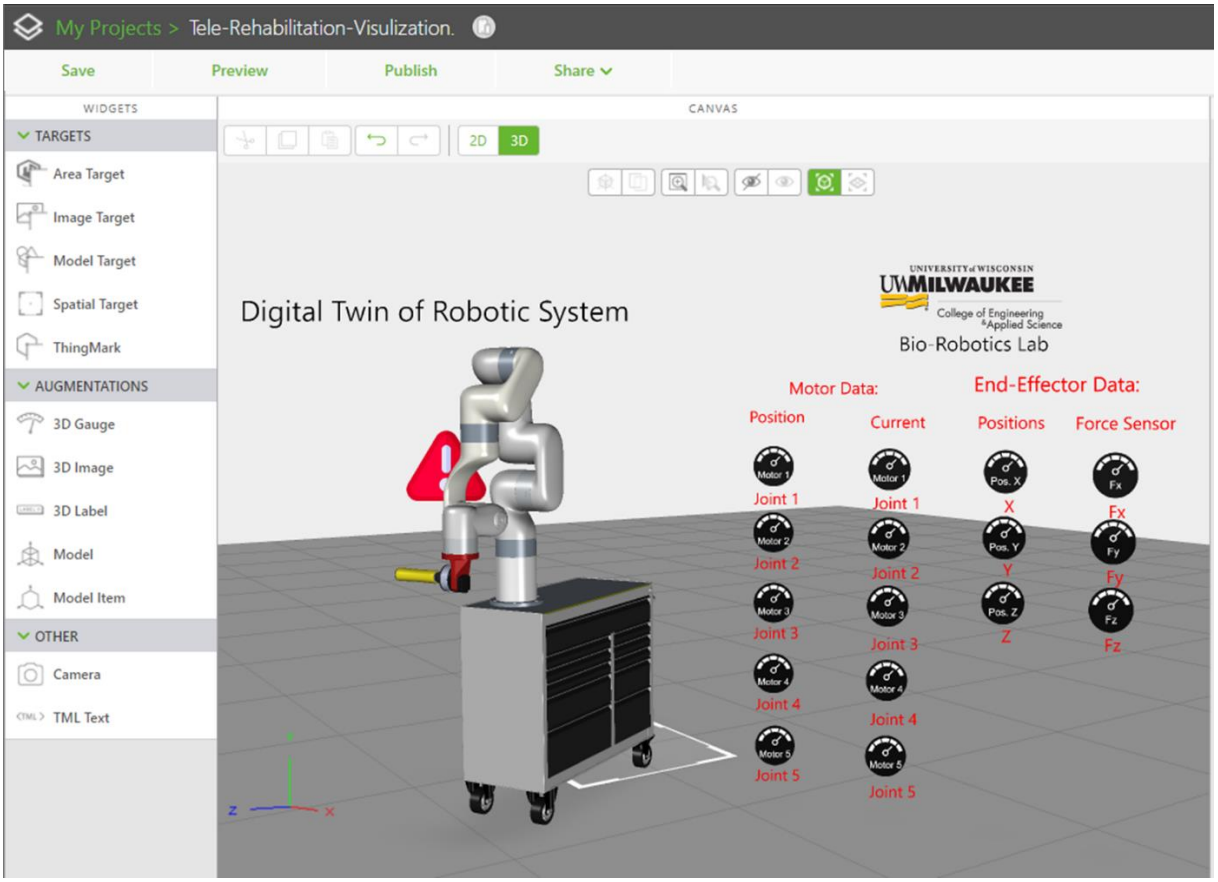


Figure 5.11 Vuforia Studio virtual environment with robot mode and 3D labels

The 2D widgets have categories such as containers, inputs, and others. 2D Containers' widgets can be used for grid layout, header, etc. The widgets in 2D inputs can be used to provide input for "Models", and "Model Items", using buttons, toggle buttons, sliders, etc., by binding them with model properties. Figure 5.12 shows the different use of 2D widgets on canvas. Also, it can be

seen from Fig. 5.12 that the user can rotate each joint in a clockwise or anticlockwise direction using sliders.

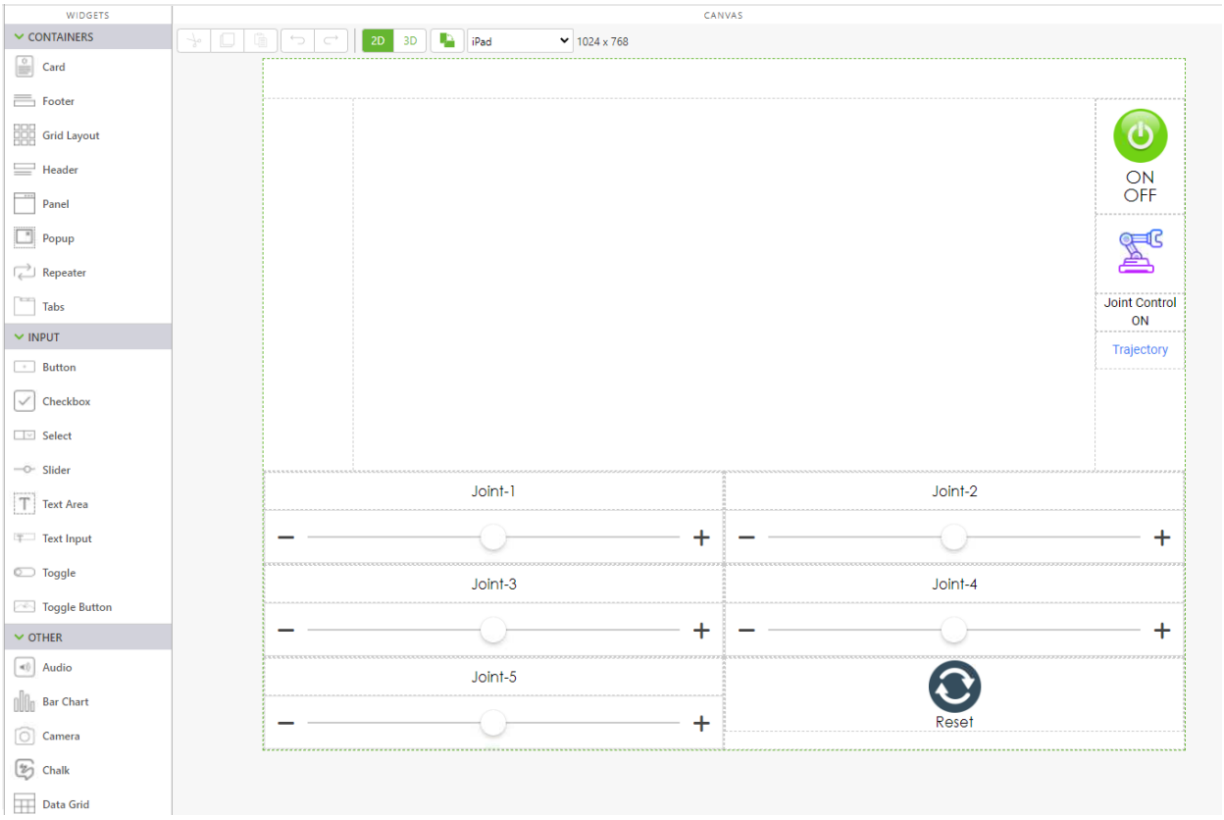


Figure 5.12 GUI developed using 2D widgets on Vuforia Studio canvas

In Digital Twin (section 5.1), it was noted that without binding CAD model with IoT data, it is not a true digital twin. Therefore, the following steps were taken to bind the robot model with ThingWorx IoT data. The data from ThingWorx can be bound with both 3D and 2D widgets. As shown in Fig. 5.13, under the "External Data" tab "xArm5-RemoteThing" is added from ThingWorx to Vuforia Studio by clicking "+". Here, the 3D gauge widget is connected with the "angle_servo1" property. Vuforia Studio developed projects can be accessed through the experience service URL (shown in section 5.5.2) by clicking the "Publish" button, and it allows users to access from any mobile device.

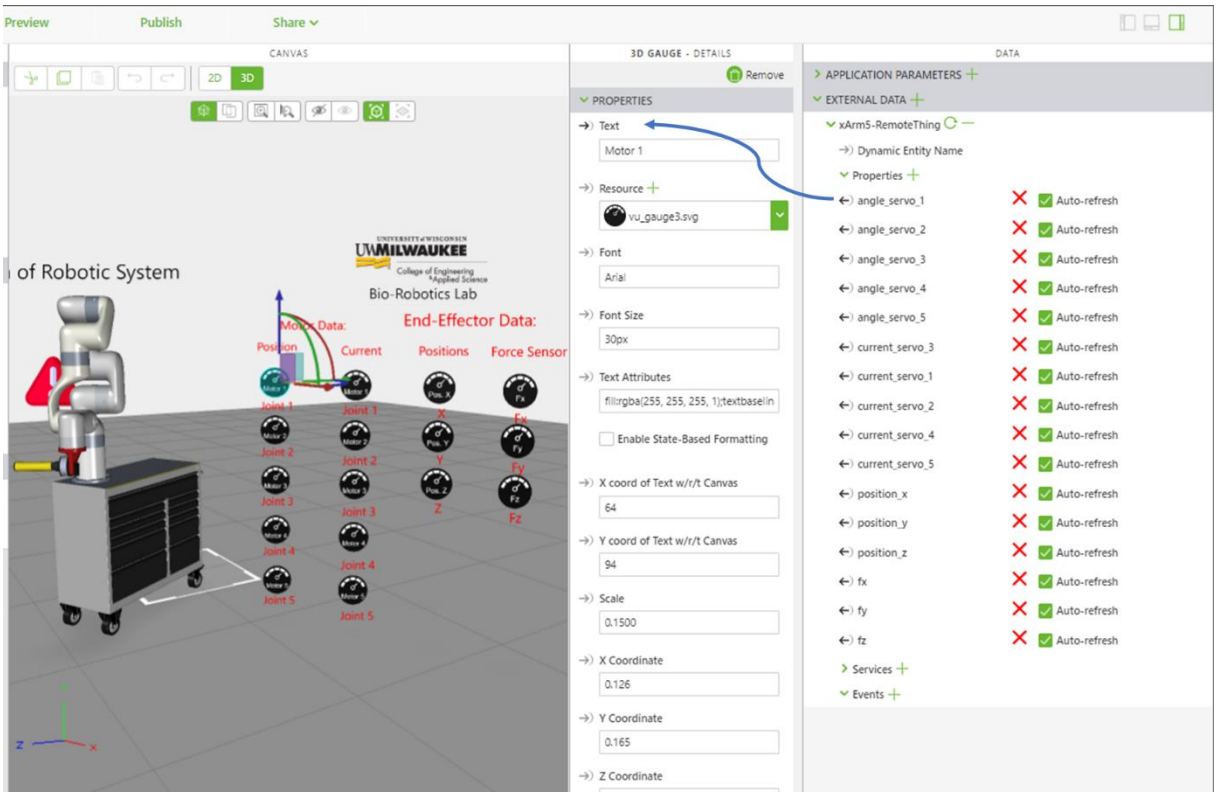


Figure 5.13 Bind 3D widgets with ThingWorx data

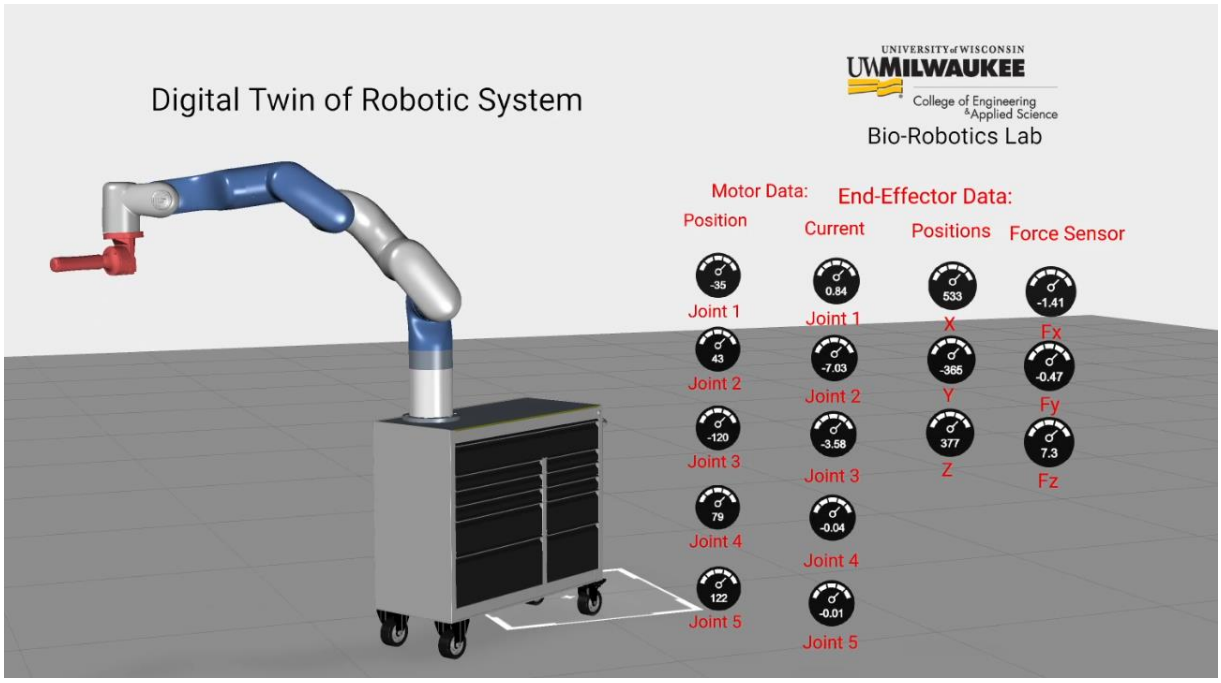


Figure 5.14 Digital Twin - Published Vuforia Studio application on experience service URL

Figure 5.14 shows the published Vuforia Studio application on the experience service URL. As a result of this step, a Digital Twin of an assistive robot (in this case, xArm-5) is generated.

To interact with the digital replica of the robot and provide telerehabilitation, we have developed a graphical user interface (GUI). The GUI contains various telerehabilitation exercises that can be delivered using different control modes are discussed in chapter 7. Also, to constantly monitor the physical robot via digital twin, we have developed a low latency connection with ThingWorx using different protocols (discussed in section 6.3).

CHAPTER 6

TELEMANIPULATION FRAMEWORK FOR HUMAN-ROBOT COLLABORATION

This chapter discusses on telemanipulation framework and control architecture of telemanipulation. The telemanipulation framework is developed using PTC ThingWorx, PTC Vuforia Studio, PTC Vuforia View as software, and Logitech Joystick as hardware.

6.1 Framework Design – High Level

This section describes the high-level framework design for telemanipulation. As shown in Fig. 6.1, it mainly focuses on four data flow components that communicate with each other: User/therapist, xArm-5 robot, PTC ThingWorx, and PTC Vuforia Studio.

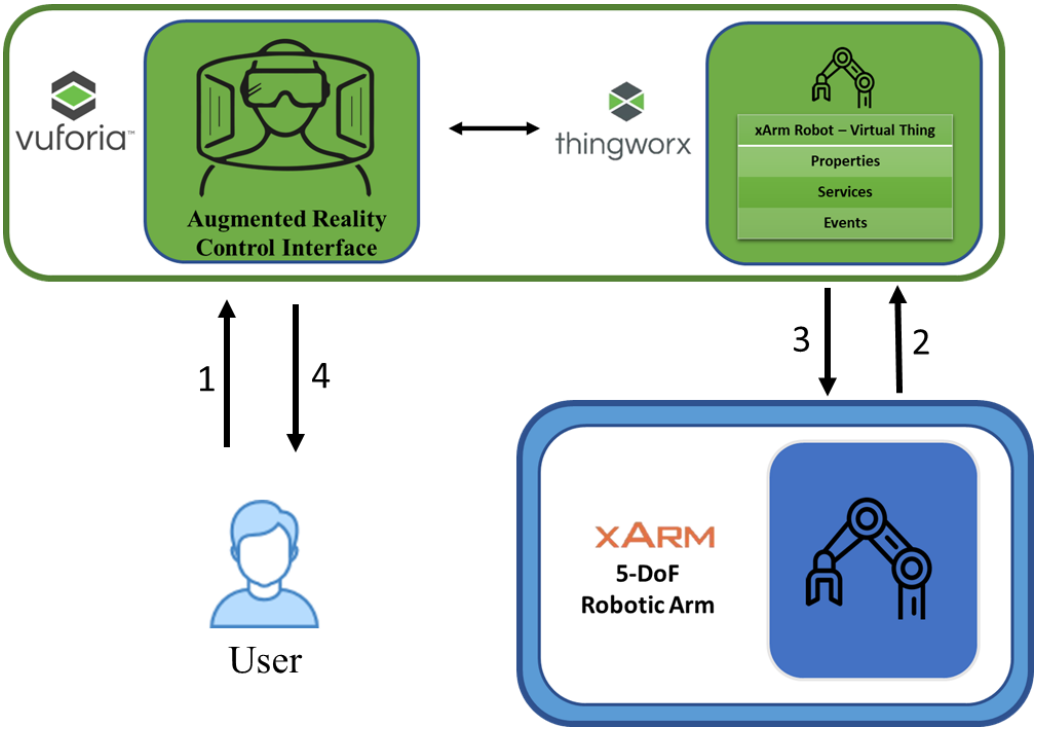


Figure 6.1 Telemanipulation framework – high level

The figure above shows the four stages of data flow. The steps are detailed as follows:

1. The user/operator interacts with the augmented reality (AR) based xArm robot “controller” in a Vuforia Studio application to modify the controller’s state of the xArm-5 robot.
2. The Vuforia Studio application conveys this controller’s state change to the ThingWorx xArm-5 robot - virtual Thing, and from there to a PC that drives the actual xArm-5 robot. The PC receives this data and physically actuates the xArm-5 robot to match the new controller state.
3. The xArm-5 robot’s PC reads the current telemetry data from the robot and sends this data to the Vuforia Studio application.
4. The user can see a different AR of the xArm-5 robot that reflects the actual orientation of the xArm-5 robot. This gives the user the ability to verify that the xArm-5 robot behaves according to expectations.

A technology that enables a collaborative group of users to teleoperate an industrial robot simultaneously via the Industrial Internet of Things (IIoT). A digital twin of the physical robot lives on the IIoT platform. As the name implies, the digital twin mirrors the physical robot as it sends telemetry data. The IIoT platform can be configured to respond to changes in data.

6.2 Framework Design – Low Level

This section provides low-level and detailed knowledge of telemanipulation control architecture. Proposed research mainly focuses on telerehabilitation application through telemanipulation, and on that basis, a control framework is developed and explained. As shown in Fig. 6.2 [77], the control architecture of telerehabilitation is mainly divided into two subsystems: The Robotic System and The IIoT Platform. Both these subsystems communicate through the internet. The xArm robot is connected to a Client PC (Windows Computer), and it delivers various telerehab

exercises (trajectories) to the xArm robot when it receives excises commands from the IIoT platform.

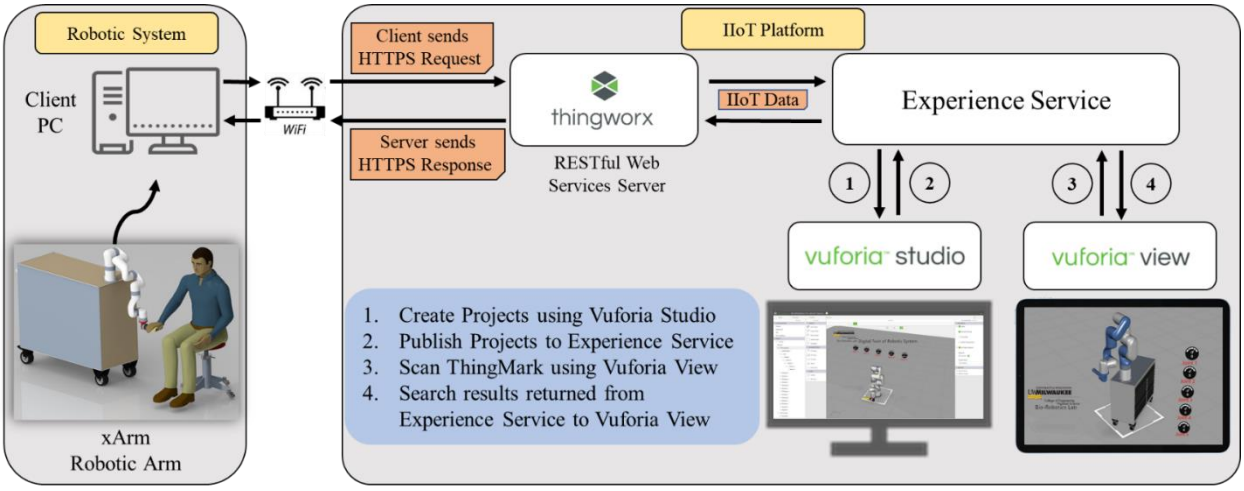


Figure 6.2 Control architecture of proposed telerehabilitation *[77]*

IIoT platform contains ThingWorx, Experience Service, Vuforia Studio, and Vuforia View. In the field of the IIoT platform, ThingWorx acts as a bridge for process IIoT data transfer between the robotic system and the Vuforia Studio application (Digital Twin of a robot). Client PC and ThingWorx communicate bidirectional using ThingWorx server secure HTTPS protocol. Client PC sends HTTPS request to ThingWorx server, and ThingWorx server sends HTTPS response back to Client PC. The 'Experience Service' is part of the IIoT platform (Fig. 6.2), and it contains rehabilitation therapy data. In our study, we built the digital twin of a robot (xArm-5 robot) with a graphical user interface (GUI) for telerehabilitation therapies using Vuforia Studio (explained in section 5.5). The developed digital twin is then published to the Experience Service URL through Vuforia Studio. For accessing the published GUI from the Windows/Android/iOS platform, we used the Vuforia View app (explained in section 6.5). Vuforia View app is connected with the experience services URL of the ThingWorx. It sends the IIoT data to the robotic system once interacting with the digital twin of a robot GUI. The advantage of using the ThingWorx IIoT cloud-

based platform is that user therapists can connect the physical joystick to the Vuforia view app and move the actual robot. At the same time, it can visualize the movements of the xArm-5 robot on the AR digital twin robot. Different types of control modes are developed in this research for telerehabilitation: pre-defined therapy / passive therapy, assistive mode, impedance control mode, trajectory recording mode, joint–control mode, end–effector control mode, joystick–control mode. Also, when a session starts, the therapist and patients join the Video calling room using Microsoft Teams for better communication. The following sections discuss the development of the proposed telemanipulation framework for telerehabilitation that uses PTC's ThingWorx, Vuforia Studio, Vuforia View, xArm-6 robot, and a Logitech Joystick.

6.3 PTC ThingWorx Platform

The ThingWorx is the Industrial Internet of Thing (IIoT) platform developed by PTC Inc. The ThingWorx platform makes it easier and faster to realize efficiencies. It reduces risk in operations such as in-home/industrial utilities, smart cities, healthcare, and factories. Getting and analyzing the data that continuously generates critical operational data is the key to a connected system. ThingWorx provides a platform where data can be structured and organized in a way that makes sense in industrial IoT to organize physical assets from digital systems such as Product Lifecycle Management (PLM), 3D CAD models, etc. ThingWorx is a platform that provides connectivity and analytics, as well as tools to create apps and Augmented Reality experiences. Using connectivity solutions in real-time, physical entities are represented by their virtual representations using the "Thing Model", the approach that underpins the entire ecosystem [82]. The ThingWorx model can be accessed using Mashups to deliver information to a web page [83]. ThingWorx architecture is described in Fig. 6.3; it shows that ThingWorx has wide use at the industrial level. In essence, the "Thing Model" is formed of seven key concepts and fundamental components;

Josef [82] has reviewed those concepts and explained them below from the source material [82][83],

- **Thing**: A Thing is a virtual data model; it represents a physical system or an entity.

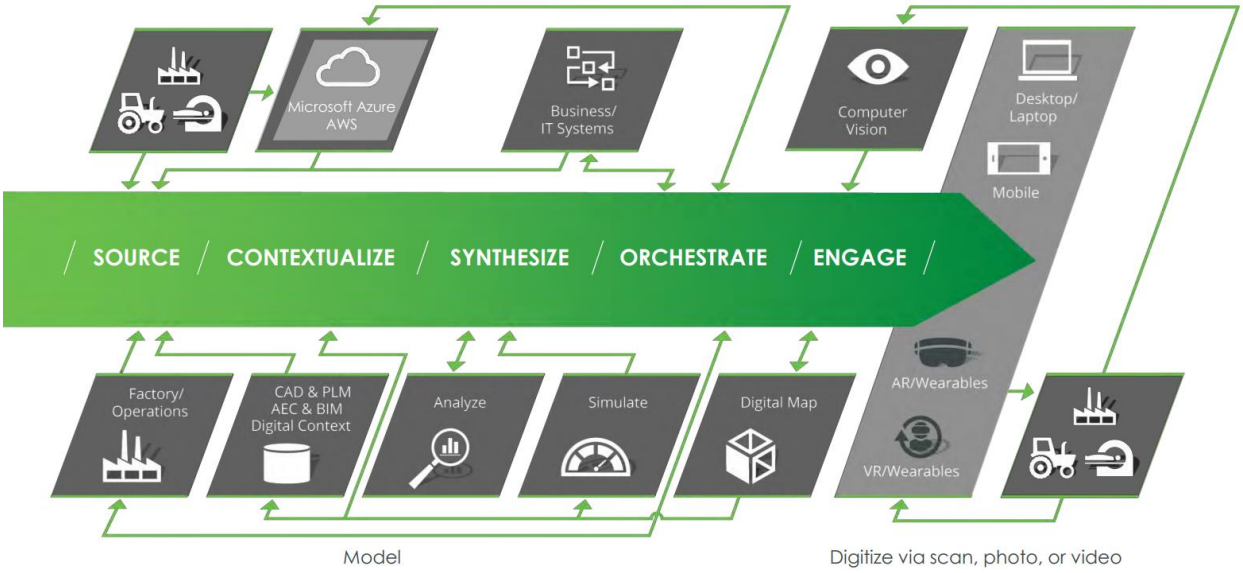


Figure 6.3 PTC ThingWorx platform architecture [84]

- **Thing Shape**: A Thing Shape is an entity class; it contains specific characteristics and methods. A Thing can call Thing Shape's methods.
- **Thing** Template: A Thing Template is a predefined template structure that can be used by new Things. This template characterizes the configurations and functions of a Thing.
- **Property**: A Property is an attribute of a Thing. Things have multiple properties/attributes that define the Thing. For example, a Thing might have properties like a name, a description, and a data type. Depending on the application, properties can be static or change over time. In most scenarios, physical sources like sensors are bound to properties through fast communication channels. This allows almost real-time virtual representation of the actual values

- **Service**: A Service is a function a Thing can execute. The ThingWorx platform provides various essential pre-implemented Services for several applications and Things.
- **Event**: An Event is a trigger-evoked function that a Thing can subscribe to. Events are primarily triggered by Services/functions. The data provided by an event is received by the subscriber of the event. For example, a timer event invokes an action once it reaches a specified number.
- **Subscription**: A Subscription is a function that gets executed once an Event is triggered. Subscriptions and Events are mostly used in combination.

6.3.1 ThingWorx Composer Configuration

This section provides information about ThingWorx composer configuration for telerehabilitation. Fig. 6.4 shows the layout of the ThingWorx composer. We started with building a connection between "Thing" from the ThingWorx Composer to the xArm-5 robot system using two methods: 1) ThingWorx REST API and 2) ThingWorx Edge Java SDK.

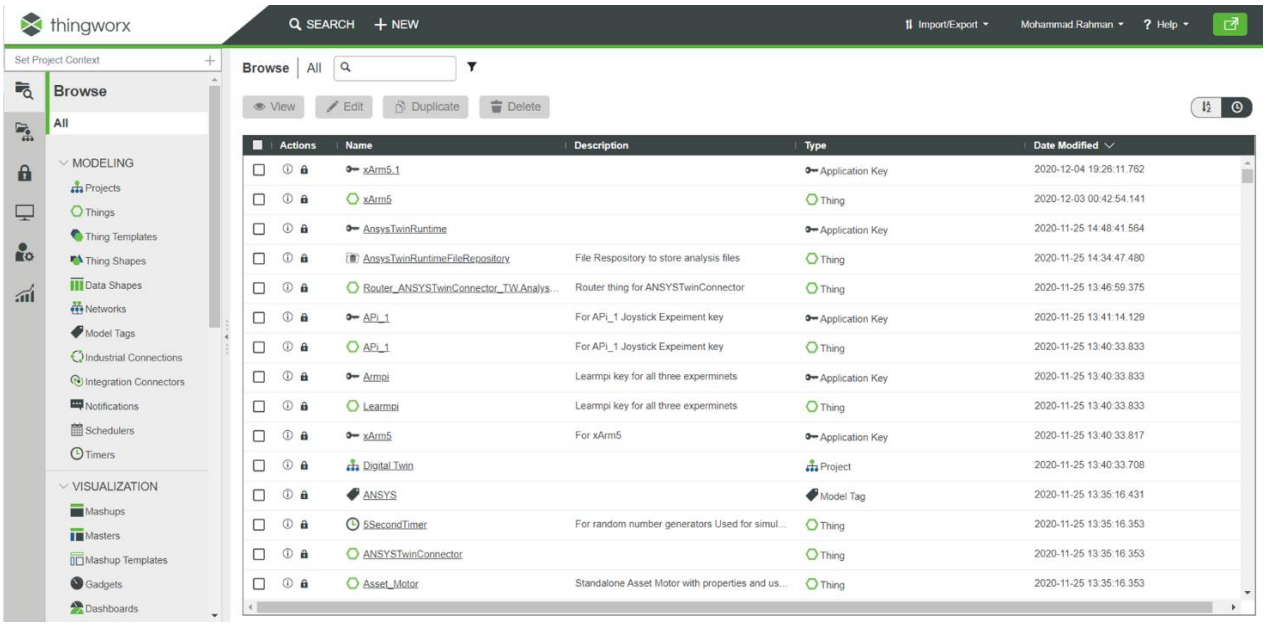


Figure 6.4 ThingWorx Composer

In this research, we used ThingWorx REST API to control the robot through Vuforia Studio/Vuforia View. We also used a remote Logitech joystick and various developed modes for robot-aided rehabilitation exercises. On another side, to visualize/remote monitoring data from the xArm robot to ThingWorx, we used ThingWorx Edge Java SDK.

- **ThingWorx URL Pattern**

  Every ThingWorx instance has its unique host and port number. It requires making a connection with the ThingWorx instance.

  ThingWorx instance URL: https://pp-2007231852eh.portal.ptc.io:8443/Thingworx/Composer

- **Application Key** [85]

  The application key is required to pass with ThingWorx URL to accept the request from the client. The application key is a common part of both methods (ThingWorx REST API and ThingWorx Edge Java SDK). It is essential for devices to be authenticated before sending or receiving data from ThingWorx. The most commonly used authentication method with ThingWorx is an application key. Each application key is assigned to a particular user and has the same permissions as that user. [85]

  For example: "358d267a-fb4a-4355-954f-318a2c7ca722"

1. **ThingWorx REST API** [86]

   REST (Representational State Transfer) is a crucial tool for communication and provides excellent communication for Web Services. One of the advantages of REST is that it functions without complexities. Due to this reason, REST is easy and convenient to work with. Also, it can be easily operated by any client having the ability to make an HTTP request. A huge number of potential clients interact and integrate with ThingWorx. The major aim of REST is

to enhance performance, reliability, and portability. This can be achieved with the help of various REST principles, such as a layered system and client-server architecture. [86]

- **REST API Security** [86]

  REST API Security is a crucial part of project development and REST APIs. It is recommended by the PTC, that the customers must cautiously review and get along with the permissions linked with the REST API to function; this helps ensure that only legitimate individuals can use it. [86]

- **REST API Design**

  Here are some steps about the REST API call applicable to ThingWorx entities and services. REST APIs often use similar HTTP request verbs:

  - **GET** to retrieve information
  - **PUT** to change the value of an existing entity

  In this research study, we have used python scripts on the xArm-5 robot computer to communicate with the ThingWorx. GET HTTP request provides us telemetry data/information from ThingWorx and then passes those received data/information to the xArm-5 robot controller to perform the task. PUT HTTP request keeps updating xArm robot parameters to ThingWorx.

2. **ThingWorx Edge Java SDK** [87]

   ThingWorx Edge Java SDK is a platform for application development that is highly portable. It provides an ease to integrate applications into the ThingWorx distributed computing view of the Internet of Things (IoT). SDK helps to connect Java-enabled devices to the ThingWorx Platform. Furthermore, SDK also allows developers to create highly advanced applications by

providing enough flexibility. It also allows the individual to generate an application for devices or machines that corresponds with the ThingWorx Platform. The ThingWorx AlwaysOn™ protocol is used by SDK for communication. A binary protocol, "AlwaysOn" uses WebSockets as its transport mechanism. This protocol permits continuous WebSocket connections that can easily operate by a firewall. The main advantage of persistency is that it allows two-way, lower-input latency communication between the device and platform. [87]

Major objectives of the Edge Java SDK

The ThingWorx Edge Java SDK consists of the compulsory Java libraries (JAR files) that help communicate with the ThingWorx Platform by writing a client in Java. The main goals of the ThingWorx Edge Java SDK are described below [87]:

- It integrates and maintains an extremely secure and safe AlwaysOn connection with a ThingWorx Platform.
- It allows programmatically simple interaction with all the services or events present on a ThingWorx Platform and exposed by the entities.
- It helps define the properties and services accessed from the ThingWorx Platform.

6.5 PTC Vuforia View

Vuforia View is a part of the PTC product line, and it is a mobile app. This app is available for Android, iOS, and Windows. It allows users to access and enjoy AR experiences that are created and developed using Vuforia Studio. This AR experience is rich with 3D content and IIoT data [88].

Figure 6.5 Steps for Vuforia View [89]

To access any of the developed and published Vuforia Studio-based experiences, we follow the steps mentioned below (Fig. 6.5). The ThingMark used for this research is attached in Appendix-A.

1. Open the Vuforia View app
2. Scan a ThingMark
3. Select an Experience from the list
4. Interact and enjoy selected the Augmented Reality Experience

6.6 Logitech Joystick

Logitech G Extreme 3D Pro Joystick was used in this research as an alternate control method to control the x-Arm-5's end-effector position. It has two axis movements (X and Y directions), 12 action buttons, an eight-way hat switch, and a rapid-fire trigger. Figure 6.6 depicts the Logitech G Extreme 3D Pro joystick [90], and Figure 6.7 shows the xArm-5 robot end-effector coordination. As shown in Fig. 6.6, the trigger button of the joystick is the main control button. Triggers must always be pressed to send and control the xArm-5's end-effector.

The X-axis of the xArm-5's end-effector can be controlled by pressing and holding the trigger button and moving the handle in a horizontal X-axis direction. The positive direction of the X-axis moves the end-effector towards the right side from the end-effector origin, and the negative direction of the X-axis moves the end-effector towards the left side from the end-effector origin.

The Y-axis of the xArm-5's end-effector can be controlled by pressing and holding the trigger button and moving the handle in a horizontal Y-axis direction. The positive direction of the Y-axis moves the xArm-5's end-effector towards the front side from the robot end-effector origin, and the negative direction of the Y-axis moves the robot end-effector towards the backward direction from the end-effector origin.

The positive Z-axis of the end-effector can be controlled by pressing and holding the trigger button and the button-2. This joystick setup moves the robot in an upward direction. The negative Z-axis of the end-effector can be controlled by pressing and holding the trigger button and the button-3. This joystick setup moves the robot in a downward direction.

Figure 6.6 Logitech Extreme 3D Pro [90]

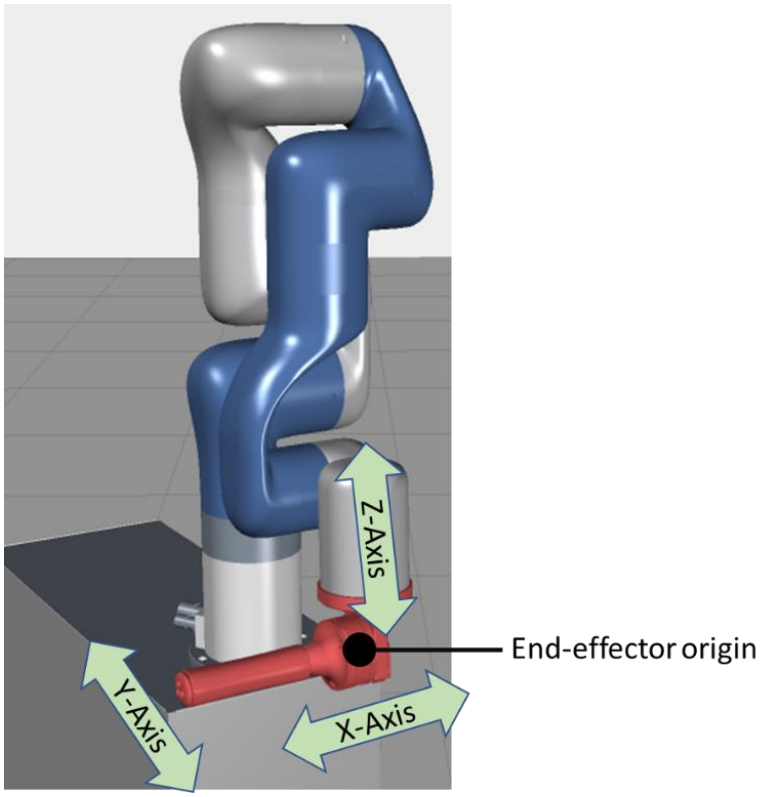


Figure 6.7 xArm-5 robot's end-effector axis coordination

6.6.1 Logitech Joystick Architecture

To control the robot end-effector remotely, we have developed Logitech joystick control architecture, which is depicted in Fig. 6.8. The operator (therapist/caregiver) connects the Logitech joystick with the computer (Windows-10).

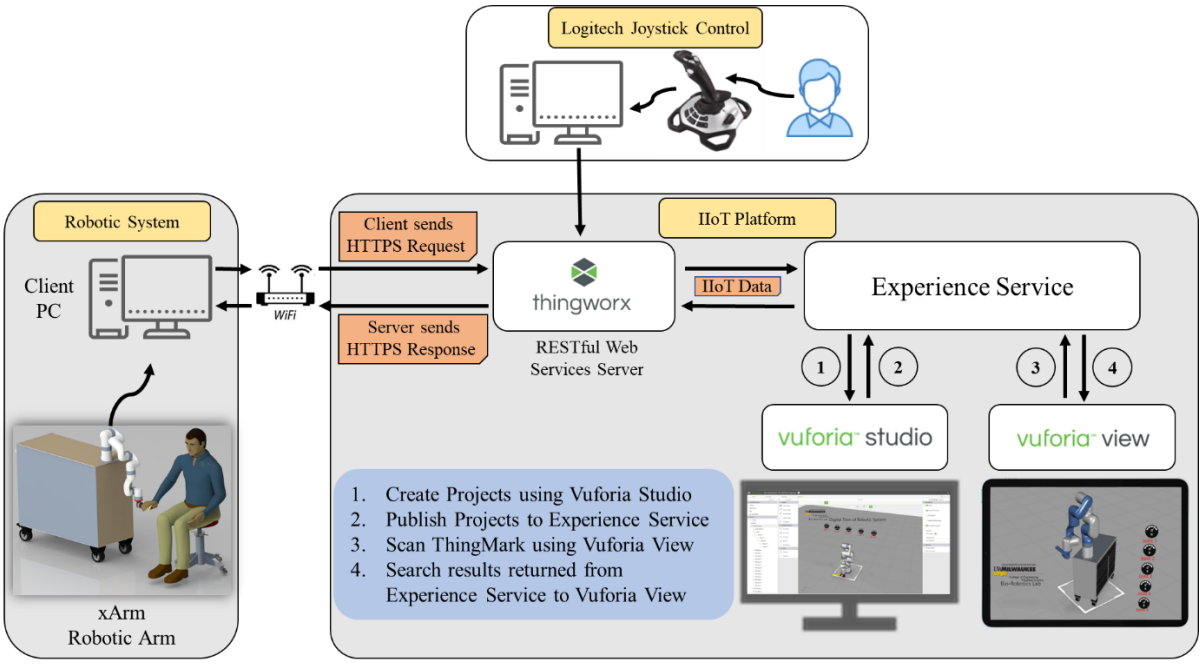


Figure 6.8 Logitech joystick architecture

REST API Python code (APPENDIX-B) pushes the data from the joystick to the PTC ThingWorx. When the ThingWorx platform receives the data, it automatically translates and reflects to the xArm-5 robot system and the Vuforia View/Vuforia Studio interface. For example, when the operator presses the trigger button and moves the joystick's handle towards the X-axis positive direction, the actual robot's end-effector also moves toward the right side from the end-effector origin. At the same time, the operator can monitor the robot's movement on the AR digital twin.

CHAPTER 7

EXPERIMENTS AND RESULTS

This chapter presents the experimental results to test and evaluate the functionality of the developed telemanipulation framework. The experiments were conducted with healthy participants to demonstrate the use of robot-assisted telerehabilitation. The developed system incorporates the principles of motor rehabilitation and neurorehabilitation (as discussed in Chapter 3). The beginning of this chapter describes the experimental setup with human-robot collaboration. The xArm-5 robot was used to provide various rehabilitation exercises (such as passive, resistive, manual teaching, etc.) via the developed telerehabilitation framework. During the experiments, the robot's joint and cartesian positions, torque, speed, and human-robot interactive forces are collected and compared with those of the digital twin of the robot. Experimental results show that the developed telerehabilitation framework can be effectively used to provide telerehabilitation exercises to the upper limb using a rehab robot.

Note: The term "Participants" refers to the healthy subjects/individuals who interact with the robot to get telerehabilitation therapy exercises during these experiments. And the word "Operator" refers to the person using the developed telemanipulation framework to provide rehabilitation therapy. An operator could be a therapist, clinician, caregiver, etc.

7.1 Experimental Setup

The experimental setup of the telerehabilitation system includes two different sides of arrangements: The operator's (therapist/caregiver) side and the participant's side. The operator's side experimental setup is depicted in Fig. 7.1, where an operator and participant communicate with each other in a video session using Microsoft Teams. With the help of this video session, an

operator can observe and interact with each participant while providing a variety of rehabilitation exercises, including passive rehab exercise (using pre-defined trajectory parameters), resistive rehab exercise (using impedance control), etc. Therapy modes need to be changed to provide a variety of rehab exercises, and to do that, an operator uses the Vuforia View app on iPad, as shown in Fig. 7.1 (developed using Vuforia Studio 3D/2D widgets). The modes are discussed in section 7.2. An operator uses the developed digital twin of an assistive robot to track the robot motion with low latency and to monitor the participant's feedback while interacting with the robot, as illustrated in Fig. 7.1.

On the other hand, the participant's side experimental setup is represented in Fig. 7.2, which includes communication devices and human-robot interaction.

As shown in Fig. 7.3, the xArm-5 robot system is mounted on a rolling cabinet to be mobile to provide telerehabilitation. Subjects sit on a chair at a specific distance from the rolling cabinet/robot for experiments. The upper arm motion must be captured to examine the human upper limb during rehabilitation exercises; for that, we have used the Kinect V2 sensor (Microsoft 2017). Figure 7.4 shows participants using a camera, Kinect sensor, and monitor to communicate with operators through Microsoft Teams during therapy sessions.

Experiments were conducted with healthy participants (age: 23–35 years, height: 1.60–1.80 m, weights: 51–110 kg) to validate various telerehabilitation techniques/approaches. Since it is telerehabilitation, we simulated a use case scenario by placing the participant's side system and operator's side system in two separate rooms. During the experiments, subjects hold the end-effector handle of the xArm-5 robot. In our experiments, the robotic system and IIoT platform experienced a low latency of around 0.16s.

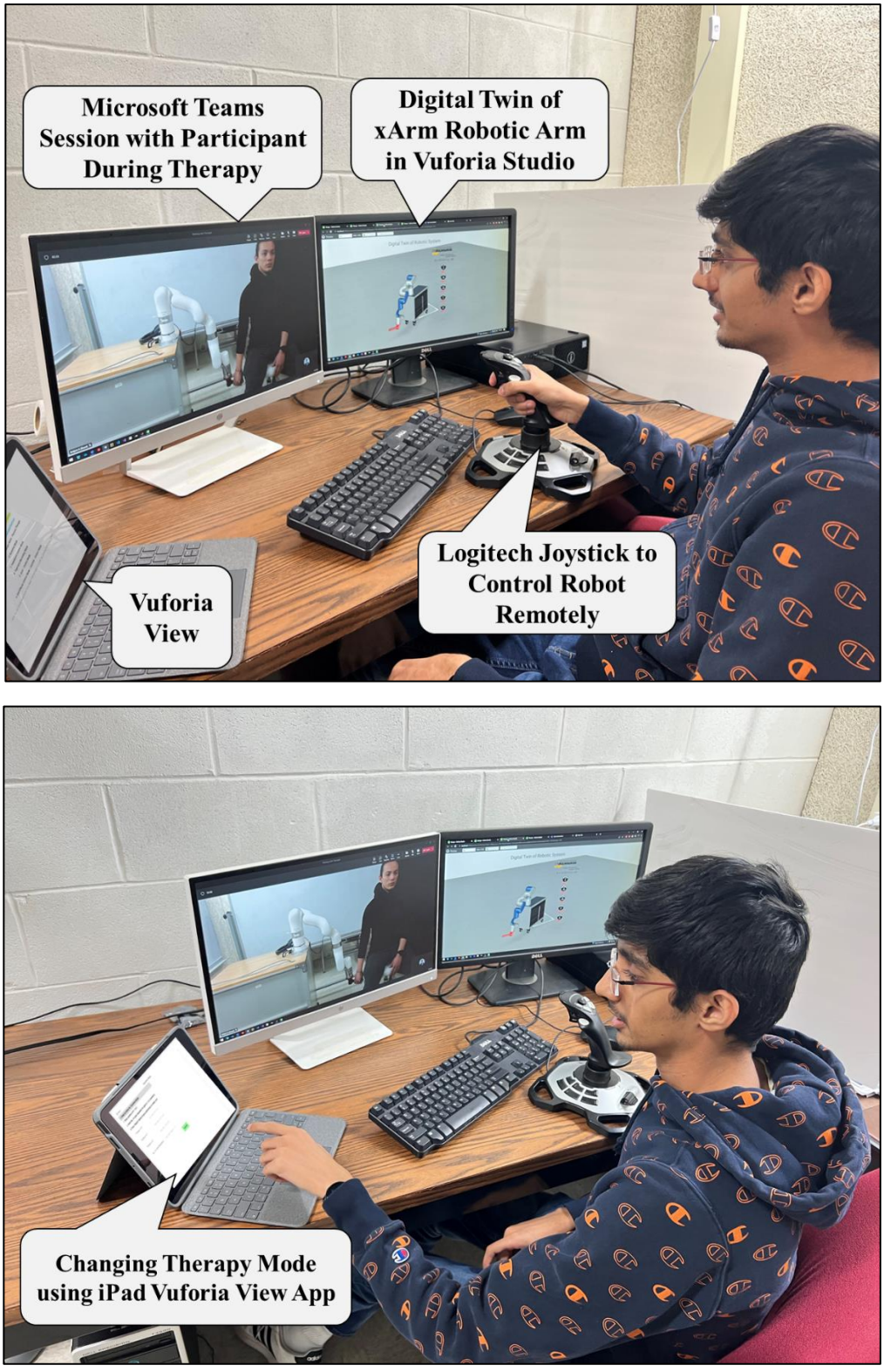


Figure 7.1 An experimental setup on the operator's end

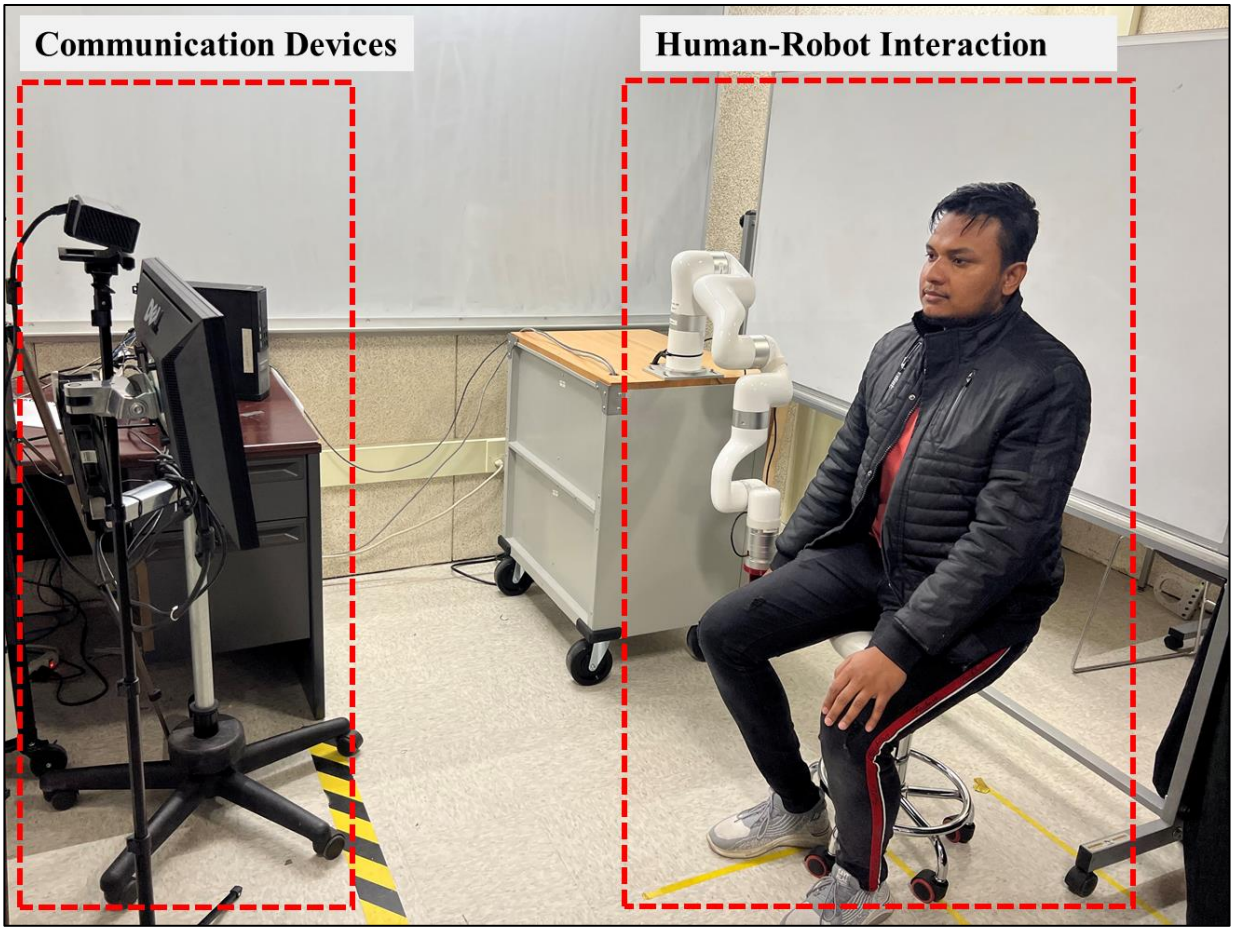


Figure 7.2 An experimental setup on the participant's end

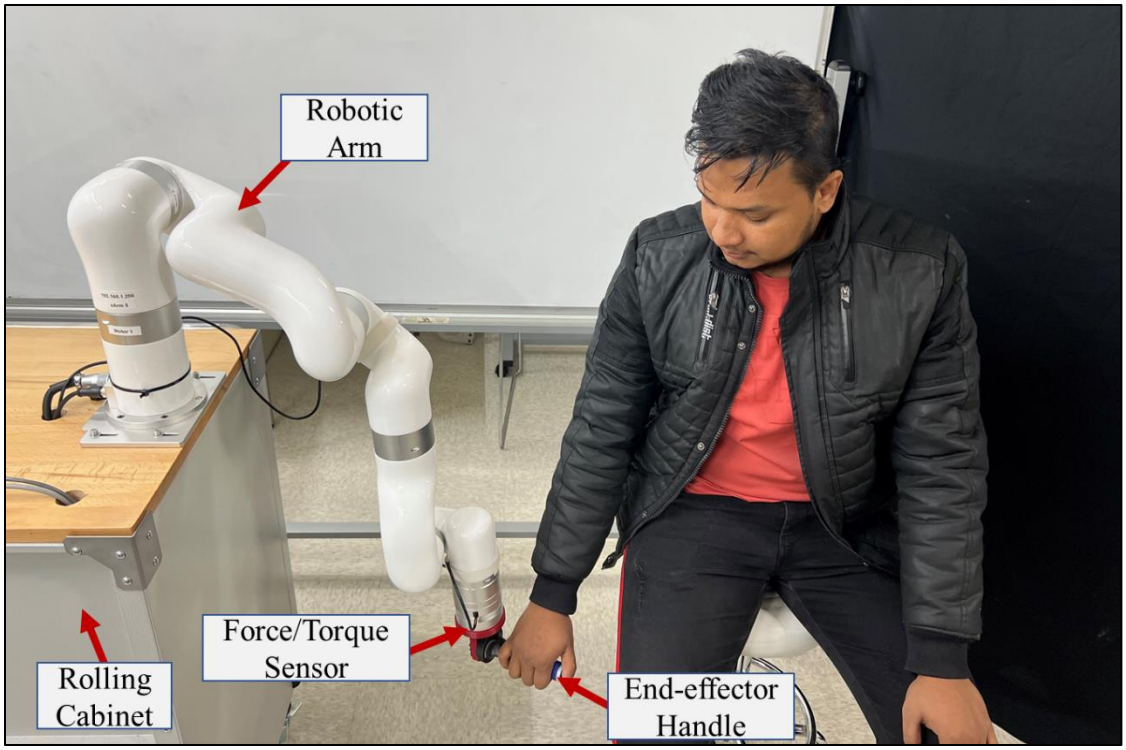


Figure 7.3 Human-robot collaboration on the participant's end

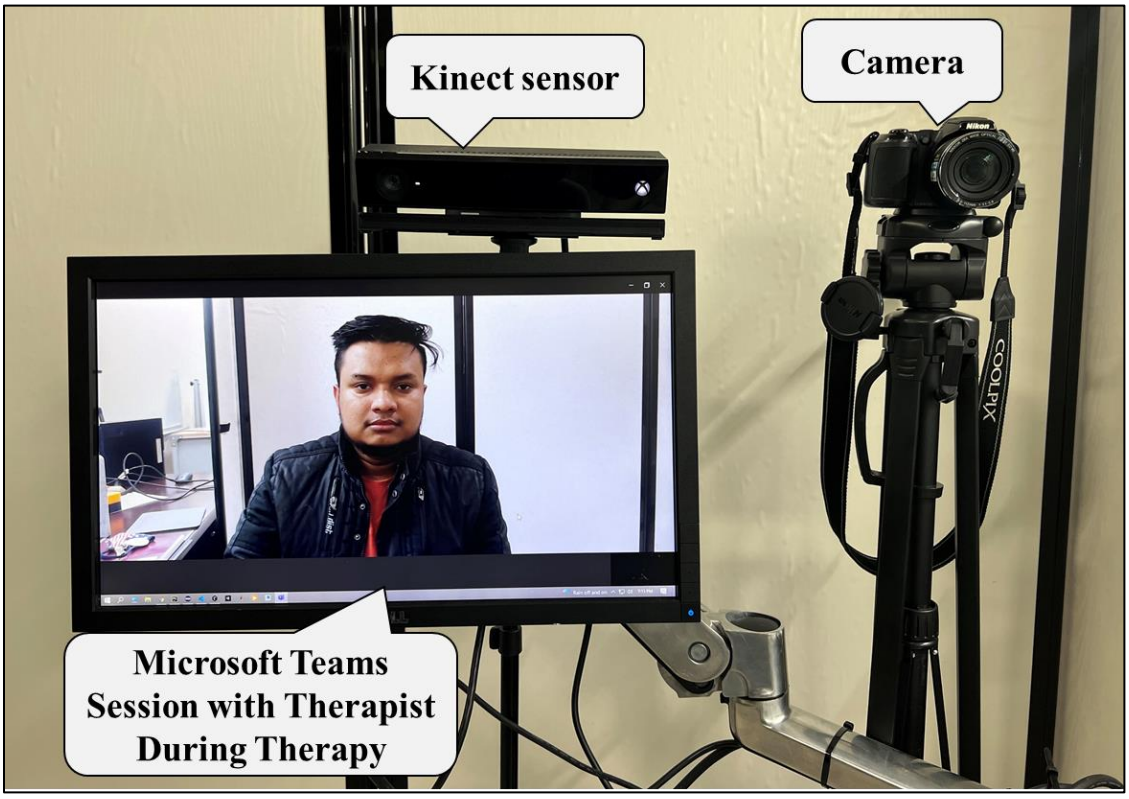


Figure 7.4 Commination device setup on the participant's end

7.2 Telerehabilitation Exercise Experiments

We developed six modes to provide different telerehabilitation exercises, including-

(i) Passive Rehab Exercise (PRE) mode using Pre-determined Trajectory,

(ii) PRE mode with Real-Time Recorded Trajectory (PREwRT^2),

(iii) Resistive Rehab Exercise (RRE) mode using Impedance Control,

(iv) Manual Teaching mode for Generating and Providing PRE,

(v) Interactive One-on-One Real-Time TeleRehabilitation Exercise (IO^{3}RT^{2}RE) Mode, Controlling Individual Joints of the Robot to Provide PRE, and

(vi) Interactive One-on-One Real-Time TeleRehabilitation Exercise (IO^{3}RT^{2}RE) Mode, Controlling Robots' End-Effector Position using a Joystick to provide PRE.

The operator can change the different exercise methods/modes during experiments through the Vuforia View user interface, as shown in Fig. 7.5. Each exercise mode has its unique user interface. Experiments and results of each therapeutic mode are broadly discussed in sections 7.2.1 – 7.2.6.

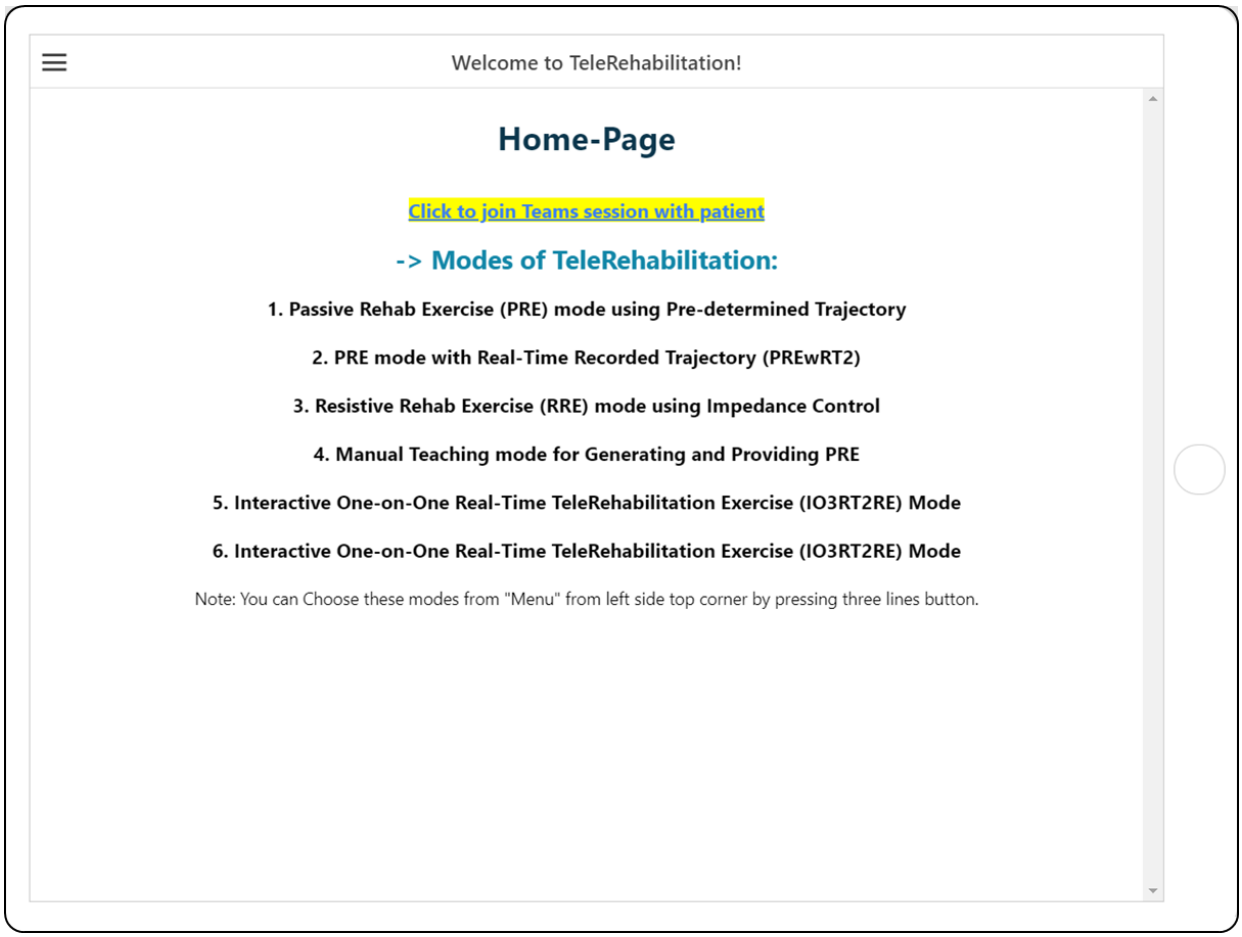


Figure 7.5 Welcome page of telerehabilitation Vuforia View app in iPad

7.2.1 Passive Rehab Exercise (PRE) Mode Using Pre-determined Trajectory

The passive mode of telerehabilitation exercises has been evaluated with the xArm-5 robot while remotely controlling, monitoring, and running the experiments. Passive mode targets the massed and repetitive practice of the exercises. In this mode, the rehab device, the xArm-5 robot, moves the participant's arm in 2D/3D space to increase the range of motion and reduce spasticity. Exercising this way can improve the effectiveness of therapy and speed up recovery. These are the robot's joint-based trajectories, which are generated by simulating the motion of the participant's arm. Figure 7.6 shows the user interfaces for administering the developed trajectories /exercises for shoulder, elbow, and forearm motion.

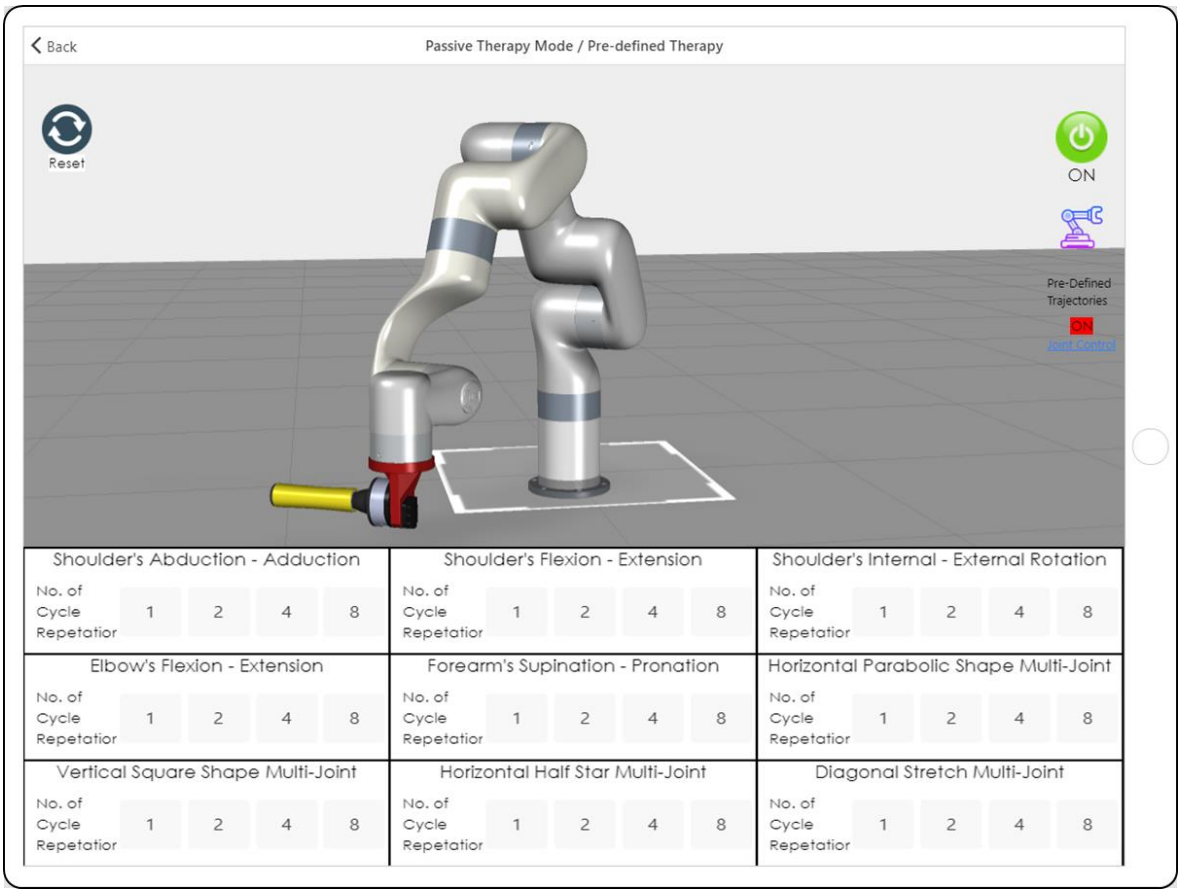


Figure 7.6 Augmented reality user interface for a passive mode of telerehabilitation exercises.

Here, the first five exercises focused on each upper limb's joint movements, including shoulder, elbow, and forearm. The other four focused on multi-joint movement exercises of the upper limb's joints. Additionally, all exercises can be repeated by clicking the repetition number under specific exercises and monitoring on AR digital twin. The preferred trajectories are:

- **Shoulder Abduction – Adduction**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 115° (joint-2), -118.9° (joint-3), 0° (joint-4), and -57.5° (Joint-5). Figures 7.8 through 7.11 show the recorded experiment data for shoulder joint abduction-adduction exercise. As shown in Figure 7.8, during the abduction/adduction exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to -32.9°, -37°, -33.2°, -37°, -32.9°, and finally returns to 0°;

  – Joint-2 moves from its initial position (115^o) to 78.1°, 50.6°, 34.2°, 50.6°, 78.1°, and finally returns to 115°;

  – Joint-3 stays close to its initial position (-118.9^o);

  – Joint-4 moves from its initial position (0^o) to 45.7°, 69.5°, 89°, 69.5°, 45.7°, and finally returns to 0°, and

  – Joint-5 moves from its initial position (-57.5^o) to -122.1°, stays at -122.1° for around 10 sec, and finally returns to -57.5°. The experiment uses one repetition.

  Figure 7.9 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the shoulder joint abduction–adduction exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.10 and 7.11, respectively.

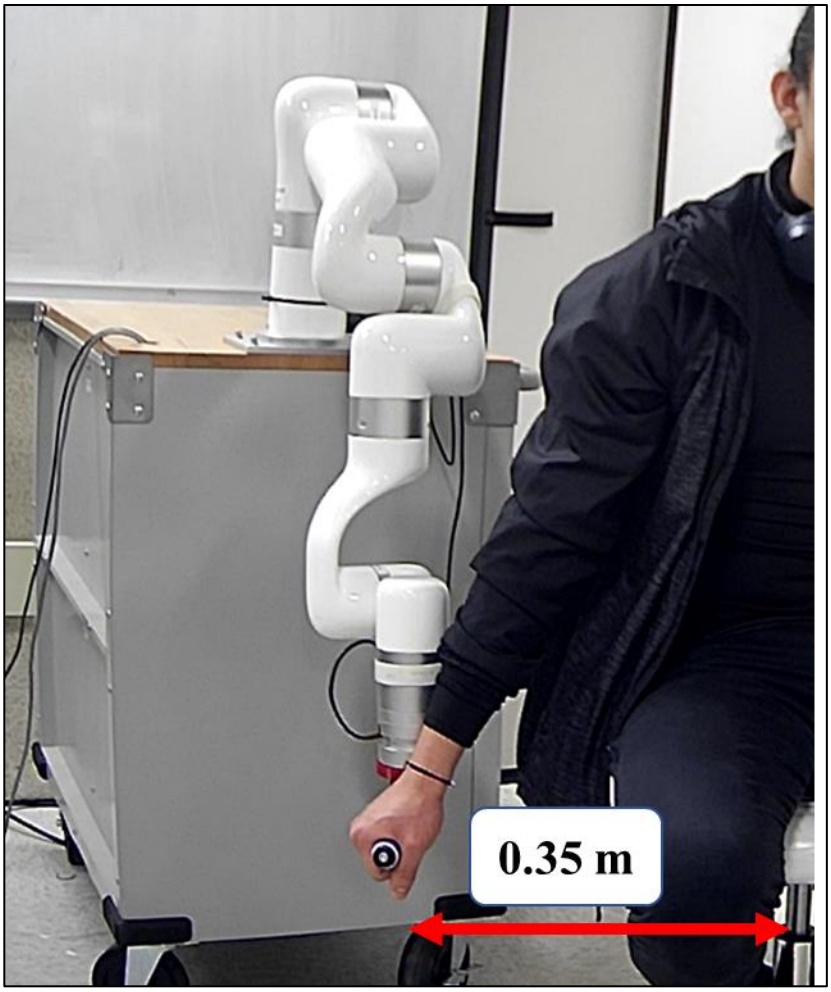


Figure 7.7 Participant sitting position during shoulder joint abduction–adduction exercise.

As shown in Fig. 7.7, a participant sat on the chair at a distance of 0.35 m from the end-effector handle of the robot and faced opposite the rolling cabinet during the shoulder joint abduction-adduction exercise.

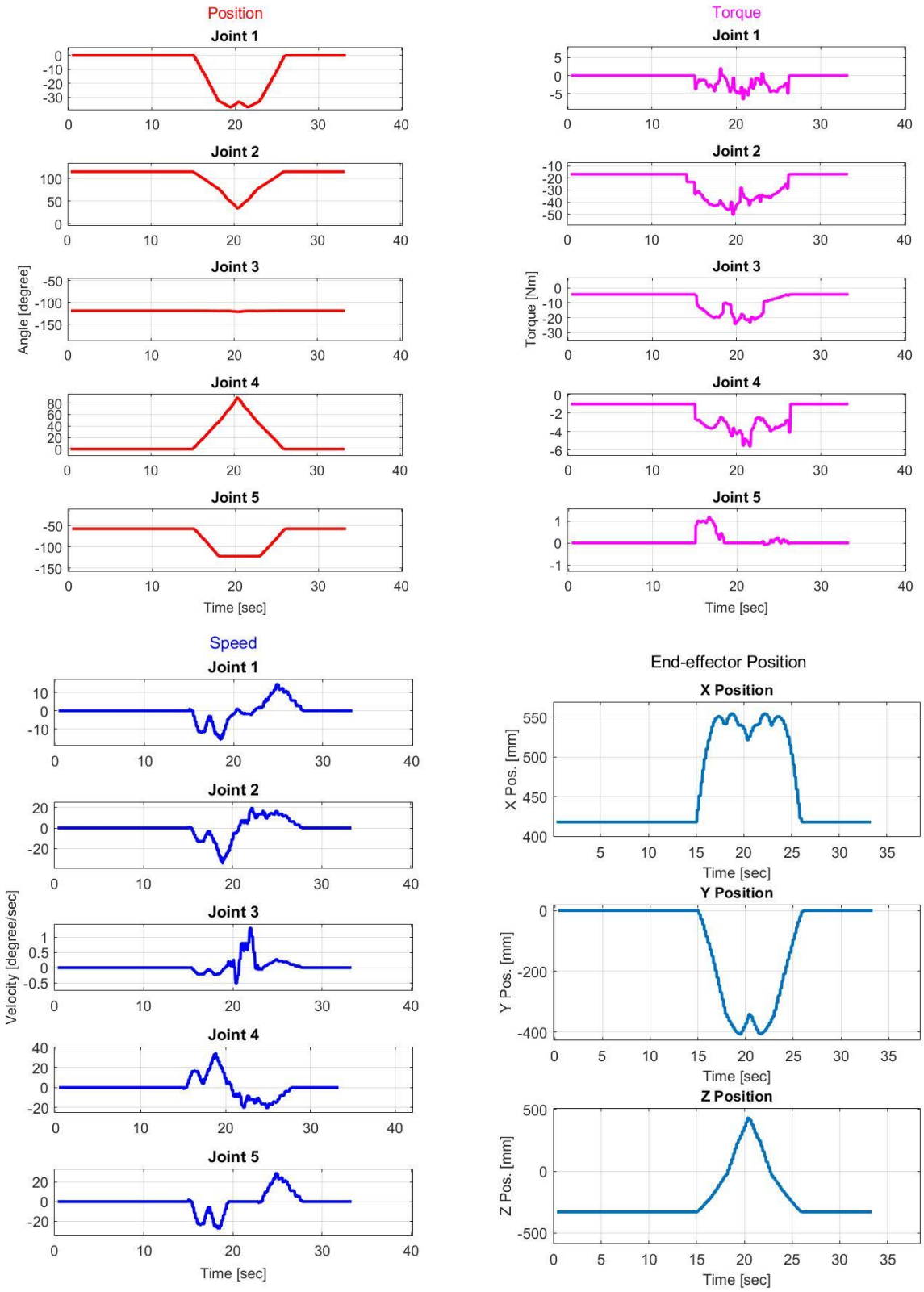


Figure 7.8 Joint angles, torques, speed, and end-effector position during shoulder joint abduction–adduction exercise.

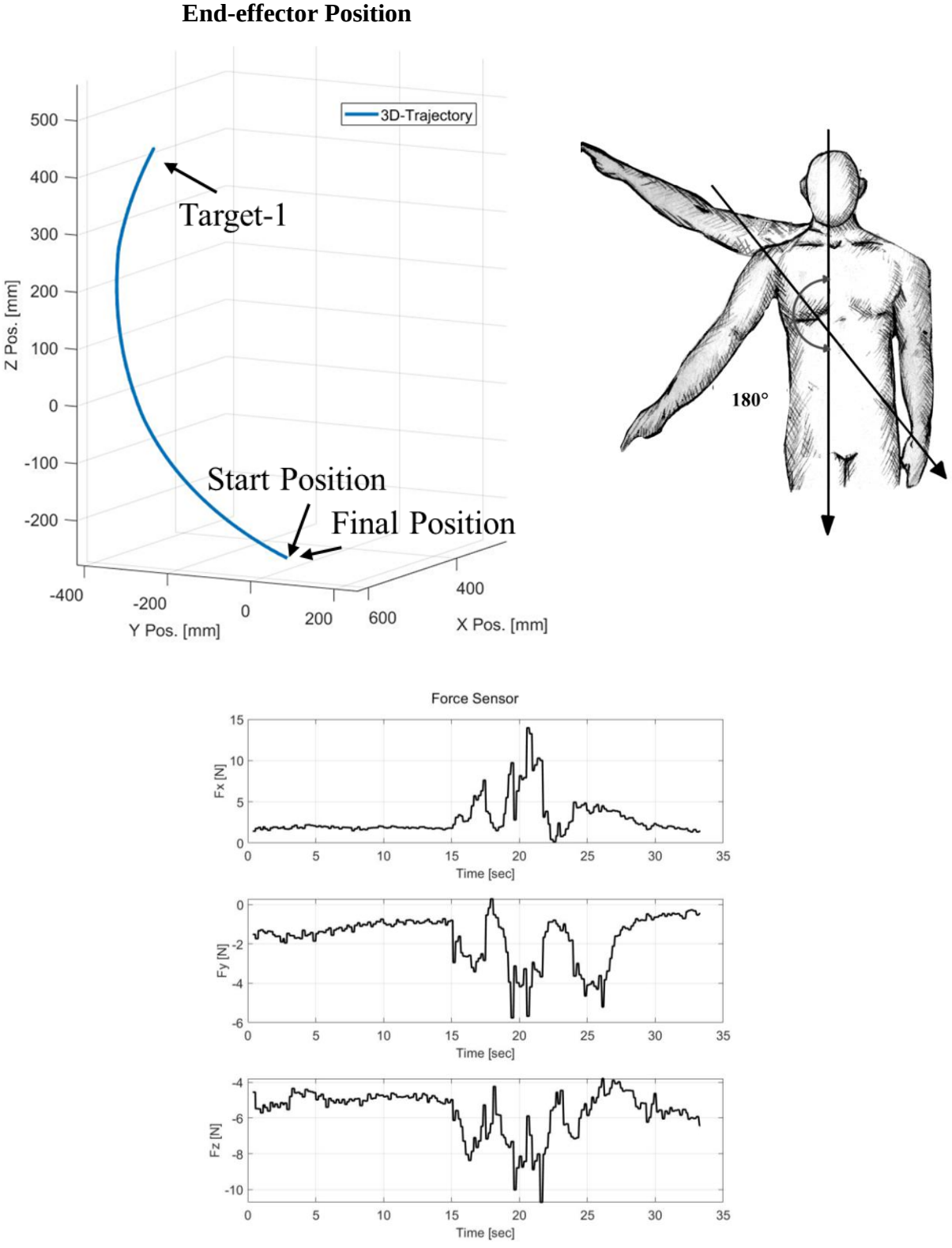


Figure 7.9 Human-robot interactive force detected from the force sensor at the end-effector during shoulder joint abduction–adduction exercise.

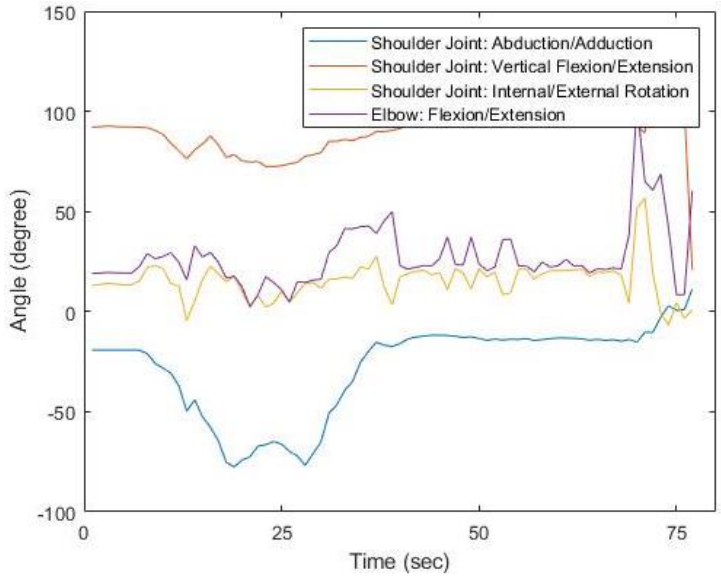


Figure 7.10 Upper-limb joint coordinate from Kinect sensor during shoulder joint abduction–adduction exercise.

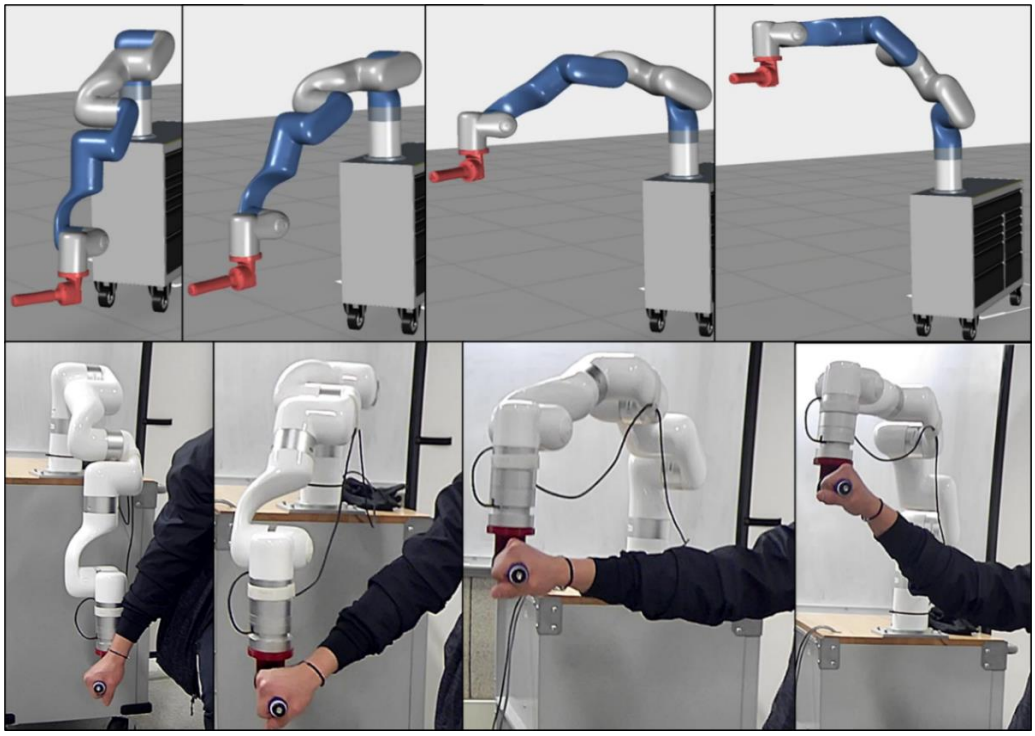


Figure 7.11 Monitoring shoulder joint abduction–adduction exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Shoulder Flexion – Extension**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 115° (joint-2), -118.9° (joint-3), 0° (joint-4), and -90° (Joint-5). Figures 7.13 through 7.16 show the recorded experiment data for shoulder joint flexion-extension exercise. As shown in Figure 7.13, during the flexion-extension exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to -32.9°, -37°, -33.2°, -15.5°, 1.4°, 36.7°, 1.4°, -15.5°, -33.2°, -37°, -32.9°, and finally returns to 0°;

  – Joint-2 moves from its initial position (115^o) to 78.1°, 50.6°, 34.2°, 17.2°, 12.2°, 26.2°, 12.2°, 17.2°, 34.2°, 50.6°, 78.1°, and finally returns to 115°;

  – Joint-3 moves from its initial position (-118.9°) to -119.5°, -119.4°, -122.1°, -130.2°, -139.1°, -144.7°, -139.1°, -130.2°, -122.1°, -119.4°, -119.5°, and finally returns to -118.9°;

  – Joint-4 moves from its initial position (0^o) to 45.7°, 69.5°, 89°, 112.8°, 124.1°, 118.8°, 124.1°, 112.8°, 89°, 69.5°, 45.7, and finally returns to 0°, and

  – Joint-5 moves from its initial position (-90^o) to -122.1°, -122.1°, -122.1°, -110.7°, -94.7°, -59.1°, -94.7°, -110.7°, -122.1°, -122.1°, -122.1°, and finally returns to -90°. The experiment uses two repetitions.

  Figure 7.14 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the shoulder joint flexion-extension exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.15 and 7.16, respectively.

Figure 7.12 Participant sitting position during shoulder joint flexion-extension exercise.

As shown in Fig. 7.12, a participant sat on the chair at a distance of 0.71 m from the rolling cabinet holding the robot's end-effector (handle) and the facing sideway of the rolling cabinet during the shoulder joint flexion-extension exercise.

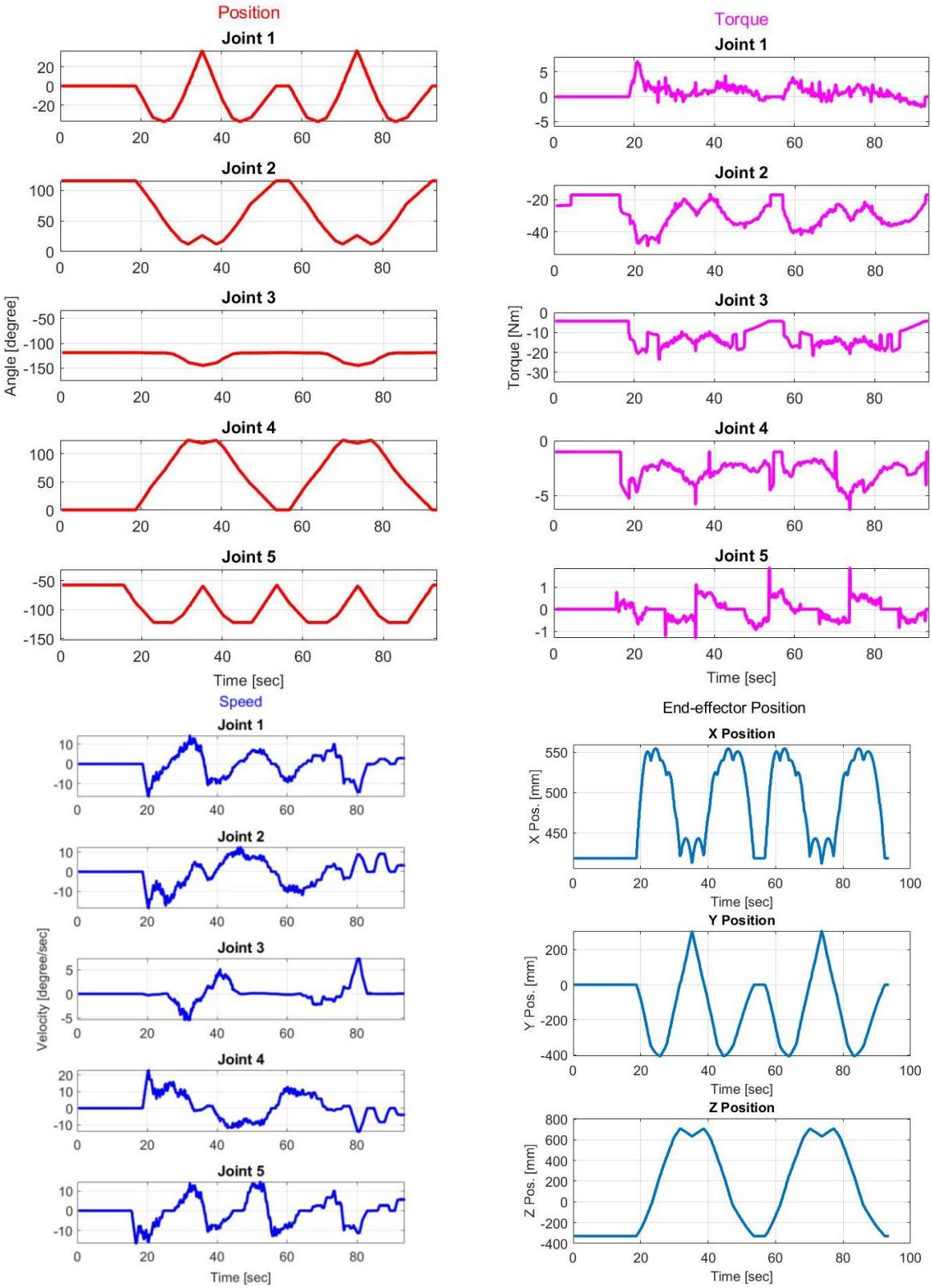


Figure 7.13 Joint angles, torques, speed, and end-effector position during shoulder joint flexion-extension exercise.

End-effector Position

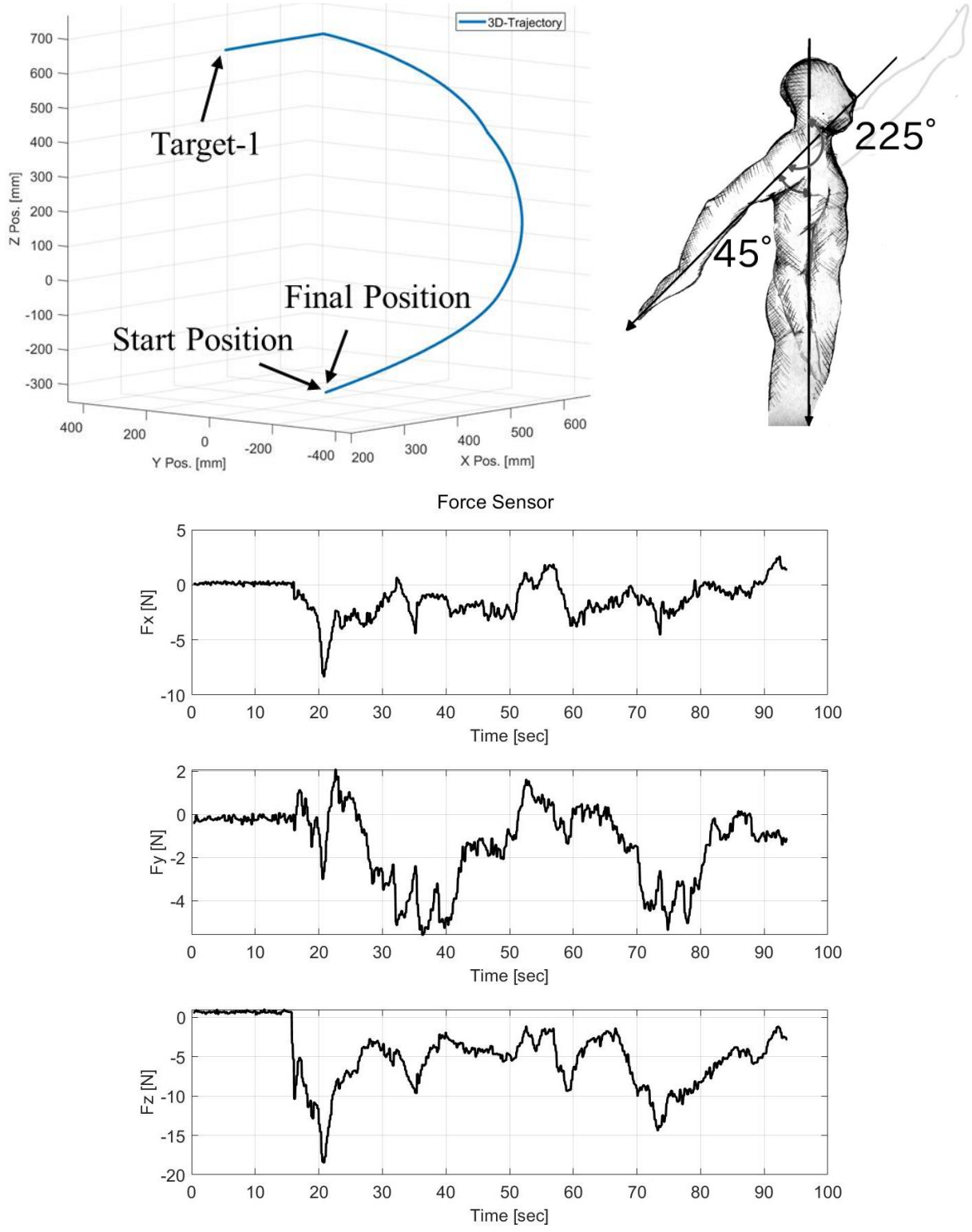


Figure 7.14 Human-robot interactive force detected from the force sensor at the end-effector during shoulder joint flexion-extension exercise.

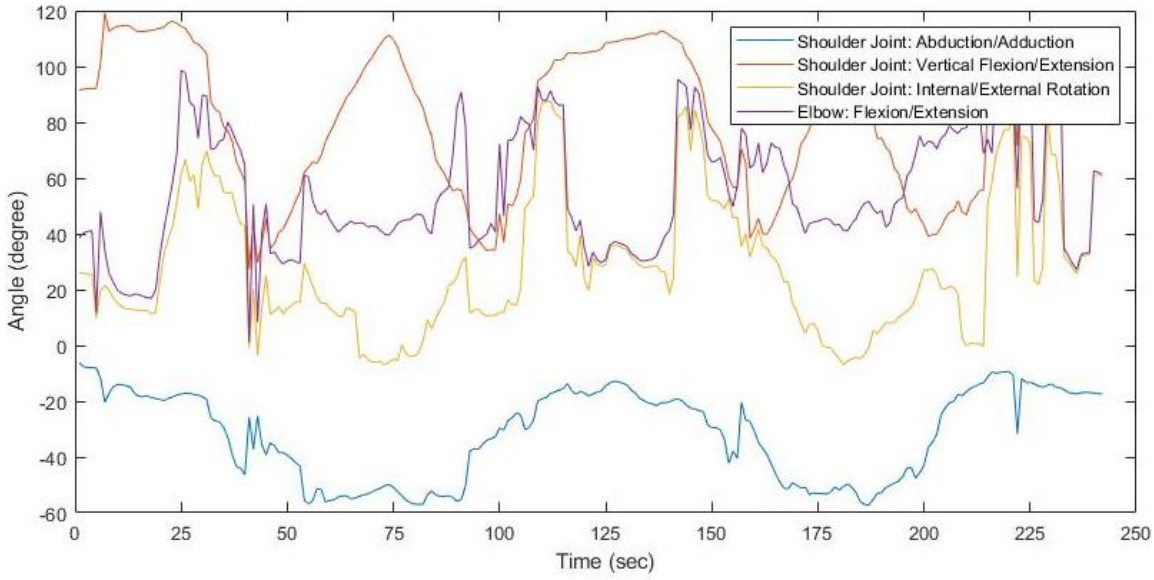


Figure 7.15 Upper-limb joint coordinate from Kinect sensor during shoulder joint flexion-extension exercise.

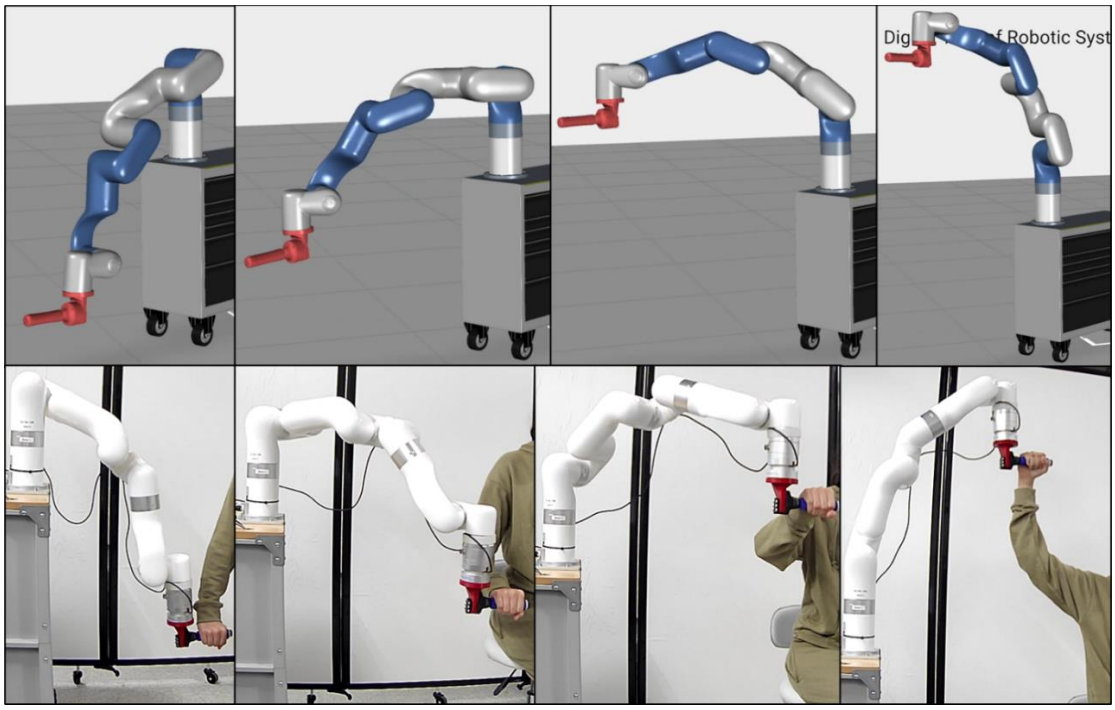


Figure 7.16 Monitoring shoulder joint flexion-extension exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Shoulder Internal – External Rotation**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 117.1° (joint-2), -121.3° (joint-3), -85.8° (joint-4), and 91.2° (Joint-5). Figures 7.18 through 7.21 show the recorded experiment data for shoulder joint internal-external rotation exercise. As shown in Figure 7.18, during the internal-external rotation exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to -27.2°, 0°, 22.3°, and finally returns to 0°;

  – Joint-2 stays in its initial position (117.1^o);

  – Joint-3 moves from its initial position (-121.3^o) to -142.8°, -121.3°, -142.3°, and finally return to -121.3^o;

  – Joint-4 moves from its initial position (-85.8^o) to -59.5°, -85.8°, -59.3°, and finally returns to -85.8°, and

  – Joint-5 stays in its initial position (91.2^o). The experiment uses four repetitions.

  Figure 7.19 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the shoulder joint internal-external rotation exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.20 and 7.21, respectively.

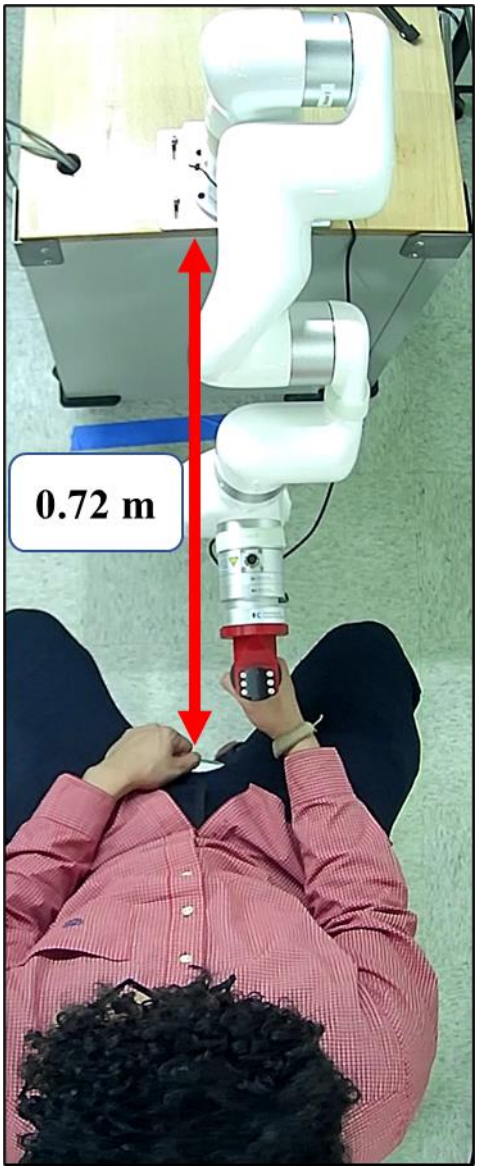


Figure 7.17 Participant sitting position during shoulder joint internal-external rotation exercise.

As shown in Fig. 7.17, a participant sat on the chair at a distance of 0.72 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the shoulder joint internal-external rotation exercise.

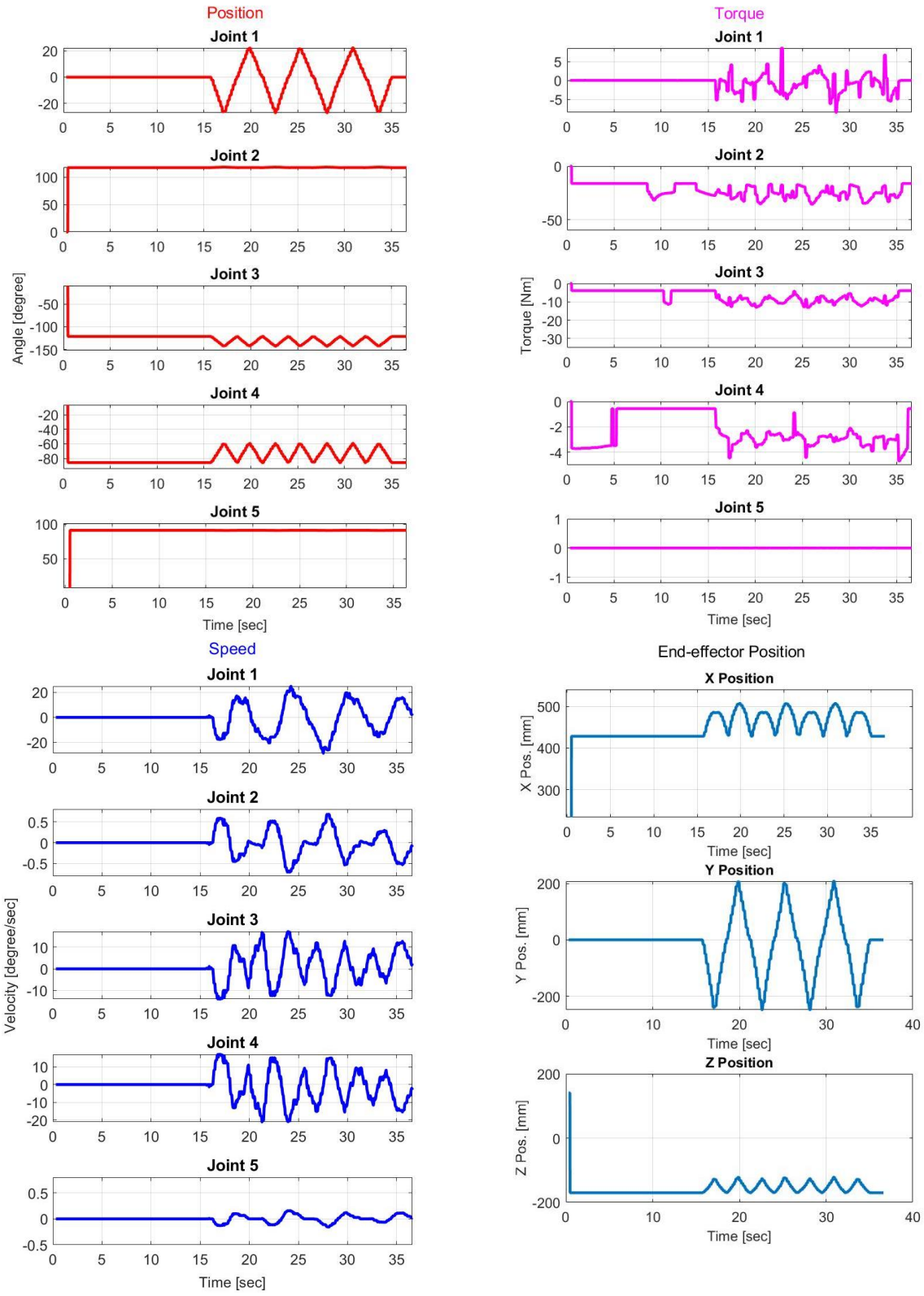


Figure 7.18 Joint angles, torques, speed, and end-effector position during shoulder joint internal-external rotation exercise.

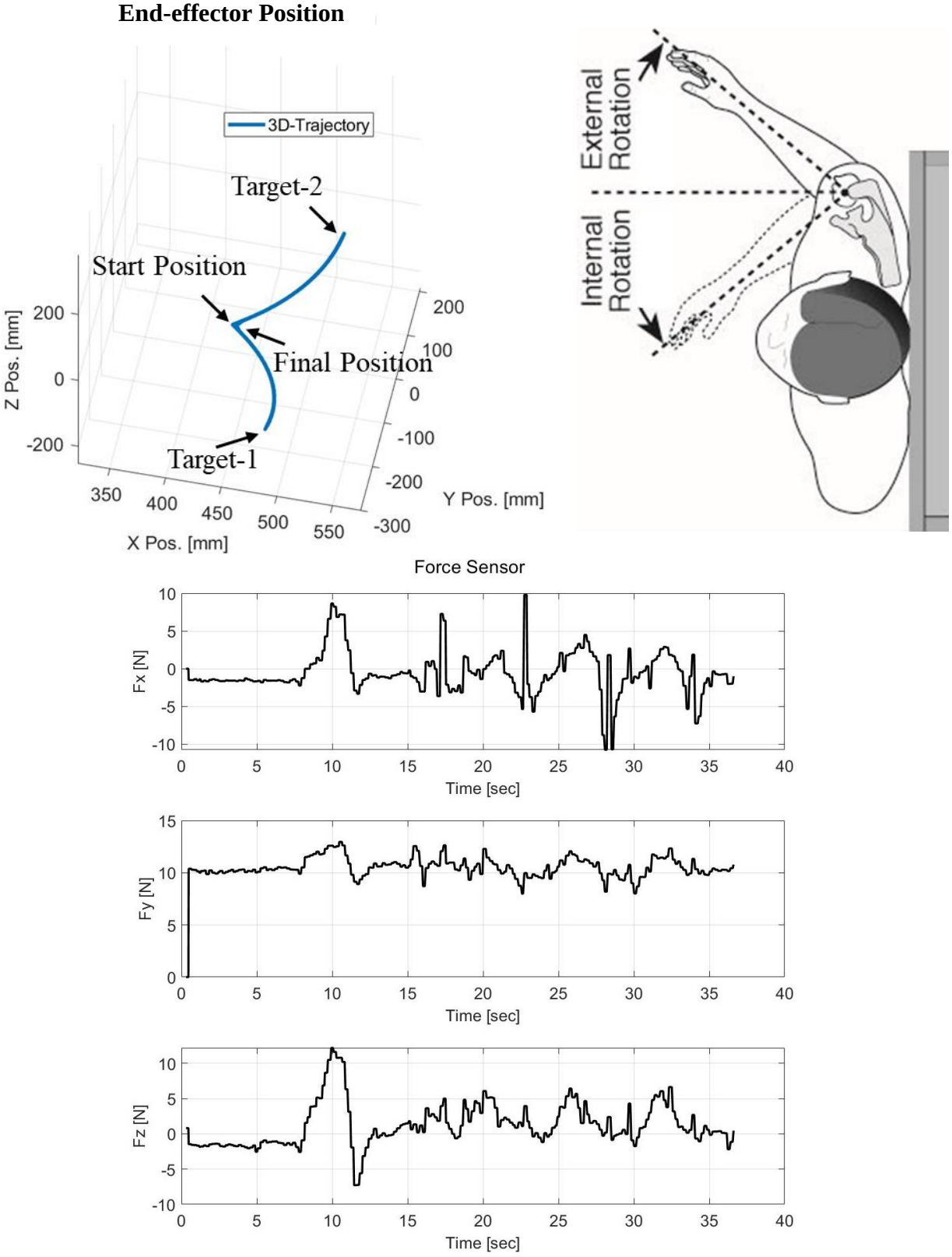


Figure 7.19 Human-robot interactive force detected from the force sensor at the end-effector during shoulder joint internal-external rotation exercise.

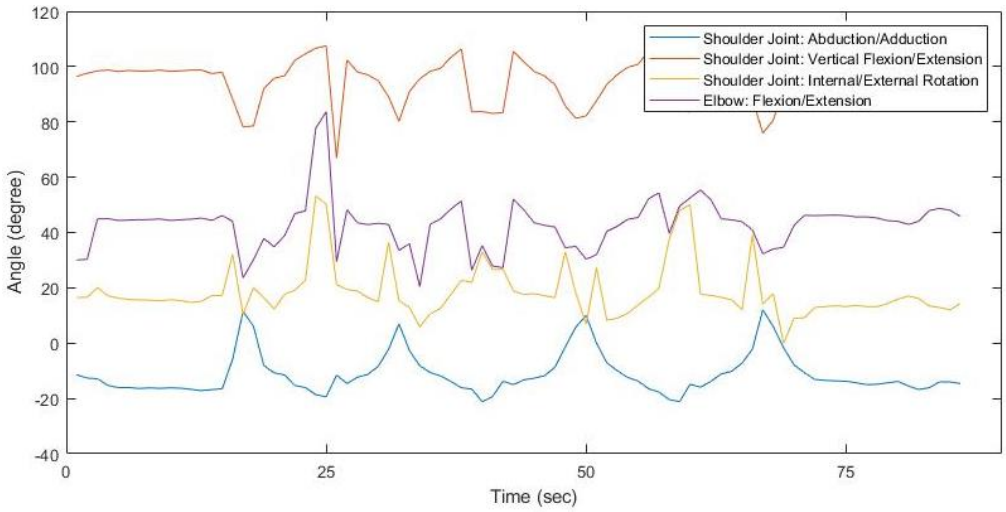


Figure 7.20 Upper-limb joint coordinate from Kinect sensor during shoulder joint internal-external rotation exercise.

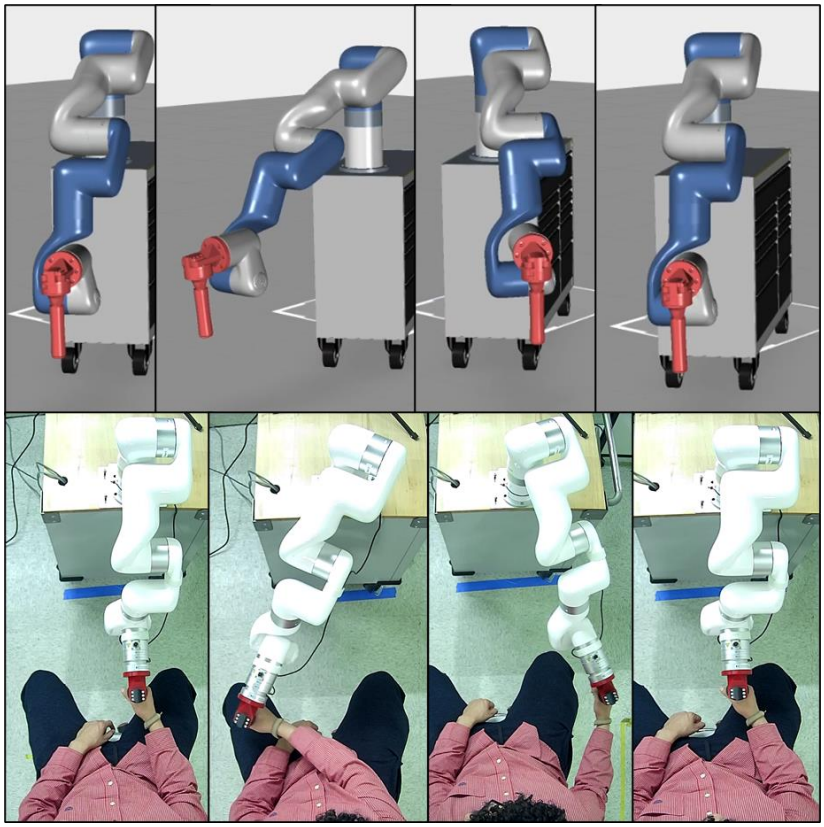


Figure 7.21 Monitoring shoulder joint internal-external rotation exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Elbow Flexion – Extension**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 117.1° (joint-2), -121.3° (joint-3), -85.8° (joint-4), and 0° (Joint-5). Figures 7.23 through 7.26 show the recorded experiment data for elbow joint flexion-extension exercise. As shown in Figure 7.23, during the flexion-extension exercise, the robot's joints are moved as follows:

  – Joint-1 stays in its initial position (0^o);

  – Joint-2 stays in its initial position (117.1^o);

  – Joint-3 moves from its initial position (-121.3^o) to -108.9°, -133.8°, -161°, -133.8°, -108.9°, and finally return to -121.3^o;

  – Joint-4 moves from its initial position (-85.8^o) to -25.5°, -75°, -89°, -75°, -25.5°, and finally returns to -85.8°, and

  – Joint-5 stays in its initial position (0^o). The experiment uses two repetitions.

  Figure 7.24 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the elbow joint flexion-extension exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.25 and 7.26, respectively.

Figure 7.22 Participant sitting position during elbow joint flexion-extension exercise.

As shown in Fig. 7.22, a participant sat on the chair at a distance of 0.66 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the elbow joint flexion-extension exercise.

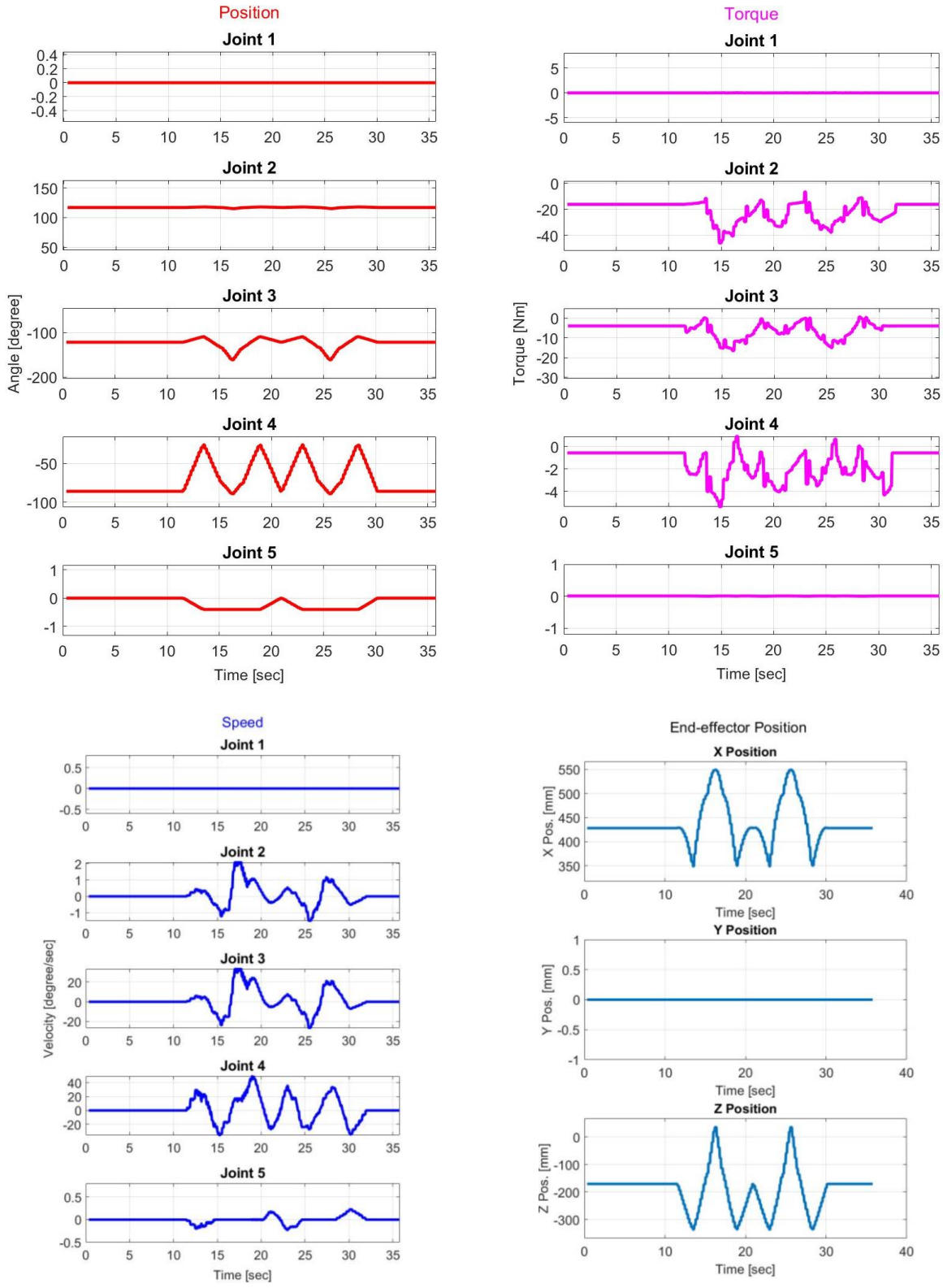


Figure 7.23 Joint angles, torques, speed, and end-effector position during elbow joint flexion-extension exercise.

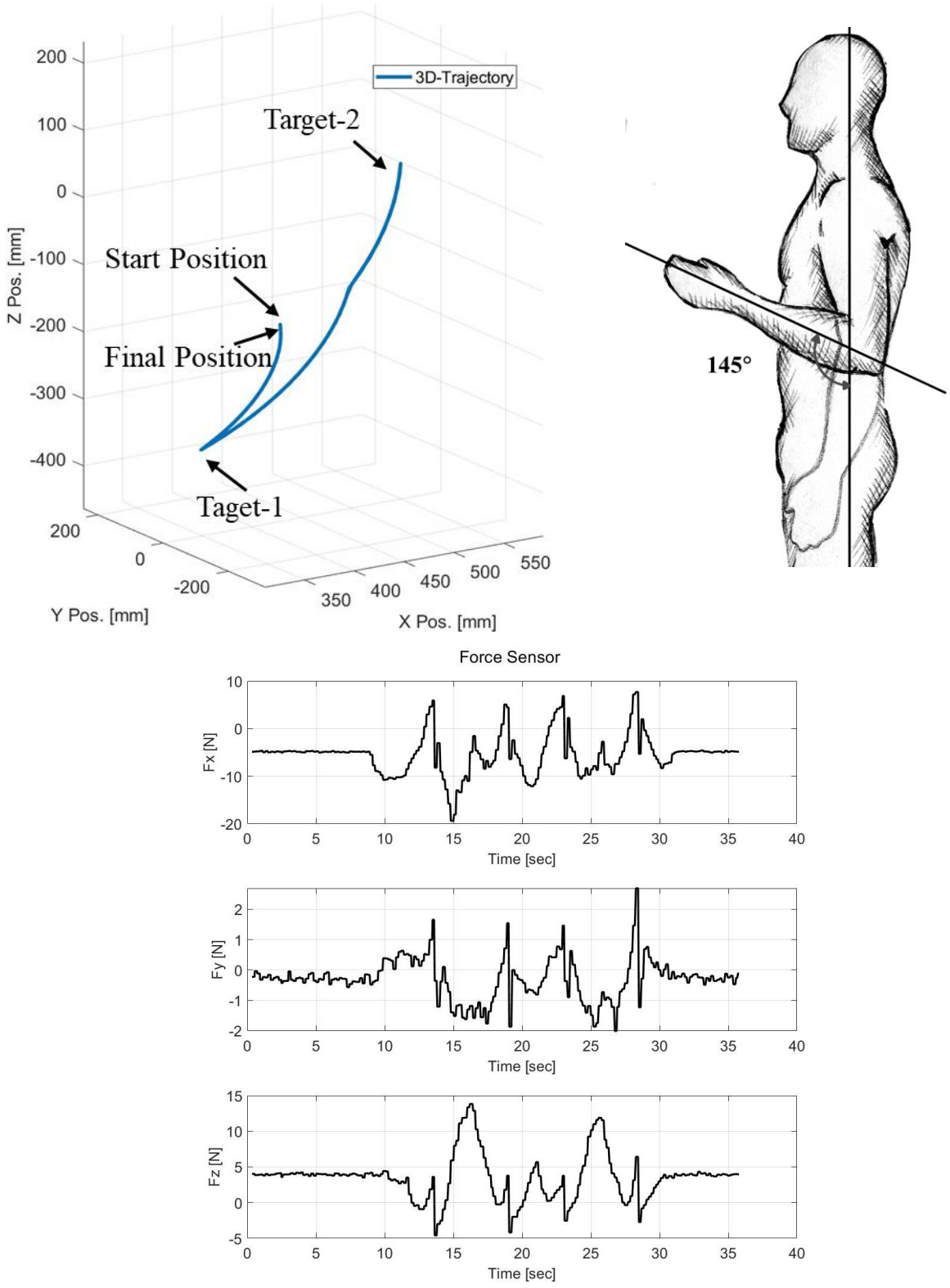


Figure 7.24 Human-robot interactive force detected from the force sensor at the end-effector during elbow joint flexion-extension exercise.

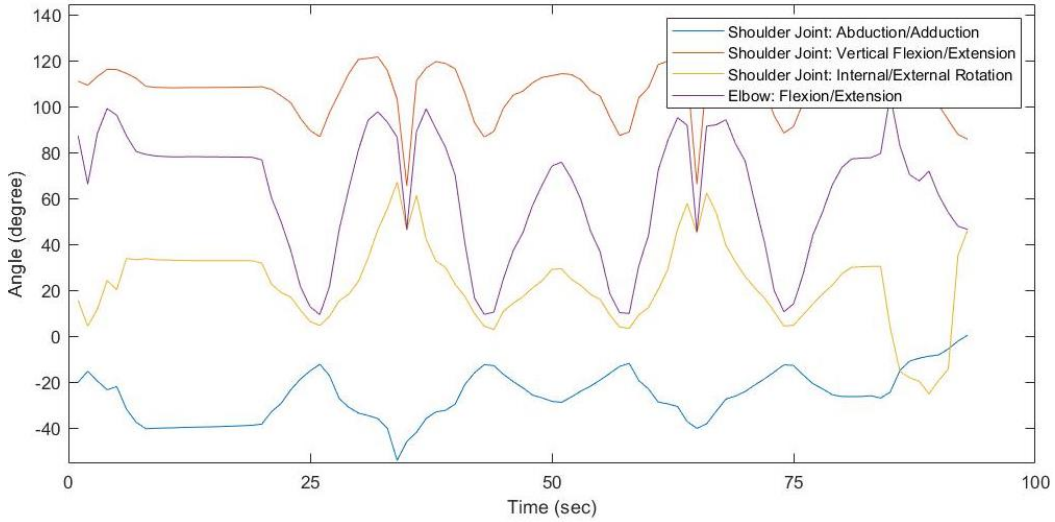


Figure 7.25 Upper-limb joint coordinate from Kinect sensor during elbow joint flexion-extension exercise.

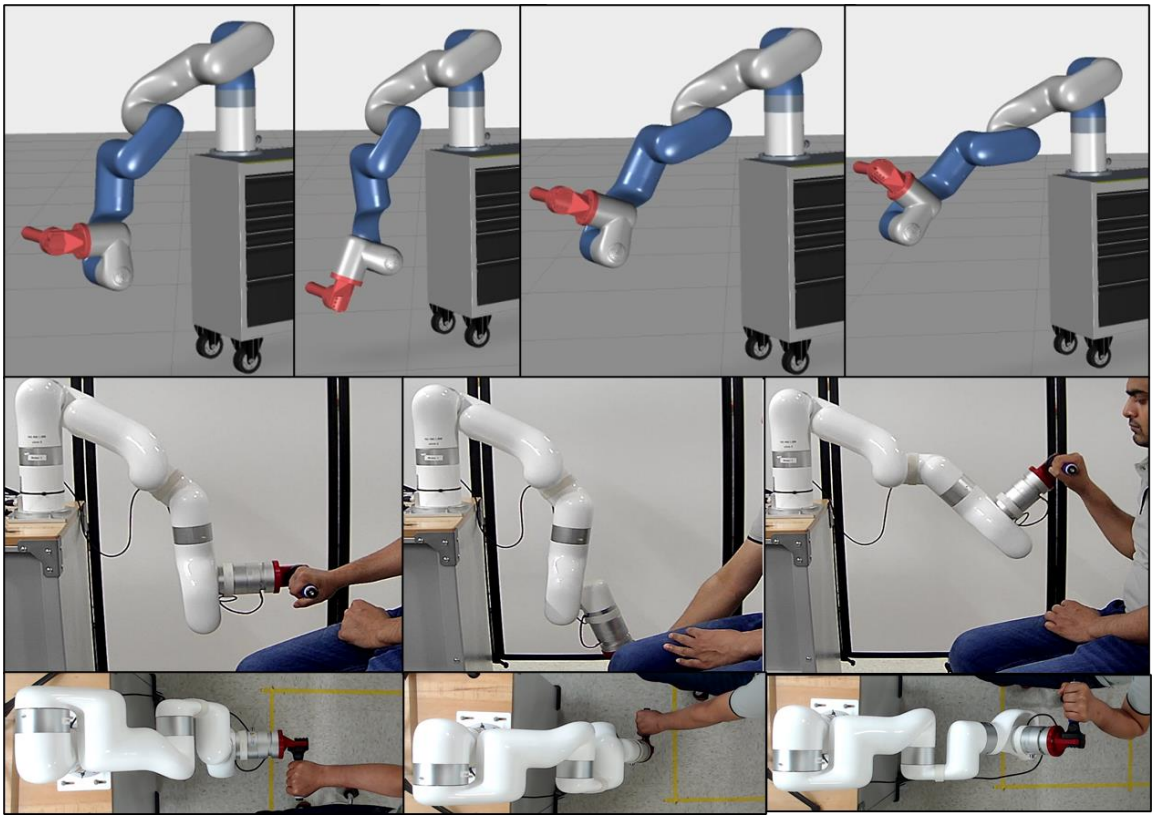


Figure 7.26 Monitoring elbow joint flexion-extension exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Forearm Supination – Pronation**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 117.1° (joint-2), -121.3° (joint-3), -85.8° (joint-4), and 0° (Joint-5). Figures 7.28 through 7.31 show the recorded experiment data for forearm joint supination-pronation exercise. As shown in Figure 7.28, during the supination-pronation exercise, the robot's joints are moved as follows:

  – Joint-1 stays in its initial position (0^o);

  – Joint-2 stays in its initial position (117.1^o);

  – Joint-3 stays in its initial position (-121.3^o);

  – Joint-4 stays in its initial position (-85.8^o), and

  – Joint-5 moves from its initial position (0^o) to -160° and returns to 0°. The experiment uses four repetitions.

  Figure 7.29 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the forearm joint supination-pronation exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.30 and 7.31, respectively.

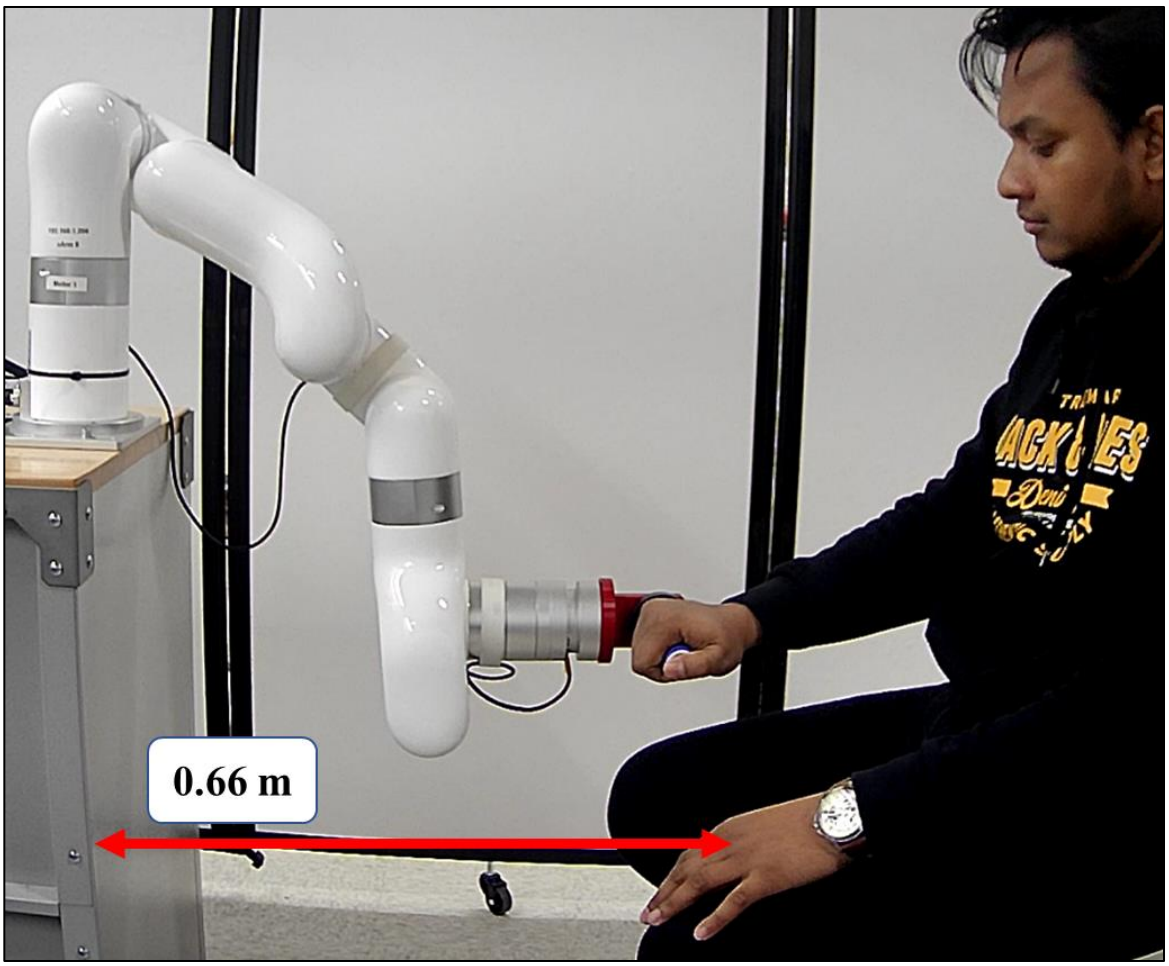


Figure 7.27 Participant sitting position during forearm joint supination–pronation exercise.

As shown in Fig. 7.27, a participant sat on the chair at a distance of 0.66 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the forearm joint supination – pronation exercise.

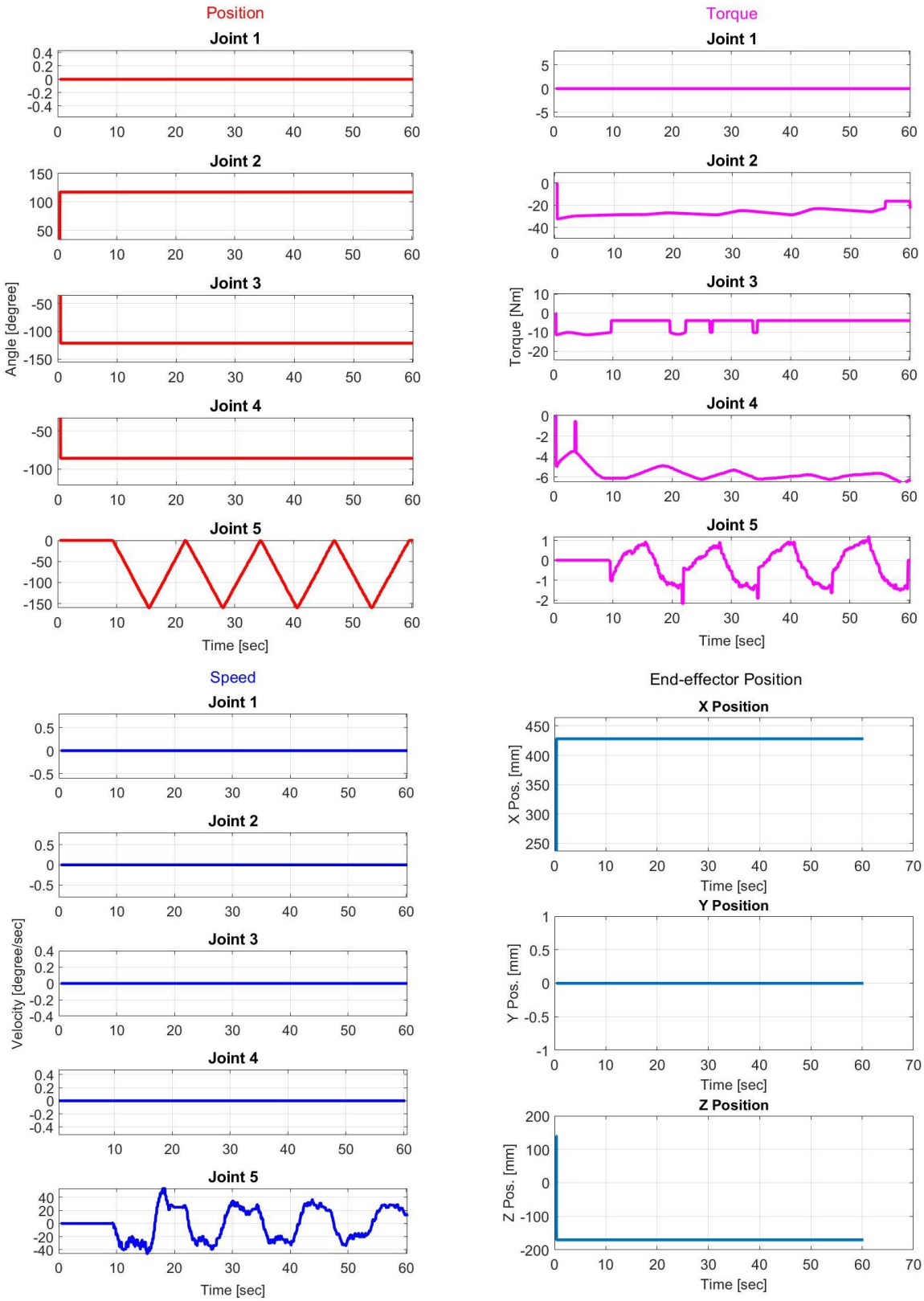


Figure 7.28 Joint angles, torques, speed, and end-effector position during forearm joint supination–pronation exercise.

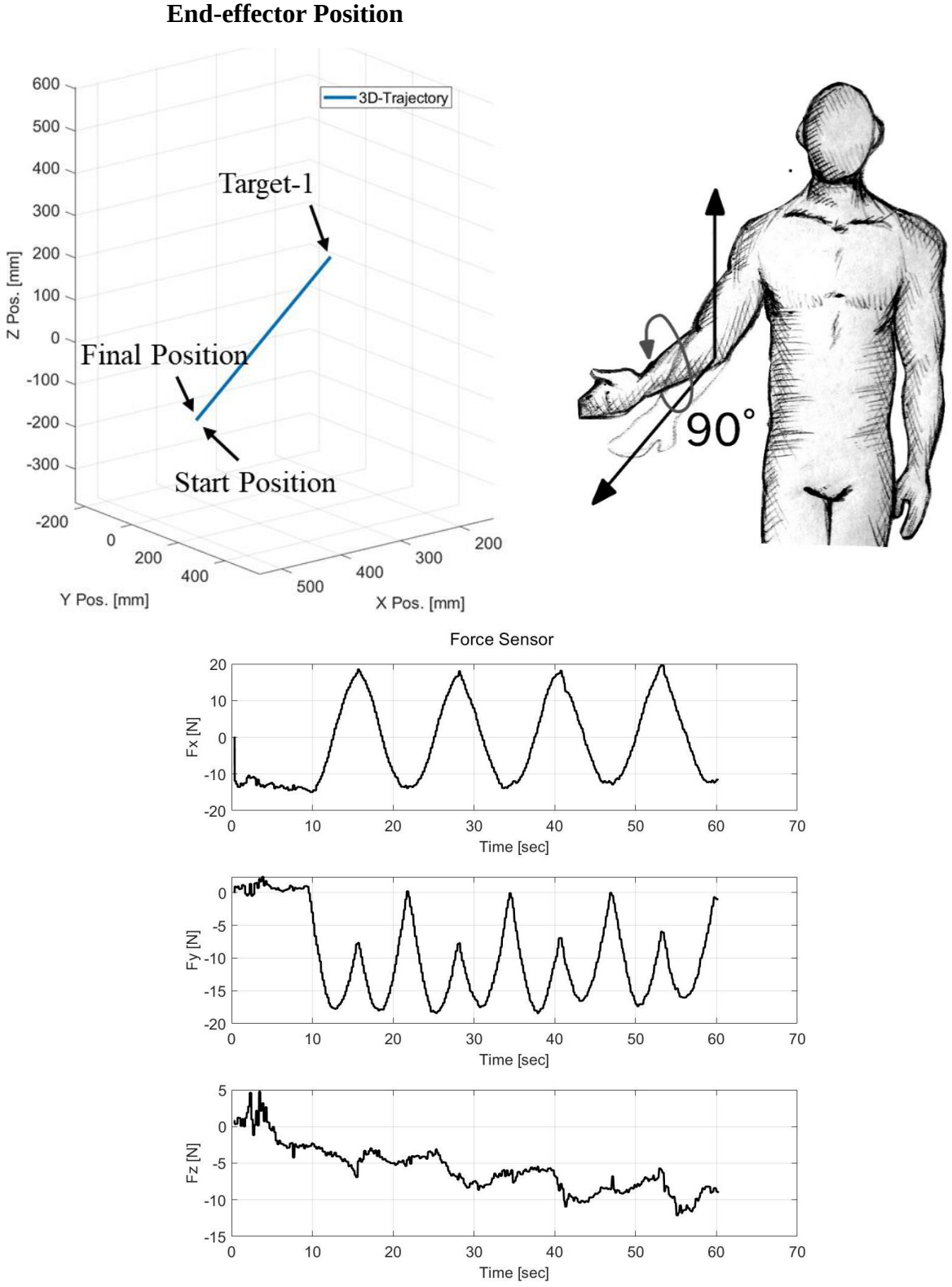


Figure 7.29 Human-robot interactive force detected from the force sensor at the end-effector during Forearm joint Supination – Pronation exercise.

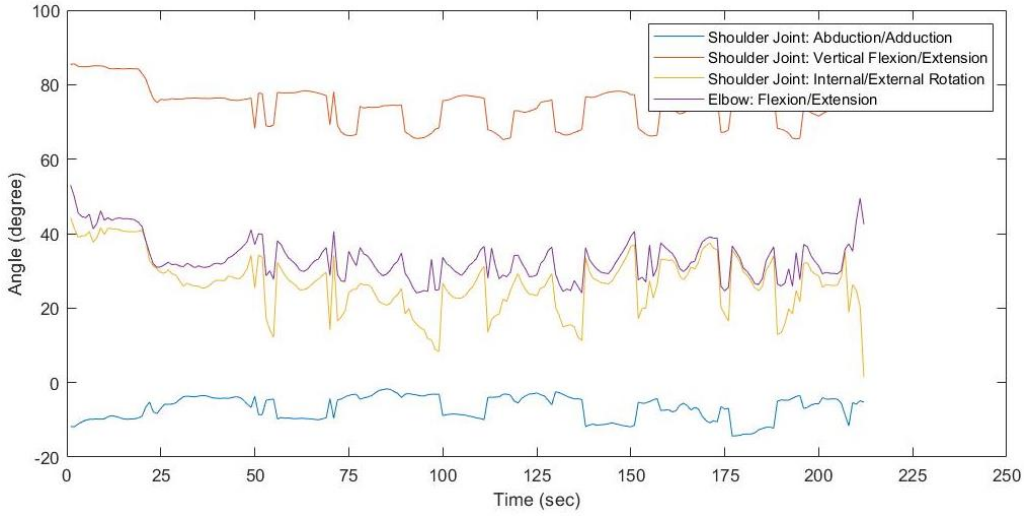


Figure 7.30 Upper-limb joint coordinate from Kinect sensor during forearm joint supination–pronation exercise.

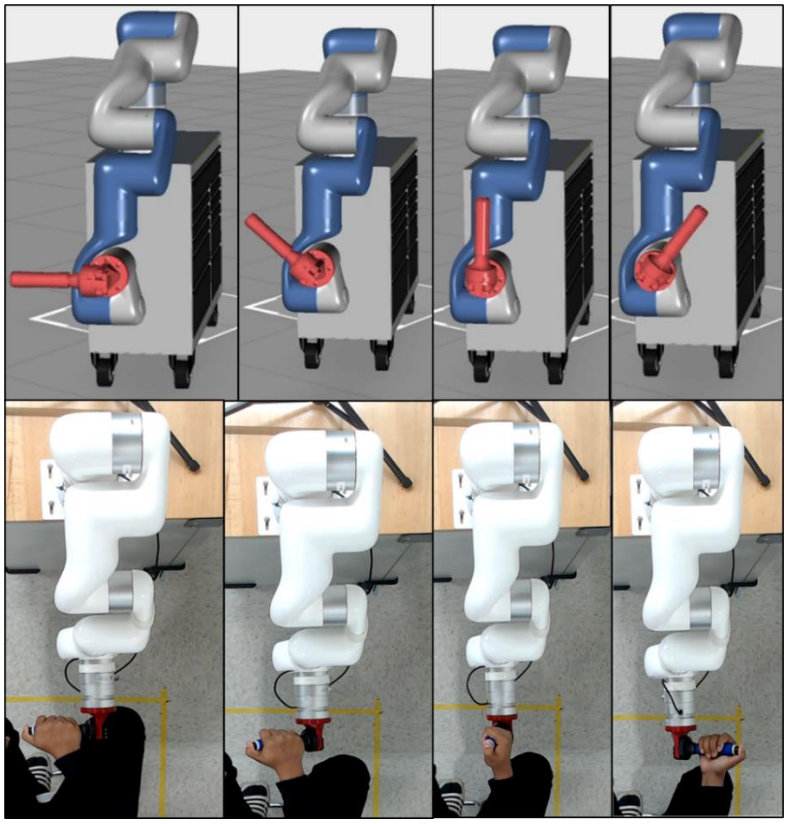


Figure 7.31 Monitoring forearm joint supination–pronation exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session

- **Horizontal Parabolic Shape Multi-Joint**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 108° (joint-2), -113° (joint-3), -84.6° (joint-4), and -90° (Joint-5). Figures 7.33 through 7.36 show the recorded experiment data for multi-joint horizontal parabolic shape exercise. As shown in Figure 7.33, during the horizontal parabolic shape exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to -56.3°, 48.9°, and finally returns to 0°;

  – Joint-2 stays to its initial position (108^o);

  – Joint-3 stays to its initial position (-113^o);

  – Joint-4 stays in its initial position (-84.6^o), and

  – Joint-5 moves from its initial position (-90^o) to -160° and returns to 0°. The experiment uses four repetitions.

  Figure 7.34 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint horizontal parabolic shape exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.35 and 7.36, respectively.

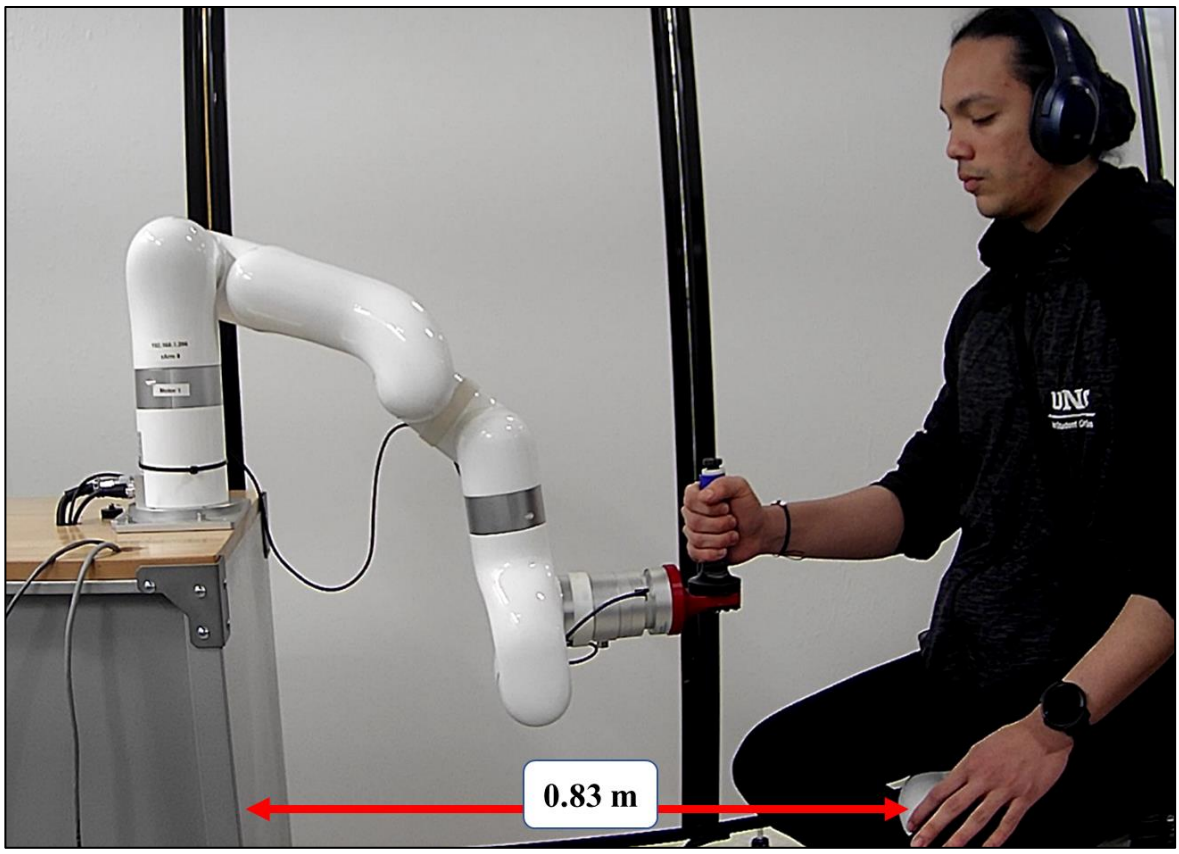


Figure 7.32 Participant sitting position during horizontal parabolic shape multi-joint exercise.

As shown in Fig. 7.32, a participant sat on the chair at a distance of 0.83 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the horizontal parabolic shape multi-joint exercise.

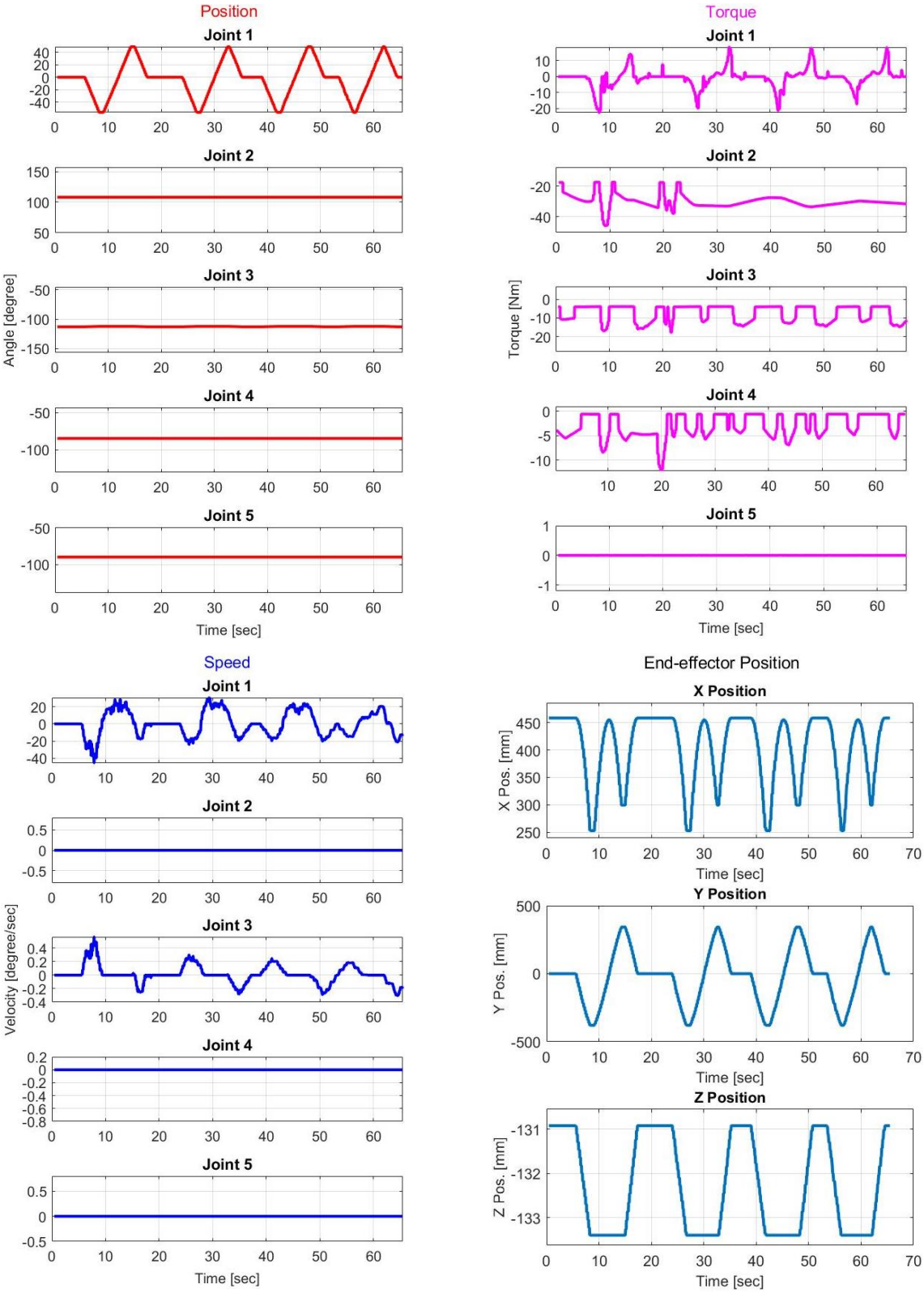


Figure 7.33 Joint angles, torques, speed, and end-effector position during horizontal parabolic shape multi-joint exercise.

End-effector Position

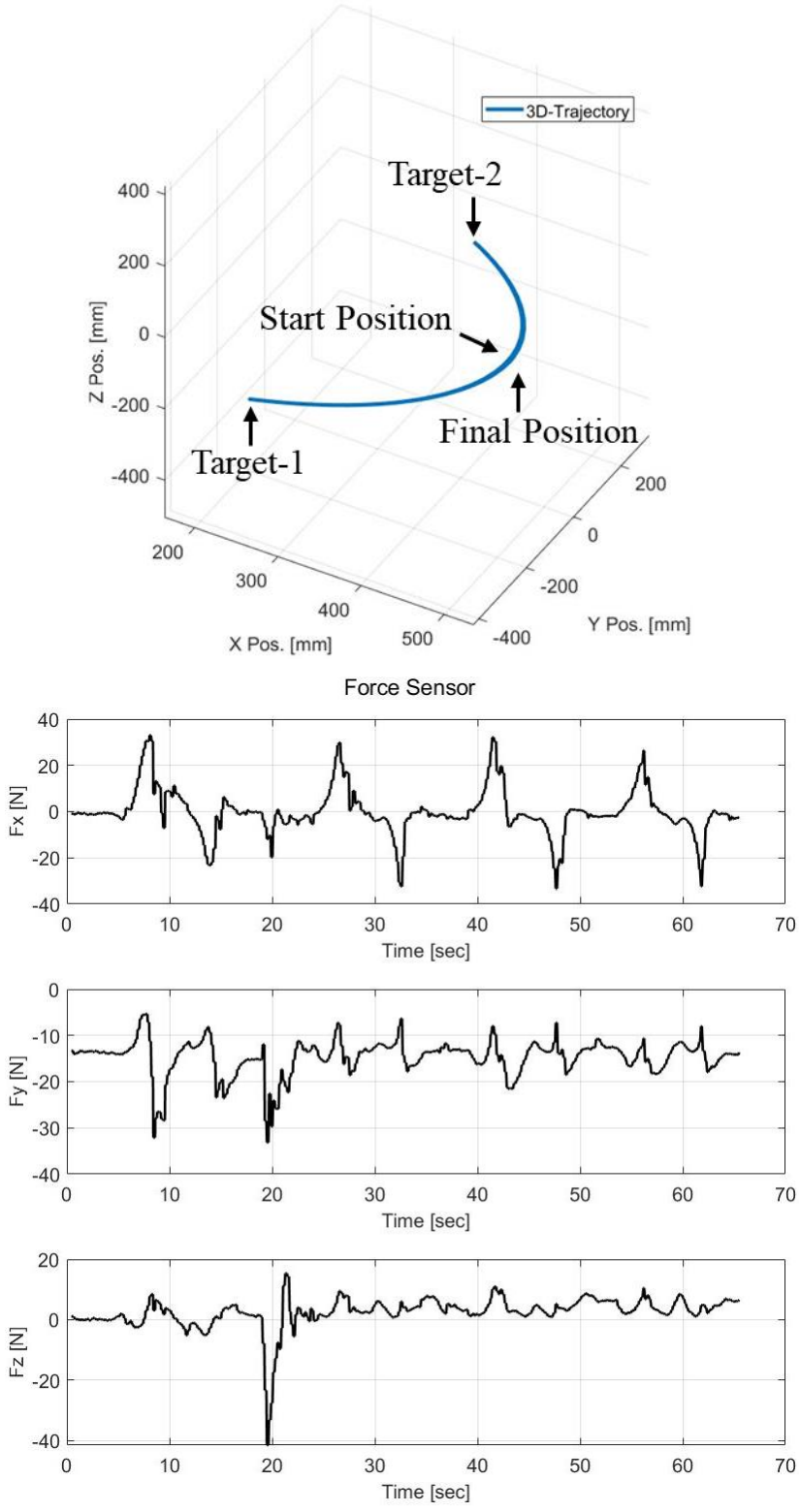


Figure 7.34 Human-robot interactive force detected from the force sensor at the end-effector during horizontal parabolic shape multi-joint exercise.

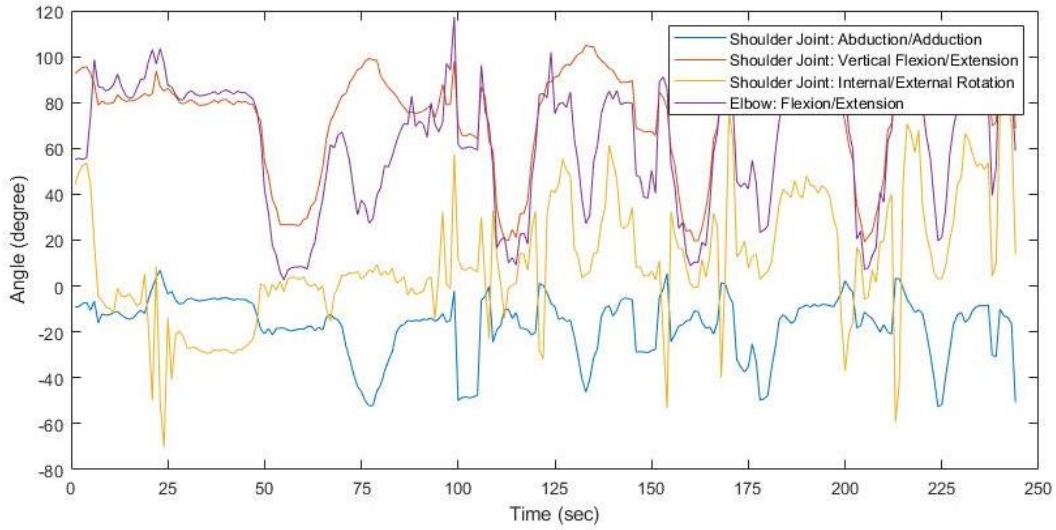


Figure 7.35 Upper-limb joint coordinate from Kinect sensor during horizontal parabolic shape multi-joint exercise.

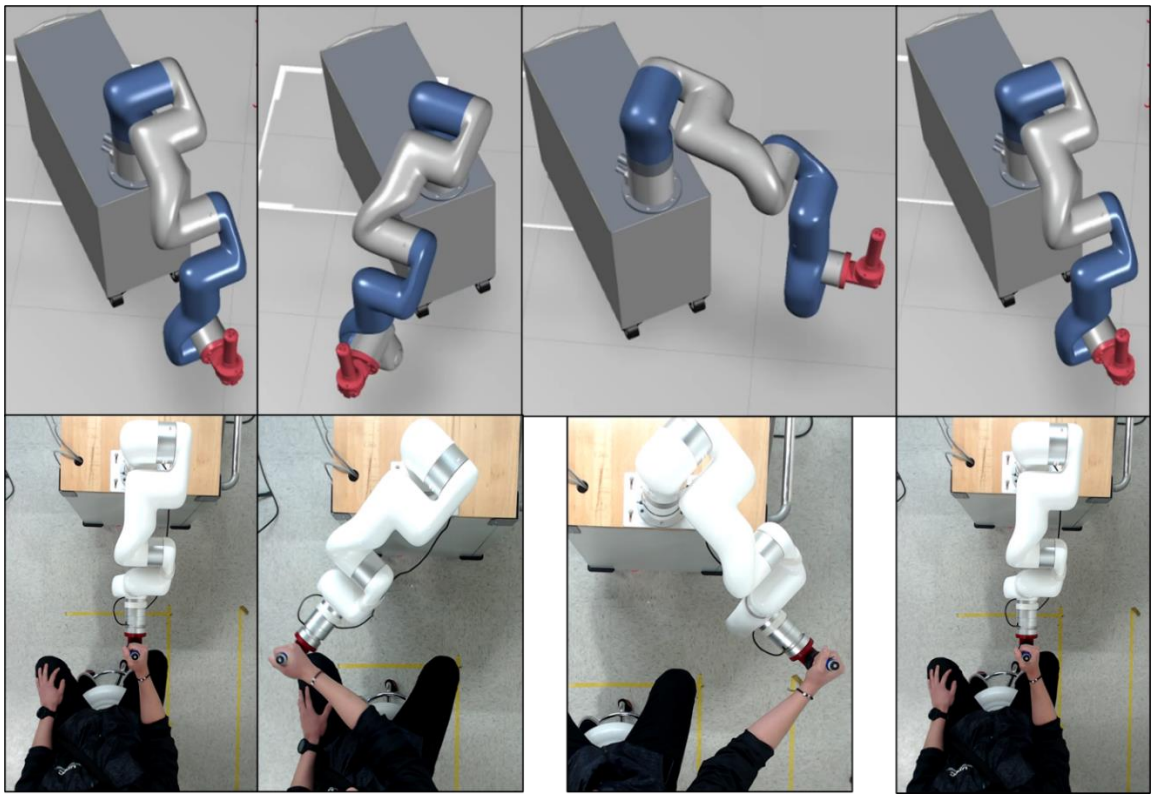


Figure 7.36 Monitoring horizontal parabolic shape multi-joint exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Vertical Square Shape Multi-Joint**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 5.8° (joint-2), -10.2° (joint-3), 4.4° (joint-4), and 0° (Joint-5). Figures 7.38 through 7.41 show the recorded experiment data for the multi-joint vertical square shape exercise. As shown in Figure 7.38, during the vertical square shape exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to -23.3°, -23.3°, 43°, 43°, and finally returns to 0°;

  – Joint-2 moves from its initial position (5.8^o) to 8.3°, -37°, -20°, 13.1°, and finally returns to 5.8^o;

  – Joint-3 moves from its initial position (-10.2^o) to -14.6°, -24°, -36.5°, -28.3°, and finally return to -10.2^o;

  – Joint-4 moves from its initial position (4.4^o) to 6.3°, 61.1°, 56.5°, 15.2°, and finally returns to 4.4°, and

  – Joint-5 moves from its initial position (0^o) to -23.3°, -23.3°, 43°, 43°, and finally returns to 0^o. The experiment uses two repetitions.

  Figure 7.39 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint vertical square shape exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.40 and 7.41, respectively.

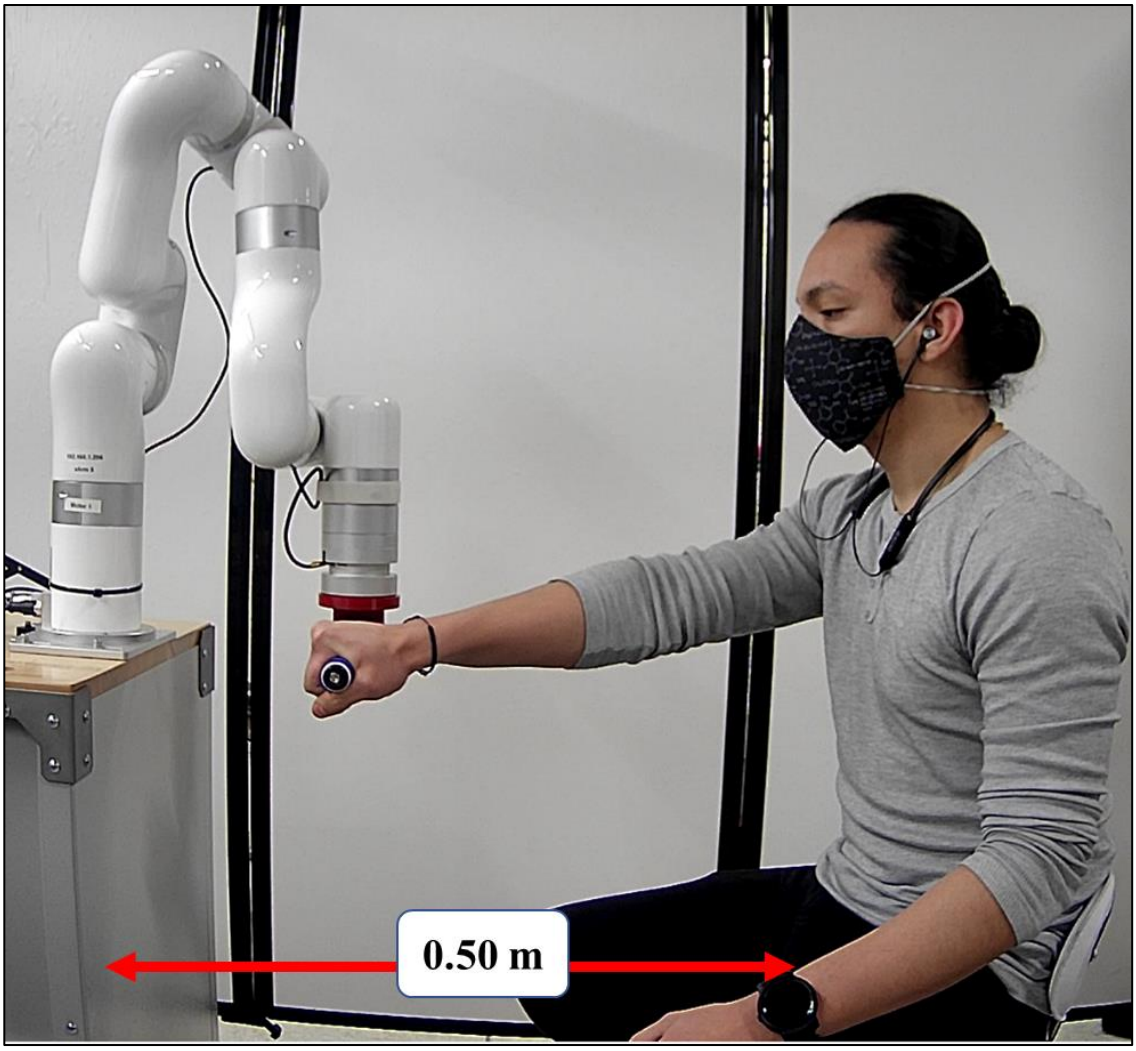


Figure 7.37 Participant sitting position during vertical square shape multi-joint exercise.

As shown in Fig. 7.37, a participant sat on the chair at a distance of 0.50 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the vertical square shape multi-joint exercise.

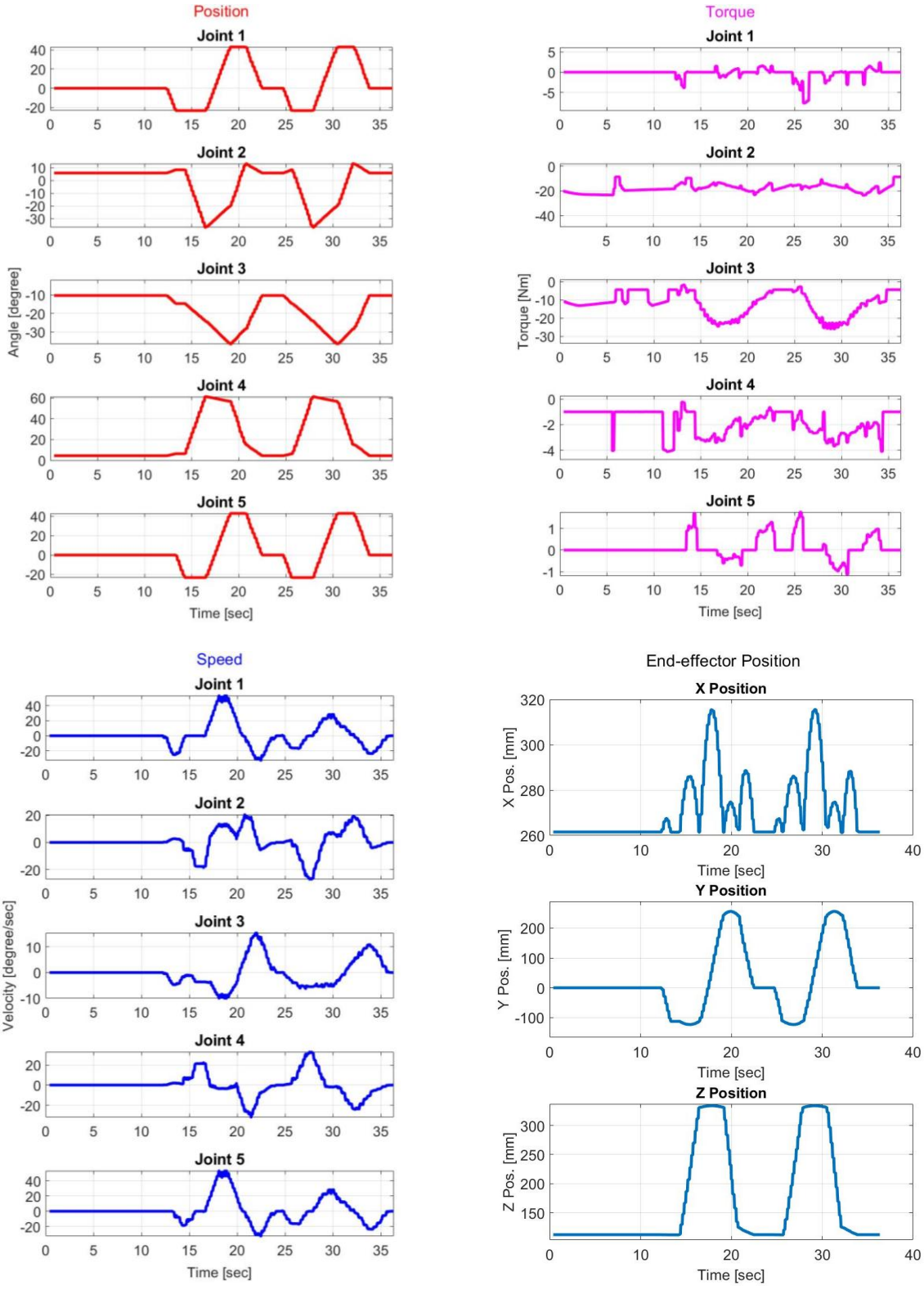


Figure 7.38 Joint angles, torques, speed, and end-effector position during vertical square shape multi-joint exercise.

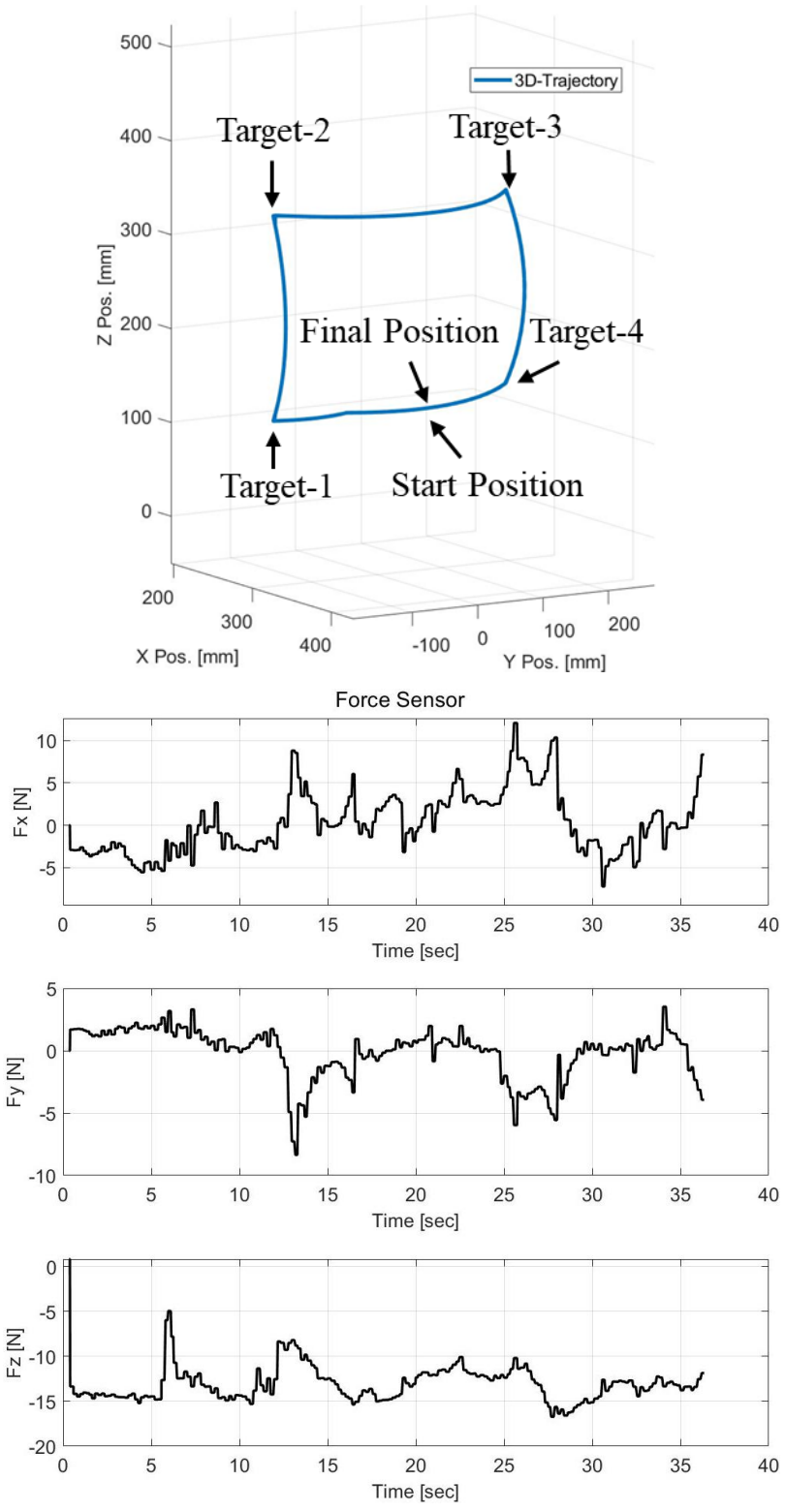


Figure 7.39 Human-robot interactive force detected from the force sensor at the end-effector during vertical square shape multi-joint exercise.

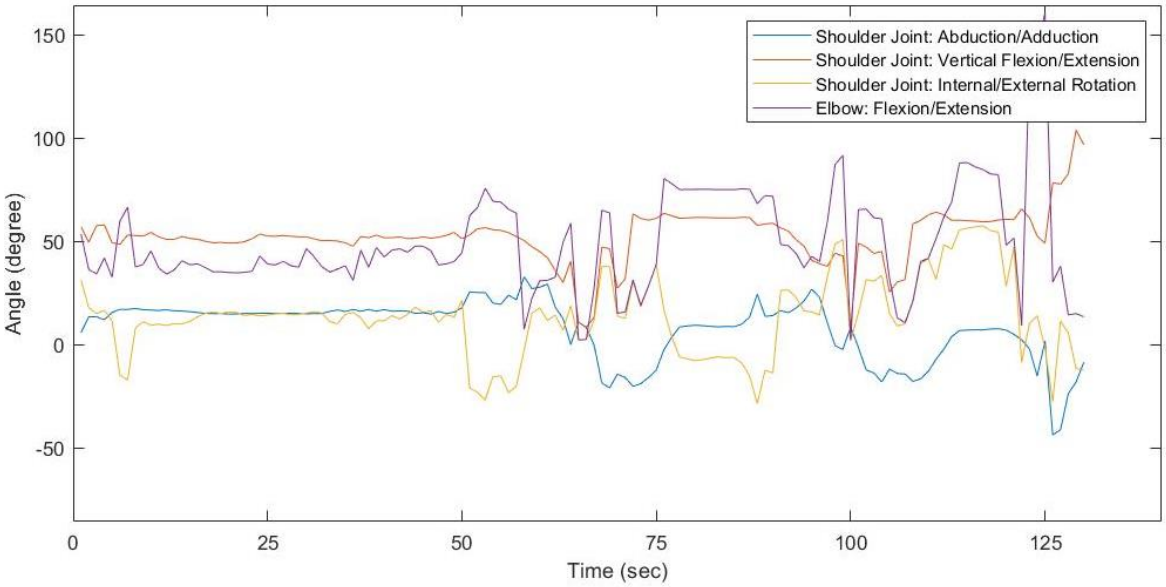


Figure 7.40 Upper-limb joint coordinate from Kinect sensor during vertical square shape multi-joint exercise.

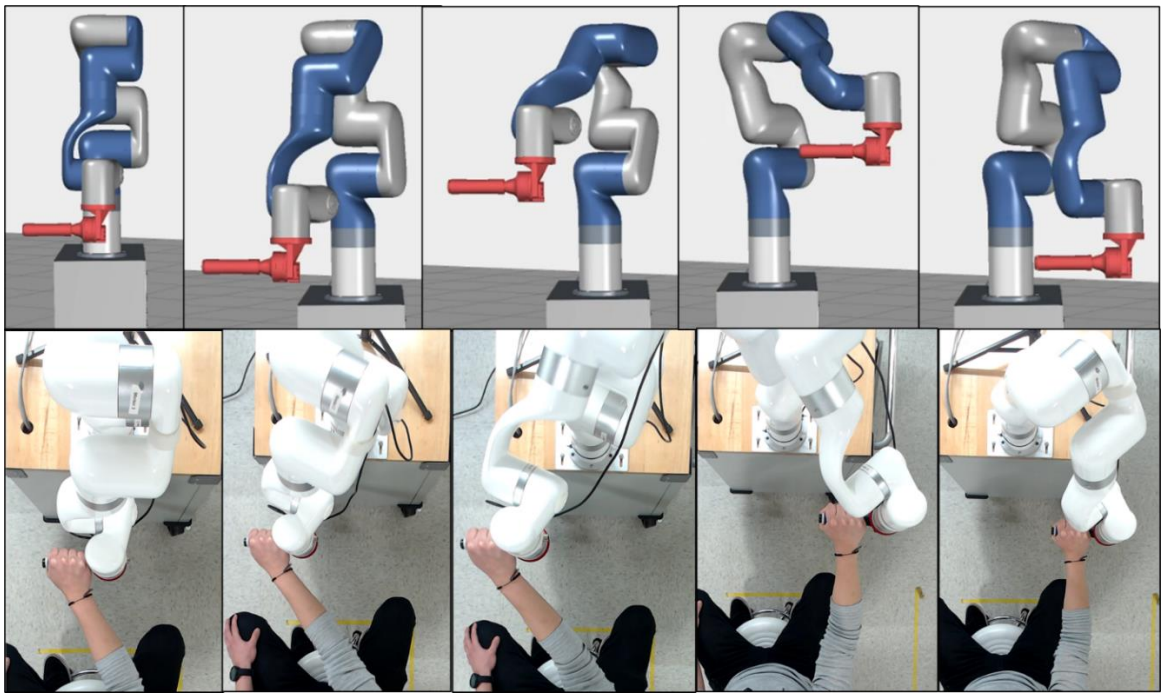


Figure 7.41 Monitoring vertical square shape multi-joint exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Horizontal Half-Star Shape Multi-Joint**

  To provide this exercise, the robot joints are initialized at angles: 0° (joint-1), 20.1° (joint-2), -36.2° (joint-3), 16.2° (joint-4), and 0° (Joint-5). Figures 7.43 through 7.46 show the recorded experiment data for the multi-joint horizontal half-star shape exercise. As shown in Figure 7.43, during the horizontal half-star shape exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (0^o) to 46.5°, 0°, 26.3°, 0°, -31.7°, 0°, -38.9°, and finally returns to 0°;

  – Joint-2 moves from its initial position (20.1^o) to 20°, 20.1°, -5.5°, 20.1°, 7.8°, 20.1°, 40.6°, and finally returns to 20.1^o;

  – Joint-3 moves from its initial position (-36.2^o) to -37°, -36.2°, 1.7°, -36.2°, -7.7°, -36.2°, -63.7°, and finally return to -36.2^o;

  – Joint-4 moves from its initial position (16.2^o) to 16.2°, 16.2°, 2.6°, 16.2°, 1.5°, 16.2°, 22.9°, and finally returns to 16.2°, and

  – Joint-5 moves from its initial position (0^o) to 68.1°, 0°, 20.8°, 0°, -51.7°, 0°, -93.7°, and finally returns to 0^o. The experiment uses one repetition.

  Figure 7.44 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint horizontal half-star shape exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.45 and 7.46, respectively.

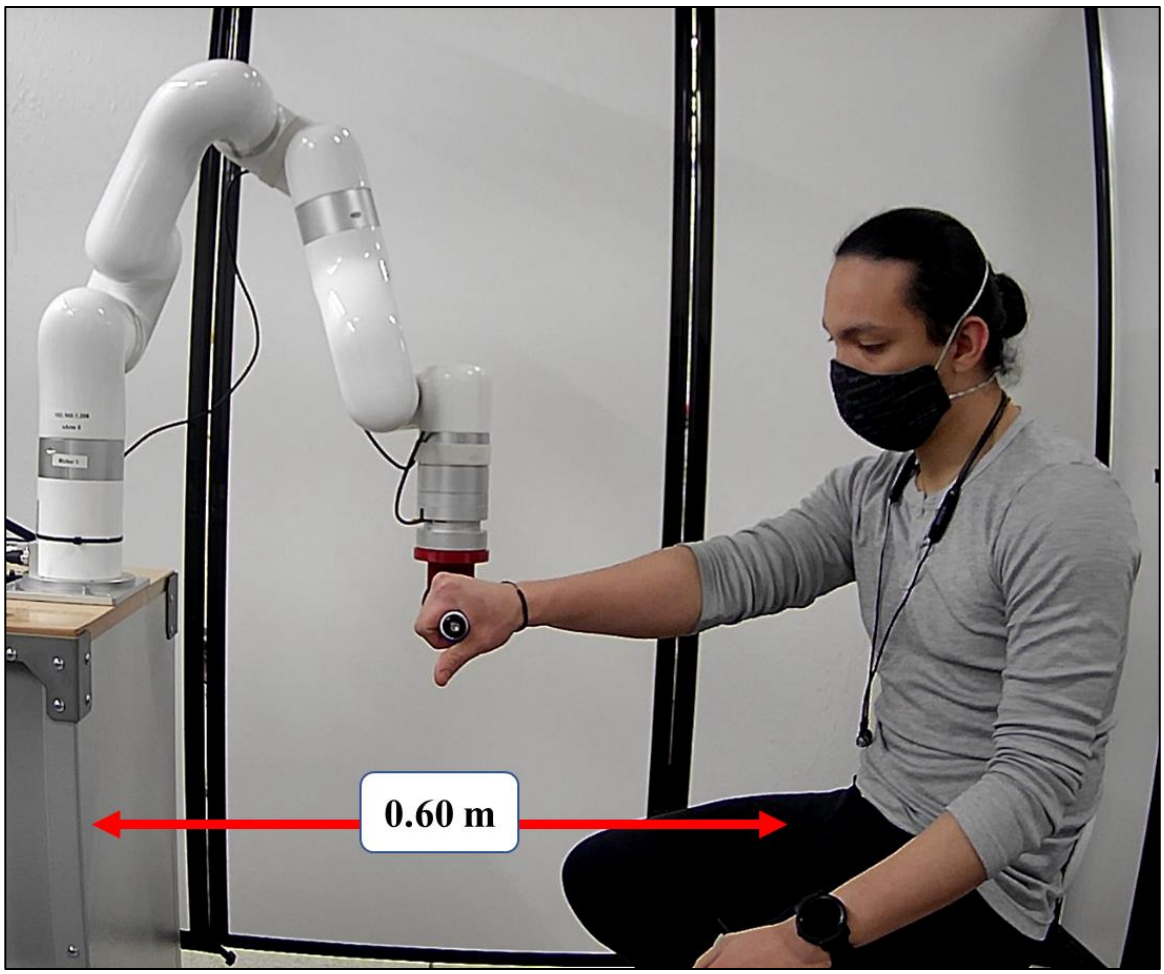


Figure 7.42 Participant sitting position during horizontal half-star shape multi-joint exercise.

As shown in Fig. 7.42, a participant sat on the chair at a distance of 0.60 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the horizontal half-star shape multi-joint exercise.

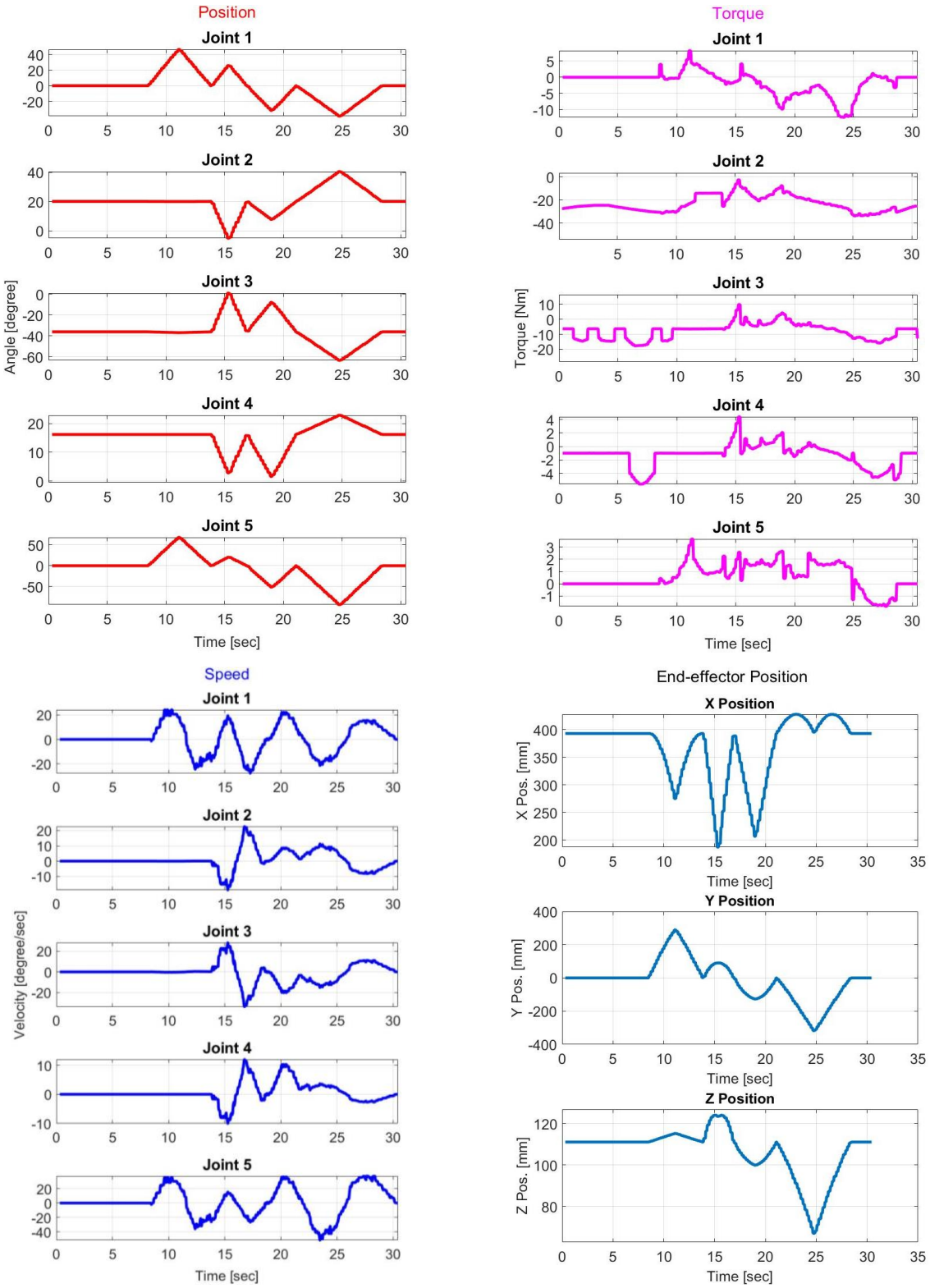


Figure 7.43 Joint angles, torques, speed, and end-effector position during horizontal half-star shape multi-joint exercise.

End-effector Position

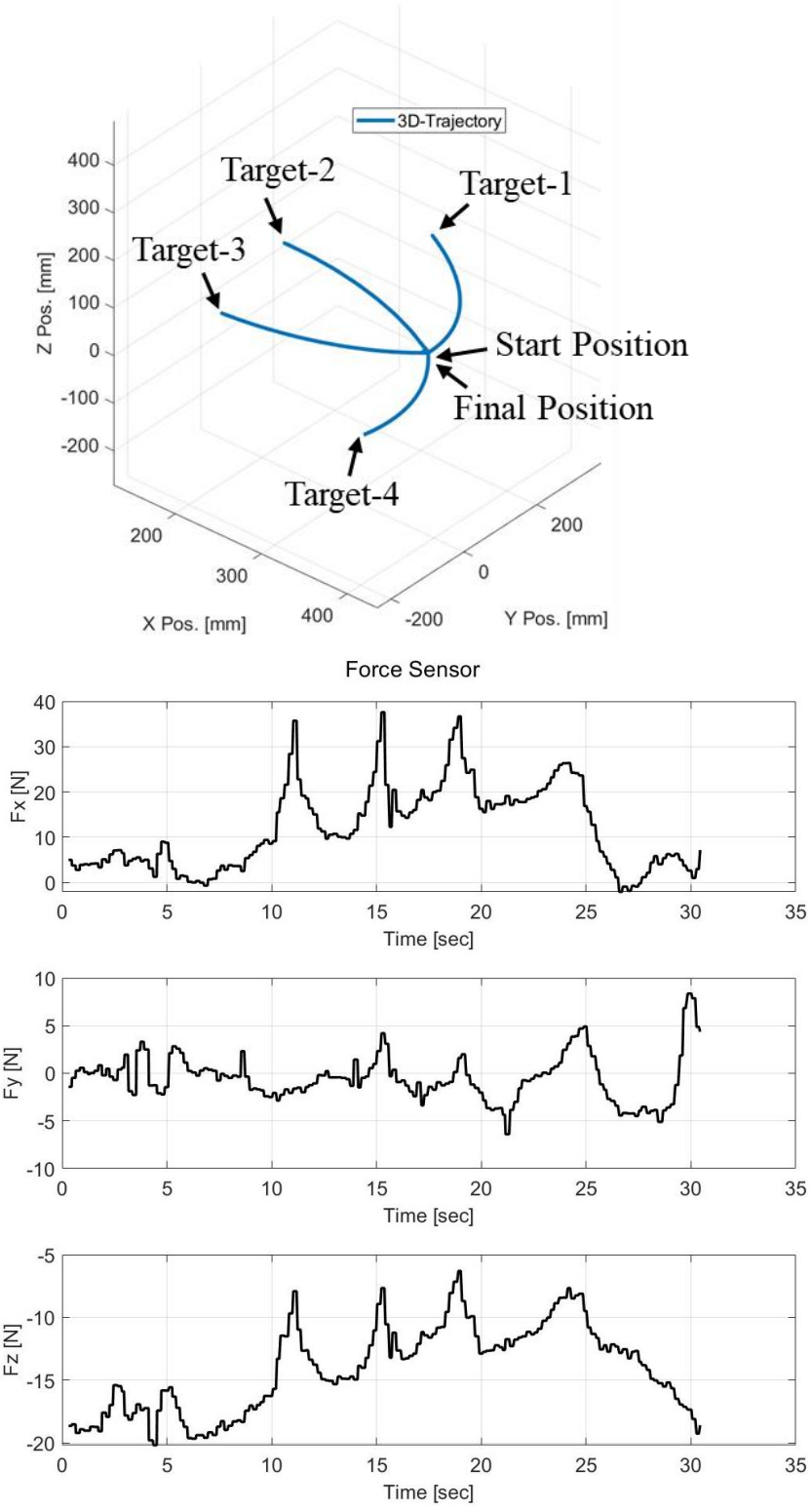


Figure 7.44 Human-robot interactive force detected from the force sensor at the end-effector during horizontal half-star shape multi-joint exercise.

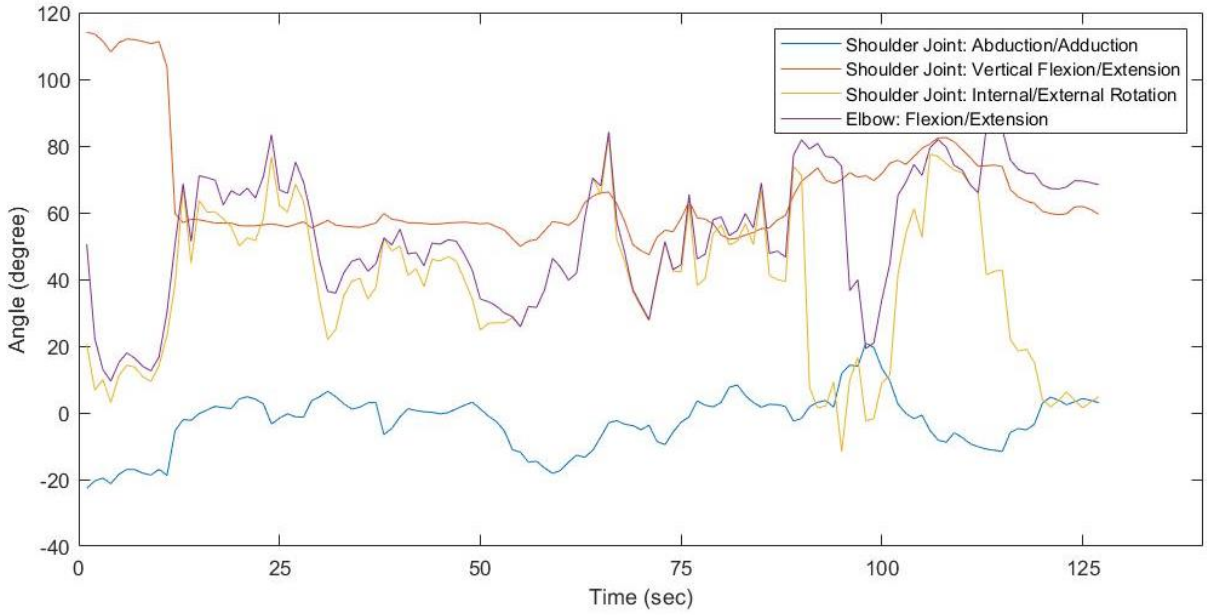


Figure 7.45 Upper-limb joint coordinate from Kinect sensor during horizontal half-star shape multi-joint exercise.

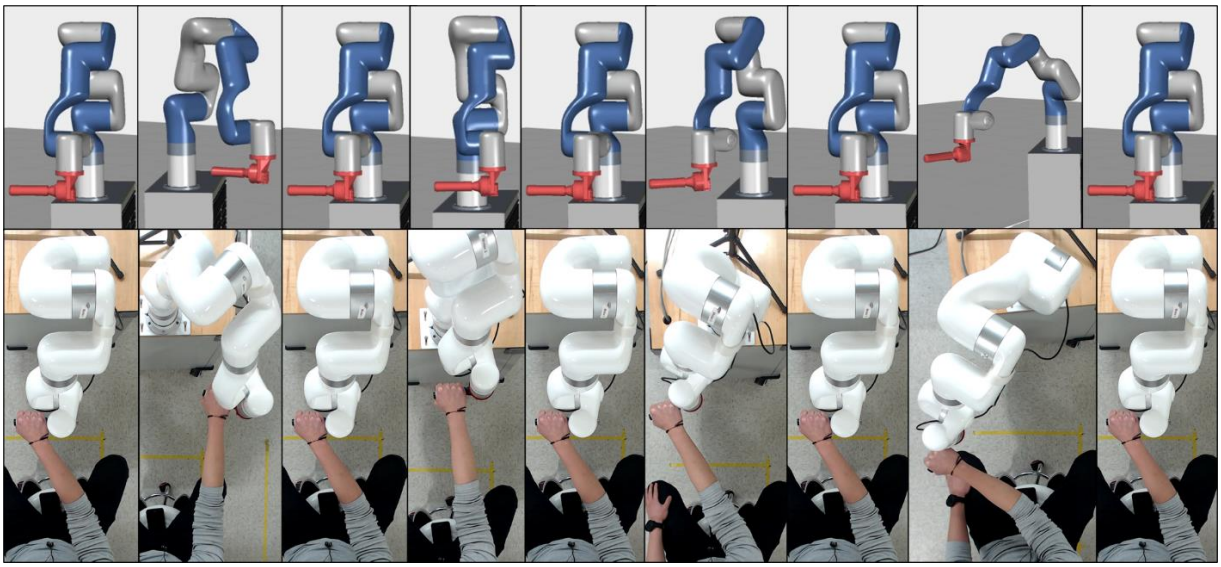


Figure 7.46 Monitoring horizontal half-star shape multi-joint exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

- **Diagonal Stretch Shape Multi-Joint**

  To provide this exercise, the robot joints are initialized at angles: -13.1° (joint-1), 26.3° (joint-2), -49° (joint-3), 22.2° (joint-4), and -12° (Joint-5). Figures 7.48 through 7.51 show the recorded experiment data for the multi-joint diagonal stretch shape exercise. As shown in Figure 7.48, during the diagonal stretch shape exercise, the robot's joints are moved as follows:

  – Joint-1 moves from its initial position (-13.1^{o}) to -29.5°, 0°, 24.8°, and finally returns to -13.1°;

  – Joint-2 moves from its initial position (26.3^{o}) to -9°, 25.4°, 41.5°, and finally returns to 26.3^{o};

  – Joint-3 moves from its initial position (-49^{o}) to -22.8°, -46.4°, -76.6°, and finally returns to -49^{o};

  – Joint-4 moves from its initial position (22.2^{o}) to 31.8°, 21°, 35.1°, and finally returns to 22.2°, and

  – Joint-5 moves from its initial position (-12^{o}) to -29.5°, 0°, 24.8°, and finally returns to -12^{o}. The experiment uses four repetitions.

  Figure 7.49 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint diagonal stretch shape exercise. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.50 and 7.51, respectively.

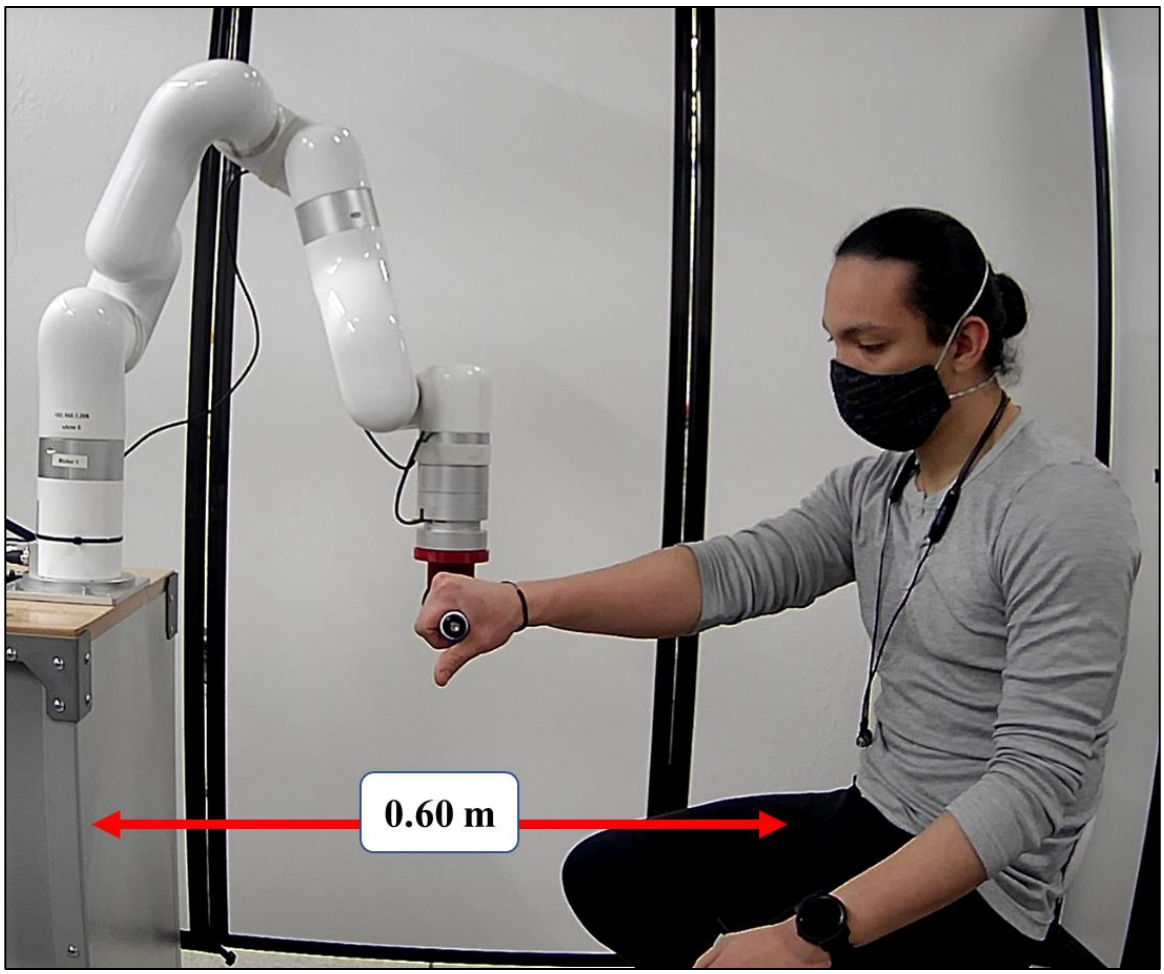


Figure 7.47 Participant sitting position during diagonal stretch shape multi-joint exercise.

As shown in Fig. 7.47, a participant sat on the chair at a distance of 0.60 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the diagonal stretch shape multi-joint exercise.

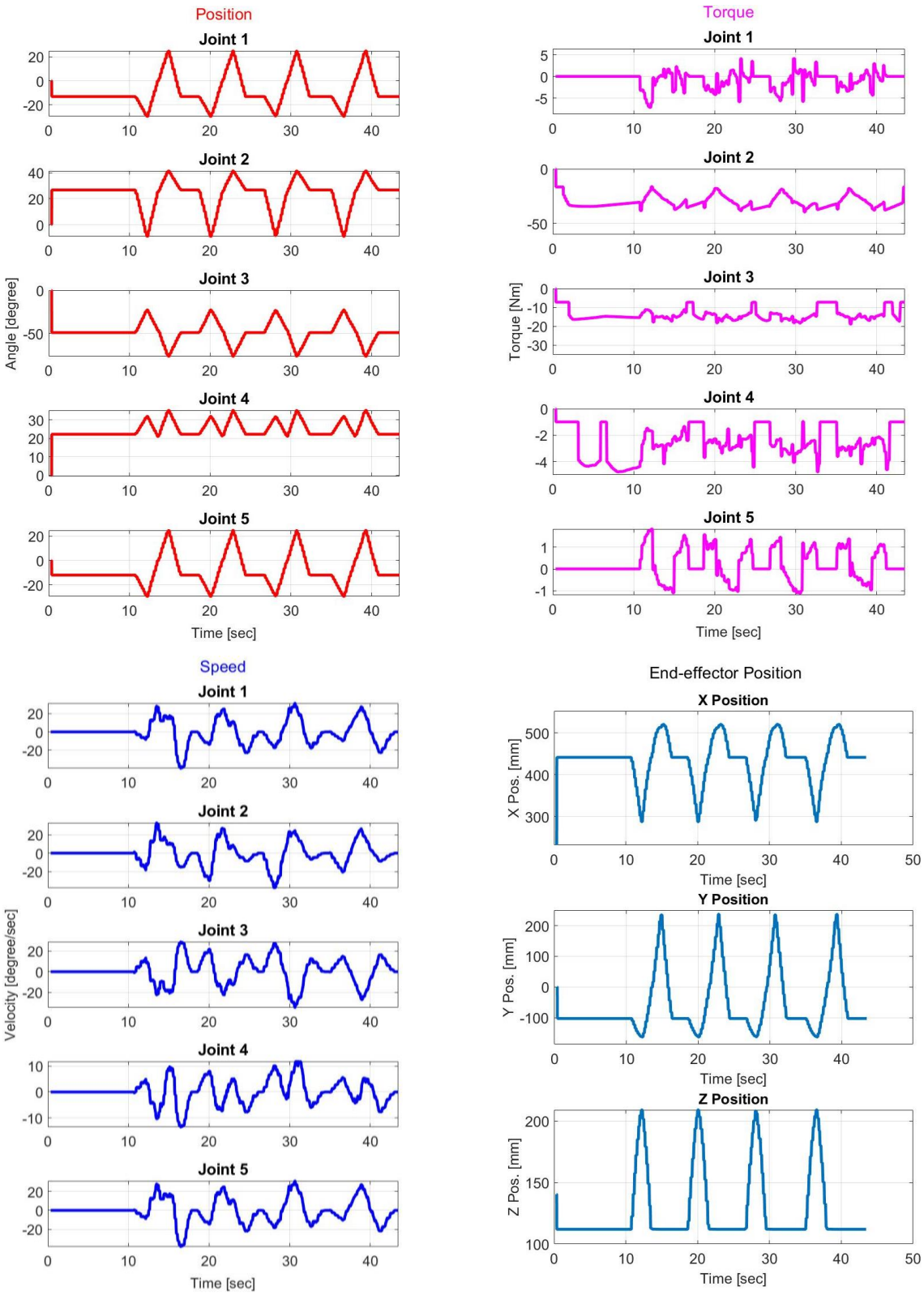


Figure 7.48 Joint angles, torques, speed, and end-effector position during diagonal stretch shape multi-joint exercise.

End-effector Position

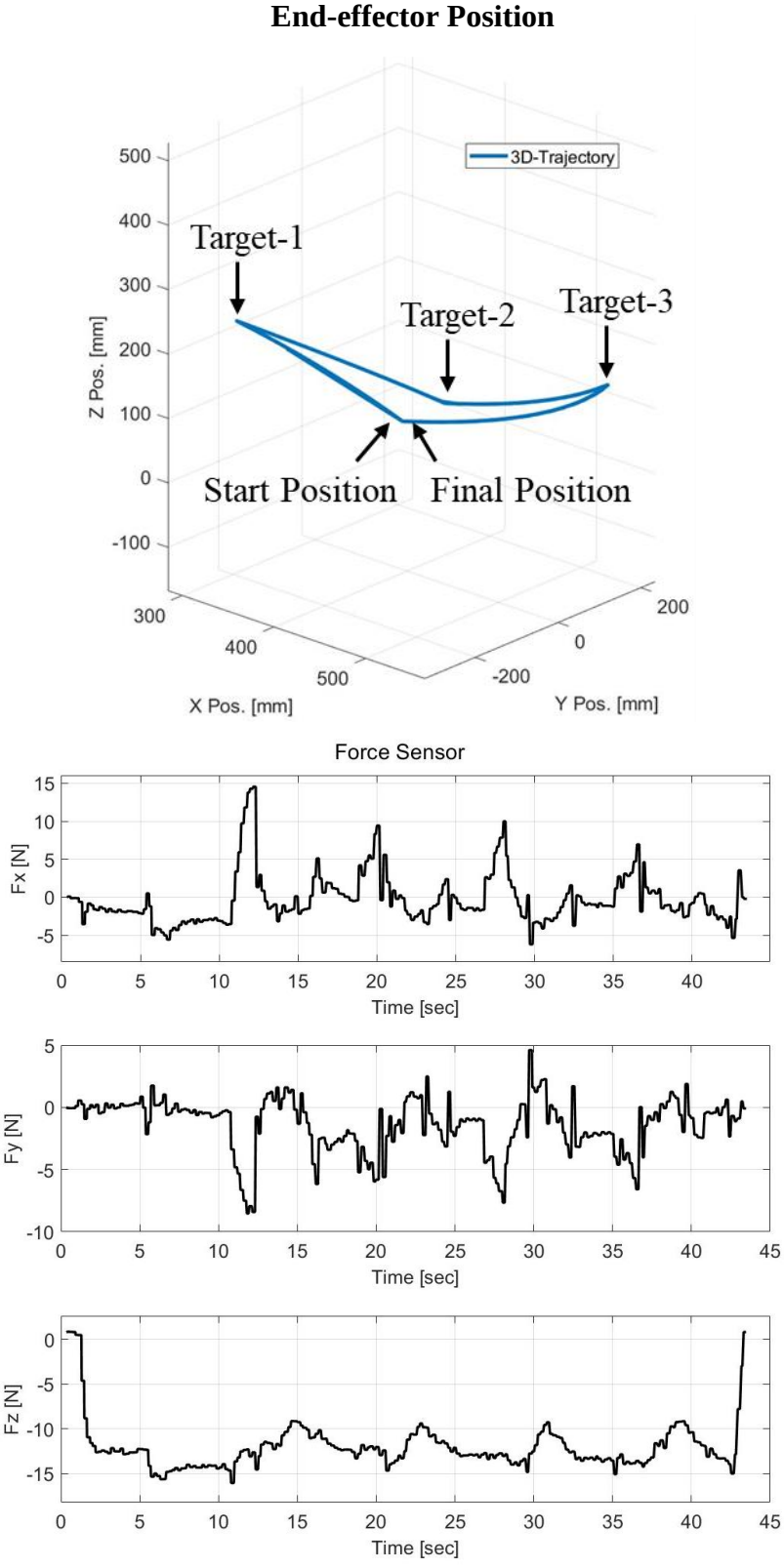


Figure 7.49 Human-robot interactive force detected from the force sensor at the end-effector during diagonal stretch shape multi-joint exercise.

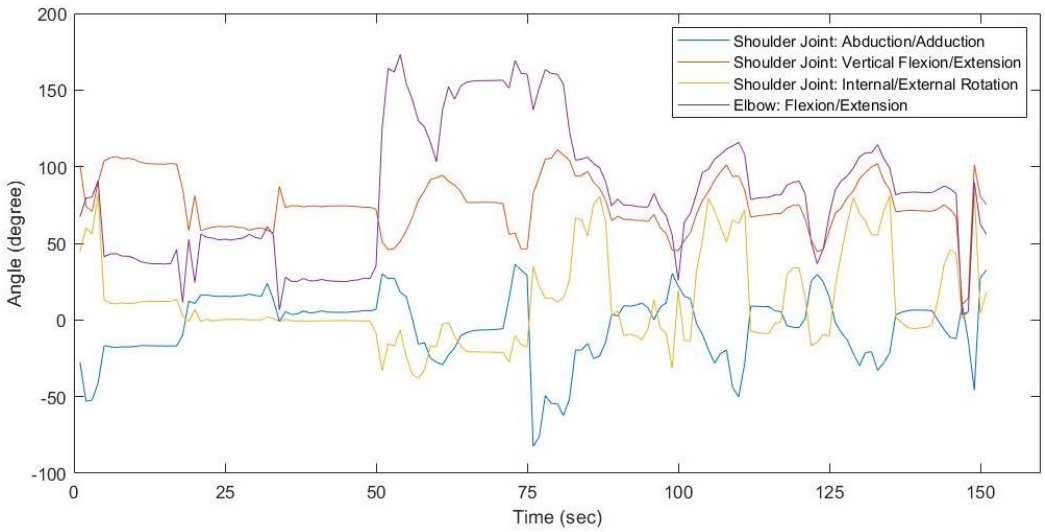


Figure 7.50 Upper-limb joint coordinate from Kinect sensor during diagonal stretch shape multi-joint exercise.

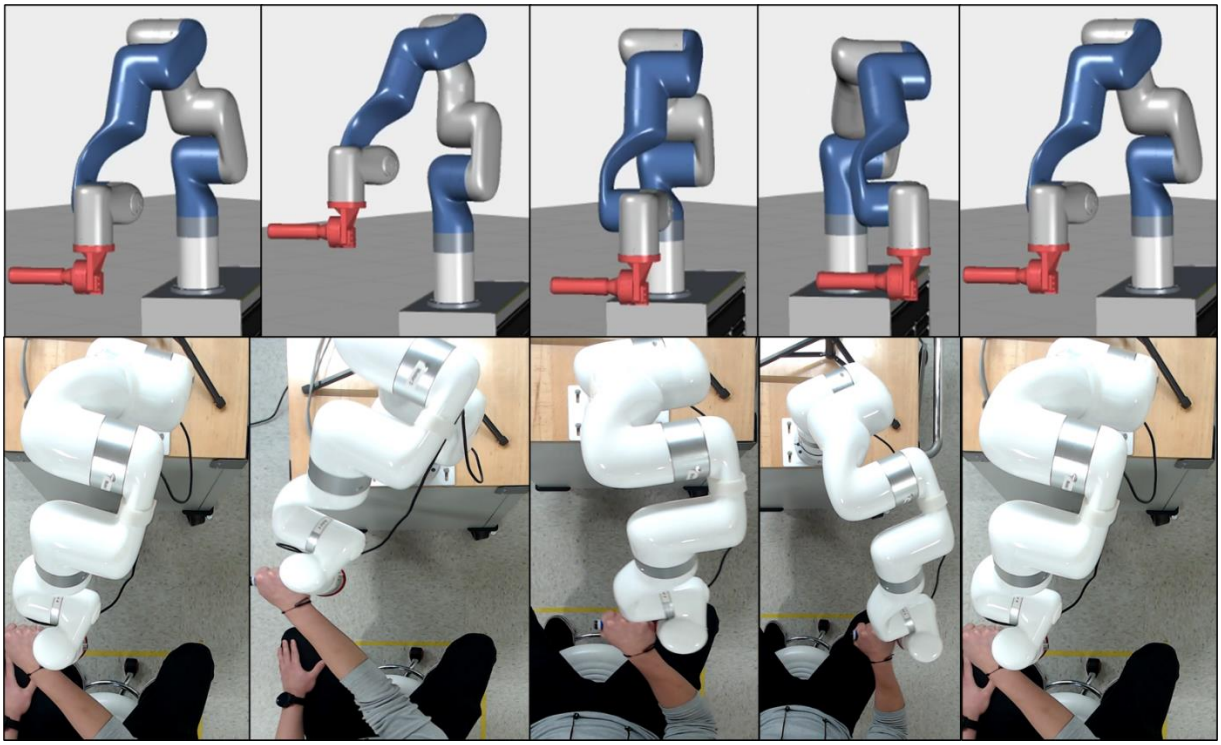


Figure 7.51 Monitoring diagonal stretch shape multi-joint exercise using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

Note: Anatomy skeletons in Fig. 7.9, Fig. 7.14, Fig. 7.24, and Fig. 7.29 are used from [13], and Fig. 7.19 is used from [91].

7.2.2 Passive Rehab Exercise (PRE) Mode with Real-Time Recorded Trajectory (PREwRT^2)

Figure 7.52 shows the user interface of Vuforia View for the PREwRT^2 telerehabilitation exercises. The exercise is given to the subject in a seated position holding the robot's end-effector (handle).

To provide this exercise, as a first step, the operator uses the joystick to remotely control the robot's end-effector position at a certain level based on the subject's upper-limb range of motion and observe the changes in the robot's (i.e., participant's) end-effector position on the AR digital twin. The operator then enters the robot/participant's end-effector position values of X, Y, and Z into the telerehabilitation exercise user interface in PREwRT^2. Note that the limiting values of X, Y, and Z positions are determined by the human-robot interactive force (HRIF). High HRIF (detected from the force sensor) indicates that the subject is resisting the motion. A threshold value of HRIF needs to be defined in future research work, which can be estimated based on the patient's pain-free range of motion. As shown in Fig. 7.43, once the robot's end-effector positions (X, Y, Z) for a particular exercise are entered into the system, the exercise can be repeated by entering the repetition number between 1-10 in the 'No. of Cycle Repetition' cell. The exercises will then be monitored on AR digital twin.

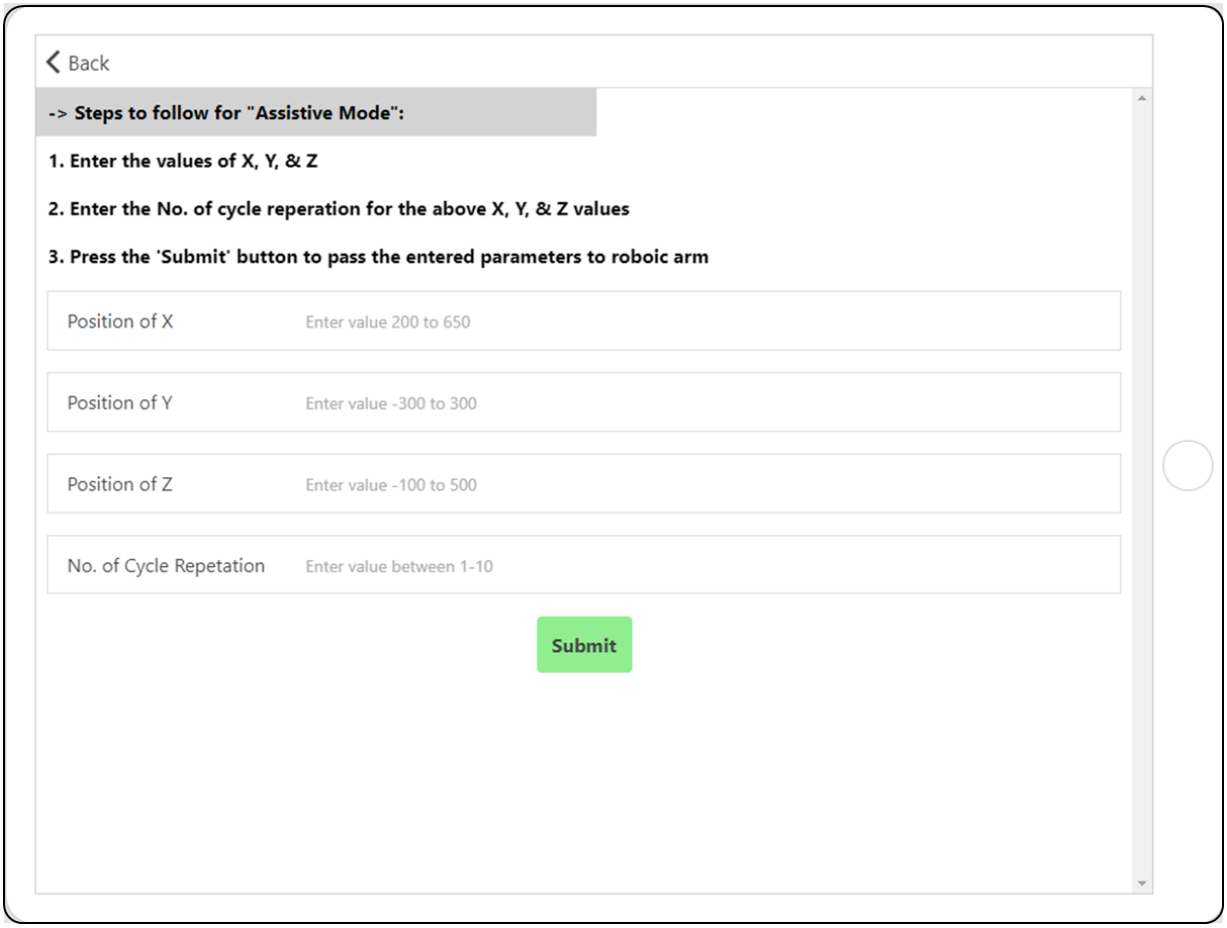


Figure 7.52 Augmented reality user interface for PREwRT^2.

We did experiments with PRMwRT^2 with two participants. Figures 7.54 through 7.57 and Fig. 7.58 through 7.61 show participant-1's and participant-2's experiments data.

For participant-1, as shown in Fig. 7.54, the operator started moving the robot's end-effector from its initial position X = 207 mm, Y = 0 mm, and Z = 112 mm using the remote joystick. From Fig. 7.54(a), it can be seen that the trajectory of the robot's end-effector reached the final position (participant's pain-free range of area) at X = 557 mm, Y = 268 mm, and Z = 280 mm. Next, the operator entered the values of the robot's end-effector from the above final position to the Vuforia View user interface and repeated the final position three times, as presented in Fig. 7.54(b). Also, it can be seen from the force plots in Fig. 7.54 that the participant exerted a force during the

PREwRT^2. Figure 7.55 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.56 illustrates the participant-1's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.54. Figure 7.57 shows the AR digital twin of the xArm-5 robot and participant-1's upper-limb movement through the Microsoft Teams video session during that PREwRT^2.

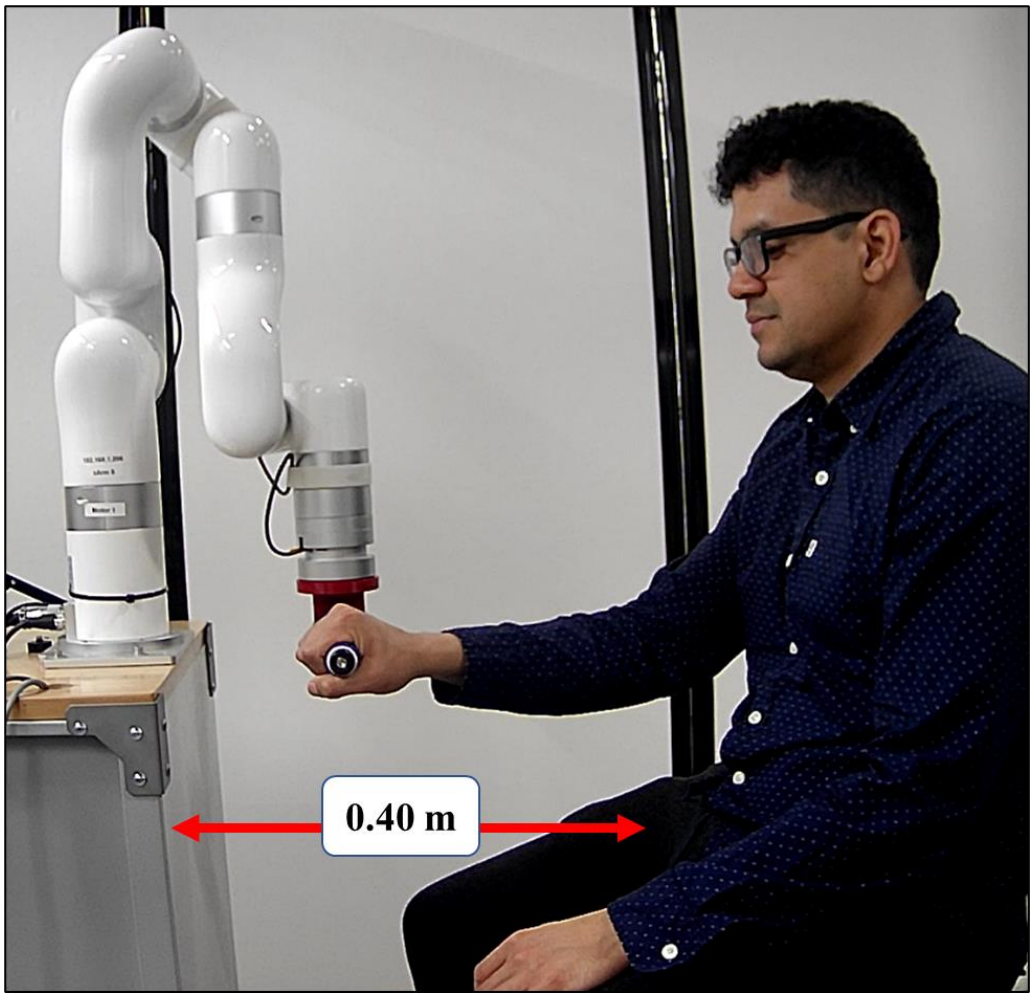


Figure 7.53 Participants sitting position during PREwRT^2.

As shown in Fig. 7.53, participants sat on the chair at a distance of 0.40 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during PREwRT^2.

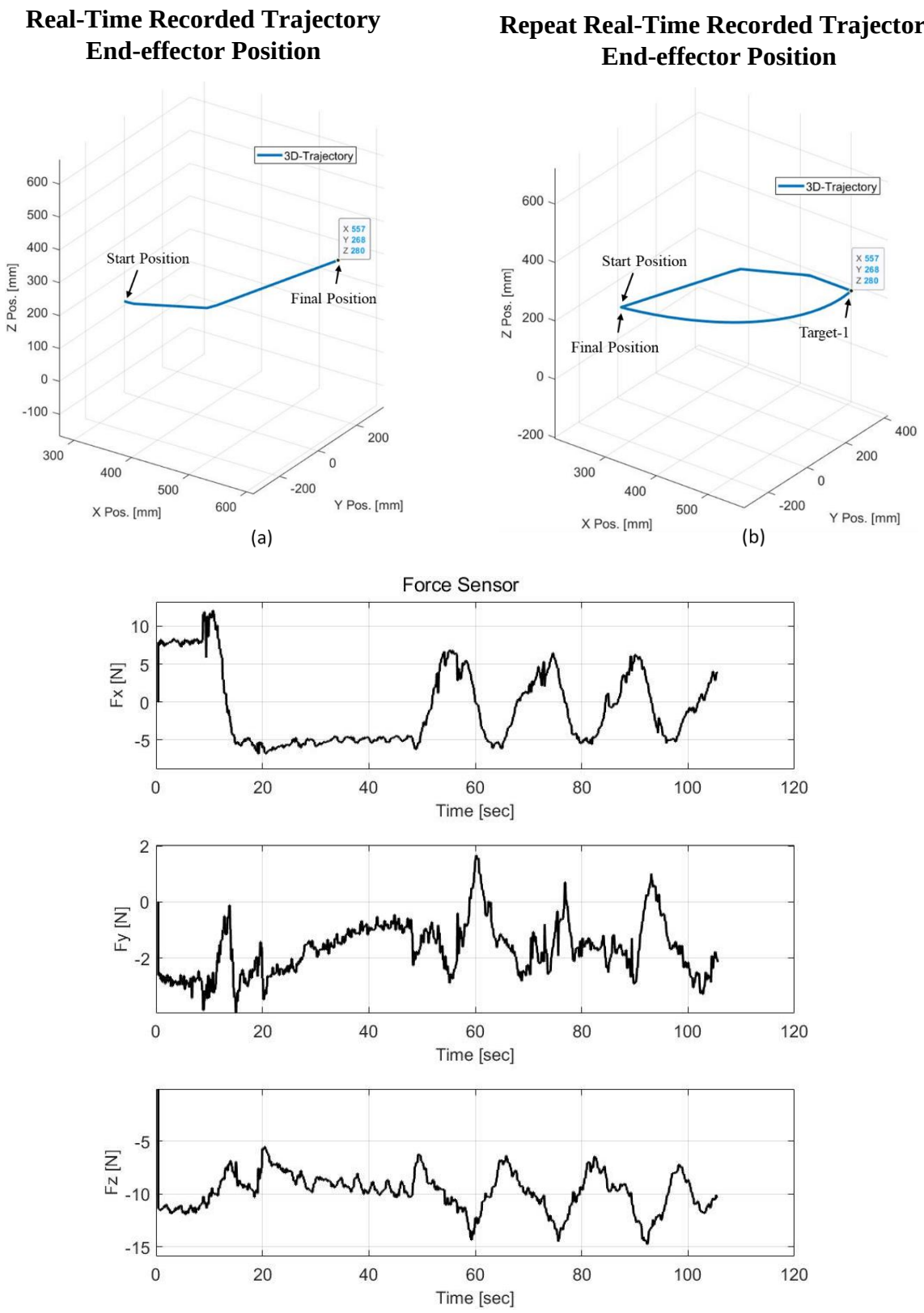


Figure 7.54 Human-robot interactive force detected from the force sensor at the end-effector during PREwRT[2].

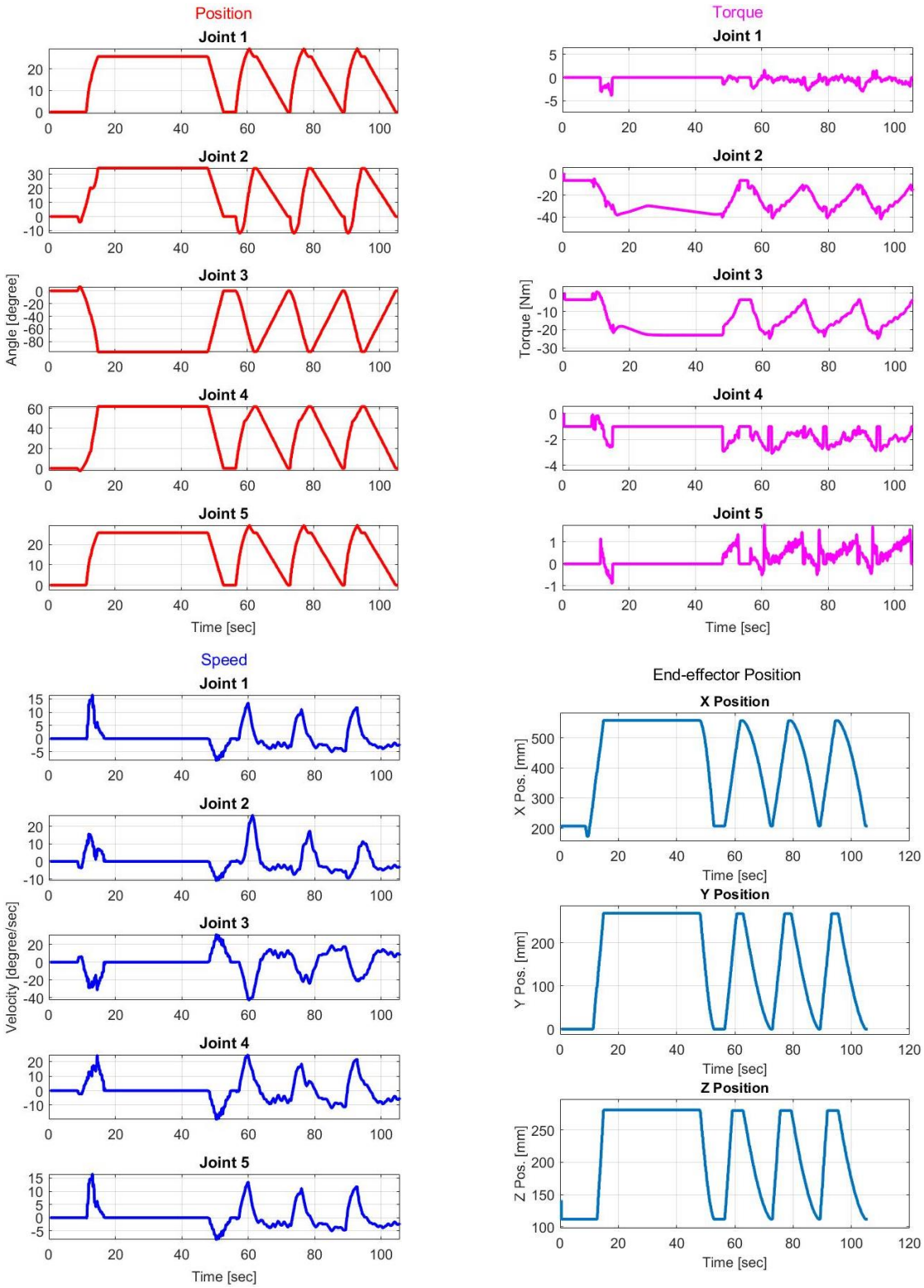


Figure 7.55 Joint angles, torques, speed, and end-effector position during PREwRT[2].

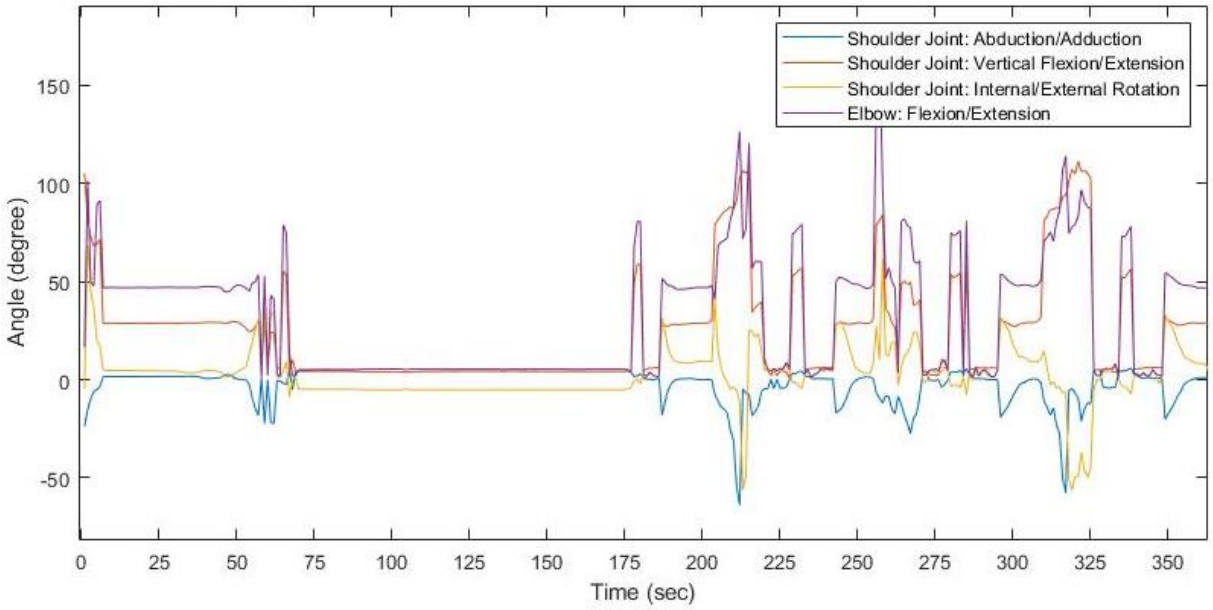


Figure 7.56 Upper-limb joint coordinate from Kinect sensor during PREwRT[2].

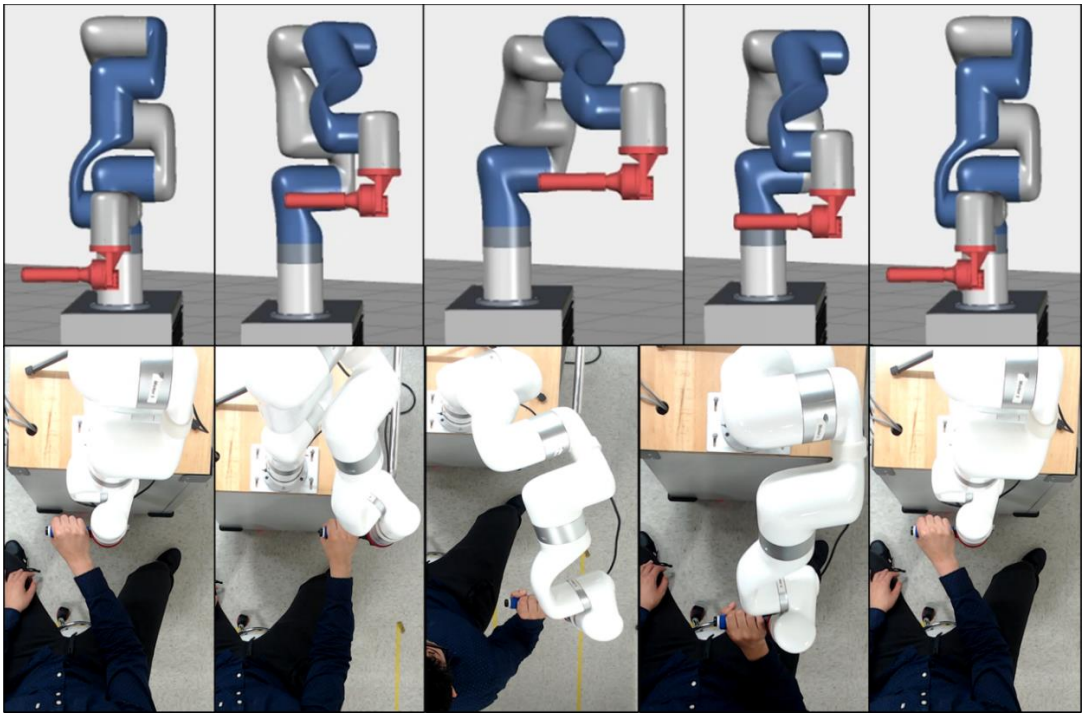


Figure 7.57 Monitoring PREwRT[2] using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

For participant-2, as shown in Fig. 7.58, the operator started moving the robot's end-effector from its initial position X = 207 mm, Y = 0 mm, and Z = 112 mm using the remote joystick. From Fig. 7.58(a), it can be seen that the trajectory of the robot's end-effector reached the final position (participant's pain-free range of area) at X = 349 mm, Y = -224 mm, and Z = 490 mm. Next, the operator entered the values of the robot's end-effector from the above final position to the Vuforia View user interface and repeated the final position five times, as presented in Fig. 7.58(b). Also, it can be seen from the force plots in Fig. 7.58 that the participant exerted a force during the PREwRT^2. Figure 7.59 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.60 illustrates the participant-2's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.58. Figure 7.61 shows the AR digital twin of the xArm-5 robot and participant-2's upper-limb movement through the Microsoft Teams video session during that PREwRT^2.

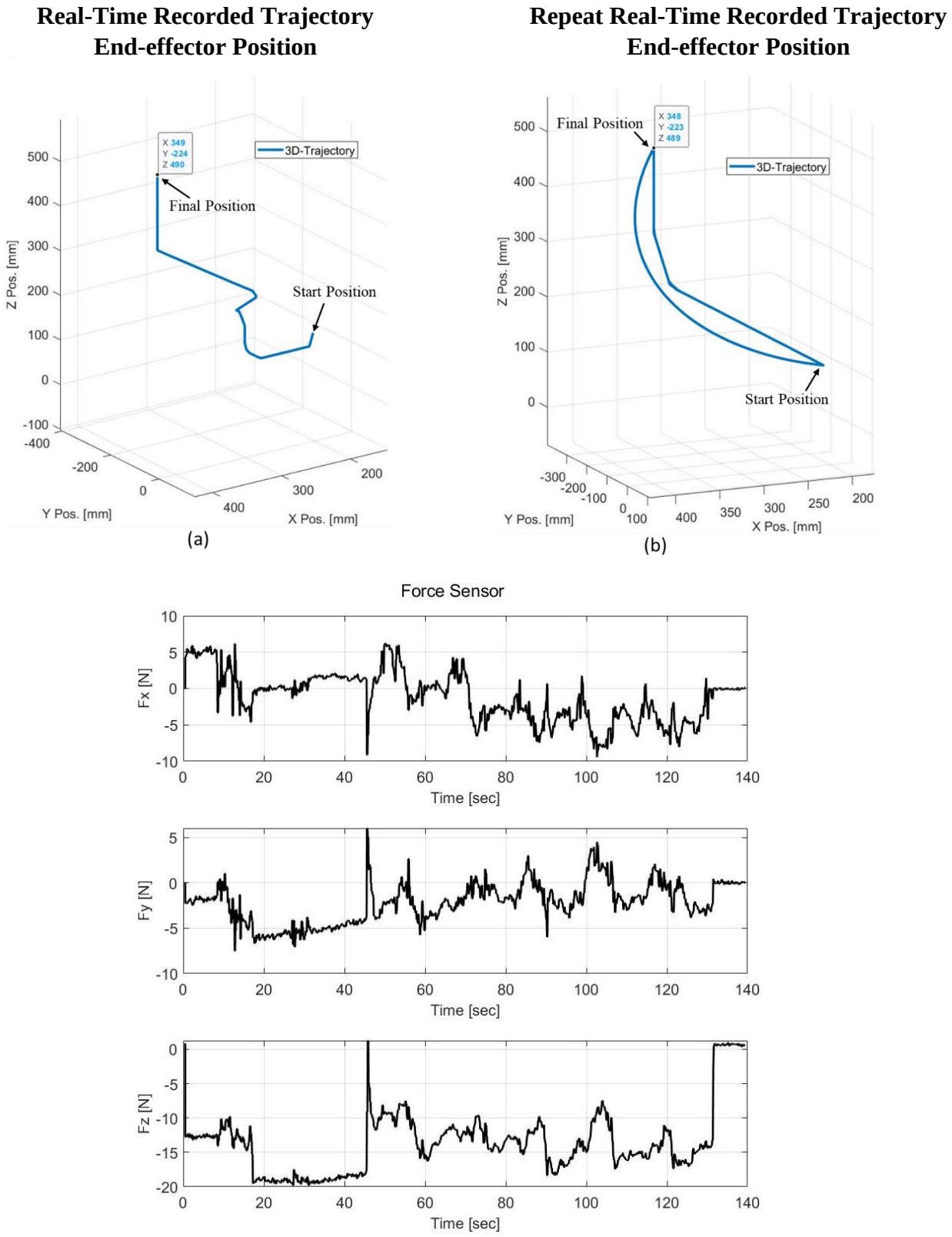


Figure 7.58 Human-robot interactive force detected from the force sensor at the end-effector during PREwRT[2].

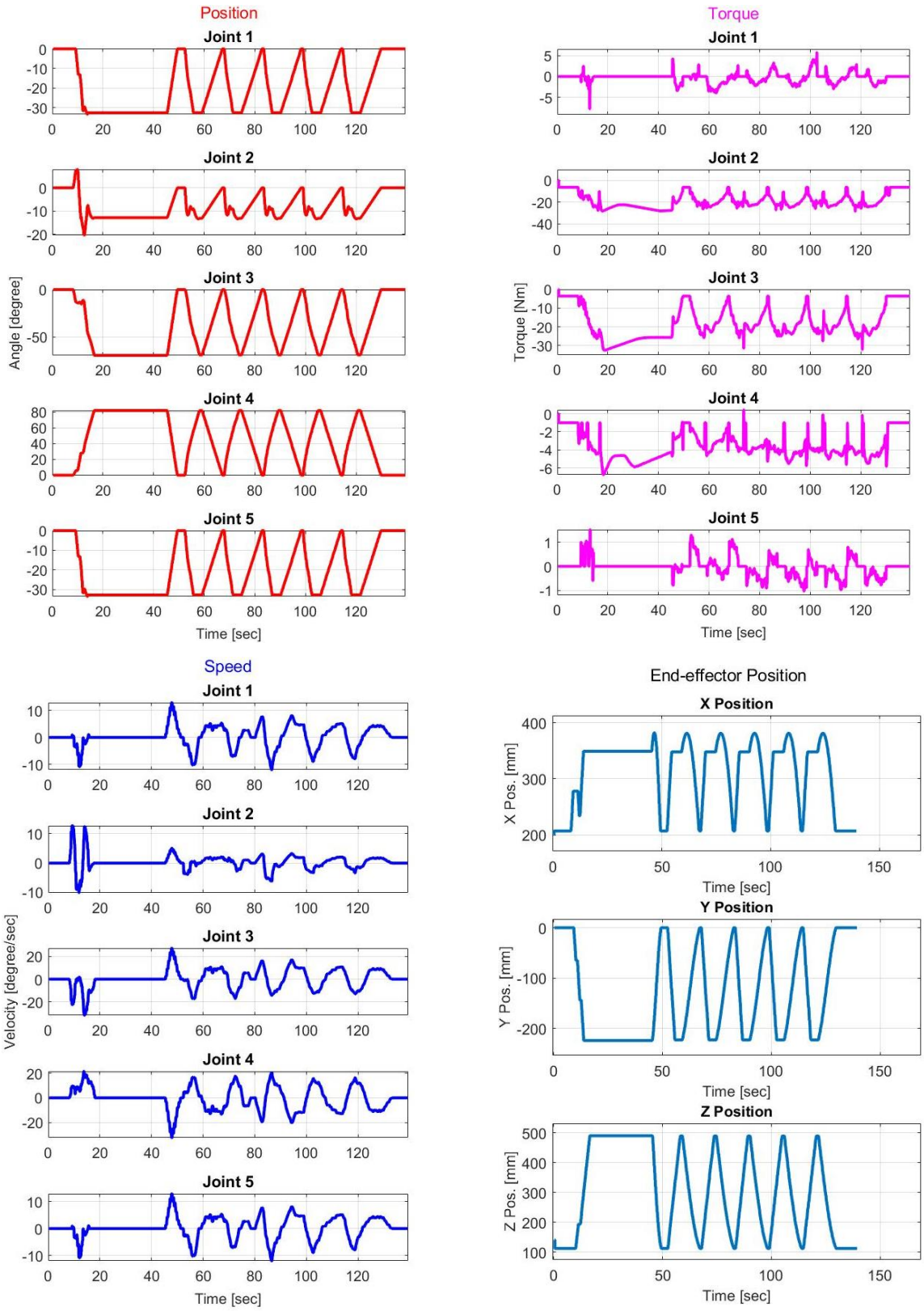


Figure 7.59 Joint angles, torques, speed, and end-effector position during PREwRT[2].

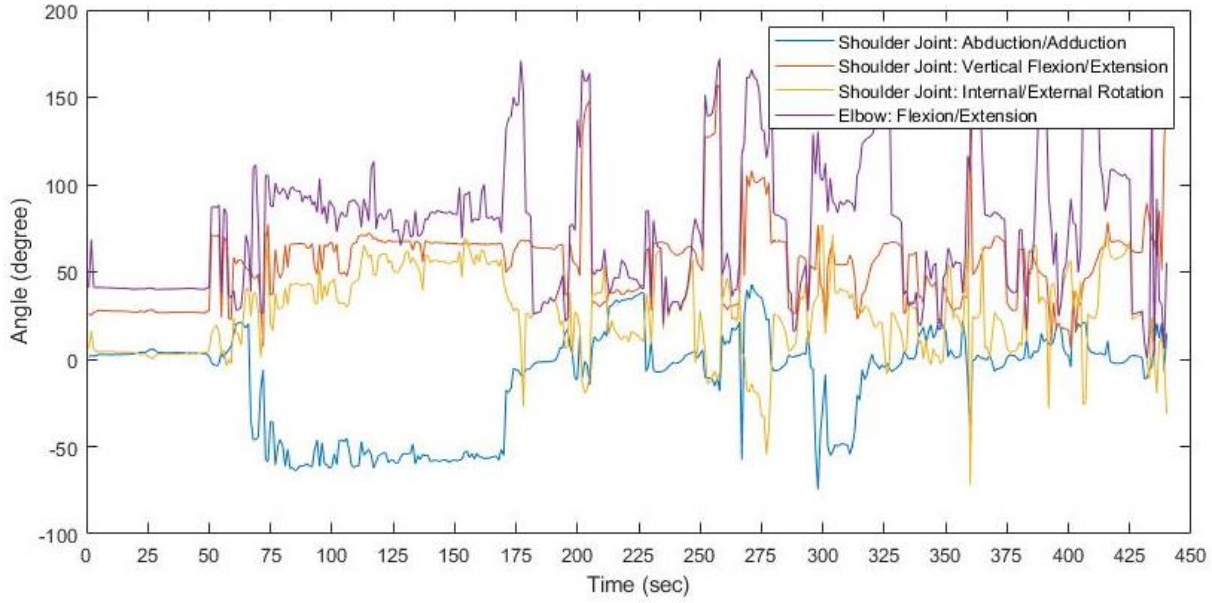


Figure 7.60 Upper-limb joint coordinate from Kinect sensor during PREwRT[2].

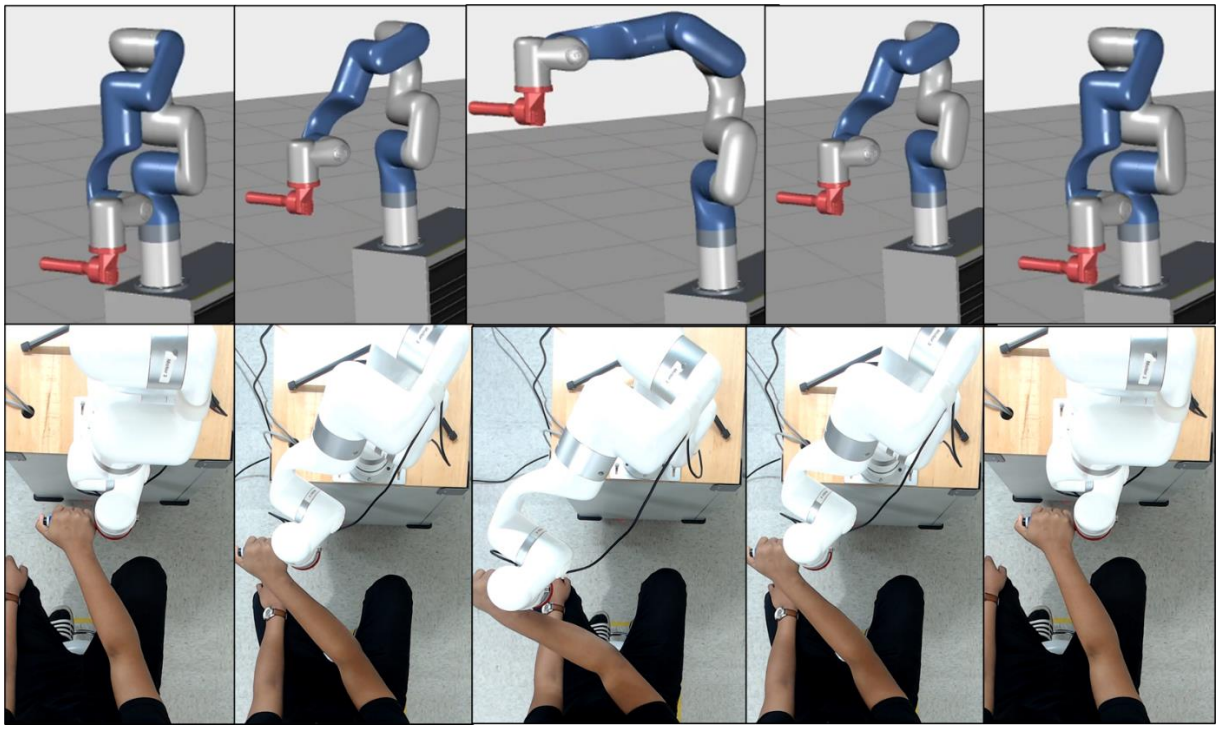


Figure 7.61 Monitoring PREwRT[2] using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session

7.2.3 Resistive Rehab Exercise (RRE) using Impedance Control

An impedance control [92] mode is used to provide resistive rehab exercise (RRE) with the developed telerehabilitation framework. During this exercise, the participant actively used muscle strength to accomplish a defined task, such as pushing/pulling the robot's end-effector to a certain target location. An operator uses the user interface shown in Fig. 7.62 to change the linear stiffness coefficient (k) value and passes this value to the robot remotely by enabling a toggle button to provide various resistance to the xArm-5's movement.

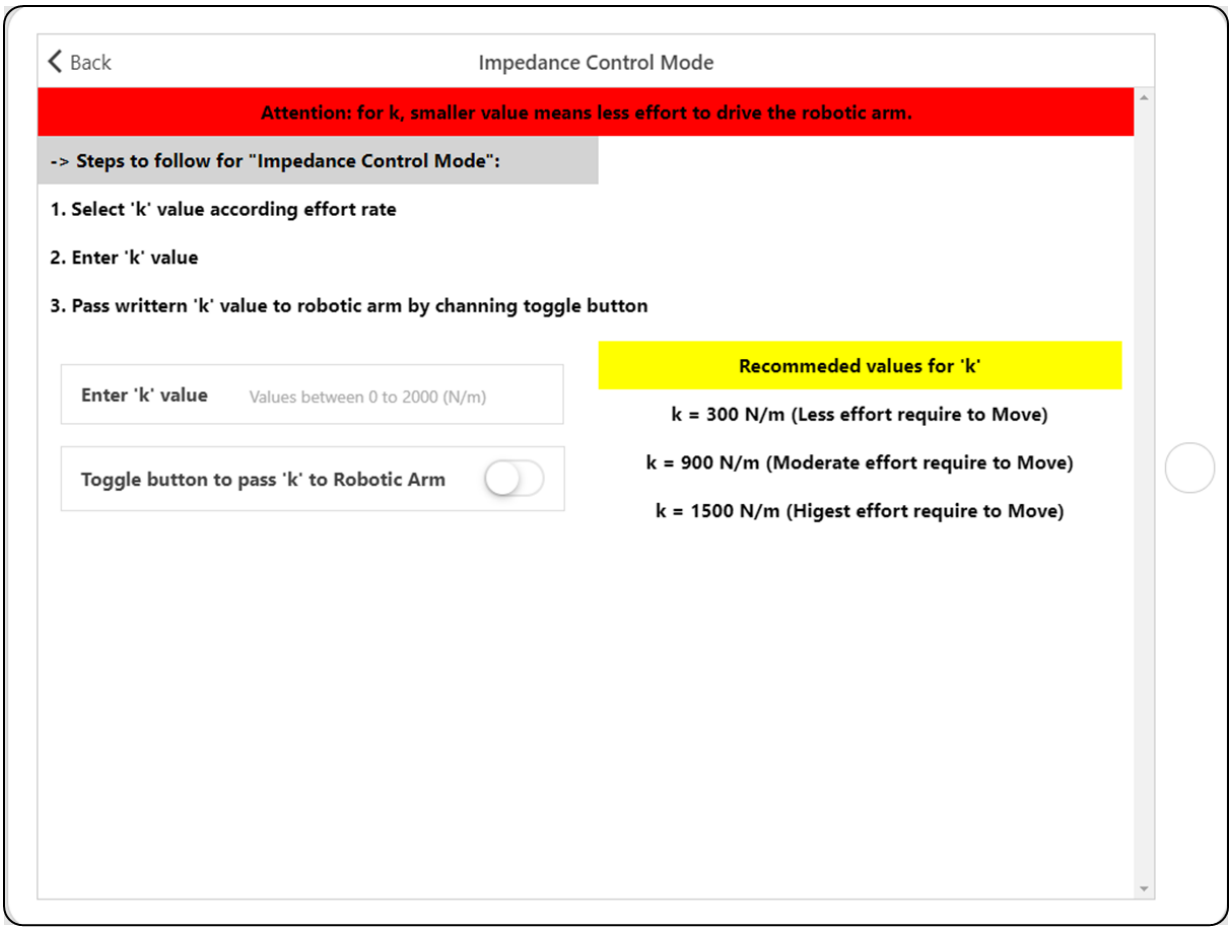


Figure 7.62 Augmented reality user interface for RRE using impedance control mode of telerehabilitation exercises

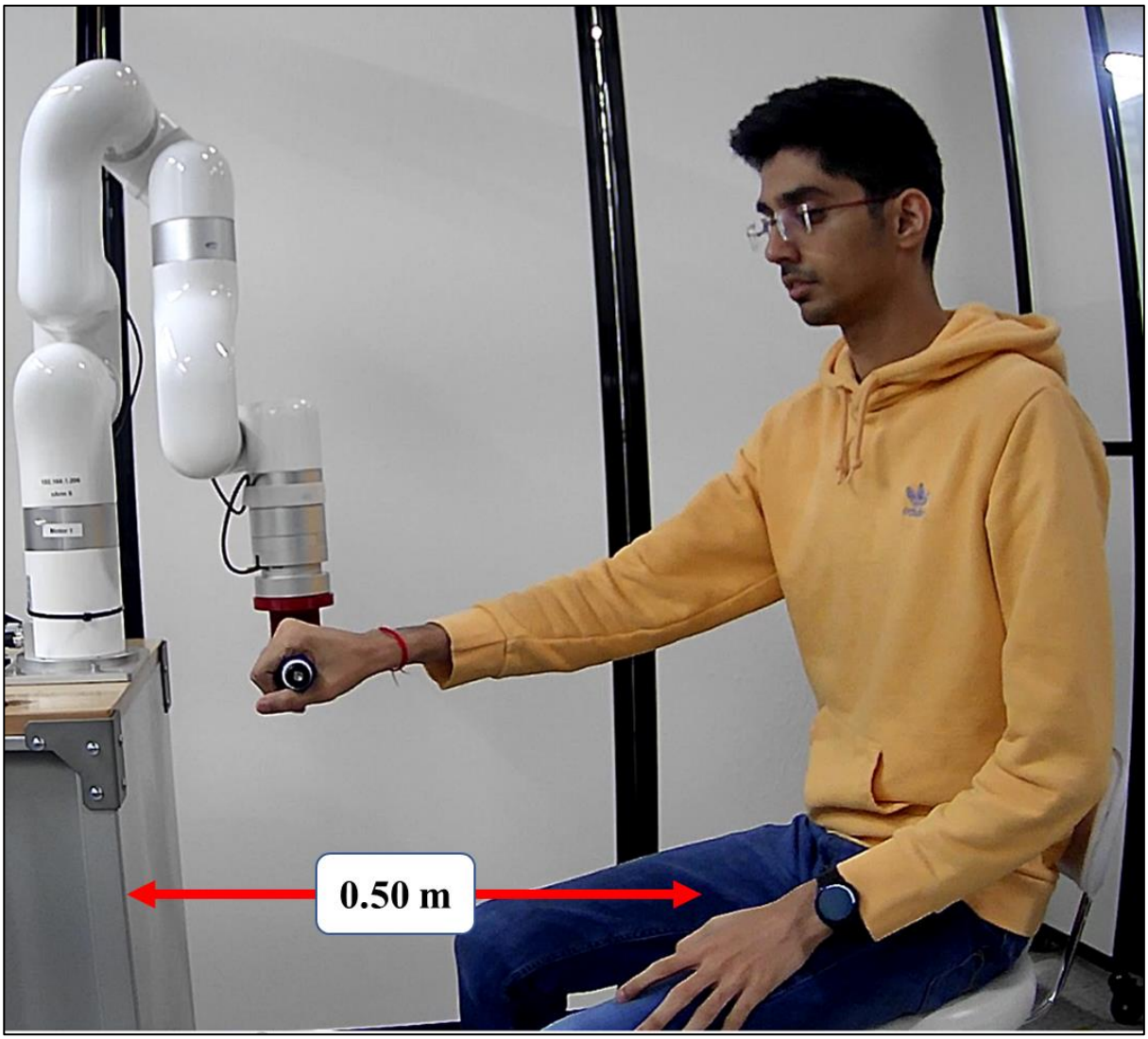


Figure 7.63 Participants sitting position during the resistive rehab exercises (k = 300 N/m, k = 900 N/m, k = 1500 N/m).

As shown in Fig. 7.63, participants sat on the chair at a distance of 0.50 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during the resistive rehab exercises (k = 300 N/m, k = 900 N/m, k = 1500 N/m).

Note that an operator can send ‘k’ values between 0 to 2000 (N/m). As the ‘k’ value increases, the robot becomes more resistant, and the participant needs to put more force on the end-effector to accomplish the task. We performed this RRE with three different participants and with three distinct stiffness ‘k’ values.

For participant-1, the stiffness was set to k = 300 N/m value. It can be seen from Fig. 7.64 that participant-1 maneuvers (pull/puss) the robot’s end-effector to reach four different targets. Also, it can be seen from the force plots in Fig. 7.64 that the participant exerted a force between 30N to 40N during the RRE. Figure 7.65 shows the joint angles, speed, and torque, of the xArm-5’s robot, and Fig. 7.66 illustrates the participant-1’s upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.64. Figure 7.67 shows the AR digital twin of the xArm-5 robot and participant-1’s upper-limb movement through the Microsoft Teams video session during that RRE.

End-effector Position

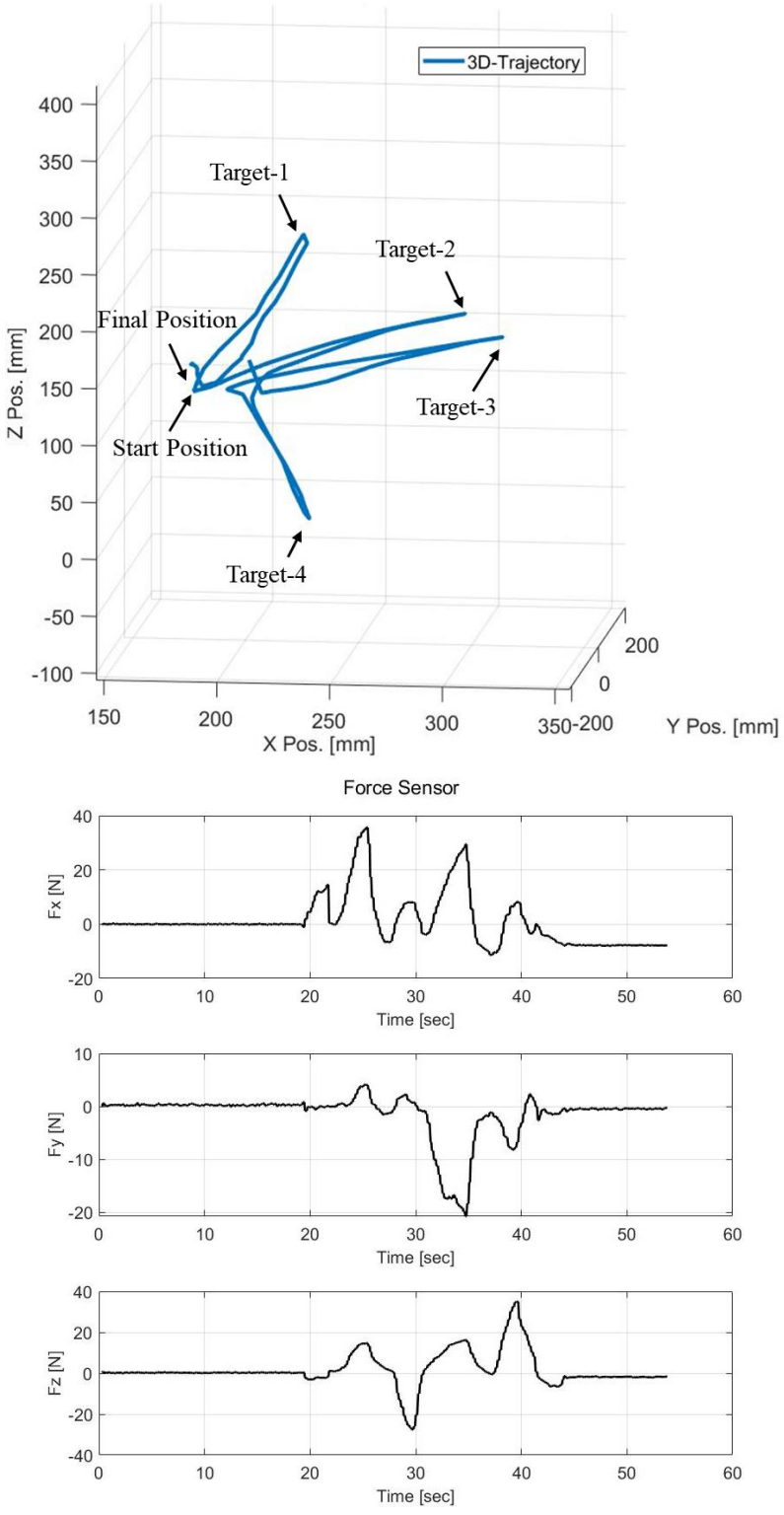


Figure 7.64 Robot's end-effector position and the human-robot interactive force during the resistive rehab exercise (k = 300 N/m) were performed by participant-1.

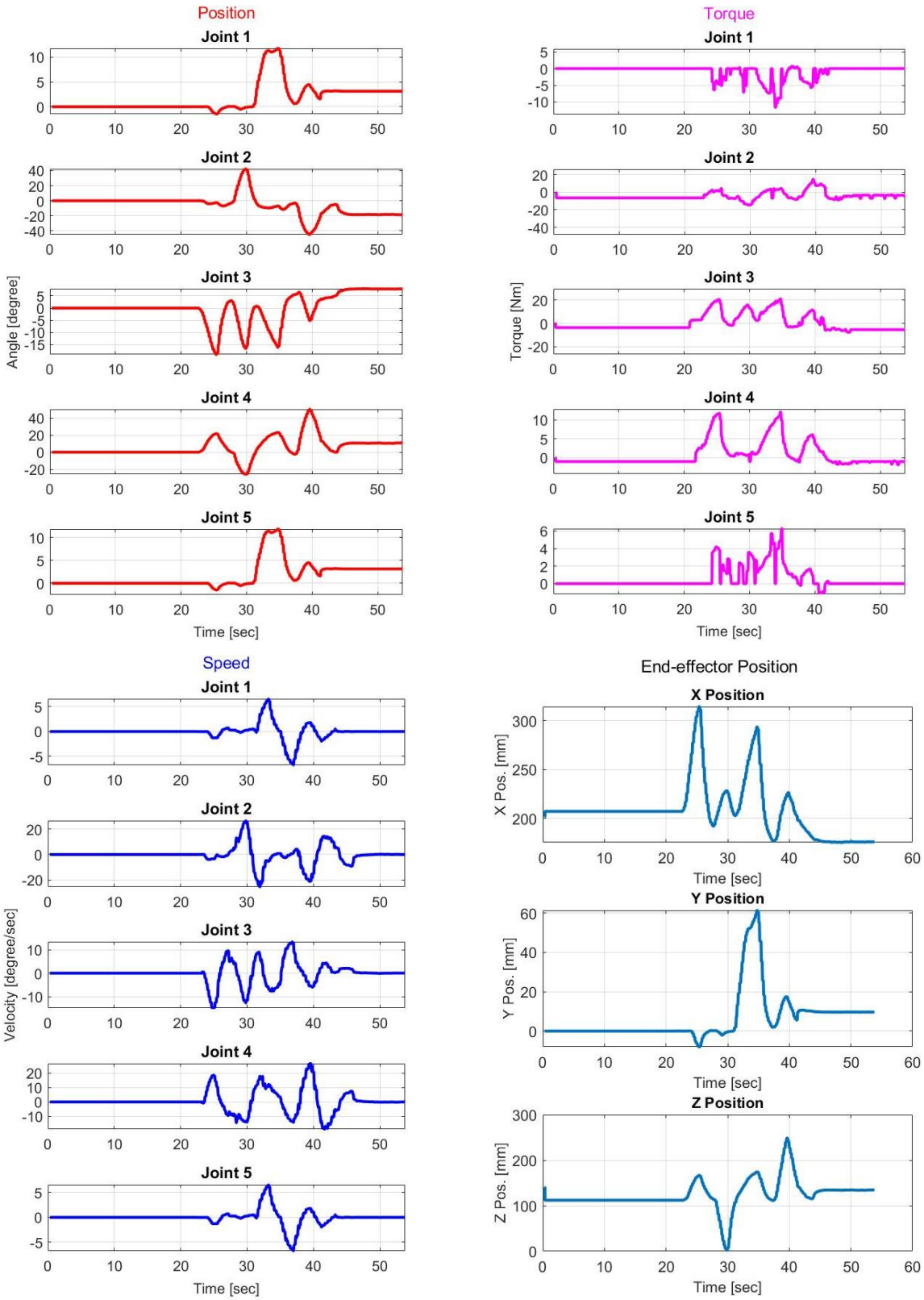


Figure 7.65 Joint angles, torques, speed, and end-effector position during the resistive rehab exercise (k = 300 N/m) performed by participant-1.

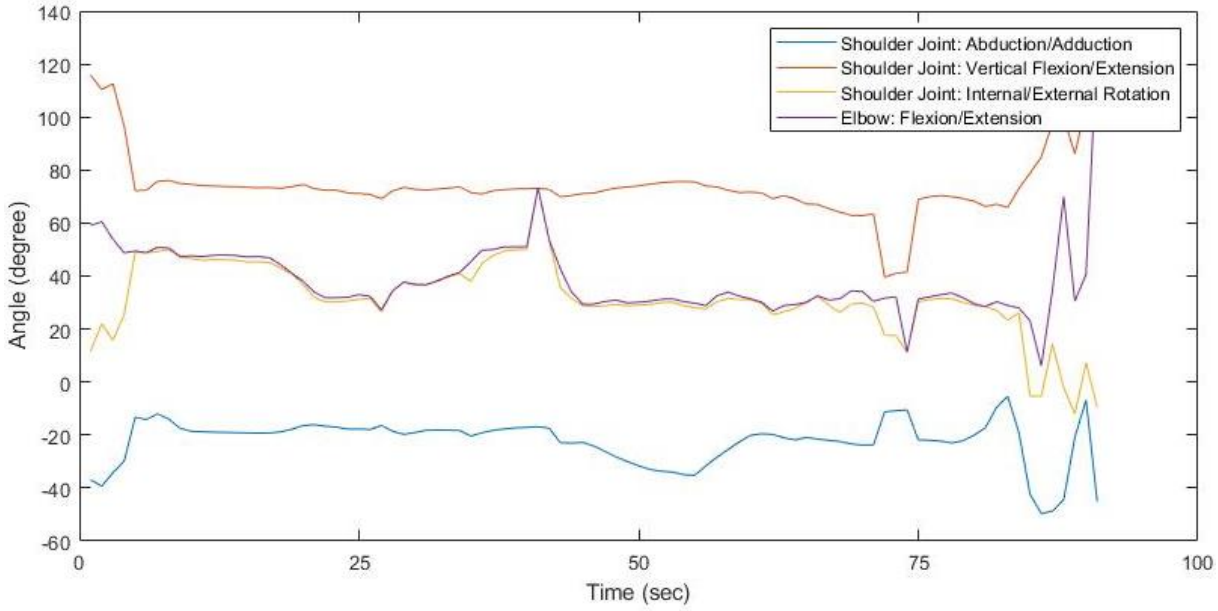


Figure 7.66 Participant-1's upper-limb joint coordinate from Kinect sensor during performing resistive rehab exercise (k = 300 N/m).

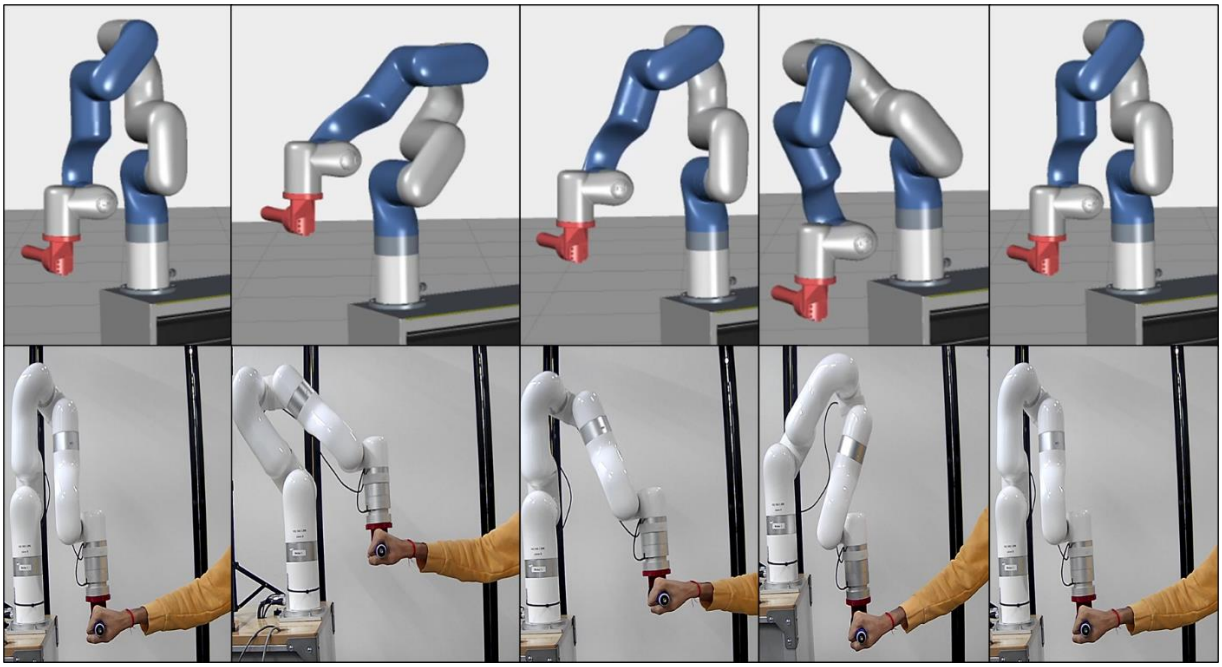


Figure 7.67 Monitoring RRE (k = 300 N/m) using Vuforia Studio AR platform and observing participant-1's upper-limb movement through Microsoft Teams video session.

For participant-2, the stiffness was set to k = 900 N/m value. It can be seen from Fig. 7.68 that participant-2 maneuvers (pull/puss) the robot's end-effector to reach four different targets. Also, it can be seen from the force plots in Fig. 7.68 that the participant exerted a force between 35N to 50N during the RRE. Figure 7.69 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.70 illustrates the participant-2's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.68. Figure 7.71 shows the AR digital twin of the xArm-5 robot and participant-2's upper-limb movement through the Microsoft Teams video session during that RRE.

End-effector Position

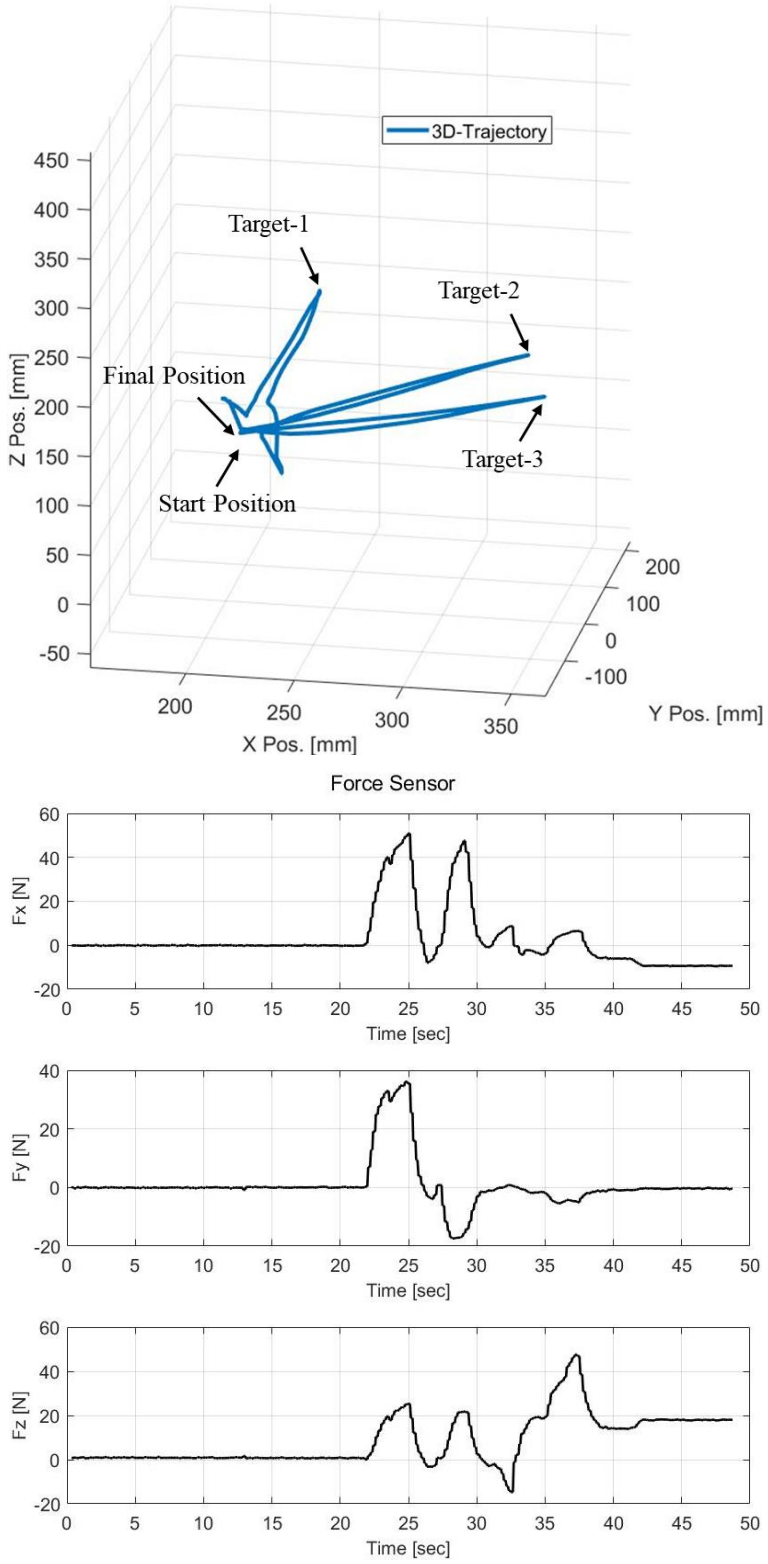


Figure 7.68 Robot's end-effector position and the human-robot interactive force during the resistive rehab exercise (k = 900 N/m) were performed by participant-2.

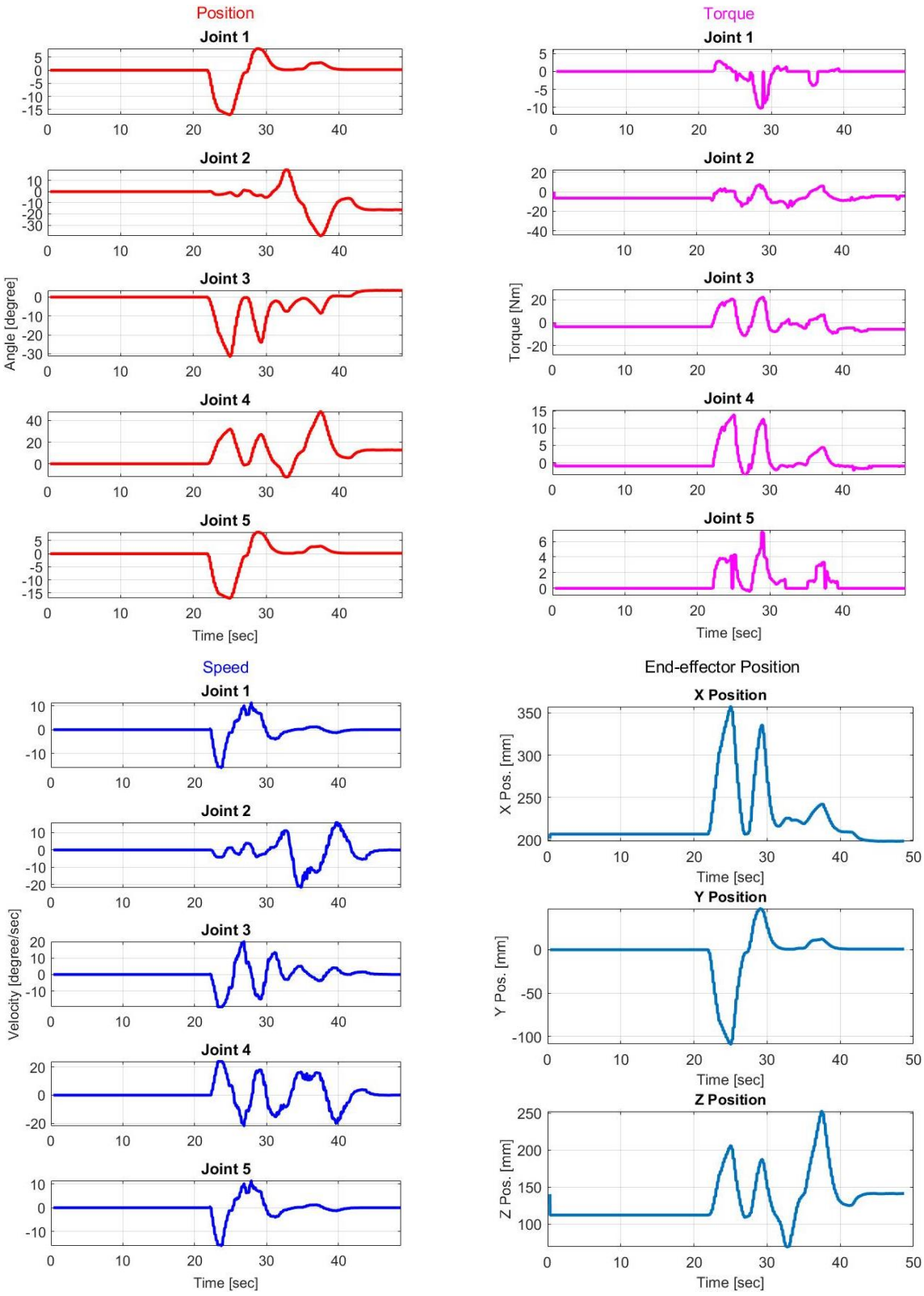


Figure 7.69 Joint angles, torques, speed, and end-effector position during the resistive rehab exercise (k = 900 N/m) performed by participant-2.

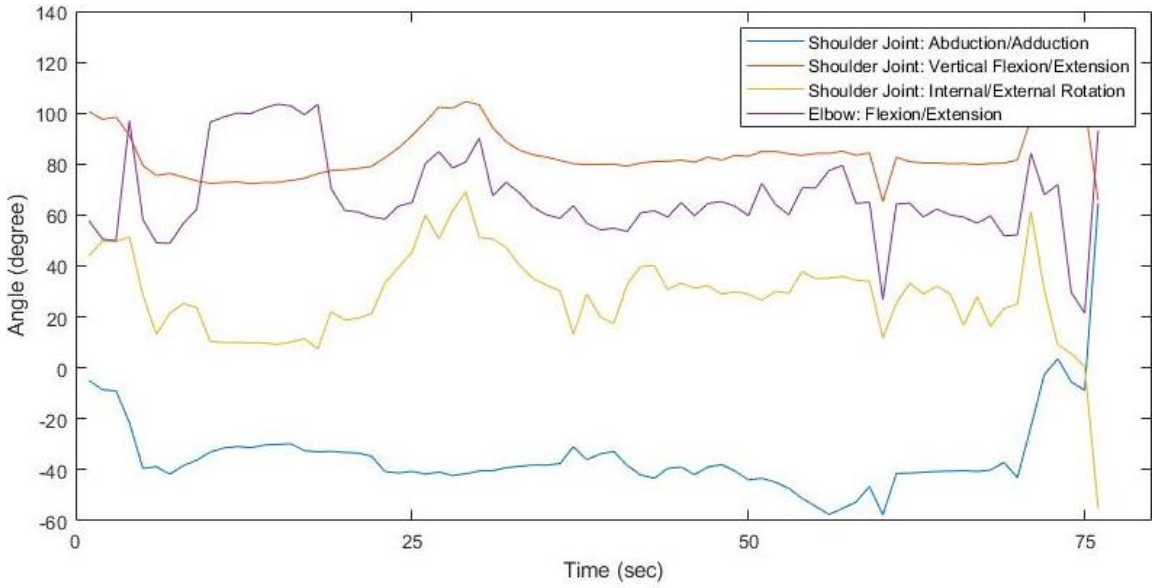


Figure 7.70 Participant-2's upper-limb joint coordinate from Kinect sensor during performing resistive rehab exercise (k = 900 N/m).

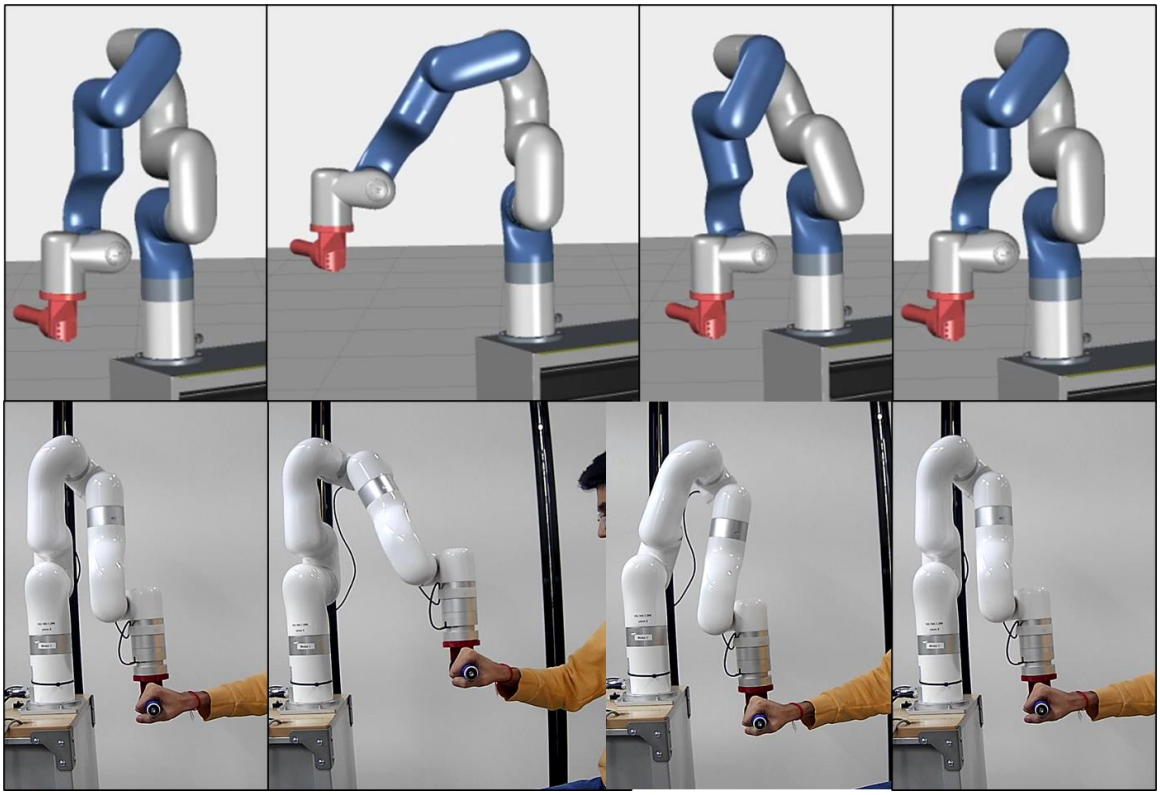


Figure 7.71 Monitoring RRE (k = 900 N/m) using Vuforia Studio AR platform and observing participant-2's upper-limb movement through Microsoft Teams video session.

For participant-3, the stiffness was set to k = 1500 N/m value. It can be seen from Fig. 7.72 that participant-3 maneuvers (pull/puss) the robot's end-effector to reach four different targets. Also, it can be seen from the force plots in Fig. 7.72 that the participant exerted a force between 50N to 60N during the RRE. Figure 7.73 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.74 illustrates the participant-3's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.72. Figure 7.75 shows the AR digital twin of the xArm-5 robot and participant-1's upper-limb movement through the Microsoft Teams video session during that RRE.

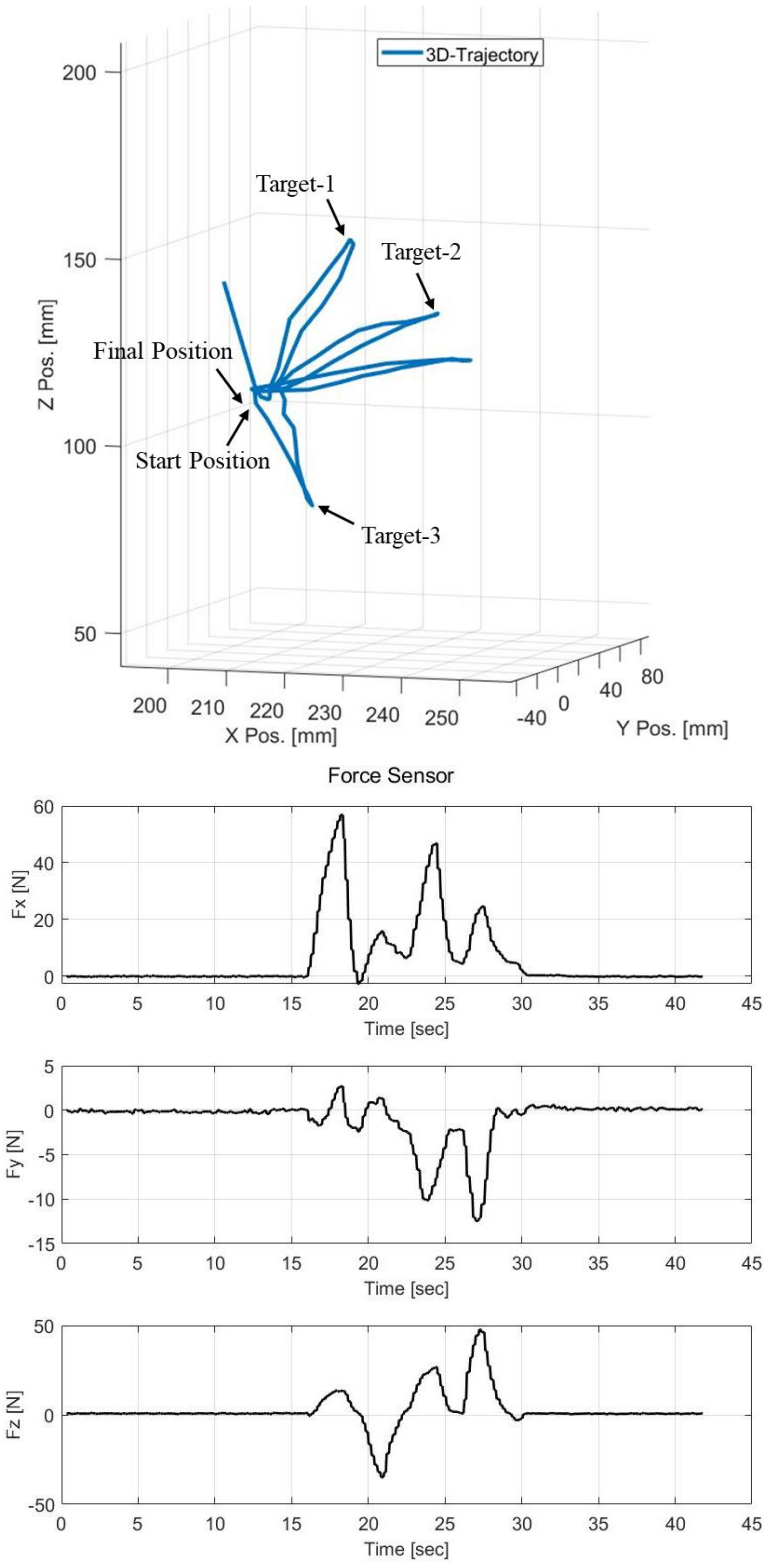


Figure 7.72 Robot's end-effector position and the human-robot interactive force during the resistive rehab exercise (k = 1500 N/m) were performed by participant-3.

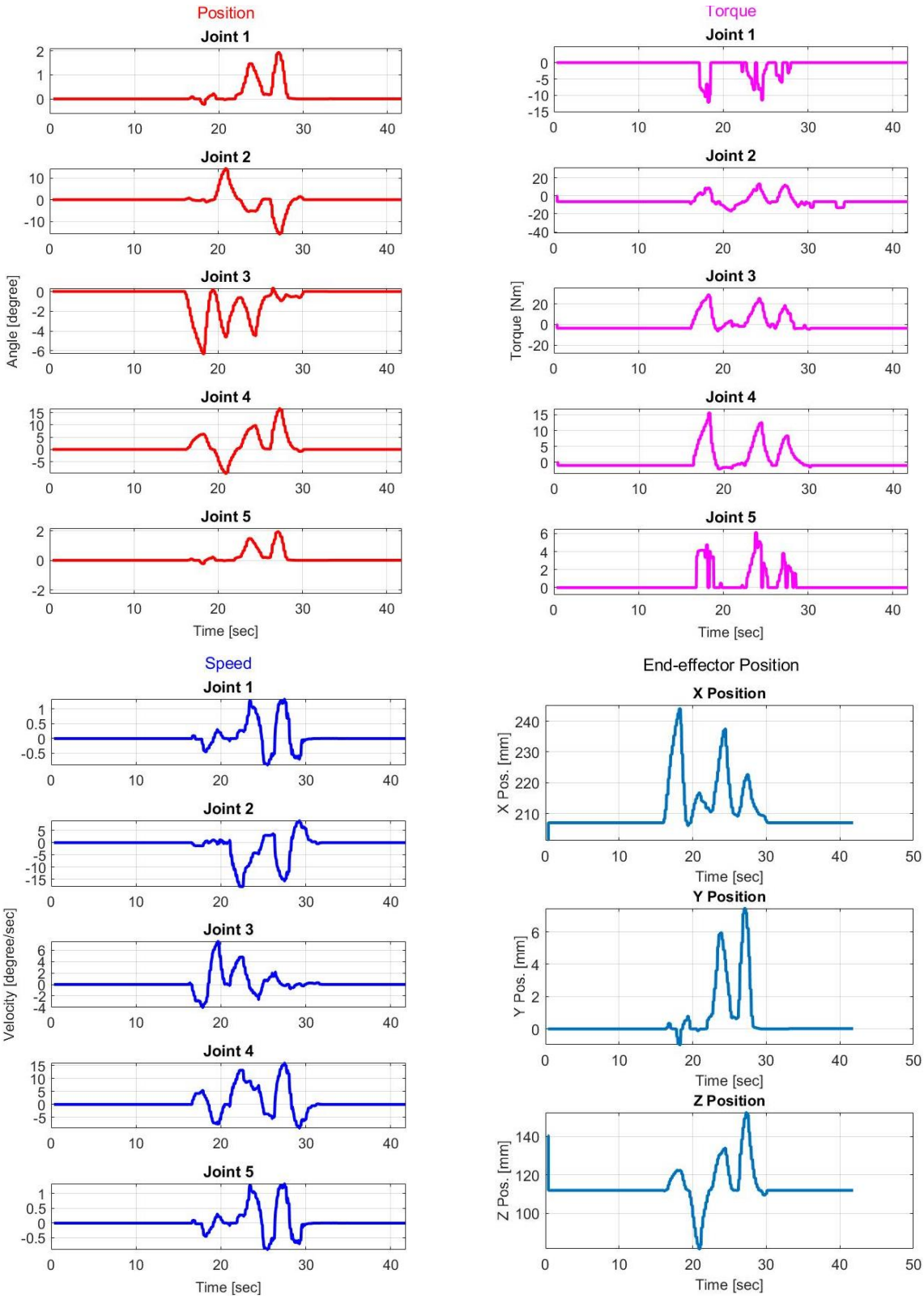


Figure 7.73 Joint angles, torques, speed, and end-effector position during the resistive rehab exercise (k = 1500 N/m) performed by participant-3.

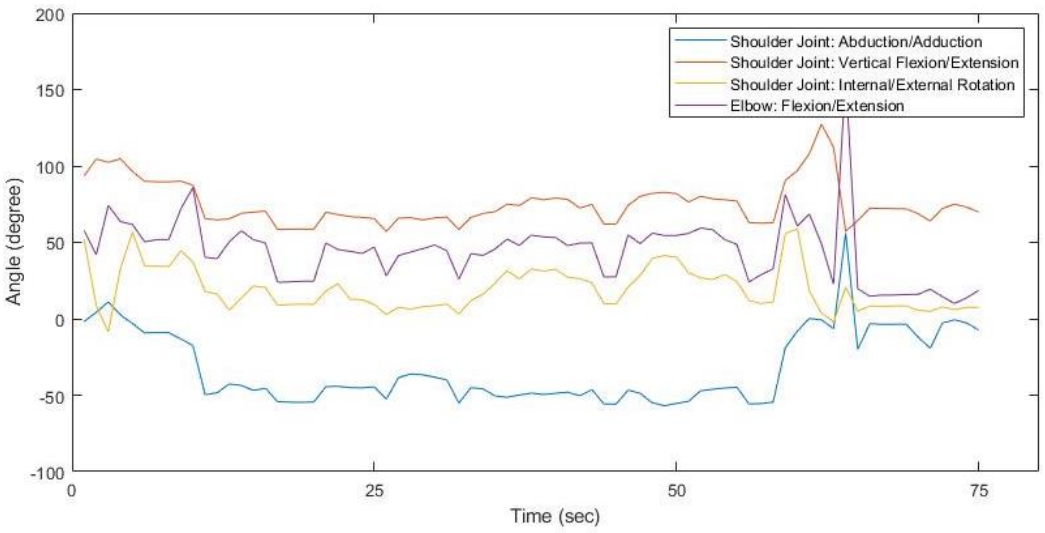


Figure 7.74 Participant-3's upper-limb joint coordinate from Kinect sensor during performing resistive rehab exercise (k = 1500 N/m).

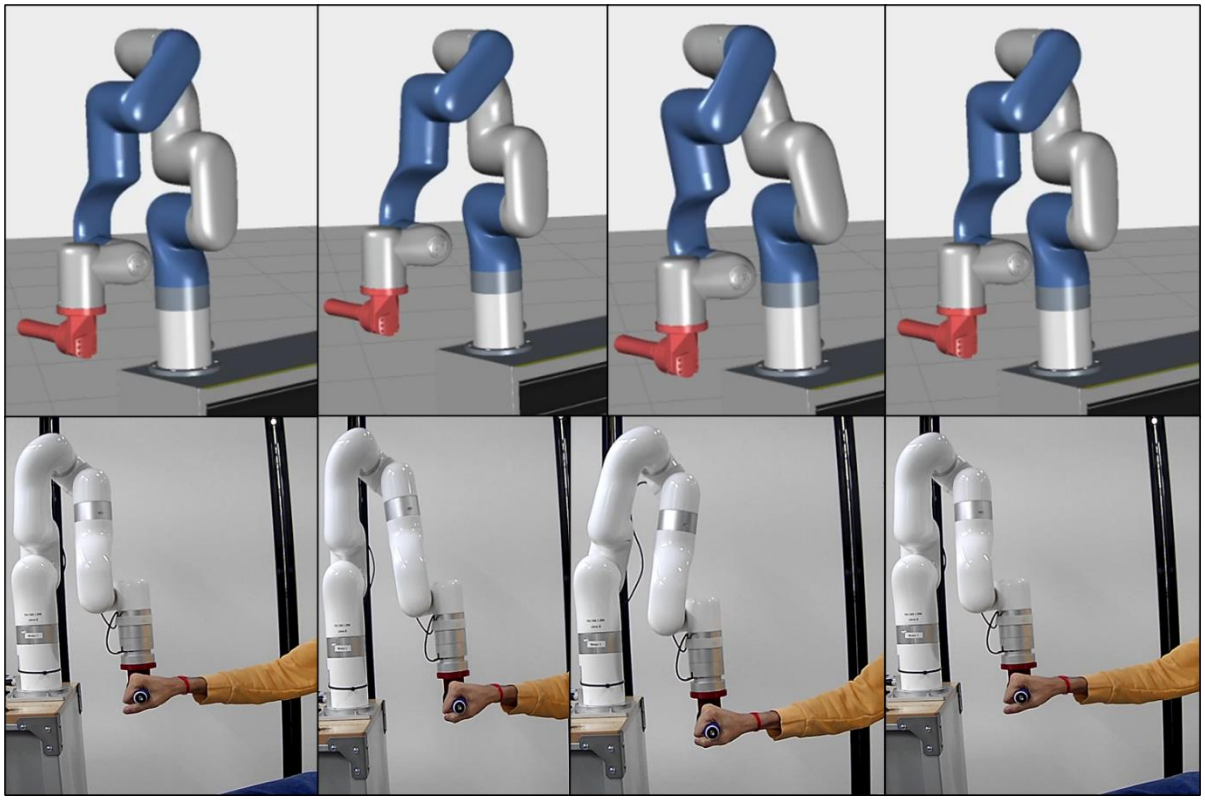


Figure 7.75 Monitoring RRE (k = 1500 N/m) using Vuforia Studio AR platform and observing participant-3's upper-limb movement through Microsoft Teams video session.

7.2.4 Manual Teaching Mode for Generating and Providing Passive Rehab Exercise (PRE)

The manual teaching mode, also known as the trajectory recording mode of telerehabilitation, allows recording the trajectory and repeating the exercises. Figure 7.76 shows the user interface of Vuforia View for the trajectory recording mode of telerehabilitation exercises. During this rehab exercise, the operator can enable the manual mode of the robot remotely by changing the status of the toggle button. Manual mode allows participants to record the trajectory by themselves or through a helper/caregiver. Once the exercise is recorded, the operator or the participant can begin repeating the recorded trajectory/exercise.

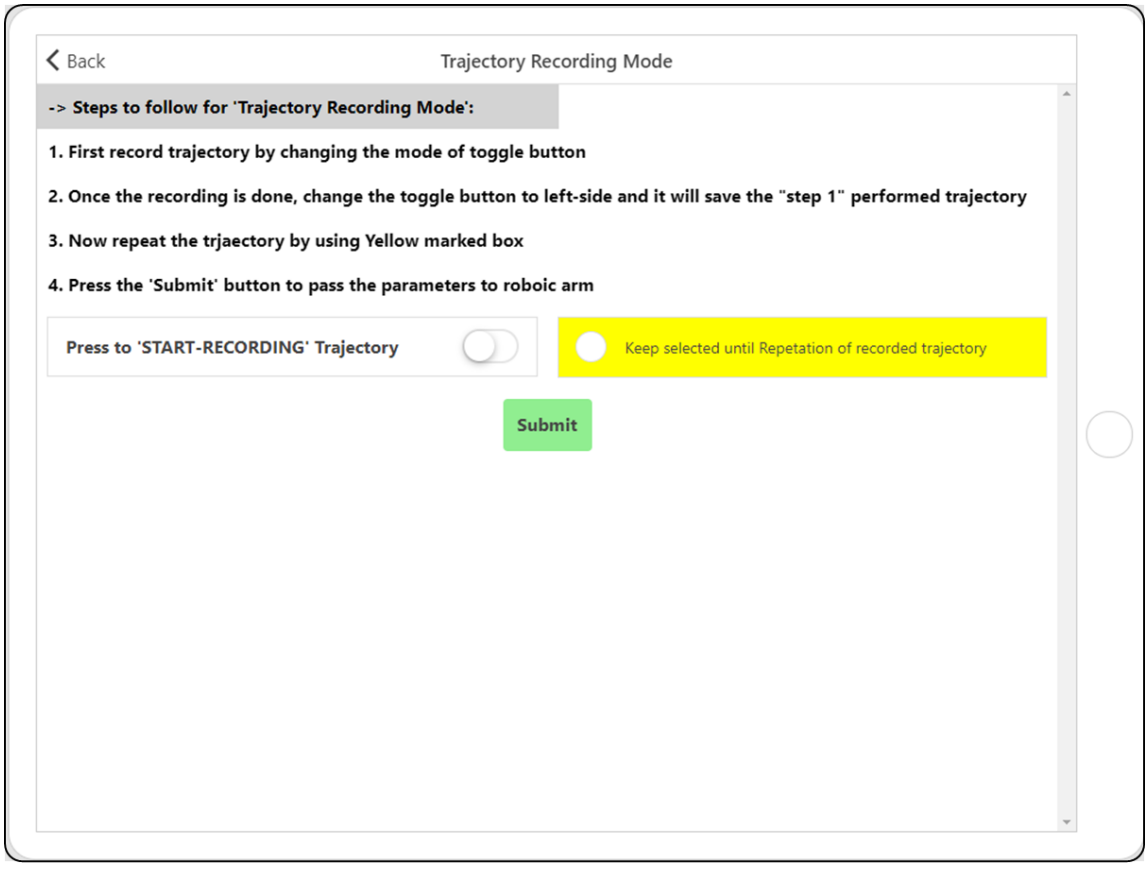


Figure 7.76 Augmented reality user interface for manual teaching mode for generating and providing passive rehab exercise (PRE) of telerehabilitation exercises.

In this therapeutic approach, we did passive rehab exercises (PRE) in two different experiments scenarios:

- Scenario-1: Participant-1 recorded the trajectory with the assistance of a caregiver/helper, and the trajectory/exercise was repeated to provide continuous repetitive PRE.
- Scenario-2: Participant-2 themself recorded the trajectory, and the trajectory/exercise was repeated to provide continuous repetitive PRE.

As shown in Fig. 7.63, participants sat on the chair at a distance of 0.50 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during multi-joint PRE trajectory recorded through helper and repeated recorded trajectory.

In all cases, the exercises were performed under the supervision of skilled personnel.

For participant-1, the operator enabled the manual mode from the Vuforia view user interface to record the trajectory. As shown in Fig. 7.77, the caregiver/helper started moving the robot's end-effector from its initial position towards target-1 (the participant's pain-free range of area) and finished recording at the final position. Also, it can be seen from the force plots in Fig. 7.77 that the participant exerted a force during the PRE. Figure 7.78 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.79 illustrates the participant-1's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.77. Figure 7.80 shows the AR digital twin of the xArm-5 robot and participant-1's upper-limb movement through the Microsoft Teams video session during that PRE. It can be seen from the experimental results of participant-1 (see Fig. 7.77 through Fig. 7.80) that the recorded trajectory/motion was repeated four times.

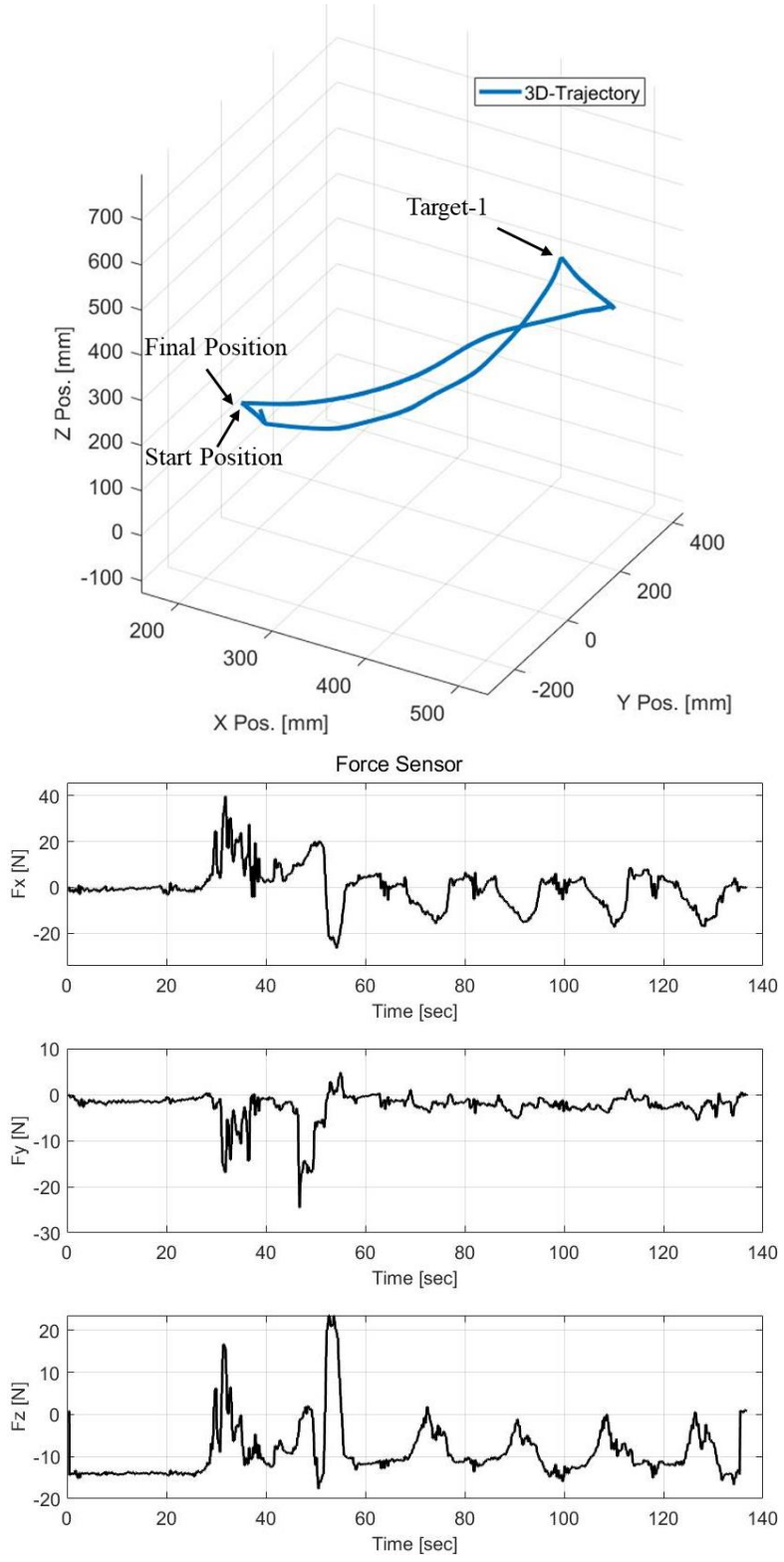


Figure 7.77 Human-robot interactive force detected from the force sensor at the end-effector during multi-joint PRE trajectory recorded through helper and repeated recorded trajectory.

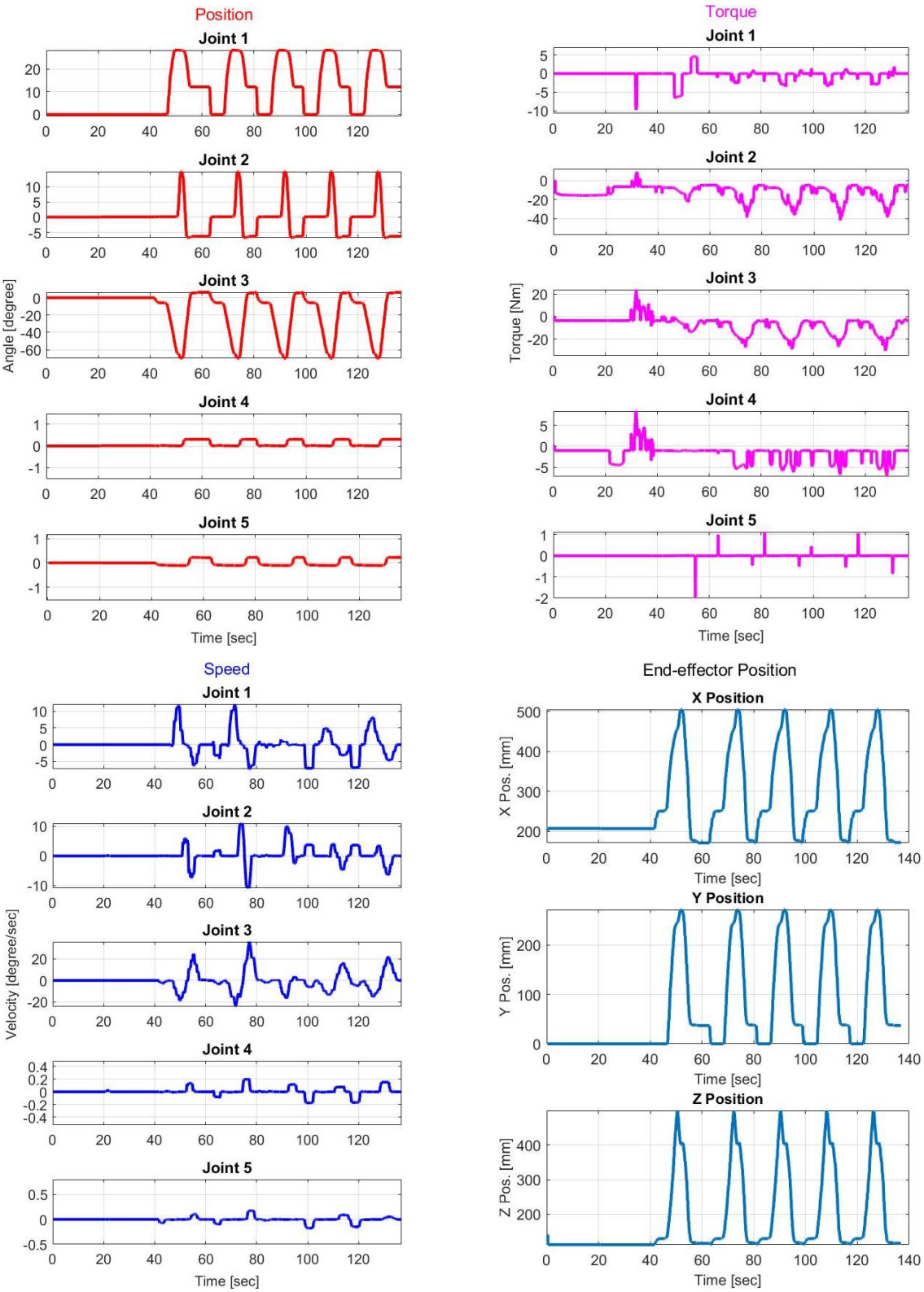


Figure 7.78 Joint angles, torques, speed, and end-effector position during multi-joint PRE trajectory recorded through helper and repeated recorded trajectory.

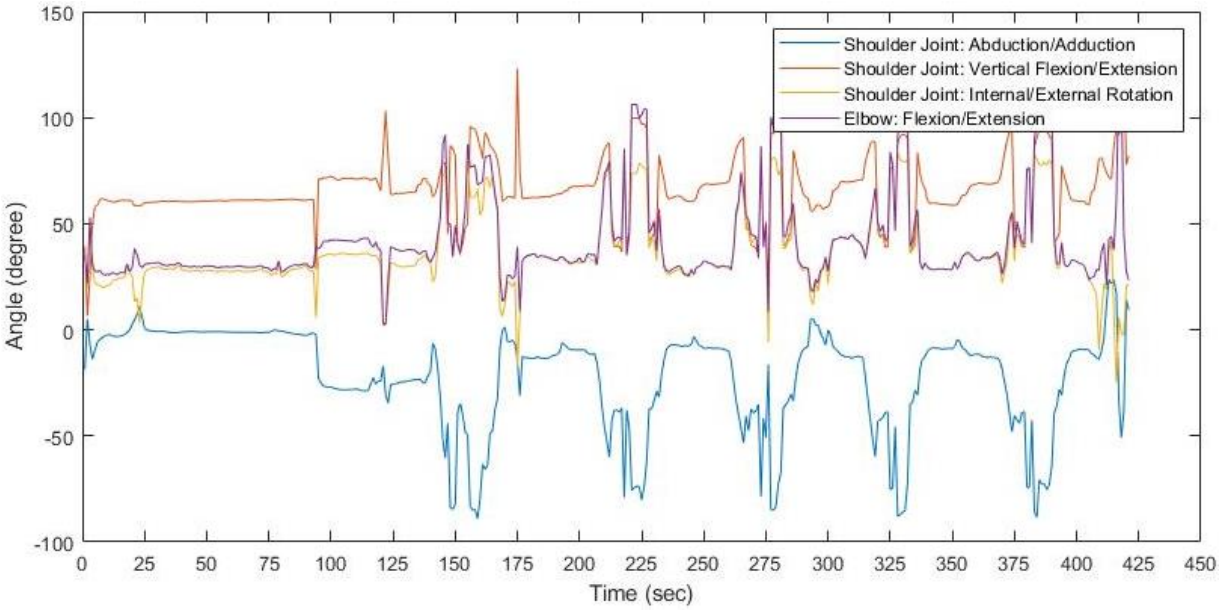


Figure 7.79 Upper-limb joint coordinate from Kinect sensor while multi-joint PRE trajectory recorded through helper and repeated recorded trajectory.

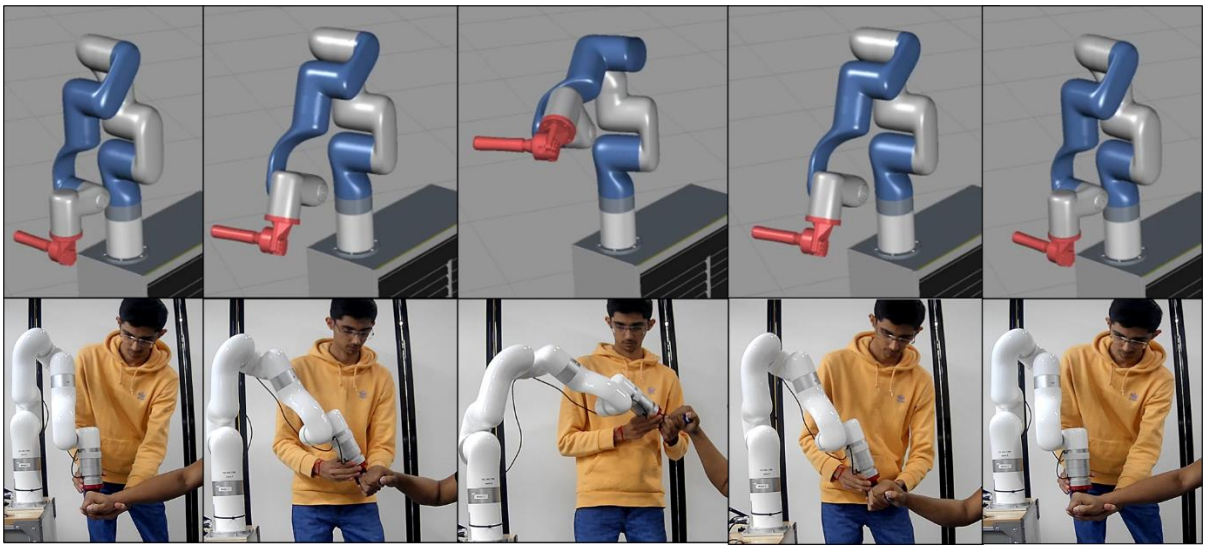


Figure 7.80 Monitoring multi-joint PRE trajectory recording mode using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

For participant-2, the operator enabled the manual mode from the Vuforia view user interface to record the trajectory. As shown in Fig. 7.81, the participant started moving the robot's end-effector from its initial position towards target-1 (the participant's pain-free range of area) and finished recording at the final position. Also, it can be seen from the force plots in Fig. 7.81 that the participant exerted a force during the PRE. Figure 7.82 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.83 illustrates the participant-2's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.81. Figure 7.84 shows the AR digital twin of the xArm-5 robot and participant-2's upper-limb movement through the Microsoft Teams video session during that PRE. It can be seen from the experimental results of participant-2 (see Fig. 7.81 through Fig. 7.84) that the recorded trajectory/motion was repeated four times.

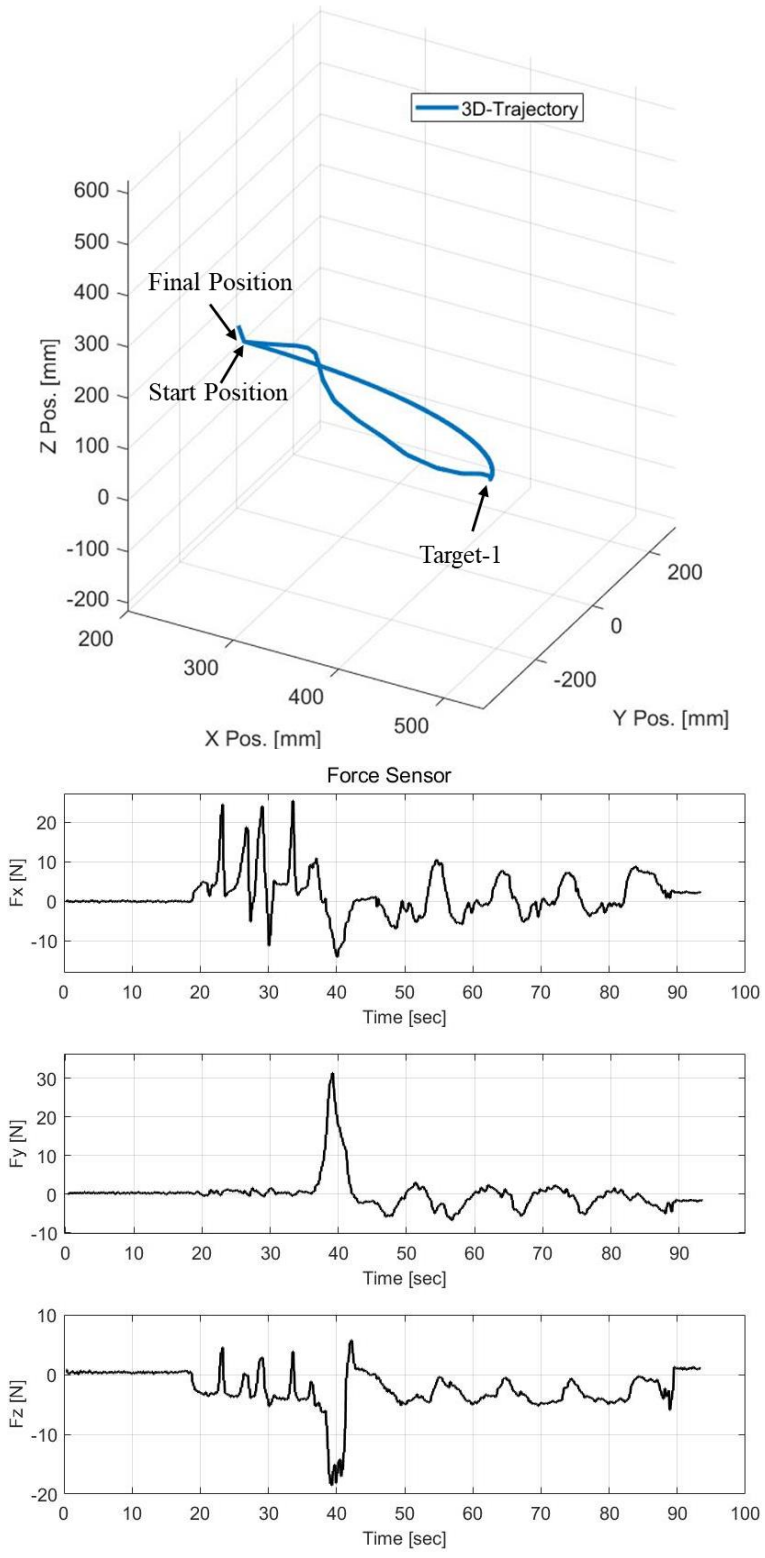


Figure 7.81 Human-robot interactive force detected from the force sensor at the end-effector during multi-joint PRE trajectory recorded by themselves and repeated recorded trajectory.

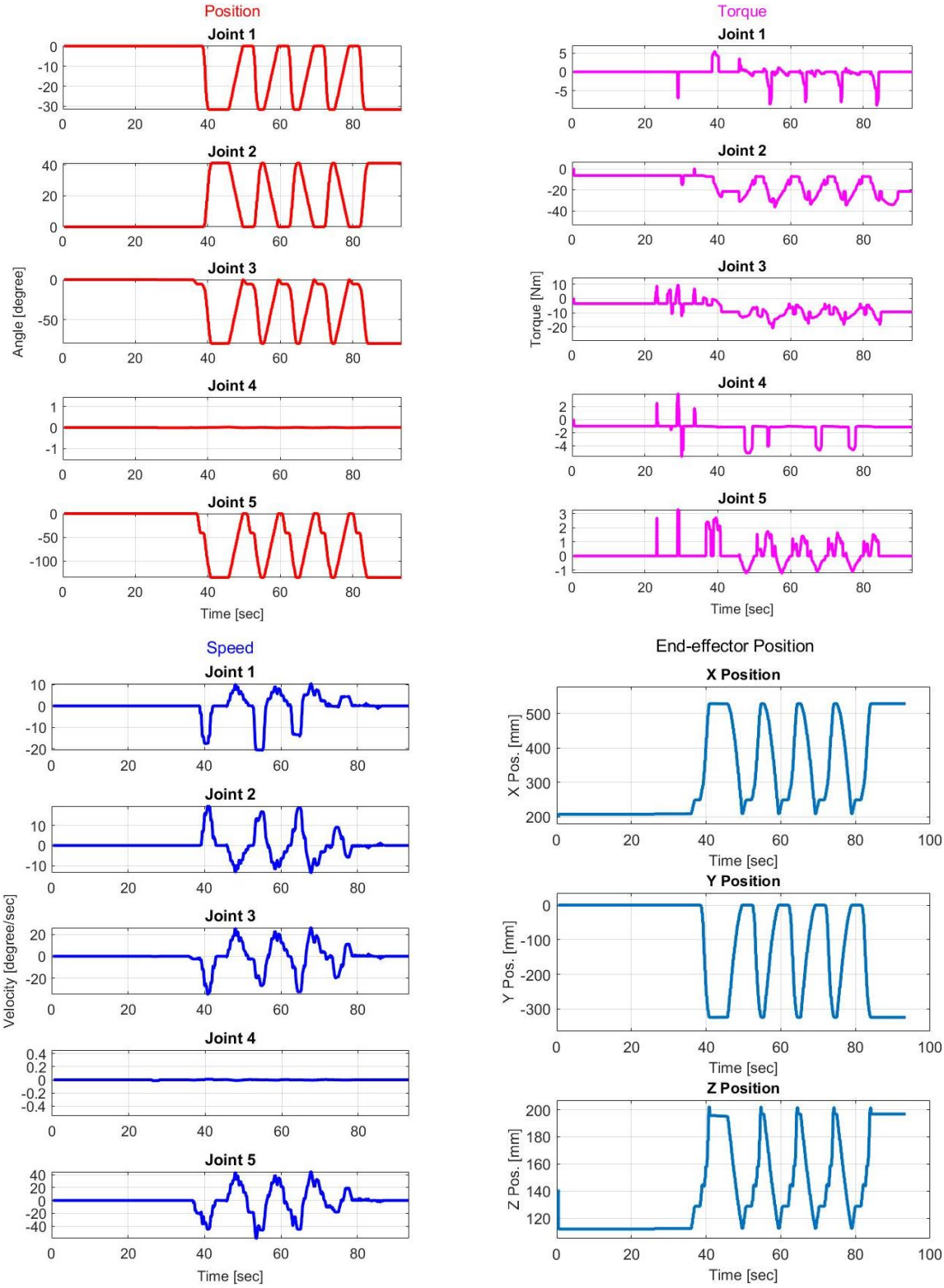


Figure 7.82 Joint angles, torques, speed, and end-effector position during multi-joint PRE trajectory recorded by themselves and repeated recorded trajectory.

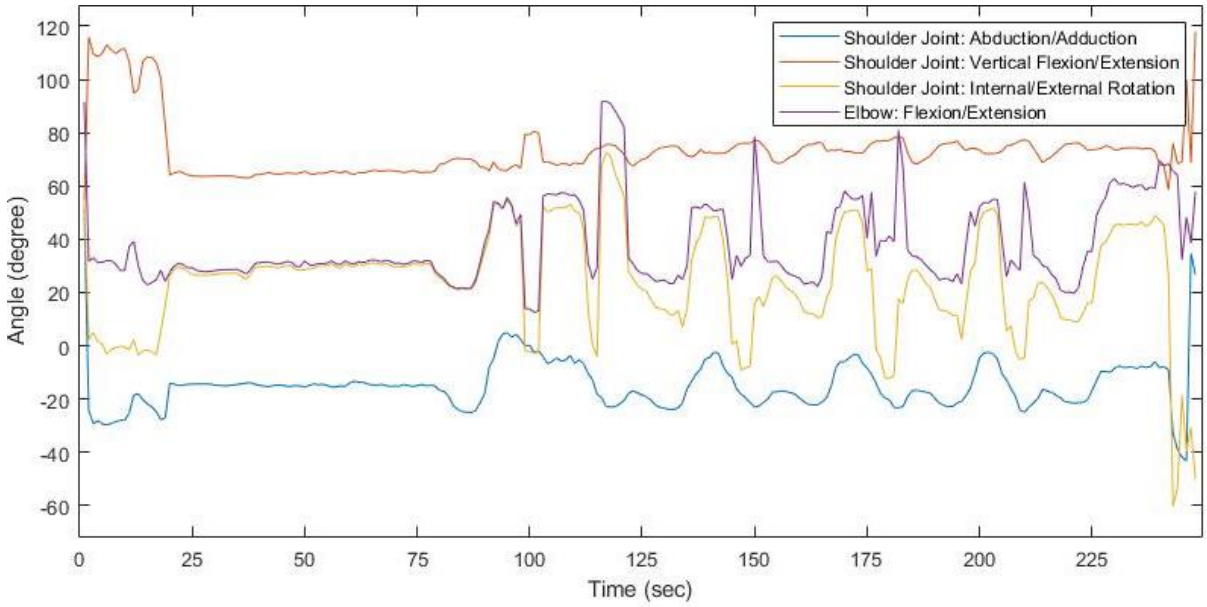


Figure 7.83 Upper-limb joint coordinate from Kinect sensor while multi-joint PRE trajectory recorded by themselves and repeated recorded trajectory.

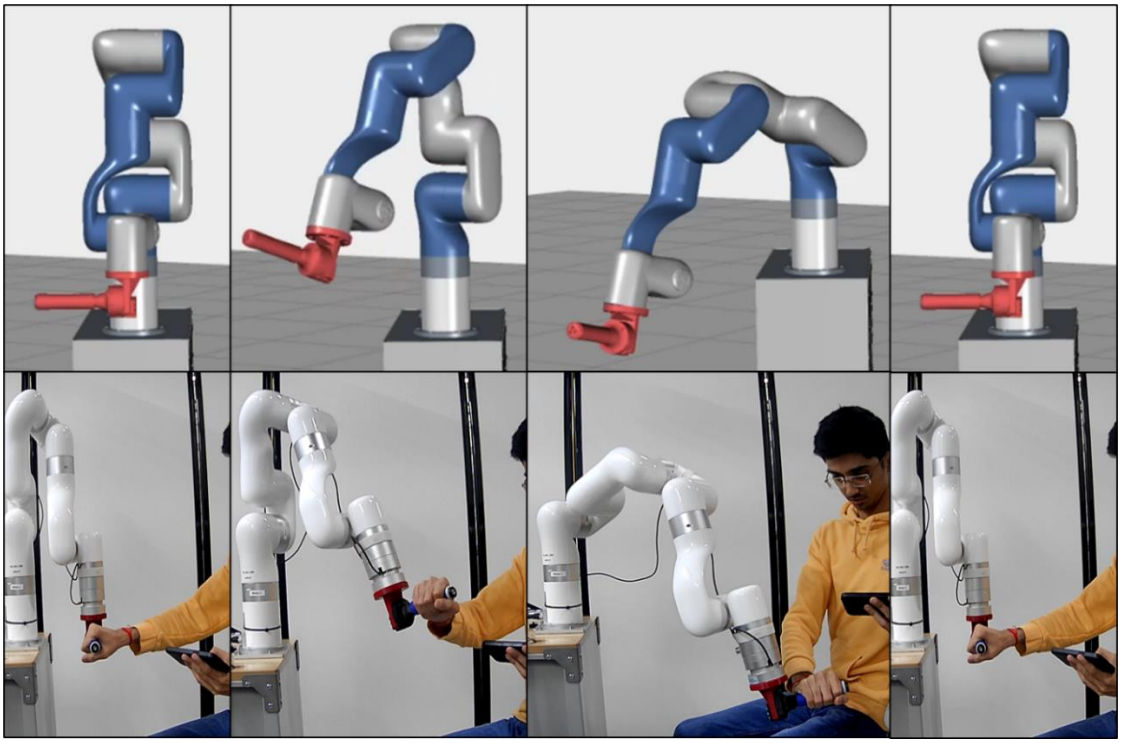


Figure 7.84 Monitoring multi-joint PRE trajectory recording mode using Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

7.2.5 Interactive One-on-One Real-Time TeleRehabilitation Exercise (IO^3RT^2RE) Mode, Controlling Individual Joints of the Robot to Provide PRE

Figure 7.85 depicts an augmented reality user interface for the IO^3RT^2RE mode of telerehabilitation exercises. Unlike other therapeutic control modes, as discussed earlier, where the robot's end-effector motion was controlled to PRE, in this mode, individual joints of the robot can be controlled using the Vuforia user interface to provide Telerehabilitation exercises (PRE). The operator can rotate the robot joint in the clockwise or anticlockwise direction by moving sliders on the positive or negative side through this user interface. Additionally, the 'Reset' button allows the rehab device to start from the robot's initial position (zero degrees for all five joints).

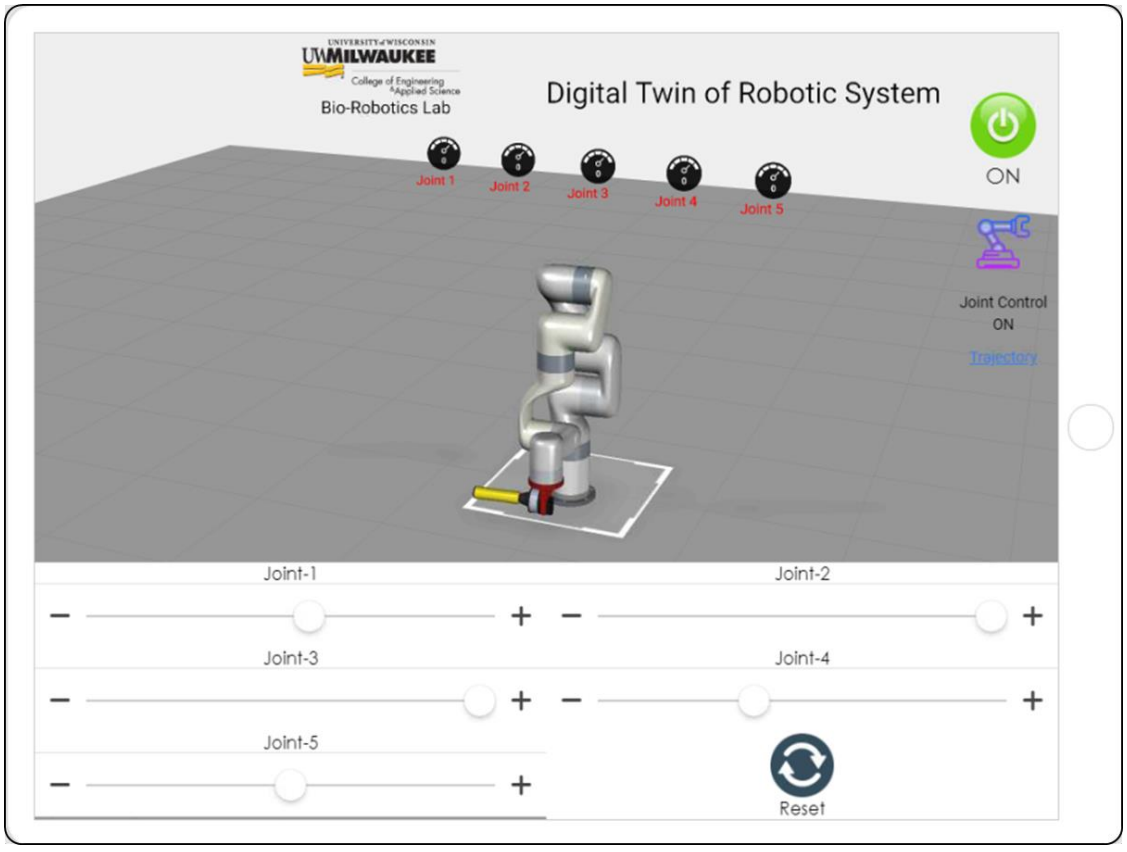


Figure 7.85 Augmented reality user interface for IO^3RT^2RE mode of telerehabilitation exercises.

We performed this IO^3RT^2RE mode with two different participants. This mode provides multi-joint PRE for the upper limb's joints. As shown in Fig. 7.63, participants sat on the chair at a distance of 0.50 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during IO^3RT^2RE mode using individual joint control of the robot to provide PRE.

For participant-1, all the robot joints are reset at zero angles. Figures 7.86 through 7.89 show the recorded experiment data for the multi-joint PRE. As shown in Figure 7.86, during the PRE, the robot's joints are moved by the operator as follows after the first 10 seconds:

- Joint-1 only moves its position to 15^o and stays at 15^o for around 13 sec, and then it returns to 0^o;
- Joint-2 moves its position to -10^o, -25^o and stays at -25^o for about 16 sec, and then it returns to 0^o;
- Joint-3 moves its position to -15^o and stays at -15^o for approximately 14 sec, and then it moves to -35^o and returns to 0^o;
- Joint-4 only moves its position to 20° and stays at 16 sec, and it returns to 0°, and
- Joint-5 stays in its initial position (0^o).

Figure 7.87 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint PRE. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.88 and 7.89, respectively.

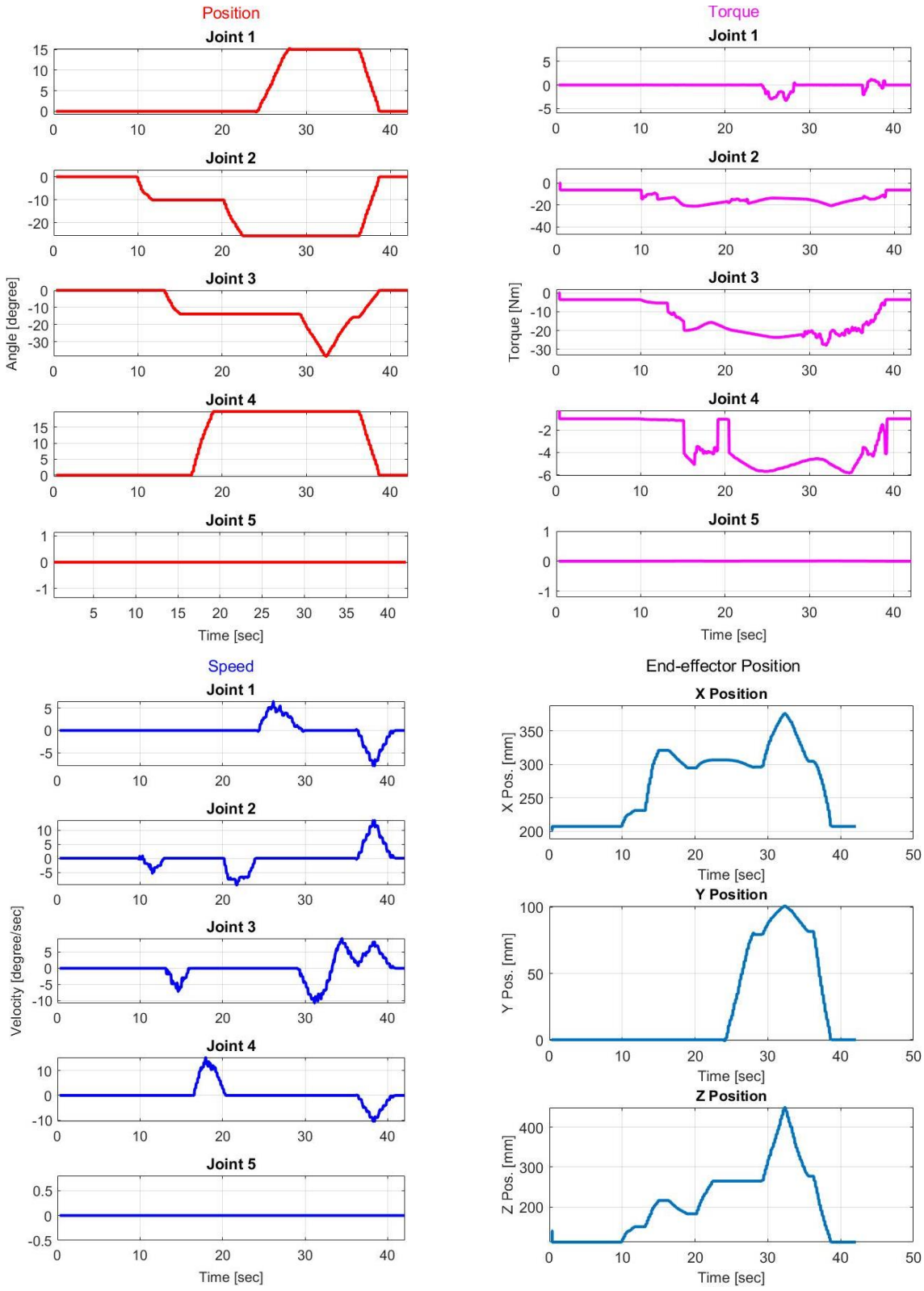


Figure 7.86 Joint angles, torques, speed, and end-effector position during IO^3RT^2RE mode using individual joint control of the robot to provide PRE.

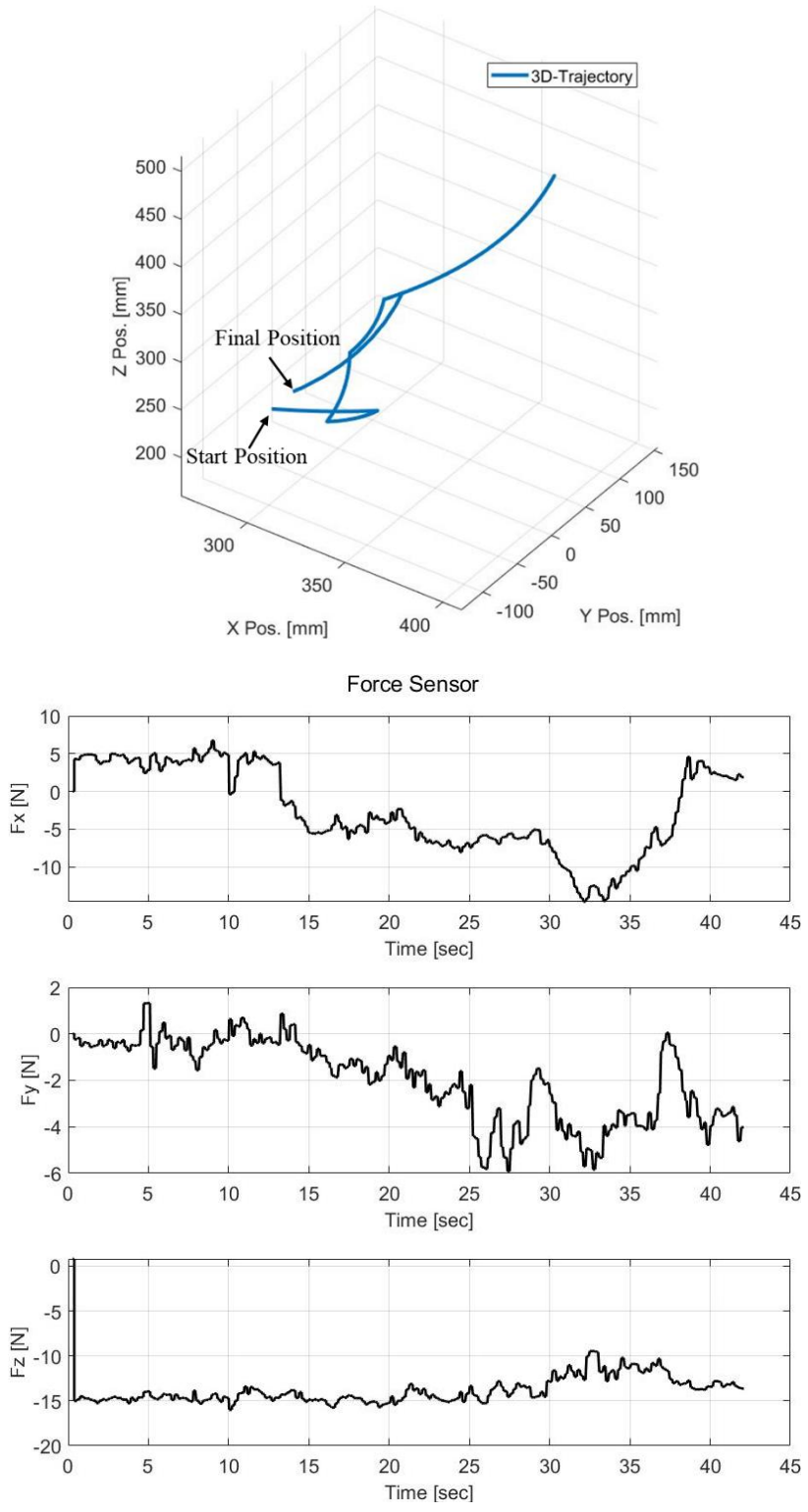


Figure 7.87 Human-robot interactive force detected from the force sensor at the end-effector during IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE.

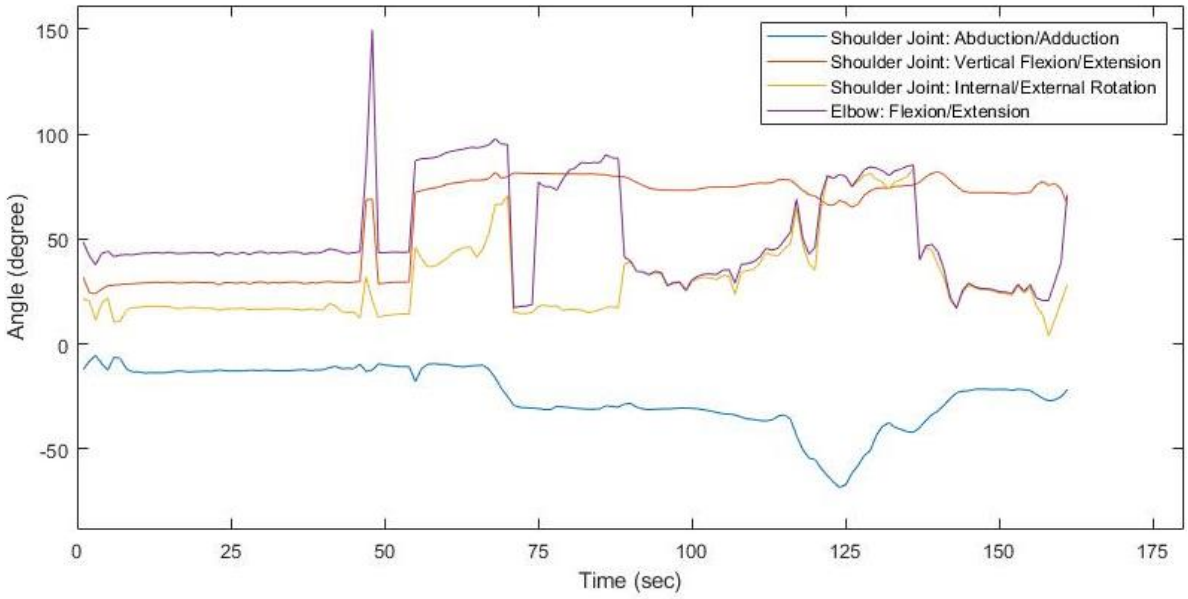


Figure 7.88 Upper-limb joint coordinate from Kinect sensor during IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE.

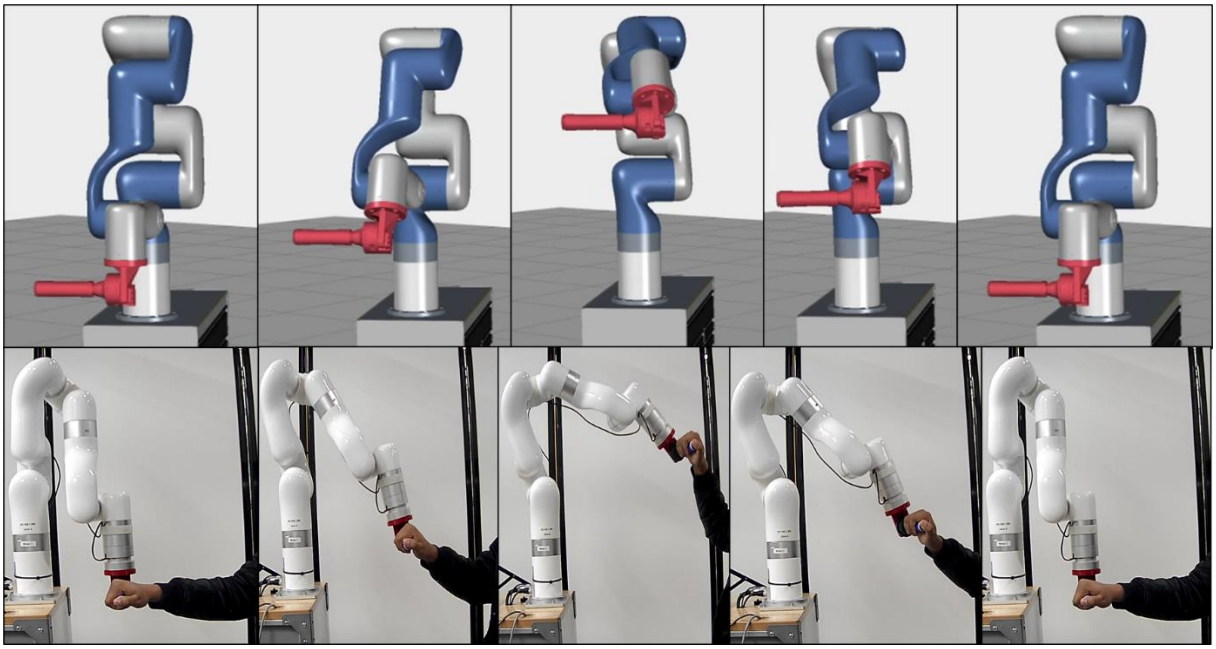


Figure 7.89 Monitoring IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE through Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

For participant-2, all the robot joints are reset at zero angles. Figures 7.90 through 7.91 show the recorded experiment data for the multi-joint PRE. As shown in Figure 7.90, during the PRE, the robot's joints are moved by the operator as follows after the first 20 seconds:

- Joint-1 only moves its position to -15^{o} and stays at -15^{o} for the rest of the trajectory;
- Joint-2 moves its position to -20^{o} and stays at -20^{o} for about 25 sec, and then it ends at -10^{o};
- Similar to Joint-1, Joint-3 moves its position to -25^{o} and stays at -25^{o};
- Joint-4 and Joint-5 move their positions to 50^{o} and -35^{o} respectively and stays there for the rest of the trajectory

Figure 7.91 shows the robot's end-effector position and the human-robot interactive force (collected from the force sensor at the end-effector) during the multi-joint PRE. Kinect sensor's data to examine human upper-limb and IIoT-based monitoring robot moments on AR robot are plotted in Fig. 7.92 and 7.93, respectively.

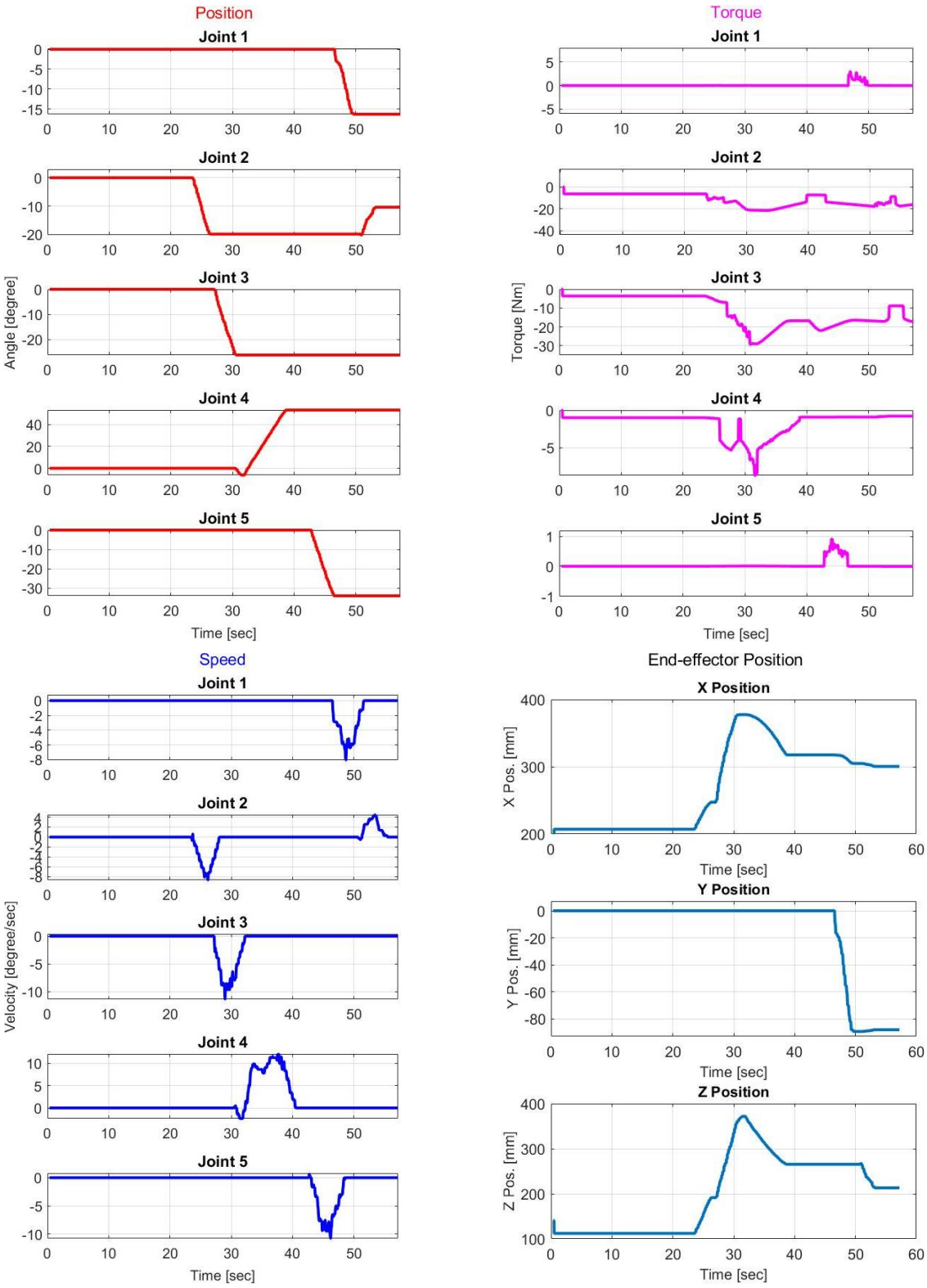


Figure 7.90 Joint angles, torques, speed, and end-effector position during IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE.

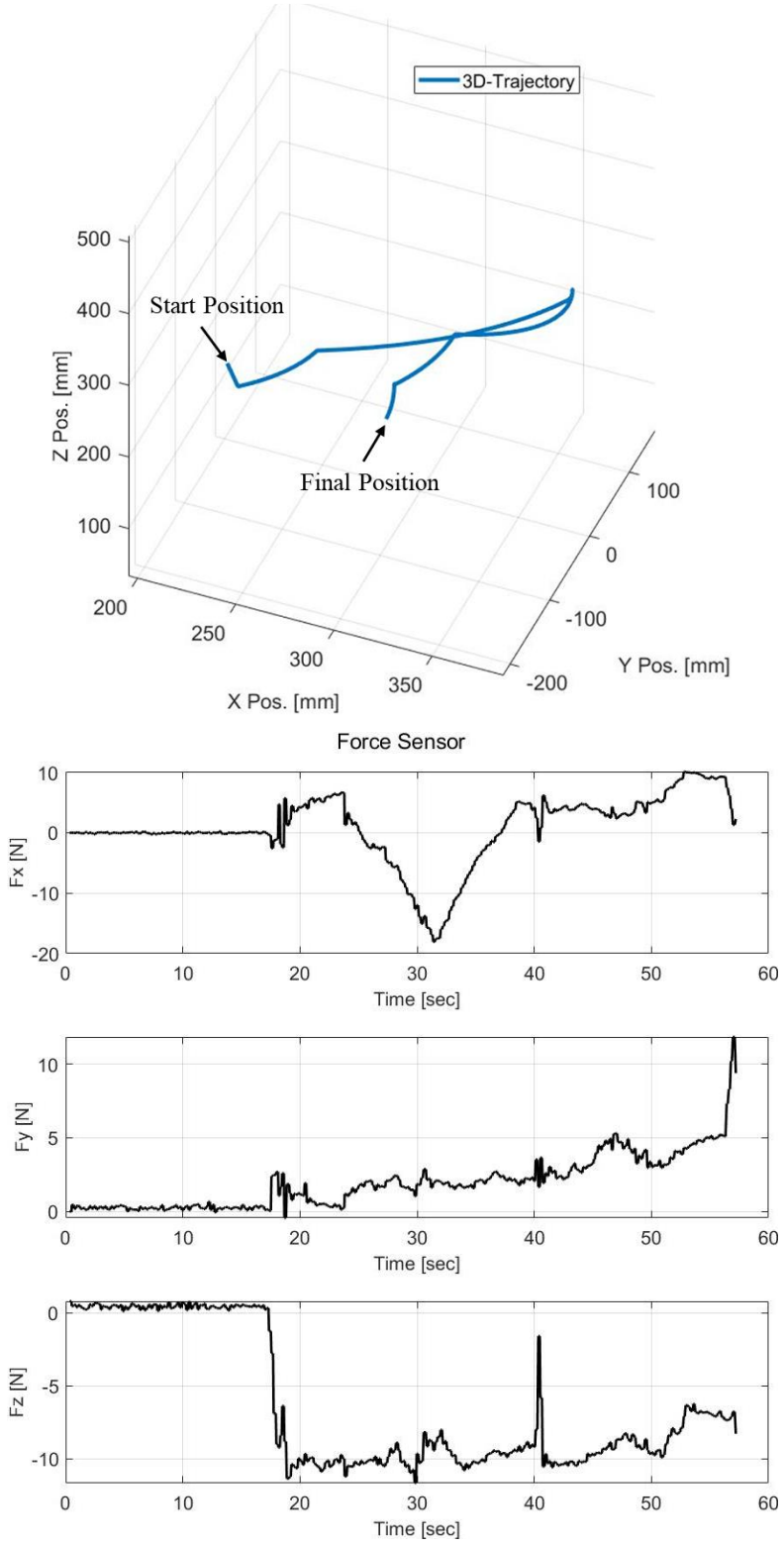


Figure 7.91 Human-robot interactive force detected from the force sensor at the end-effector during IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE.

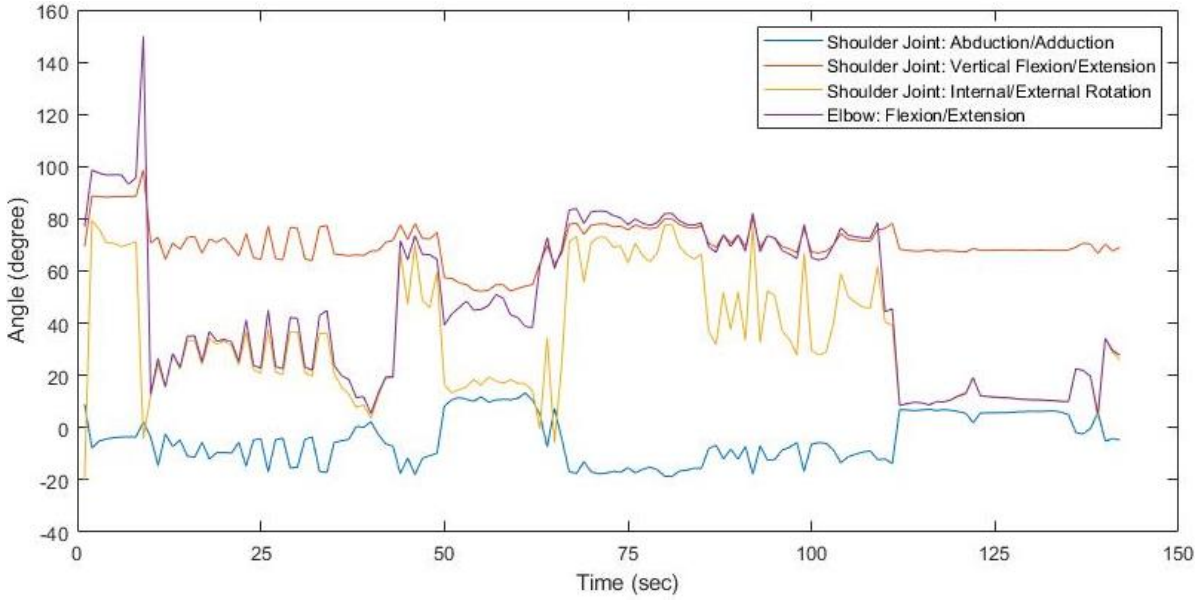


Figure 7.92 Upper-limb joint coordinate from Kinect sensor during IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE.

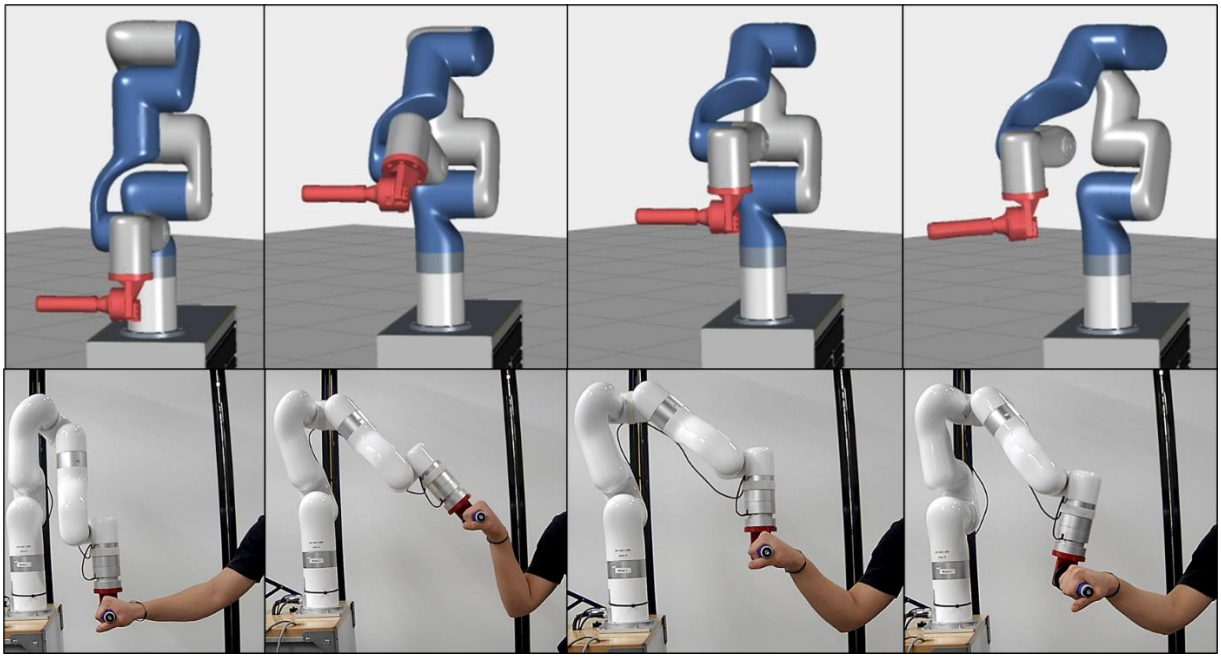


Figure 7.93 Monitoring IO^{3}RT^{2}RE mode using individual joint control of the robot to provide PRE through Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

7.2.6 Interactive One-on-One Real-Time TeleRehabilitation Exercise (IO^3RT^2RE) Mode, Controlling Robot's End-Effector Position using a Joystick to Provide PRE

This IO^3RT^2RE mode uses a Logitech joystick to control the end-effector of the xArm-5 robot. The control scheme to operate the xArm-5 robot remotely using a Logitech joystick is explained in Section 6.6.

To provide this exercise, the operator uses the joystick to remotely control the robot's end-effector position at a certain level based on the subject's upper-limb range of motion and observe the changes in the robot's (i.e., participant's) end-effector position on the AR digital twin. Note that the limiting values of X, Y, and Z positions are determined by the human-robot interactive force (HRIF). High HRIF (detected from the force sensor) indicates that the subject is resisting the motion. A threshold value of HRIF needs to be defined in future research work, which can be estimated based on the patient's pain-free range of motion. The exercises will then be monitored on AR digital twin.

As shown in Fig. 7.63, participants sat on the chair at a distance of 0.50 m from the rolling cabinet holding the robot's end-effector (handle) and facing toward the rolling cabinet during IO^3RT^2RE mode using the robot's end-effector position control using a joystick to provide PRE.

For participant-1, as shown in Fig. 7.94, the operator started moving the robot's end-effector from its initial position X = 207 mm, Y = 0 mm, and Z = 112 mm using the remote joystick. From Fig. 7.94, it can be seen that the trajectory of the robot's end-effector reached the final position (participant's pain-free range of area) at X = 325 mm, Y = 350 mm, and Z = 350 mm. Also, it can be seen from the force plots in Fig. 7.94 that the participant exerted a force during the IO^3RT^2RE. Figure 7.95 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.96 illustrates the participant-1's upper-limb joint coordinate (detected using the Kinect sensor)

corresponding to the exercises presented in Fig. 7.94. Figure 7.97 shows the AR digital twin of the xArm-5 robot and participant-1's upper-limb movement through the Microsoft Teams video session during that IO^3RT^2RE.

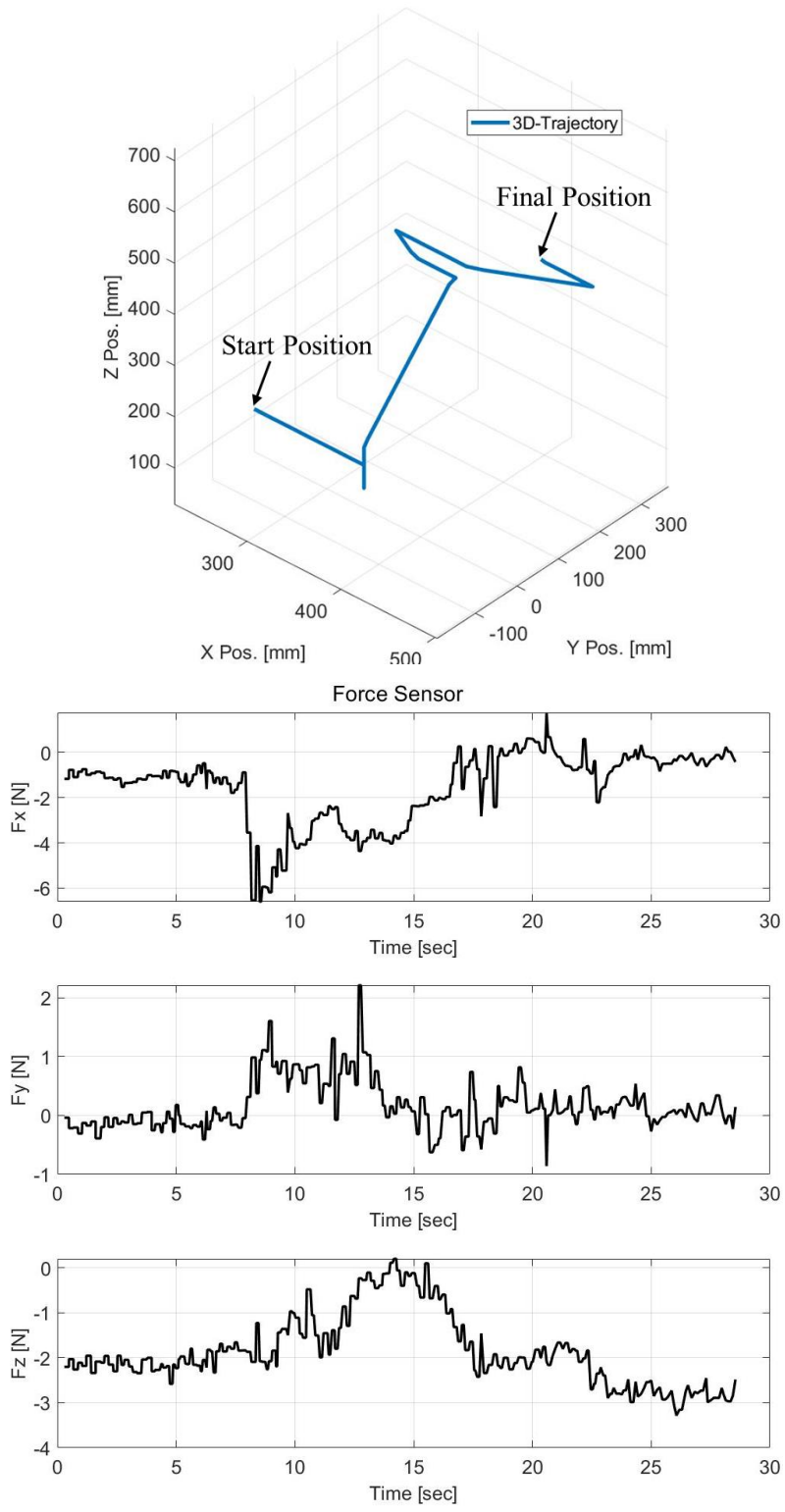


Figure 7.94 Human-robot interactive force detected from the force sensor at the end-effector during IO^{3}RT^{2}RE mode using the robot's end-effector position control using a joystick to provide PRE.

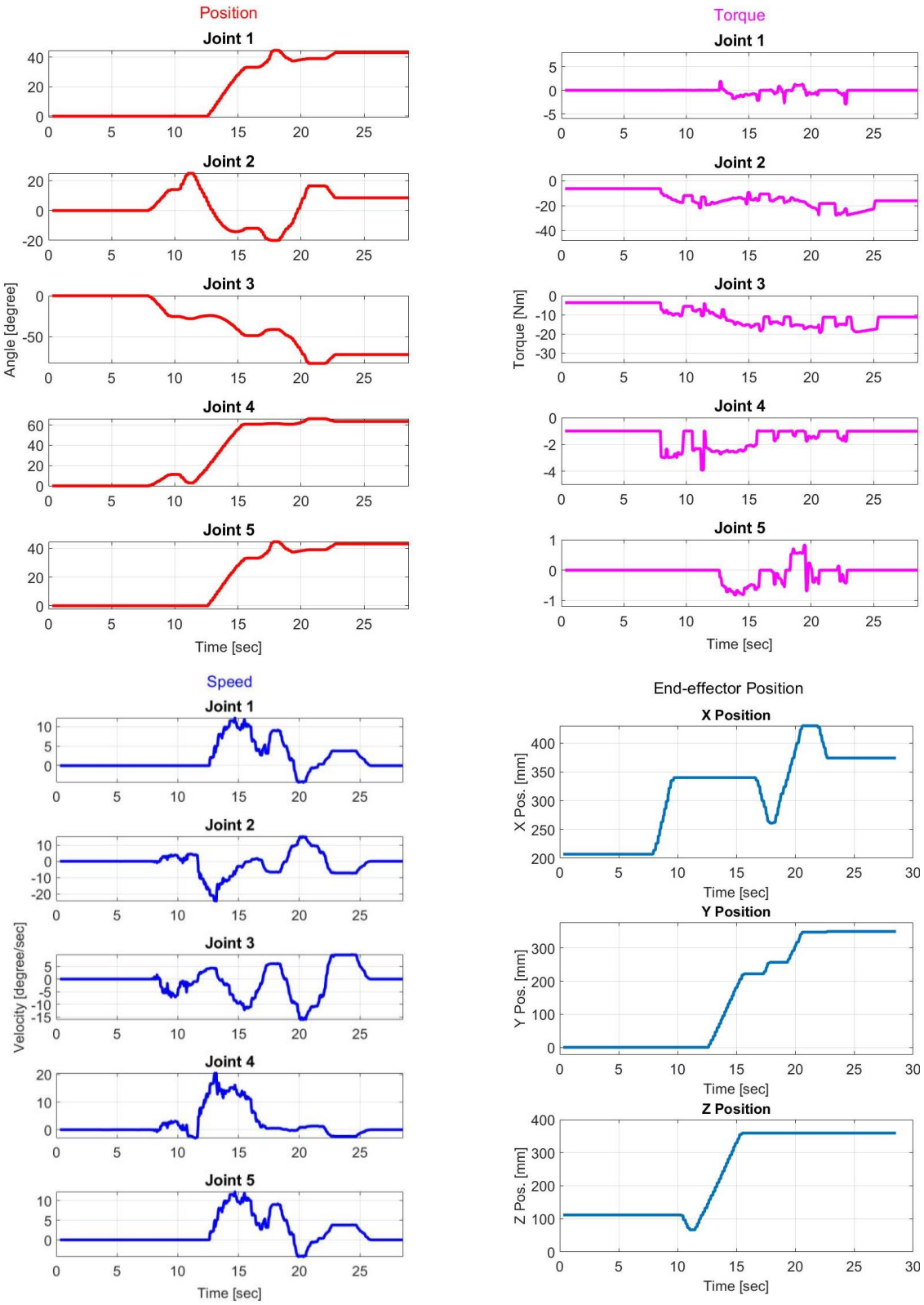


Figure 7.95 Joint angles, torques, speed, and end-effector position during IO^{3}RT^{2}RE mode using robot's end-effector position control using a joystick to provide PRE.

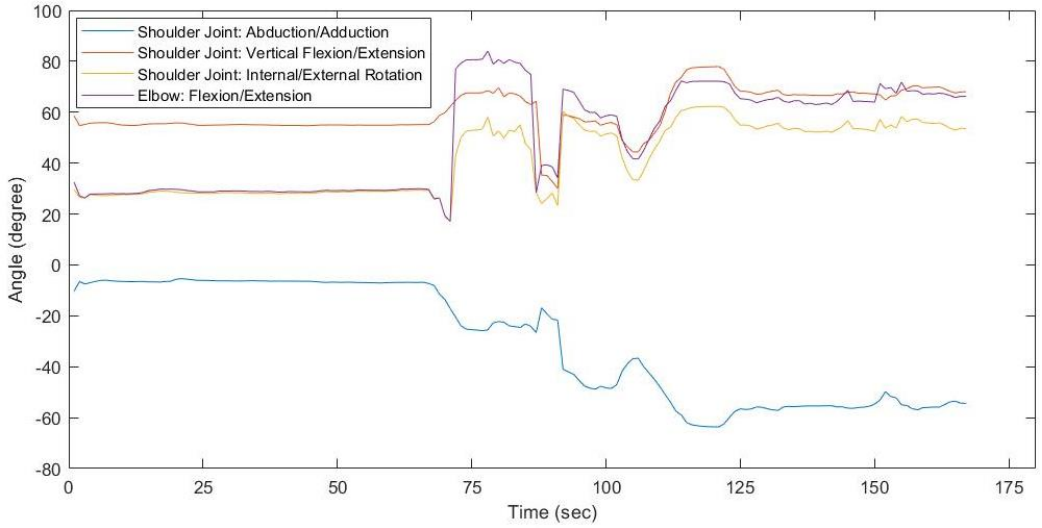


Figure 7.96 Upper-limb joint coordinate from Kinect sensor during IO³RT²RE mode using robot's end-effector position control using a joystick to provide PRE.

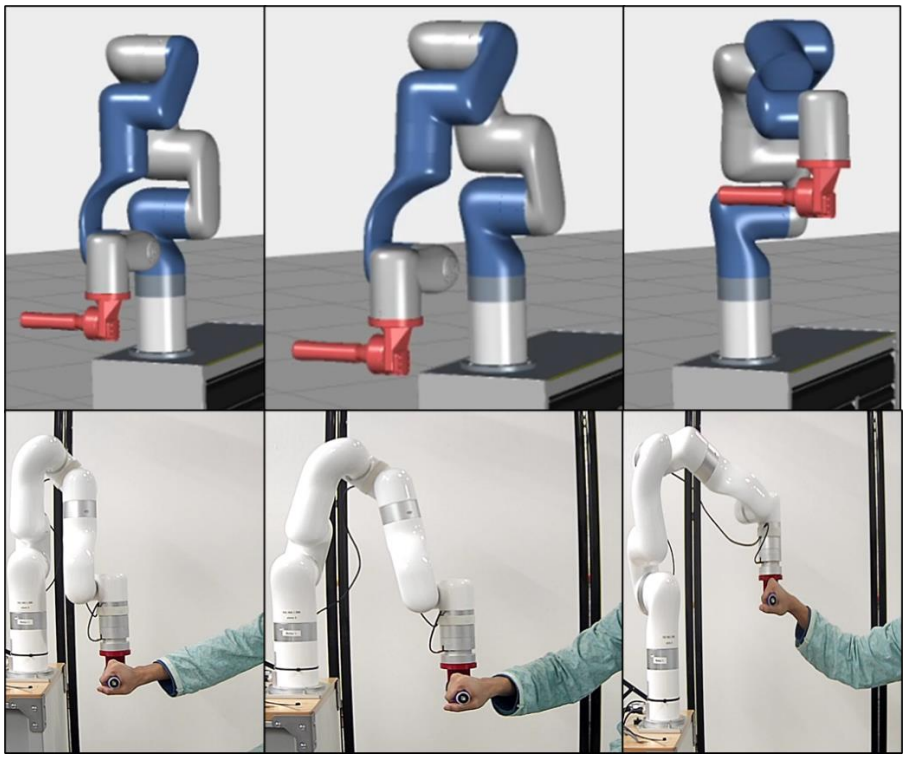


Figure 7.97 Monitoring IO³RT²RE mode using robot's end-effector position control using a joystick to provide PRE through Vuforia Studio AR platform and observe participant's upper-limb movement through Microsoft Teams video session.

For participant-2, as shown in Fig. 7.98, the operator started moving the robot's end-effector from its initial position X = 207 mm, Y = 0 mm, and Z = 112 mm using the remote joystick. From Fig. 7.98, it can be seen that the trajectory of the robot's end-effector reached the final position (participant's pain-free range of area) at X = 190 mm, Y = 250 mm, and Z = 175 mm. Also, it can be seen from the force plots in Fig. 7.98 that the participant exerted a force during the IO^3RT^2RE. Figure 7.99 shows the joint angles, speed, and torque, of the xArm-5's robot, and Fig. 7.100 illustrates the participant-2's upper-limb joint coordinate (detected using the Kinect sensor) corresponding to the exercises presented in Fig. 7.98. Figure 7.101 shows the AR digital twin of the xArm-5 robot and participant-2's upper-limb movement through the Microsoft Teams video session during that IO^3RT^2RE.

End-effector Position

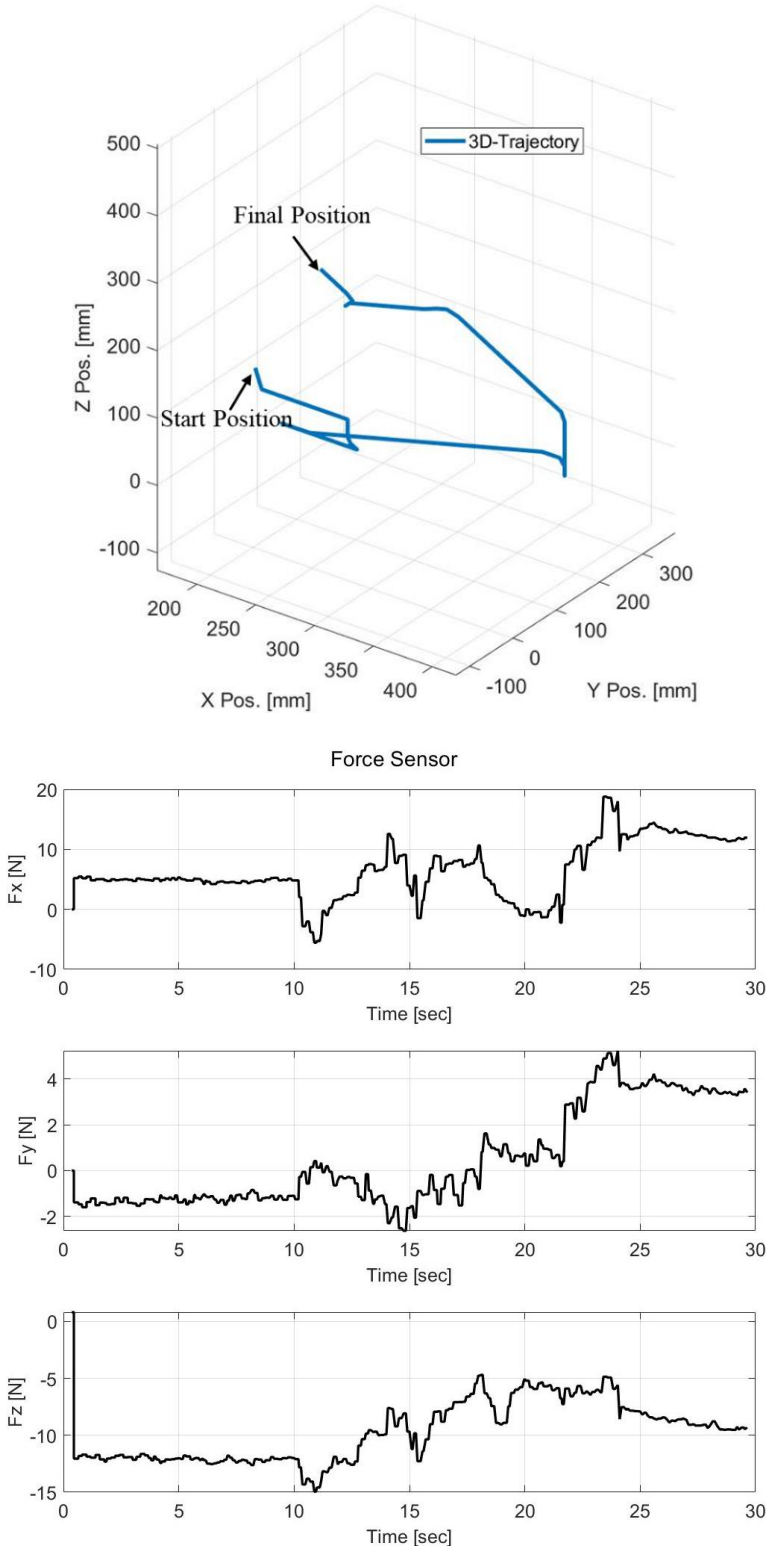


Figure 7.98 Human-robot interactive force detected from the force sensor at the end-effector during IO^{3}RT^{2}RE mode using the robot's end-effector position control using a joystick to provide PRE.

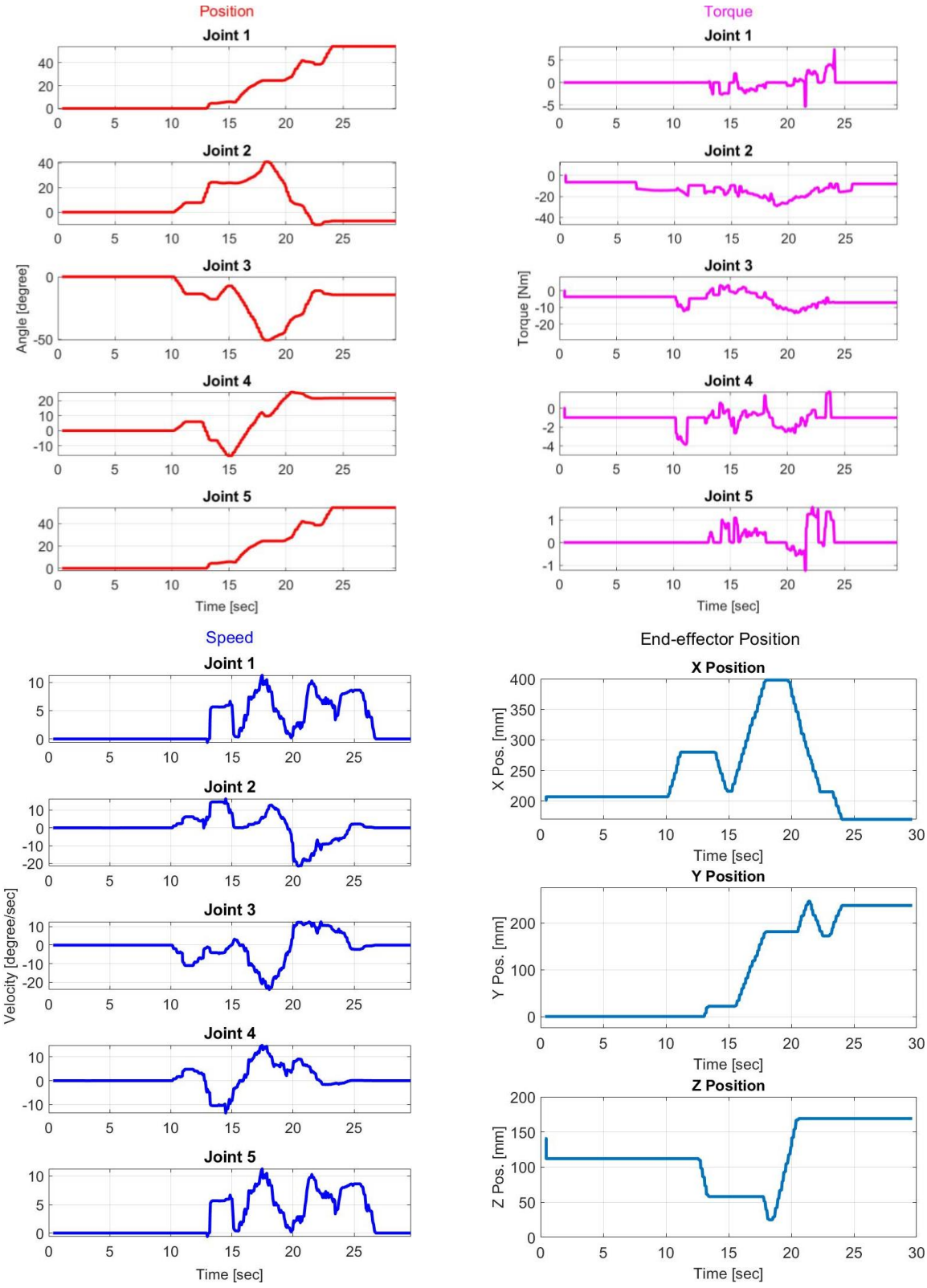


Figure 7.99 Joint angles, torques, speed, and end-effector position during IO^{3}RT^{2}RE mode using robot's end-effector position control using a joystick to provide PRE.

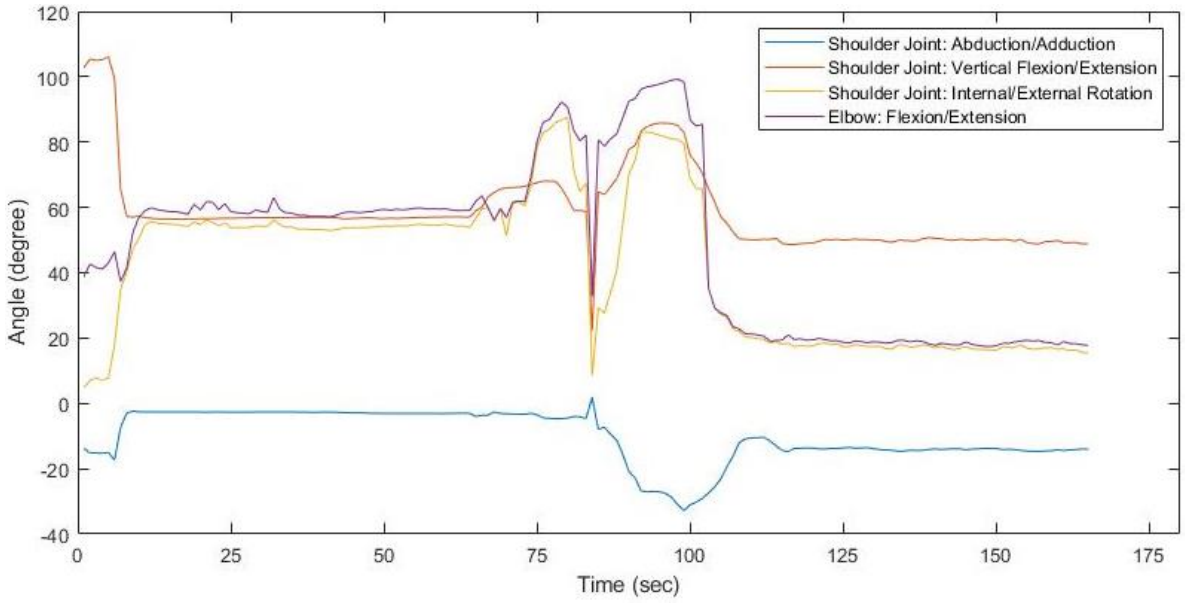


Figure 7.100 Upper-limb joint coordinate from Kinect sensor during IO³RT²RE mode using robot's end-effector position control using a joystick to provide PRE.

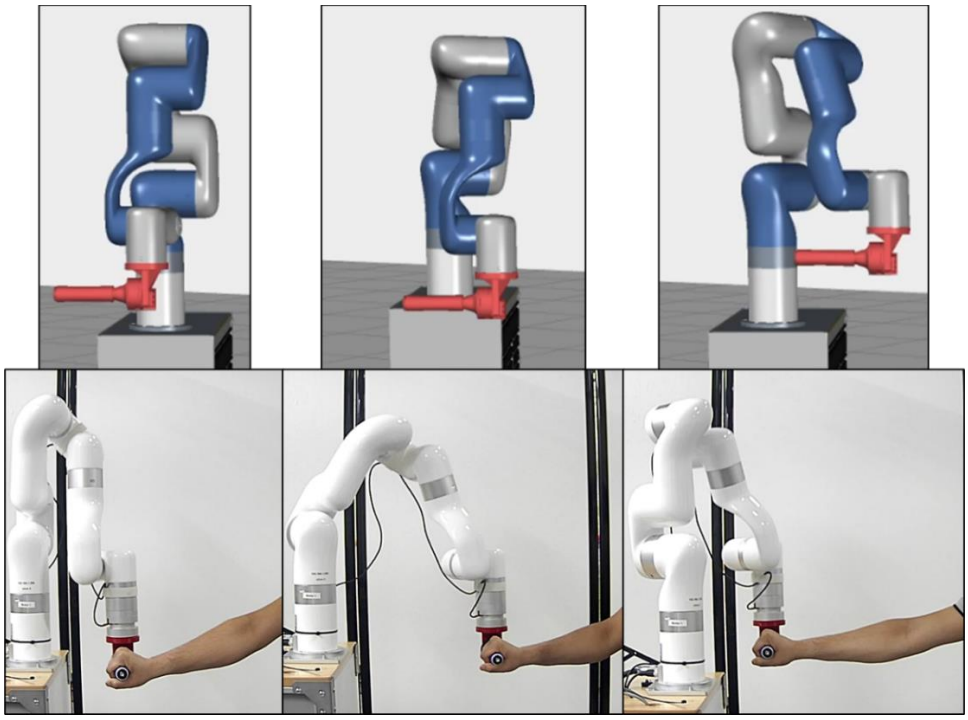


Figure 7.101 Monitoring IO³RT²RE mode using robot's end-effector position control using a joystick to provide PRE through Vuforia Studio AR platform and observing participant's upper-limb movement through Microsoft Teams video session.

Experimental results presented in Fig. 7.6 to Fig.7.101 are evident that the developed IIoT and augmented reality-based framework can be effectively used to telemanipulate a robotic manipulator for human-robot collaboration and telerehabilitation. While the proposed IIoT and augmented reality-based framework promises a stable control system for human-robot collaboration, there should be a continuous and stable connection to utilize the framework. The system becomes unstable and inoperable if the connection is lost.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this research, we develop a novel telemanipulation framework for human-robot collaboration. Several state-of-the-art technologies were utilized to develop the proposed framework, including the PTC ThingWorx IIoT platform, Vuforia Studio Augmented Reality platform, PTC ThingWorx experience services, and Vuforia View app on iPad. Experimental results validated that the developed IIoT and augmented reality-based framework for human-robot collaboration can be used for different telehealth applications such as telerehabilitation. The current methods of providing rehabilitation therapy require patients to visit a clinic/rehab center for each therapy session. A therapist can only train one patient at a time, leading to the high demand for therapists. On the other hand, the COVID-19 pandemic has affected all aspects of health care, including rehabilitation care for post-stroke patients. Telerehabilitation can address this issue. The developed telemanipulation framework for human-robot collaboration can be used to provide various telerehabilitation exercises to individuals with upper-limb impairment. Six different ways to provide telerehabilitation are developed and tested in this study. Six methods are introduced to provide passive arm movement therapy with an end-effector type robot, xArm-5. To provide resistive therapy, impedance control was used and experimentally validated.

8.2 Future Work

Future works include performing experiments with stroke participants and integrating machine learning and AI to provide patient-specific therapy based on the longitudinal data of the patients. For improving the framework, the focus will be given to stable communication and low latency real-time control and monitoring by the expert.

REFERENCES

[1] M. Burns, Z. Zavoda, R. Nataraj, K. Pochiraju, and R. Vinjamuri, "HERCULES: A Three Degree-of-Freedom Pneumatic Upper Limb Exoskeleton for Stroke Rehabilitation," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 4959-4962: IEEE.

[2] J. Bai, A. Song, B. Xu, J. Nie, and H. Li, "A novel human-robot cooperative method for upper extremity rehabilitation," *International Journal of Social Robotics*, vol. 9, no. 2, pp. 265-275, 2017.

[3] Y. Bleyenheuft and A. M. Gordon, "Precision grip in congenital and acquired hemiparesis: similarities in impairments and implications for neurorehabilitation," *Frontiers in human neuroscience*, vol. 8, p. 459, 2014.

[4] S. M. Hatem *et al.*, "Rehabilitation of motor function after stroke: a multiple systematic review focused on techniques to stimulate upper extremity recovery," *Frontiers in human neuroscience*, vol. 10, p. 442, 2016.

[5] F. Liu, X. Han, M. Lin, X. Wu, Q. Sun, and A. Song, "Remote Upper Limb Exoskeleton Rehabilitation Training System Based on Virtual Reality," in *2019 16th International Conference on Ubiquitous Robots (UR)*, 2019, pp. 323-327: IEEE.

[6] P. Langhorne, F. Coupar, and A. Pollock, "Motor recovery after stroke: a systematic review," *The Lancet Neurology*, vol. 8, no. 8, pp. 741-754, 2009.

[7] D. Novak, A. Nagle, U. Keller, and R. Riener, "Increasing motivation in robot-aided arm rehabilitation with competitive and cooperative gameplay," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, pp. 1-15, 2014.

[8] B. Brahmi, M. Saad, C. O. Luna, P. S. Archambault, and M. H. Rahman, "Passive and active rehabilitation control of human upper-limb exoskeleton robot with dynamic uncertainties," *Robotica,* vol. 36, no. 11, pp. 1757-1779, 2018.

[9] B. Brahmi, M. Saad, C. Ochoa-Luna, M. H. Rahman, and A. Brahmi, "Cartesian sliding mode control of an upper extremity exoskeleton robot for rehabilitation," in *New Developments and Advances in Robot Control*: Springer, 2019, pp. 201-220.

[10] M. H. Rahman, M. J. Rahman, O. Cristobal, M. Saad, J.-P. Kenné, and P. S. Archambault, "Development of a whole arm wearable robotic exoskeleton for rehabilitation and to assist upper limb movements," *Robotica,* vol. 33, no. 1, pp. 19-39, 2015.

[11] D. Yun *et al.*, "Handling subject arm uncertainties for upper limb rehabilitation robot using robust sliding mode control," *International Journal of Precision Engineering and Manufacturing,* vol. 17, no. 3, pp. 355-362, 2016.

[12] R. Gopura, D. Bandara, K. Kiguchi, and G. K. Mann, "Developments in hardware systems of active upper-limb exoskeleton robots: A review," *Robotics and Autonomous Systems,* vol. 75, pp. 203-220, 2016.

[13] J. Sanjuan *et al.*, "Cable driven exoskeleton for upper-limb rehabilitation: A design review," *Robotics and Autonomous Systems,* vol. 126, p. 103445, 2020.

[14] R. Gopura and K. Kiguchi, "Mechanical designs of active upper-limb exoskeleton robots: State-of-the-art and design difficulties," in *2009 IEEE International Conference on Rehabilitation Robotics,* 2009, pp. 178-187: IEEE.

[15] S. H. Lee *et al.*, "Comparisons between end-effector and exoskeleton rehabilitation robots regarding upper extremity function among chronic stroke patients with moderate-to-severe upper limb impairment," *Scientific reports*, vol. 10, no. 1, pp. 1-8, 2020.

[16] L. H. Schwamm *et al.*, "A review of the evidence for the use of telemedicine within stroke systems of care: a scientific statement from the American Heart Association/American Stroke Association," *Stroke*, vol. 40, no. 7, pp. 2616-2634, 2009.

[17] B. Nyika, "Telerehabilitation as means to improve elderlys' independence while living at home," 2013.

[18] T. Nef, M. Mihelj, and R. Riener, "ARMin: a robot for patient-cooperative arm therapy," *Medical & biological engineering & computing*, vol. 45, no. 9, pp. 887-900, 2007.

[19] (2022). *Accelerate Success With ThingWorx IIoT Solutions Platform*. Available: https://www.ptc.com/en/products/thingworx

[20] (2022). *PTC Products*. Available: https://www.ptc.com/en/products

[21] D. Kairy, P. Lehoux, C. Vincent, and M. Visintin, "A systematic review of clinical outcomes, clinical process, healthcare utilization and costs associated with telerehabilitation," *Disability and rehabilitation*, vol. 31, no. 6, pp. 427-447, 2009.

[22] A. L. Lee and R. S. Goldstein, "According to global population-based studies, the rising prevalence of COPD, particularly in those aged⩾ 65 years [1], is associated with a significant social and economic burden worldwide [2, 3]. The greatest proportion of expenditure is due to hospital costs for patients admitted with AECOPD [4, 5], with a readmission rate of 63% over 1year in those with severe disease [6]. As the severity of COPD impacts on healthcare-resource utilisation, providing early and effective healthcare management is essential to slowing disease progression, improving health-

related quality of life (HRQoL)," *Controversies in COPD: ERS Monograph,* vol. 69, p. 269, 2015.

[23] Z. Seidman *et al.,* "People attending pulmonary rehabilitation demonstrate a substantial engagement with technology and willingness to use telerehabilitation: a survey," *Journal of physiotherapy,* vol. 63, no. 3, pp. 175-181, 2017.

[24] S. L. Wolf *et al.,* "The HAAPI (Home Arm Assistance Progression Initiative) trial: a novel robotics delivery approach in stroke rehabilitation," *Neurorehabilitation and neural repair,* vol. 29, no. 10, pp. 958-968, 2015.

[25] C. O. B. Cherry *et al.,* "Expanding stroke telerehabilitation services to rural veterans: a qualitative study on patient experiences using the robotic stroke therapy delivery and monitoring system program," *Disability and Rehabilitation: Assistive Technology,* vol. 12, no. 1, pp. 21-27, 2017.

[26] J. Chen *et al.,* "Effects of home-based telesupervising rehabilitation on physical function for stroke survivors with hemiplegia: a randomized controlled trial," *American journal of physical medicine & rehabilitation,* vol. 96, no. 3, pp. 152-160, 2017.

[27] D. M. Brennan, S. Mawson, and S. Brownsell, "Telerehabilitation: enabling the remote delivery of healthcare, rehabilitation, and self management," in *Advanced technologies in rehabilitation*: IOS Press, 2009, pp. 231-248.

[28] G. Riva, "Virtual reality in neuro-psycho-physiology: Cognitive, clinical and methodological issues in assessment and rehabilitation," 1997.

[29] M. Rogante, M. Grigioni, D. Cordella, and C. Giacomozzi, "Ten years of telerehabilitation: A literature overview of technologies and clinical applications," *NeuroRehabilitation,* vol. 27, no. 4, pp. 287-304, 2010.

[30] R. V. Kenyon, J. Leigh, and E. A. Keshner, "Considerations for the future development of virtual technology as a rehabilitation tool," *Journal of neuroengineering and rehabilitation,* vol. 1, no. 1, pp. 1-10, 2004.

[31] J. R. Carey *et al.*, "Comparison of finger tracking versus simple movement training via telerehabilitation to alter hand function and cortical reorganization after stroke," *Neurorehabilitation and neural repair,* vol. 21, no. 3, pp. 216-232, 2007.

[32] B. Parmanto and A. Saptono, "Telerehabilitation: State-of-the-art from an informatics perspective," *International Journal of Telerehabilitation,* vol. 1, no. 1, p. 73, 2009.

[33] M. Zampolini *et al.*, "Tele-rehabilitation: present and future," *Annali dell'Istituto superiore di sanita,* vol. 44, no. 2, pp. 125-134, 2008.

[34] M. J. Ackerman, R. Filart, L. P. Burgess, I. Lee, and R. K. Poropatich, "Developing next-generation telehealth tools and technologies: patients, systems, and data perspectives," *Telemedicine and e-Health,* vol. 16, no. 1, pp. 93-95, 2010.

[35] E. B. Larson, M. Feigon, P. Gagliardo, and A. Y. Dvorkin, "Virtual reality and cognitive rehabilitation: a review of current outcome research," *NeuroRehabilitation,* vol. 34, no. 4, pp. 759-772, 2014.

[36] P. Weiss, M. Heldmann, A. Gabrecht, A. Schweikard, T. M. Münte, and E. Maehle, "A low cost tele-rehabilitation device for training of wrist and finger functions after stroke," in *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, 2014, pp. 422-425.

[37] A. Radman, W. Ismail, and M. Bahari, "Robotic devices for upper limb stroke rehabilitation: Potential research trends," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2017, pp. 383-388: IEEE.

[38] J. Chen, D. Hu, W. Sun, X. Tu, and J. He, "A novel telerehabilitation system based on bilateral upper limb exoskeleton robot," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019, pp. 391-396: IEEE.

[39] S. Zhang, S. Guo, B. Gao, H. Hirata, and H. Ishihara, "Design of a novel telerehabilitation system with a force-sensing mechanism," *Sensors,* vol. 15, no. 5, pp. 11511-11527, 2015.

[40] D. Kairy *et al.*, "Maximizing post-stroke upper limb rehabilitation using a novel telerehabilitation interactive virtual reality system in the patient's home: study protocol of a randomized clinical trial," *Contemporary clinical trials,* vol. 47, pp. 49-53, 2016.

[41] H. Tchero, M. Tabue-Teguo, A. Lannuzel, and E. Rusch, "Telerehabilitation for stroke survivors: systematic review and meta-analysis," *Journal of medical Internet research,* vol. 20, no. 10, p. e10867, 2018.

[42] E. Appleby, S. T. Gill, L. K. Hayes, T. L. Walker, M. Walsh, and S. Kumar, "Effectiveness of telerehabilitation in the management of adults with stroke: A systematic review," *PloS one,* vol. 14, no. 11, p. e0225150, 2019.

[43] C. Camden, G. Pratte, F. Fallon, M. Couture, J. Berbari, and M. Tousignant, "Diversity of practices in telerehabilitation for children with disabilities and effective intervention characteristics: results from a systematic review," *Disability and Rehabilitation,* vol. 42, no. 24, pp. 3424-3436, 2020.

[44] P. Seron *et al.*, "Effectiveness of telerehabilitation in physical therapy: A rapid overview," *Physical therapy,* vol. 101, no. 6, p. pzab053, 2021.

[45] P. P. Modi, A. M. Kosaraju, and M. H. Rahman, "Development of Novel IIoT Based Framework for Teleoperation of Collaborative Robot."

[46] W. H. O. Who. (2021). *Rehabilitation*. Available: https://www.who.int/news-room/fact-sheets/detail/rehabilitation

[47] C. t. W. projects. (2021). *Physical medicine and rehabilitation - Wikipedia*. Available: https://en.wikipedia.org/w/index.php?title=Physical_medicine_and_rehabilitation&oldid=1059549987

[48] C. t. W. projects. (2022). *Neuroplasticity - Wikipedia*. Available: https://en.wikipedia.org/w/index.php?title=Neuroplasticity&oldid=1084303126

[49] P. Voss, M. E. Thomas, J. M. Cisneros-Franco, and É. de Villers-Sidani, "Dynamic brains and the changing rules of neuroplasticity: implications for learning and recovery," *Frontiers in psychology*, vol. 8, p. 1657, 2017.

[50] J. A. Kleim and T. A. Jones, "Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage," 2008.

[51] M. Maier, B. R. Ballester, and P. F. Verschure, "Principles of neurorehabilitation after stroke based on motor learning and brain plasticity mechanisms," *Frontiers in systems neuroscience*, vol. 13, p. 74, 2019.

[52] T. D. Lee, *Motor control in everyday actions*. Human Kinetics, 2011.

[53] F. B. Horak, "Assumptions underlying motor control for neurologic rehabilitation," in *Contemporary management of motor control problems: Proceedings of the II STEP conference*, 1991, pp. 11-28: Foundation for Physical Therapy Alexandria, Va.

[54] G. Wulf and W. Prinz, "Directing attention to movement effects enhances learning: A review," *Psychonomic bulletin & review*, vol. 8, no. 4, pp. 648-660, 2001.

[55] C.-y. Wu, C. A. Trombly, K.-c. Lin, and L. Tickle-Degnen, "A kinematic study of contextual effects on reaching performance in persons with and without stroke:

influences of object availability," *Archives of physical medicine and rehabilitation,* vol. 81, no. 1, pp. 95-101, 2000.

[56] G. M. Lage, H. Ugrinowitsch, T. Apolinário-Souza, M. M. Vieira, M. R. Albuquerque, and R. N. Benda, "Repetition and variation in motor practice: a review of neural correlates," *Neuroscience & Biobehavioral Reviews,* vol. 57, pp. 132-141, 2015.

[57] P. Redgrave and K. Gurney, "The short-latency dopamine signal: a role in discovering novel actions?," *Nature reviews neuroscience,* vol. 7, no. 12, pp. 967-975, 2006.

[58] M. A. Guadagnoli and T. D. Lee, "Challenge point: a framework for conceptualizing the effects of various practice conditions in motor learning," *Journal of motor behavior,* vol. 36, no. 2, pp. 212-224, 2004.

[59] M. Andrieux, J. Danna, and B. Thon, "Self-control of task difficulty during training enhances motor learning of a complex coincidence-anticipation task," *Research quarterly for exercise and sport,* vol. 83, no. 1, pp. 27-35, 2012.

[60] G. H. Gendolla, "Self-relevance of performance, task difficulty, and task engagement assessed as cardiovascular response," *Motivation and Emotion,* vol. 23, no. 1, pp. 45-66, 1999.

[61] M. A. Meredith and B. E. Stein, "Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration," *Journal of neurophysiology,* vol. 56, no. 3, pp. 640-662, 1986.

[62] E. Làdavas, "Multisensory-based Approach to the recovery of unisensory deficit," *Annals of the New York Academy of Sciences,* vol. 1124, no. 1, pp. 98-110, 2008.

[63] J. M. Ross and R. Balasubramaniam, "Physical and neural entrainment to rhythm: human sensorimotor coordination across tasks and effector systems," *Frontiers in human neuroscience*, vol. 8, p. 576, 2014.

[64] S. Rossignol and G. M. Jones, "Audio-spinal influence in man studied by the H-reflex and its possible role on rhythmic movements synchronized to sound," *Electroencephalography and clinical neurophysiology*, vol. 41, no. 1, pp. 83-92, 1976.

[65] C. Nombela, L. E. Hughes, A. M. Owen, and J. A. Grahn, "Into the groove: can rhythm influence Parkinson's disease?," *Neuroscience & Biobehavioral Reviews*, vol. 37, no. 10, pp. 2564-2570, 2013.

[66] S. Ghai, "Effects of real-time (sonification) and rhythmic auditory stimuli on recovering arm function post stroke: a systematic review and meta-analysis," *Frontiers in neurology*, vol. 9, p. 488, 2018.

[67] A. W. Salmoni, R. A. Schmidt, and C. B. Walter, "Knowledge of results and motor learning: a review and critical reappraisal," *Psychological bulletin*, vol. 95, no. 3, p. 355, 1984.

[68] C. J. Winstein, "Knowledge of results and motor learning—implications for physical therapy," *Physical therapy*, vol. 71, no. 2, pp. 140-149, 1991.

[69] B. I. Molier, E. H. Van Asseldonk, H. J. Hermens, and M. J. Jannink, "Nature, timing, frequency and type of augmented feedback; does it influence motor relearning of the hemiparetic arm after stroke? A systematic review," *Disability and rehabilitation*, vol. 32, no. 22, pp. 1799-1809, 2010.

[70] (2022). *UFACTORY xArm 5 Lite*. Available: https://www.ufactory.cc/product-page/ufactory-xarm-5-lite

[71] (2022). *xArm Robot Download | UFACTORY*. Available: https://www.ufactory.cc/download-xarm-robot

[72] (2022). *Kinematic and Dynamic Parameters of UFACTORY xArm Series*. Available: http://help.ufactory.cc/en/articles/4330809-kinematic-and-dynamic-parameters-of-ufactory-xarm-series

[73] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.

[74] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[75] R. Hartenberg and J. Danavit, *Kinematic synthesis of linkages*. New York: McGraw-Hill, 1964.

[76] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[77] (2022). *Vuforia Studio Help*. Available: https://support.ptc.com/help/vuforia/studio/en/index.html#page/Studio_Help_Center/common/Security_Architecture.html

[78] (2022). *What is a Digital Thread?* Available: http://support.ptc.com/help/vuforia/studio/en/index.html#page/Studio_Help_Center%2Fdigital_twin%2Fdigital_twin_101_digital_thread.html%23

[79] (2022). *What Do Digital Twins Enable?* Available: http://support.ptc.com/help/vuforia/studio/en/index.html#page/Studio_Help_Center%2Fdigital_twin%2Fdigital_twin_101_digital_twin_advantages.html%23wwID0EW2JU.

[80] (2022). *Home*. Available: https://www.ptc.com

[81] (2022). *Get Familiar with Vuforia Studio | PTC*. Available: https://www.ptc.com/en/success-paths/develop-first-vuforia-studio-experience/plan/get-familiar-with-vuforia-studio

[82] A. Josef, "Augmented Reality approach for industrial process data monitoring using Microsoft HoloLens," University of Applied Sciences Technikum Wien, 2019.

[83] (2022). *Welcome to ThingWorx 8*. Available: https://support.ptc.com/help/thingworx_hc/thingworx_8_hc/en/index.html#

[84] (2019). *HINGWORX INDUSTRIAL INNOVATION PLATFORM*. Available: https://www.isax.com/files/2019/04/ThingWorx-Industrial-Innovation-Platform-Presentation.pdf

[85] (2022). *Use REST API to Access ThingWorx | Developer Portal : ThingWorx*. Available: https://developer.thingworx.com/en/resources/guides/thingworx-rest-api-quickstart

[86] (2022). *ThingWorx REST API*. Available: https://support.ptc.com/help/thingworx_hc/thingworx_8_hc/en/index.html#page/ThingWorx/Help/REST_API/ThingWorxRESTAPI.html#wwID0E2GDXB

[87] (2022). *Welcome to the ThingWorx Edge Java SDK Help Center*. Available: https://support.ptc.com/help/thingworx/edge_sdk_java/en/#

[88] (2022). *Vuforia View*. Available: https://apps.apple.com/us/app/vuforia-view/id1076700285?platform=ipad

[89] (2022). *Vuforia View | Apps | 148Apps*. Available: https://www.148apps.com/app/1076700285

[90] (2022). *Extreme 3D Pro Joystick*. Available: https://www.logitechg.com/en-us/products/space/extreme-3d-pro-joystick.963290-0403.html

[91] (2022). *Is the shoulder stiff?* Available: http://shoulderarthritis.blogspot.com/2011/03/is-shoulder-stiff.html

[92] Y. Yang, L. Wang, J. Tong, and L. Zhang, "Arm rehabilitation robot impedance control and experimentation," in *2006 IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 914-918: IEEE.

APPENDIX – A

Vuforia view ThingMark

APPENDIX – B

REST API Python code

```
from sys import stdout
import pygame
import time
import requests
import json
import random
from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
thing = "Logitech_joystick_1.1"
url = "https://pp-2007231852eh.portal.ptc.io:8443/Thingworx"
appKey = "fbc7afc6-9ac4-4a5e-b577-e9f66ed2310b"
headers = { 'Content-Type': 'application/json', 'appKey': appKey }
# Initialize pygame for joystick support
pygame.display.init()
pygame.joystick.init()
controller = pygame.joystick.Joystick(0)
controller.init()
while True:
    # Get next pygame event
    pygame.event.pump()
    stdout.write(controller.get_name())
    ######### All Axis #################
    YAxisOut=round(controller.get_axis(0),1) #Y = axis 0
    XAxisOut=round(-controller.get_axis(1),1) #x = axis 1
    TwistAxisOut=controller.get_axis(2) #Twist = axis 2
    ThrottleAxisOut=controller.get_axis(3) #Throttle = axis 3
```

```
########## All Buttons ##############
Button1=controller.get_button(0) # trigger
Button2=controller.get_button(1) # Button-2
Button3=controller.get_button(2) # Button-3
Button4=controller.get_button(3) # Button-4
Button5=controller.get_button(4) # Button-5
Button6=controller.get_button(5) # Button-6
Button7=controller.get_button(6) # Button-7
Button8=controller.get_button(7) # Button-8
Button9=controller.get_button(8) # Button-9
Button10=controller.get_button(9) # Button-10
Button11=controller.get_button(10) # Button-11
payload={
    'Axis_X' : XAxisOut,
    'Axis_Y' : YAxisOut,
    'Twist' : TwistAxisOut,
    'Throttle' : ThrottleAxisOut,
    'Trigger': Button1,
    'Button_2': Button2,
    'Button_3': Button3,
    'Button_4': Button4,
    'Button_5': Button5,
    'Button_6': Button6,
    'Button_7': Button7,
    'Button_8': Button8,
    'Button_9': Button9,
    'Button_10': Button10,
    'Button_11': Button11,
```

```
    }

    response = requests.put(url + '/Things/' + thing + '/Properties/*', headers=headers,
json=payload, verify=False)

    print(response)
```