Robotics and Autonomous Systems 197 (2026) 105305

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

An adaptable framework for remotely controlling a telepresence robot in a dynamic environment

Barnali Das [a,*], Gordon Dobie [b]

[a] School of Computer Science and Engineering, Faculty of Business and Technology, University of Sunderland, Sunderland, United Kingdom
[b] Centre for Ultrasonic Engineering, Electronic and Electrical Engineering, University of Strathclyde, Glasgow, United Kingdom

ARTICLE INFO

Keywords:
Telepresence system
state estimation
time delay
filter theory
predictive display
robot simulator

ABSTRACT

This paper presents a framework for telepresence robot navigation in dynamic environments with network-induced time delays. The proposed system introduces a predictive control model that processes sensor data, implements real-time control algorithms, and transmits commands to enable robust remote navigation. To address visual and control discrepancies caused by latency, a state estimation model is employed to minimise the visual disparity between the robot's actual and perceived positions. Additionally, a simulation-based predictive controller anticipates operator commands to improve teleoperation accuracy. A key contribution of this work is the development of a low-cost, simulation-based telepresence platform that enables controlled experiments without relying on expensive physical infrastructure. The system is designed for flexibility, allowing parameter adjustments to suit a range of experimental conditions. By integrating predictive technologies and addressing latency-related challenges, this research advances the state-of-the-art in telepresence robotics and provides a practical, reproducible foundation for future studies in remote robot navigation.

1. Introduction

Telepresence systems in telerobotics involve controlling robots from a distance, typically using wireless connections. In this paper, we present a framework that utilises established control algorithms and offers a user-friendly interface for human operators who may not be familiar with robot control theory. The framework outlined here is designed for robot manipulation tasks that involve long distances and time delays.

Telepresence refers to a set of technologies that allow a person to feel as if they were present in a location other than their true location or to have an effect at that location. The telepresence system enables a human operator to control and navigate a mobile robot in a remote environment, often interacting with audiences through video conferencing [1]. The system typically consists of a local site where a human operator uses a hand-controller device, a remote site where a mobile robot interacts with the physical world, and a communication channel that connects both sites. It provides interactive two-way audio and video communication, as well as physical manipulation, allowing communication between two people in different places.

Telepresence robots encounter significant challenges when navigating in remote sites due to varying communication time delays [2] often caused by network conditions. Additionally, the distance between the human operator and the remote sites of the telepresence system introduces time-varying delays, which can distort reference commands, response time, and feedback signals, leading to system instability or poor performance. The time lapse between making a decision and perceiving its consequences introduces uncertain time delays, resulting in a mismatch between the received robot's state at the operator's side and the actual robot's state in the remote environment. This mismatch negatively impacts the human operator's performance. Compensating for these time delays is advantageous for the robust navigation and manipulation of a telepresence robot, as demonstrated by Das and Dobie [3].

In the literature, various approaches are used to deal with the time delay in telepresence systems. These approaches include increasing levels of automation, adding more sensors to the robot, and using predictive technology [4]. Das and Dobie [3,5] developed a state estimation algorithm to predict the robot's current state by using the robot's delayed feedback information and the timing of the operator's command transmissions. In the proposed research, we present a graphical representation of the real-life scenario of a telepresence system along with the state estimation model.

In order to address the gap, we propose a framework that is designed to benefit two specific groups of people. The first group includes robotics researchers or industry manufacturers who can use the experimental framework for controlled experiments. The framework allows

* Corresponding author.
E-mail address: barnali.das@sunderland.ac.uk (B. Das).

https://doi.org/10.1016/j.robot.2025.105305
Received 26 November 2024; Received in revised form 13 October 2025; Accepted 15 December 2025
Available online 18 December 2025

for the conversion of a real-life telepresence system into a graphical environment, which can be customised by adjusting the model parameters and control algorithm to fit personalised experiment scenarios. When utilising this framework, a human operator is responsible for providing all the parameters and control commands to design their experimental environment. For this group, the aim is to reduce development time, minimise the need for robotics experts, and facilitate the construction and validation of advanced robotics applications before real-world execution or manufacturing. Modifying or incorporating a commercial robot into a defined experimental scenario is extremely challenging for research purposes. Robot companies provide very limited design information due to their company policy, which restricts the choice of commercial robots for research purposes. Previous research has been based on sensor-captured data, but there has been very limited research on its control mechanism.

The second group consists of operators who need a user-friendly mechanism to remotely manoeuvre telepresence robots in known and unknown dynamic environments. This is especially true when a human operator controls a robot in a crowded area, may be busy with other work, or is in a rush. Our proposed framework is primarily designed for application in areas where human operators are predominantly academics delivering lectures or attending meetings, healthcare professionals tending to patients, or in a corporate setting where a senior employee is visiting an overseas company or participating in a group discussion. In such scenarios, we assume that human operators may not be very familiar with controlling a robot in a remote location and may not be able to manoeuvre the robot in a crowded space effectively.

There is another reason for developing a simulation-based framework. Due to COVID-19 restrictions, access to the University lab and hardware was limited, which halted the development of the framework in a real environment. It was too risky to perform experiments in the robotics lab because of the limited security system. At that time, in order to continue the experimental work, we needed to find an alternative route: creating a simulation environment that could replicate the risky and expensive real-life system, which we think has benefits beyond COVID-19 even today and in the future. Additionally, the adaptable nature of the framework could facilitate further experiments with different robots and scenarios by adjusting its graphical model and system control algorithms.

The proposed framework has a wide range of applications in semi-autonomous robotics, including space robotics, service robotics, social robotics, and experimental cases. It is designed to accommodate diverse use cases, from tasks requiring full autonomy over extended periods to requiring more manual human-in-the-loop control. In all these scenarios, the framework ensures robust safety measures.

The main contributions of this paper are:

1. Developing a predictive framework involves superposing real-environment robot feedback data with simulated data to compensate for uncertain time delays. This allows for the prediction of real-time robot position, displaying both real and predicted information on the same screen.
2. The framework is validated through a series of experiments using a state-of-the-art differential-drive telepresence robot in a simulation environment, considering the real robot's graphical model and the dynamic nature of the experimental scenarios.
3. The Configurable and modular design of the simulation system allows for manipulating and simulating various types of robots and environments by adjusting the robot's physical and experimental parameters. This flexibility opens up a wide range of potential application areas in the future.

While the foundational components of this system are well-established, the key contribution of this work is the integration of these elements into a cohesive, adaptable framework that facilitates delay-compensated telepresence experiments in both real and simulated environments. We will start by reviewing related work in Section 2, followed by a description of our framework and its components in Section 3. Next, we will discuss the experimental results in Section 4 and present the conclusions in Section 5.

2. Background

A telepresence system enables a person to control a robot and interact in a remote environment via a communication channel. To ensure a smooth telepresence experience, it is important to include human sensory elements like vision, sound, and remote manipulation. However, the effectiveness of the telepresence system can vary based on the level of detail, and there are challenges that affect precise and reliable robotic control for the human operator in a dynamic environment. Key factors affecting navigation include limited field of view, managing multiple cameras and screens, depth perception, low frame rate, video quality, and time delay. Additionally, the nonlinear behaviour of robot navigation presents further challenges in predicting the robot's true state in the dynamic environment.

Our research focuses on time delay or latency, which is the time gap between the human operator's input commands and the robot's feedback. Various factors influence the telepresence system's total time delay, including network switching delays, bandwidth limitations, communication drop-outs, hardware processing delays, and the mobile robot's slow dynamics. Time delays can result from physical obstacles or the distance between the operator and the robot. The total time delay can be continuous and uncertain. It is important to note that the causes of the time delay are beyond the scope of this work. We are specifically interested in the total perceived time delay from the moment the human operator sends a command until the human operator visually perceives the robot's reaction to the feedback information.

When there is a time delay, the display in front of the human operator shows incorrect information. As a result, the human operator predicts the robot's position based on the displayed information and sends control commands. However, when the human operator receives the robot's position from the sensor feedback, it does not match the predicted robot position. The two different displays cause conflicts in human perception. When a human operator controls a robot in a known or unknown environment, they rely on remote perception and remote manipulation to understand the robot's surroundings. Remote manipulation is limited by the operator's performance and human motor skills, including distance estimation, obstacle detection, environment awareness, and command generation [6]. Dealing with a virtual display that differs from the physical environment around the operator makes remote perception very challenging. Teleoperating in a known or unknown environment with time delay is extremely difficult and stressful for the human operator, often leading to mental fatigue [6,7].

One potential solution to these issues is a framework that includes a state estimation algorithm and a simulation-based predictive display. In this framework, a human operator sends control commands to the robot based on the display, while a robot state estimation algorithm runs in the background. This algorithm superposes the robot's predicted position and the robot's actual position based on delayed feedback. The combined information is immediately displayed graphically on the simulation-based predictive display, showing the robot's estimated pose without any time delay in front of the human operator. The term "predictive display" refers to visualising the robot's pose in response to the operator's commands without waiting for delayed information.

2.1. State estimation algorithm

In robot navigation, the Extended Kalman Filter (EKF) is commonly used to estimate the true state from noisy measurements. However, there is a fundamental communication time when a filtering processor is connected to a sensor through a network. Additionally, if raw sensor data requires post-processing to update the state of the dynamical system, there will be an additional post-processing time, causing a delay between the measurement acquisition and its availability to the filter. In a real environment scenario, we cannot assume the total time delay between sending a control command to the robot and the moment when the received sensor measurement data is entered in the state estimator. We assumed that the measurement time delay was uncertain. In the previous research paper, we proved that regular EKF could not handle uncertain time delays when the robot changes its direction [3].

In our research, we utilised an Augmented State Extended Kalman Filter to estimate the position of telepresence robots while accounting for uncertain delayed sensor measurements. We augmented the current and past states into multiple augmented state vectors to accurately estimate the robot's true position. The current measurement state, which includes information from past measurement states, directly adjusts the augmented state vectors. This process allows us to determine the corresponding past state in the augmented state vector within a delayed system. Subsequently, we updated the past state using delayed measurement data while simultaneously correcting the current state in the augmented state vector. It is important to note that the algorithm initially considers a specific time delay, which is then extended to compensate for uncertain delays using Probability Density Functions (PDFs). For a one-time step delay, the prediction equation was modified as

$$\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} = \begin{bmatrix} f(x_k, u_k, w_k) \\ x_k \end{bmatrix}, \tag{1}$$

where, $\begin{bmatrix} x_{k+1}^T & x_k^T \end{bmatrix}^T$ was the augmented state vector. The measurement equation was

$$z_k = h\left(\begin{bmatrix} 0 & I \end{bmatrix}\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix}, v_k\right), \tag{2}$$

where, I was the identity matrix, the current measurement z_k was used to update $\begin{bmatrix} x_{k+1}^T & x_k^T \end{bmatrix}^T$. For multi-step delays, the prediction equation is defined as

$$\begin{aligned} \mathcal{X}_{(k+1)} &= \begin{bmatrix} f(x_k, u_k) \\ \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & I & 0 \end{bmatrix} X_k \end{bmatrix} + \begin{bmatrix} w_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \\ &\equiv f(\mathcal{X}_k, \mathcal{U}_k, \mathcal{W}_k), \end{aligned} \tag{3}$$

where, $\mathcal{X}_{(k)}$ was the augmented state vector defined by $\begin{bmatrix} x_k^T & x_{k-1}^T & \cdots & x_{k-n}^T \end{bmatrix}^T$ and n was the maximum number of delayed time steps. The measurement equation was rewritten as

$$\begin{aligned} \mathcal{Z}_k &= h\left[\begin{bmatrix} 0 \\ \vdots \\ I \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} x_k \\ \vdots \\ x_{k-\tau_k} \\ \vdots \\ x_{k-n} \end{bmatrix}\right] + \begin{bmatrix} 0 \\ \vdots \\ v_{k-\tau_k} \\ \vdots \\ 0 \end{bmatrix}, \\ &\equiv h(\mathcal{X}_k, \mathcal{V}_k), \end{aligned} \tag{4}$$

In the given context, τ_k represents the time delay, which is less than n, and I is placed at the corresponding time step $k - \tau_k$. If the time delay τ_k is known or certain, the augmented state vector can be estimated recursively using the EKF algorithm. To assess delay compensation performance, we conducted experiments applying both the EKF and AS-EKF algorithms and compared the results to achieve a desirable predictive display. While MATLAB was used for proof-of-concept development due to its flexibility in algorithm design, future work will involve implementing the AS-EKF in C++ for deployment in commercial or real-time embedded systems.

2.2. Predictive display

Our research aims to develop a flexible, user-friendly framework for effective robot control tasks. This framework will pave the way for further research on robot navigation without needing real-world experimental setups. According to existing literature, using predictive displays can improve human performance in robot navigation by approximately 20% [4]. We aim to create a framework that allows for seamless control of a telepresence robot for remote manipulation. Predictive display for robot navigation can be dissected into the following categories:

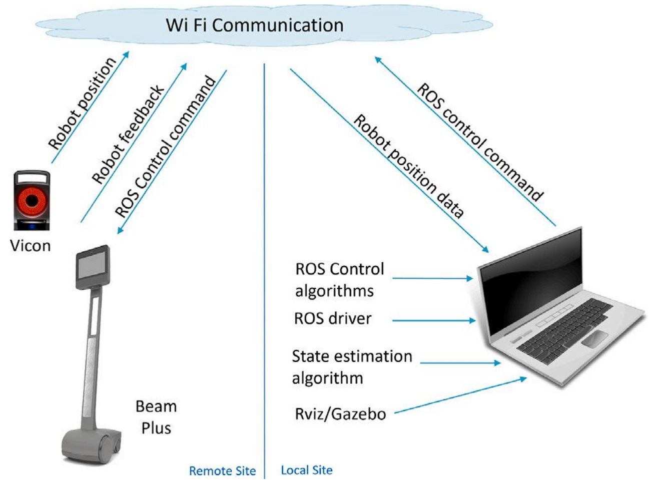


Fig. 1. The experimental framework included the camera display for two-way human communication and the simulation display with predicted robot position for navigation.

With and without a priori model of the remote environment:

The traditional method for predictive display involved using hand-modelled wire frames and solid-model overlays of the robot manipulator and scene objects on delayed video in highly pre-calibrated settings [8]. This approach was developed for a known environment and a non-moving external camera. However, more recent work aims to create photo-realistic predictive displays in less calibrated scenarios where a priori 3D model of the environment is not required [9]. Jin et al. [10] has developed a vision-based semi-autonomous teleoperation system designed for long-range teleoperation tasks under time delay network conditions. This system does not require prior knowledge of the remote scene. It starts with a self-exploration behaviour that uses a freely mounted eye-in-hand webcam to sense the remote surroundings. Burkert et al. [11] have created an online depth-fusion technique for predictive display. This technique acquires a dense 3D geometry model using a stereo camera. Yerex et al. [12,13] introduced an image-based method for predictive display. This method involves capturing the scene's geometry and appearance, compressing and transmitting it, and generating immediate feedback in response to the operator's movements, ensuring a smooth teleoperation experience.

SLAM-based predictive displays:

Hu et al. [14] proposed a solution based on sparse 3D points provided by simultaneous localisation and mapping (SLAM) and physically generated the correct surface for these point sets. Rachmielowski et al. [9] reconstructed a coarse 3D geometry model using online SLAM. They generated a predictive display by texturing from key-frame images and rendering from the current operator's viewpoint. This method does not require a pre-existing 3D model and can be used in different unknown environments. Lovi et al. [15] proposed a method that utilises Parallel Tracking and Mapping (PTAM) [16] along with the free-space carving technique [17] to create a telepresence interface. This interface includes a predictive display to generate a rendered intermediate image for human control guidance. However, this method has drawbacks, such as requiring previous images for projective texture

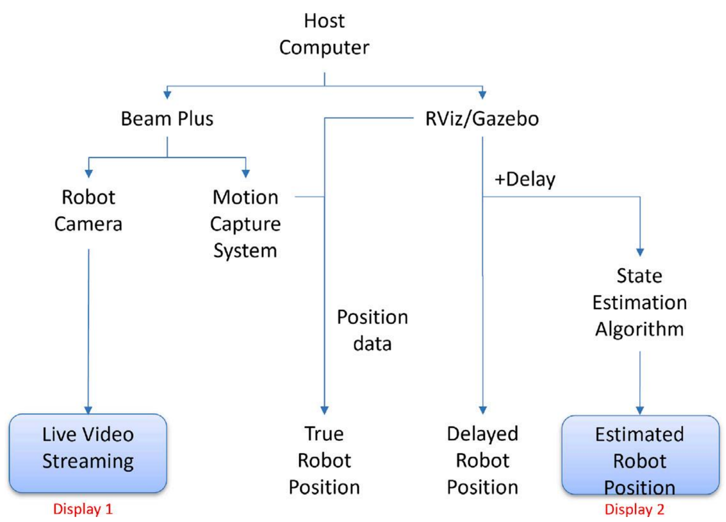


Fig. 2. This is a flow diagram of the proposed experimental environment, including three simulation displays for the robot's true position, delayed sensor measurement, and predicted robot pose.

rendering, which leads to high network consumption. Additionally, fully human-supervised control poses practical difficulties.

Multi-sensor input predictive display:

Recently, algorithms have been developed to fuse data from various sensors, such as images and LIDAR data, for predictive displays. Kelly et al. [18] created a real-time photo-realistic system by combining LIDAR data with images to reconstruct a 3D model. Livatino et al. [19] has proposed a new mixed-reality visual context for robot teleoperation interfaces to improve performance by enhancing the operator's situational awareness. This concept has been implemented on a real mobile robotic system operating outdoors. The system is equipped with onboard internal and external sensors, GPS, and an assistant drone's reconstructed 3D graphical model.

Computational hardware:

In order to improve the performance of predictive display systems, efforts were made to optimise the hardware. Schmid et al. [20] suggested a dense mapping approach for a UAV using an FPGA (Field Programmable Gate Array) to implement frame-by-frame dense stereo. In contrast, Hu et al. [14] proposed a solution that operates using a standard low-power CPU. They implemented a system that reduces bandwidth by transmitting geometry and texture information instead of video. Time delays quantitatively characterised the system's performance and qualitatively assessed using NASA TLX analysis [21].

Human operator performance measurement:

Predictive displays are designed to assist human operators in navigating robots smoothly. A recent study by Dybvik et al. [4] demonstrated that even a simple predictive display could significantly improve performance in teleoperated remote vehicle operation (up by 20% $\uparrow$) or gaming (up by 30% $\uparrow$). The study involved $N = 57$ subjects and considered three conditions: (1) Latency, (2) Latency with predictive display (PD), and (3) Baseline (no added latency). Orlosky et al. [22] examined the impact of panoramic view reconstruction, as opposed to a simple screen, in reducing perceived latency in a humanoid robot. The primary aim was to facilitate flexible head control from the robot's perspective. Zhang et al. [23] presented a framework that enhanced the guidance control of predicted trajectories by incorporating the eye movement of the teleoperator into the intended trajectory prediction. This helps to offset delays in teleoperation systems. A simulation platform involving human interaction was developed to assess the teleoperation performance of the proposed method under various delay conditions.

While the concept of a predictive display-based framework model for robot navigation is well-developed, there are only a few works in the literature [10] that address delay compensation in the system. This area requires further exploration. In this work, we proposed to create an adaptable framework that can compensate for visual time delays by adjusting the operator's viewpoint. The framework utilises a delay-compensated model to predict the robot's position in advance, giving the operator the feeling of being physically present at the remote location and directly controlling the manipulations.

Our focus has been on developing a proof of concept that can be adapted to a real system. Creating a simulation environment has the potential to replicate an expensive real-life system. Real-life telepresence systems are known to be costly and extremely challenging to establish, as the local and remote sites are often far apart, sometimes separated by cities, countries, continents, or even planets. Therefore, a simulation environment is necessary for any controlled experiment. Developing the proposed framework not only addresses this challenge for our research but also for the broader research community.

The proposed framework, designed with robustness in mind, offers a simulation environment with real-life parameters and model-based personalised control algorithms for conducting accurate predictions before implementing them in the real world. We have conducted experimental validation of the framework's performance using established prediction models. The adaptive control model and rapid robot position prediction capabilities of our framework are essential for achieving robust control of the telepresence robot in a dynamic environment.

The proposed simulation platform can seamlessly integrate and evaluate alternative delay-compensation strategies. These include adaptive controllers with parameter convergence [24], explicit predictor-based methods [25,26], a motion prediction approach [27], and learning-driven adaptive PD schemes [28]. Future enhancements may introduce SLAM-based predictive visualisation techniques, such as projected video streams [29] and mesh reconstruction with decoupled viewpoint control [30]. These approaches have been shown to enhance operator performance even under conditions of delay.

Collectively, these studies underscore the potential of predictive displays and adaptive control for latency-tolerant teleoperation. Our modular framework is well-positioned to incorporate and assess these techniques in future iterations.

The proposed framework was designed to be algorithm-agnostic and modular, making it compatible with these compensation techniques. Its ROS-based software architecture allows integration of external control modules and parameter tuning through configuration files, enabling fair comparative studies under consistent experimental conditions. In future work, we plan to incorporate and evaluate these alternative strategies within the same simulation–real environment,

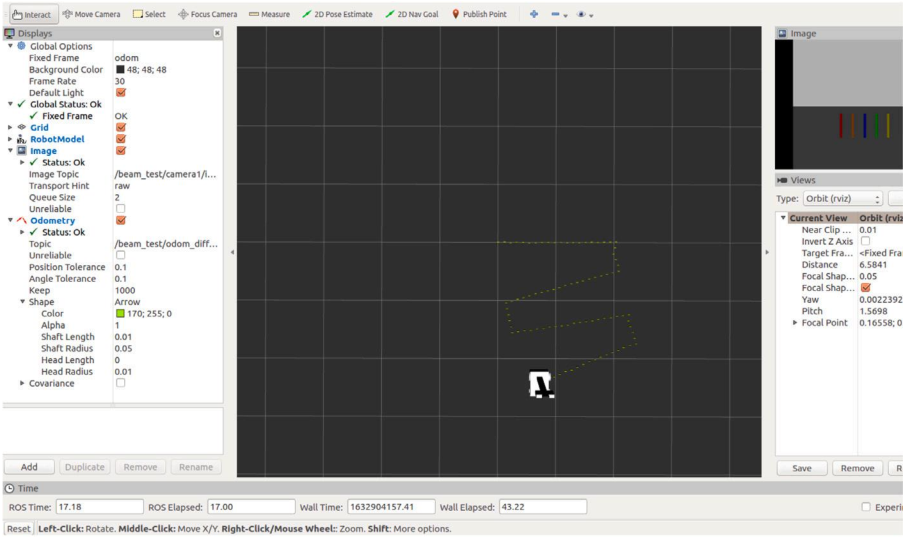
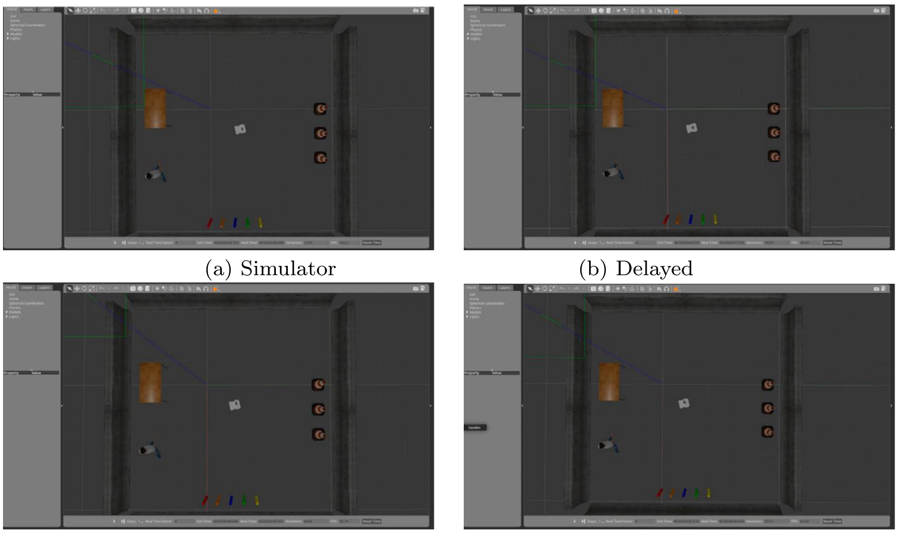


Fig. 3. Raster scan navigation path in RViz created in this work to visualise the robot model, robot path, and camera display information.



(a) Simulator (b) Delayed

(d) AS-EKF (c) EKF

Fig. 4. Simulation environment in Gazebo for a certain time delay (τ = 50). Panels (a), (b), (c), and (d) illustrate the robot positions within the simulation environment without delay, with delay, with EKF compensation, and with AS-EKF compensation, respectively.

providing quantitative benchmarking of accuracy, latency reduction, and computational overhead. This capacity for flexible algorithm evaluation establishes the framework as a testbed for generalisable teleoperation research, extending beyond the AS-EKF method demonstrated in this paper.

3. Designing the framework

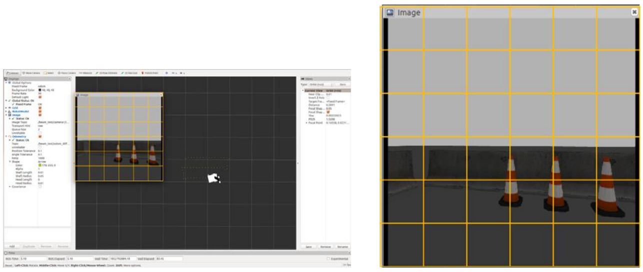
The proposed framework is based on our existing experimental framework, as described in [3]. The new experimental framework includes two research components: (a) a state estimation algorithm for predicting the robot's positions and (b) a predictive display shown in Fig. 1. This setup consists of a screen with two displays: the first is for two-way communication, and the second is for manoeuvring the robot.

The rest of this section contains essential details of the experimental framework, including both the real and simulation environments, followed by experimental results obtained using the proposed predictive framework.

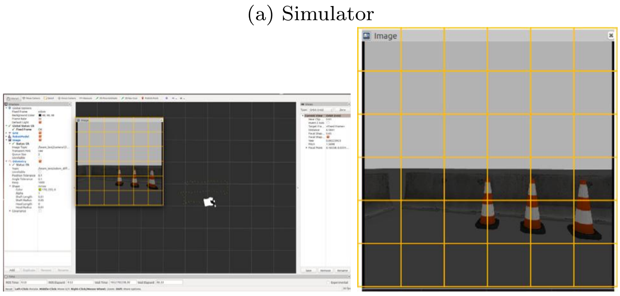
3.1. Framework components

The proposed framework contains multiple components. Although the necessity of this research drives them, we aim to develop it so that it is modular and could be adopted easily for any other type of telepresence navigation system. The proposed framework has requirements that encompass both hardware and software components. The hardware components include:

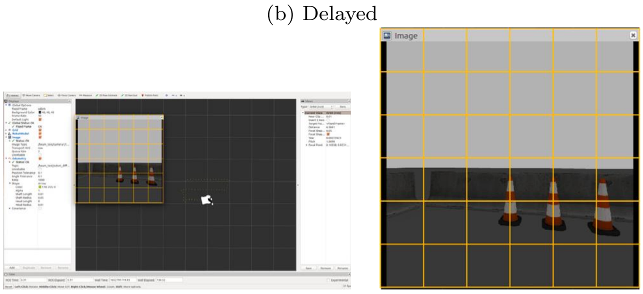
– To better evaluate different usage scenarios in our research and to have more control over the telepresence robot, we chose a mobile robot with a differential drive system that allows individual control of the wheels. Specifically, we used the Beam Plus telepresence robot [31], which is a Linux-based state-of-the-art and market-leading option for our experiments.
  In a differential-drive mobile robot, incremental odometry errors are typically caused by the system's kinematic imperfections. The two most significant errors stem from unequal wheel diameters and uncertainty about the effective wheelbase. Because the telepresence robot used in this research was a differential drive robot,



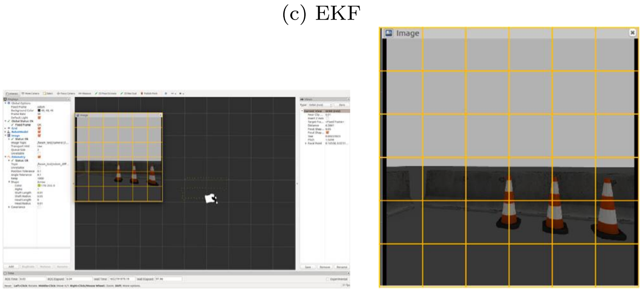
(a) Simulator



(b) Delayed



(c) EKF



(d) AS-EKF

Fig. 5. Display in the RViz for a certain time delay of (τ = 50), along with the robot's view of the environment. The figures in the left column show the robot positions in the simulation environment for (a), (b), (c) and (d) representing without delay, with delay, with EKF compensation and AS-EKF compensation, respectively. The right column shows the robot's view of the environment, indicating that the original without delay and AS-EKF have very similar environmental views, proving the effectiveness of the proposed algorithm and the use of this simulator.

it exhibited a significant amount of dead-reckoning errors during navigation. We measured the dead-reckoning accuracy of the robot using the UMBMark [32] method to determine the variance in the robot's navigation and incorporated it into the control algorithm model.

– We collected the robot's navigation data using the Vicon motion capture system [33]. Twelve motion-capture cameras were installed and calibrated in the lab, capable of tracking the robot's true state during the experiment.
– A Linux-based computer is used to establish communication between the hardware components, preprocess the navigation data and run the control and state estimation algorithms.

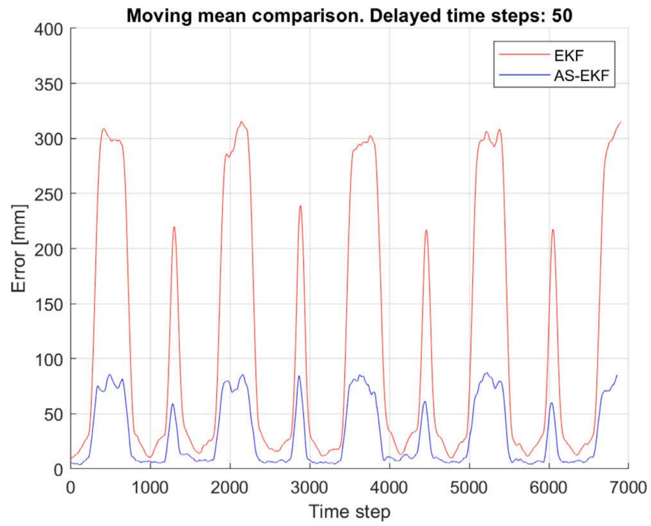


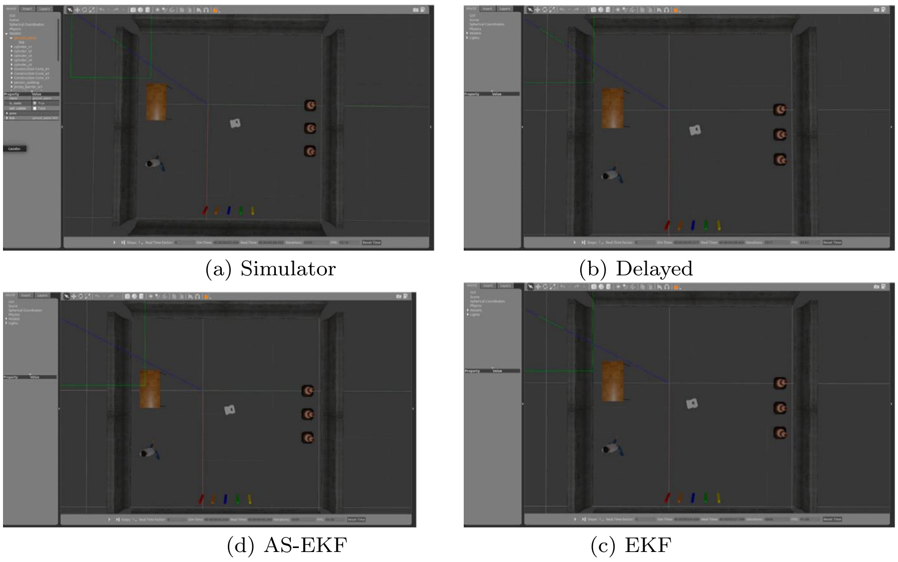
Fig. 6. Comparison of RMSE errors between EKF and AS-EKF estimations with a certain time delay. The legend shows that the red and blue lines represent results for EKF and AS-EKF, respectively. AS-EKF demonstrates significant improvement with lower error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The framework comprises three software components: (a) a robot control algorithm developed with ROS, (b) a state estimation algorithm developed, and (c) a predictive display.

– The proposed framework utilised ROS for autonomous navigation and robot control in controlled experiments. While ROS is widely used in robotics, controlling commercial robots with a closed ecosystem using ROS presents challenges. We developed algorithms in C++ using ROS commands to guide the robot in following a predefined trajectory, monitoring its progress, stopping or redirecting it as necessary, and receiving notifications of success or failure.
– The state estimation algorithm, known as the Augmented State Extended Kalman Filter (AS-EKF), which compensates for time delay, was implemented in MATLAB. The state estimation algorithms were implemented for two purposes: (a) to estimate the robot's next position and (b) to simulate various scenarios by introducing noise and delay, which helped to develop the experimental setup for robust navigation.
– The Gazebo [34] software creates a 3D scenario on the computer with robots, obstacles, and other objects. In addition, RViz [35], a powerful 3D robot visualisation tool for ROS applications, is employed. RViz's robust GUI allows for the visualisation of sensor data, robot models, and environment maps, which significantly enhances the development and debugging of the robot controllers. Gazebo is a 3D simulator, while ROS serves as the robot's interface—the combination of Gazebo and ROS results in a powerful robot simulator.

3.2. Building the framework

The proposed framework is illustrated in the flow diagram shown in Fig. 2. Our work involves modelling delays, including both certain and uncertain types, and compensating for them using the AS-EKF algorithm proposed by Das and Dobie [3]. Subsequently, the navigation paths are displayed in the simulation environment for different scenarios: true/actual robot pose, delayed robot pose, and delay-compensated estimated robot pose. The expectation is that the robot's estimated poses should be closely aligned with its actual poses.



(a) Simulator (b) Delayed

(d) AS-EKF (c) EKF

Fig. 7. Simulation environment in Gazebo for an uncertain time delay with Gamma distribution (average $\tau = 50$). Panels (a), (b), (c) and (d) illustrate the robot positions within the simulation environment without delay, with delay, with EKF compensation and AS-EKF compensation, respectively.

The software modules are built with ROS-standard interfaces and parameterised configurations, enabling support for various mobile robot platforms beyond the Beam Plus. Ongoing work includes extending support for wheeled and tracked robot models to demonstrate broader applicability.

3.2.1. Real environment

The host computer, based on Linux, connects to the telepresence robot using a ROS driver. It sends the ROS control command to the robot to create a predefined raster-scan navigation path. It receives 3D positional data of the robot's navigation captured by the Vicon motion cameras over a wireless connection.

We conducted the experiments in the robotics laboratory at our university. The goal was to create a raster-scan path for a mobile robot with a horizontal travel distance of 2000 mm, a vertical travel distance of 500 mm, and an orientation of 90°. We conducted multiple runs to collect measurement data using different combinations of linear velocities (ranging from 100 to $500\ \mathrm{mm/s}$) and angular velocities (ranging from 100 to $500\ \mathrm{mm/s}$). The starting position of the mobile robot was kept constant for all the experiments.

The position and orientation data were recorded and used as measurement data in the state estimation algorithm to estimate the robot's true position. The robot's wheel diameters and wheelbase were adjusted using the correction factor calculated by the dead-reckoning method. The captured robot navigation data using Vicon cameras have a low variance ($3.58\ \mathrm{mm}^2$) as reported in [36], which was also used to validate the prediction model. It is important to note that the Vicon-captured positional data was used to simulate noisy measurements by introducing random white noise.

3.2.2. Simulation environment

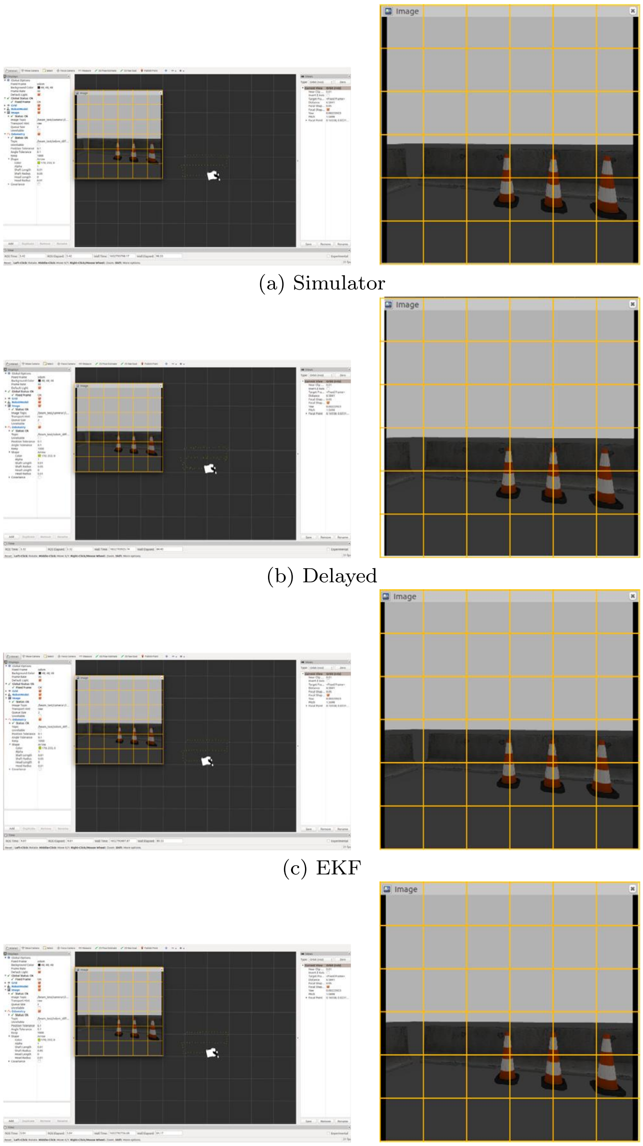
To design the simulation environment, we used three major open-source software frameworks:

1. The Robot Operating System (ROS) is utilised for developing robot control algorithms.
2. RViz, is a 3D visualisation tool for ROS that tracks a robot's position and displays what the robot sees and
3. Gazebo is a 3D robotics simulator that creates a simulated environment using graphical models similar to the real experimental setup. It allows you to evaluate and test your robot in difficult or dangerous scenarios without risking any harm to your robot. Using a simulator is typically faster than running the entire scenario on your real robot.

A telepresence robot featuring a differential drive system, similar in appearance and configuration to the Beam Plus, was simulated in Gazebo. It was controlled using ROS commands with custom-built control plugins. An algorithm for controlling the formation of navigation paths based on raster scanning was developed in C++ with ROS commands. The robot's position was exported from the simulator to use as simulated measurement data for estimating the robot's true position. The simulation framework workflow includes the following:

– Creating a simulation environment or world that includes graphical models of real-world items such as obstacles.
– Developing a differential drive telepresence robot that closely resembles the Beam Plus, the real robot used in this research work.
– Controlling and navigating the robot can be done through ROS commands or a predefined control algorithm, which is helpful for controlled experiments.
– Tracking and logging robot navigation path information and visualising the remote environment through camera sensors placed on the robot, both for simulation and the real environment.
– Including external navigation data (*i.e.,* Vicon data from real Beam Plus) in the simulation environment for the robot's position estimation.

In the simulator workflow, the simulated robot receives control commands from the host computer while operating in a real environment. The control algorithm, meticulously designed, manoeuvres the robot to its intended position and follows a predetermined trajectory that mirrors the path in the real environment. Control commands are used to navigate the simulated robot in a Gazebo environment, while RViz records the robot's position data. The control algorithm establishes a connection between Gazebo and RViz, allowing the robot's position data to be shared for visualisation of the robot's trajectory on the RViz display. We have introduced a small delay to the robot



(a) Simulator

(b) Delayed

(c) EKF

(d) AS-EKF

Fig. 8. Display in the RViz for uncertain time delay with Gamma distribution of (average $\tau = 50$), along with the robot's view of the environment. The figures in the left column show the robot positions in the simulation environment for (a), (b), (c) and (d) representing without delay, with delay, with EKF compensation and AS-EKF compensation, respectively. The right column shows the robot's view of the environment, indicating that the original without delay and AS-EKF have very similar environmental views, proving the effectiveness of the proposed algorithm and the use of this simulator.

position data that was logged, including data from both the simulation and real-life Vicon-captured robot data. This delayed data was utilised in controlled experiments instead of raw measurement data to avoid uncontrollable wireless delays. The AS-EKF-based state estimation algorithm, a robust and reliable algorithm, was used to process the delayed measurement data in order to minimise the time delay. The algorithm then produced an estimated robot position, which was displayed on the RViz.

Within the scope of this work, we intend to make a proof of the concept, and therefore, the development was done using both the ROS-RViz and Gazebo environments for simulation and visualisation, whereas the delay modelling and AS-EKF were done in MATLAB. Future development will involve translating the AS-EKF algorithm into C++ for efficient real-time applications in embedded systems and commercial robotic platforms. The integration of both and the creation of a single, unified environment is a promising aspect of our future work.

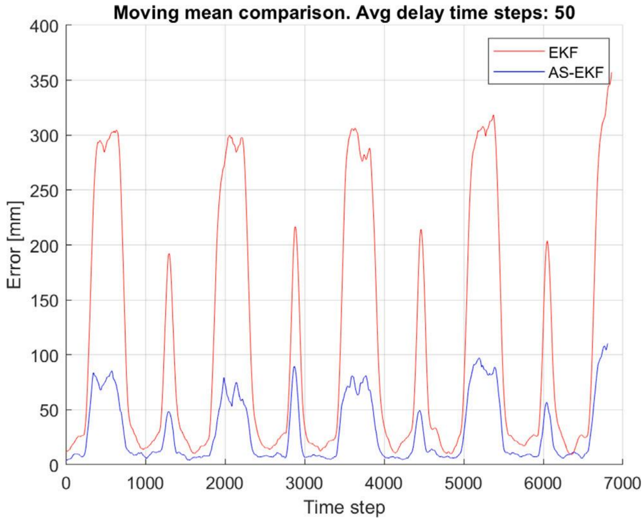


Fig. 9. Comparison of RMSE errors between EKF and AS-EKF estimations with an uncertain time delay (Gamma distribution). The legend shows that the red and blue lines represent results for EKF and AS-EKF, respectively. AS-EKF demonstrates significant improvement with lower error. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Results and discussions

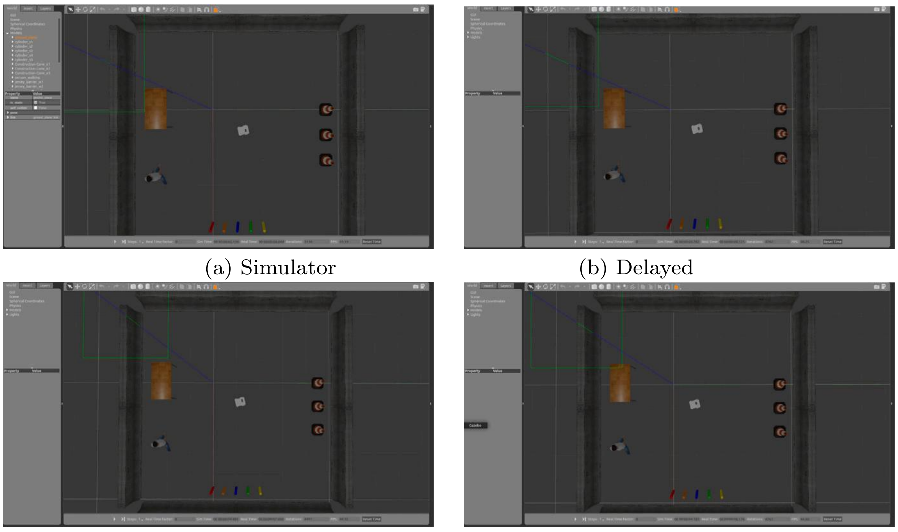
In this section, we performed a series of controlled experiments using the proposed predictive framework. The primary goal was to create and implement a reliable experimental framework that could be used by both researchers and non-expert human operators. This section confirms the framework's effectiveness by conducting a set of controlled experiments simultaneously in both simulated (using simulated Beam Plus in Gazebo) and real environments (using real Beam Plus).

4.1. Experimental workflow

In all experiments, we followed a standardised workflow/ experimental process, which involved the following steps:

1. Manoeuvring the telepresence robot using the ROS control algorithm in the real world.
2. Logging the robot's position using information from ROS feedback and Vicon cameras during navigation.
3. Incorporating certain or uncertain time delays into the recorded data to model delays.
4. Compensating for delays using the previously proposed AS-EKF state estimation algorithm.
5. Visualising the robot's navigation and path formation in the simulated environment using Gazebo.
6. Visualise the robot's path and environment using its camera in RViz.

This work mainly focused on proving the concept rather than creating a fully integrated system. After completing *Step 1*, the telepresence robot was manoeuvred, and in *Step 2*, its position information was exported to CSV files using appropriate ROS commands. These CSV files were then utilised in MATLAB to carry out *Step 3*. The result of *Step 4*, the estimated robot position after implementing the AS-EKF



(a) Simulator (b) Delayed (c) AS-EKF (d) EKF

Fig. 10. Simulation environment in Gazebo for an uncertain time delay with a Gaussian distribution (average $\tau = 50$). Panels (a), (b), (c) and (d) illustrate the robot positions within the simulation environment without delay, with delay, with EKF compensation and AS-EKF compensation, respectively.

algorithm, was exported to a CSV file and used as input to Gazebo and RViz through a custom-built control plugin.

When creating the robot model in Gazebo, we used robot parameters similar to those of the Beam Plus. We pragmatically formed a similar raster scan navigation path using another custom-built control plugin in Gazebo, as shown in Fig. 3.

We obtained two sets of results in various delayed scenarios:

- Average error in the time series between the original robot path and the delayed and corrected robot paths, and
- Visualisation of the robot's surrounding environments for the original path, delayed path, and corrected path.

In the first set of results, we expect the overall mean square error to be significantly reduced, and we should see this improvement reflected in the visualisation. In the later part of the process, we stabilised the robot after specific steps, allowing us to observe the surrounding items in a way that clearly showed a visible difference between the original and delayed path, providing a clear and unmistakable indication of our progress. However, after applying AS-EKF correction, the path should become much closer to the original one.

The following subsection presented results for both certain and uncertain delays. We looked at two types of uncertain delays: (1) Gamma and (2) Gaussian, respectively. To provide a clear visualisation, we introduced an average of 50 time step delays in all cases, translating to a 0.1-second delay in reality. However, this can be adjusted as an experimental parameter to simulate different scenarios.

4.2. Scenario I: Certain time delay

In the experiment, we introduced a time delay of ($\tau = 50$) for each robot pose and compared the moving mean errors between the original and delayed robot poses, as well as the EKF and AS-EKF predictions of the robot pose. All other parameters remained the same as in the previous experimental work. Fig. 4 shows the top view visuals of the robot's position in the Gazebo simulation environment for all four scenarios.

In order to visualise the environment from the robot camera, we used RViz and the results are shown in Fig. 5. It is evident that there is a difference between the original robot pose and the delayed robot pose in terms of what the robot can see at that particular point. Similar to the research work presented by Das and Dobie [3], we can also see that EKF cannot improve the prediction in the simulated environment. As expected, AS-EKF performed well, compensated for the delay and offered a near-original robot pose. Finally, we also measure and compare the RMSE error between EKF and AS-EKF in Fig. 6, which shows expected significant improvements in the case AS-EKF.

Table 1
RMSE error comparison for certain and uncertain time delays (average $\tau = 50$). Clearly, the AS-EKF demonstrates significant improvement over the EKF, which does not account for a delay in its filtering steps.

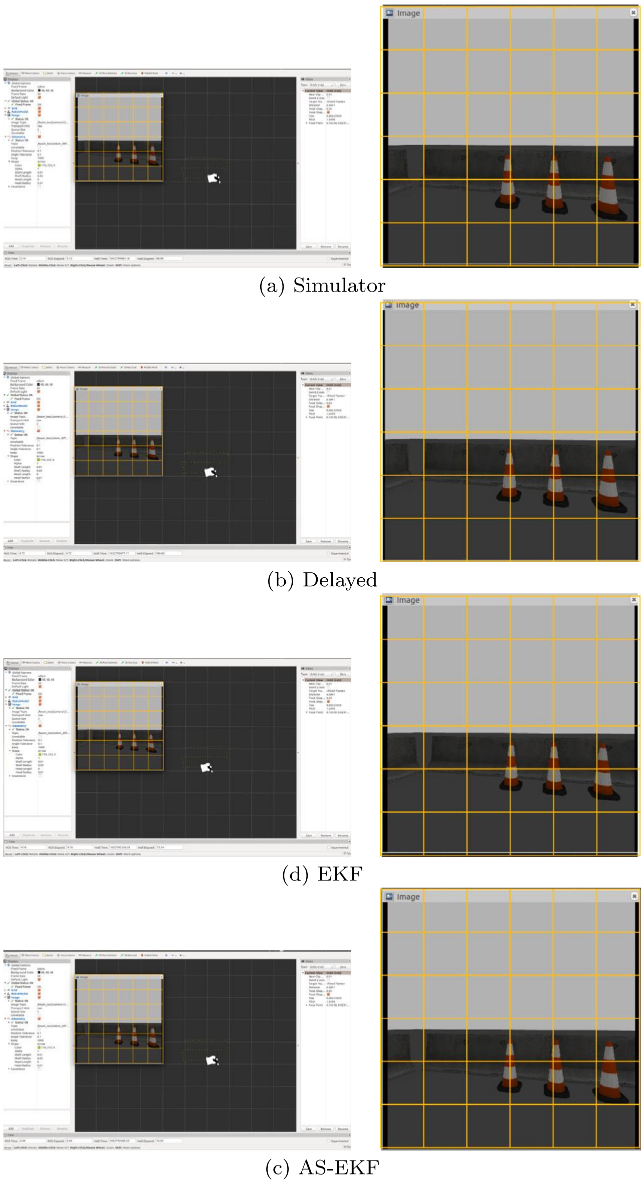
Delay type	EKF	AS-EKF	Improvement
Certain	114.82	30.84	73.15%
Uncertain (Gamma)	111.73	30.62	72.60%
Uncertain (Gaussian)	112.15	31.79	71.65%

4.3. Scenario II: Uncertain time delay

We have modelled the delay using Gamma and Gaussian distributions with an average delay of $\tau = 50$ time steps for uncertain delay. Similar to Scenario I, we captured the visualisation of Gazebo and RViz and compared the error or differences in the visualisation for the scenarios without delay, with delay, with EKF compensation, and with AS-EKF compensation, respectively.

The robot's position visualisations in Gazebo are displayed in Fig. 7 for the Gamma distribution and Fig. 10 for the Gaussian distribution. Similarly, the visualisations in RViz are displayed in Figs. 8 and 11 for the Gamma and Gaussian distributions, respectively. Additionally, the comparison between the EKF and AS-EKF performances can be seen in Figs. 9 and 12 for the Gamma and Gaussian distributions. Finally, Table 1 presents the RMSE error comparison for all three scenarios. It is evident that AS-EKF consistently performed well in all cases, as expected. The results were similar to the real-world outcomes in our previous work. AS-EKF also demonstrated better position prediction that closely matched the original robot path. These results validate the proposed predictive framework's capabilities, including visualisation, robust usability, and controlled experiments, which can be tested in a simulation environment before implementation in the real world.

The differential drive telepresence robot, similar in appearance and configuration to the Beam Plus, was tested in both real and simulated environments to develop a framework that can bring practical benefits to both robotics researchers and non-researcher human operators. The



(a) Simulator

(b) Delayed

(d) EKF

(c) AS-EKF

Fig. 11. Display in the RViz for uncertain time delay with a Gaussian distribution of (average $\tau = 50$), along with the robot's view of the environment. The figures in the left column show the robot positions in the simulation environment for (a), (b), (c) and (d) representing without delay, with delay, with EKF compensation and AS-EKF compensation, respectively. The right column shows the robot's view of the environment, indicating that the original without delay and AS-EKF have very similar environmental views, proving the effectiveness of the proposed algorithm and the use of this simulator.

framework was developed in a real environment, alongside simulations in Gazebo and was controlled through ROS with custom-built control plugins. A raster scan-based robot navigation path was designed using a C++ algorithm with ROS commands, and the robot's positional information was captured through the motion capture system connected using a ROS driver to the host computer. This positional information was utilised to model both certain and uncertain time delays using Gamma and Gaussian distributions. The delays were compensated for using standard EKF and previously proposed AS-EKF algorithms. Finally, the outputs were visualised in Gazebo and RViz to replicate a real-life telepresence system with a predictive display without any time delays.

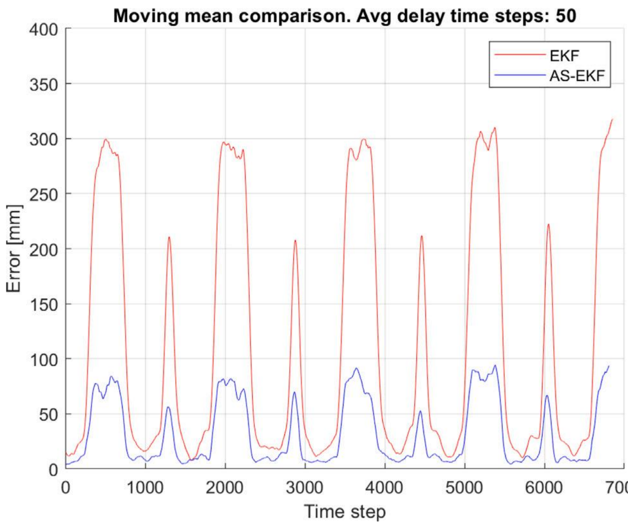


Fig. 12. Comparison of RMSE errors between EKF and AS-EKF estimations with an uncertain time delay (Gaussian distribution). The legend shows that the red and blue lines represent results for EKF and AS-EKF, respectively. AS-EKF demonstrates significant improvement with lower error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The results from various scenarios demonstrate the effectiveness and requirement of the proposed simulated framework compared to real-environment implementation, highlighting the superiority of the AS-EKF algorithm and the need for a controlled experimental setup. The similar performance trends observed in both simulated and real environments, particularly the consistent performance gain of AS-EKF (over 70% error reduction), validate the simulation framework's fidelity in representing real-world robot behaviour. Although this paper focuses on a single robot model and structured indoor settings, future evaluations will compare this approach against adaptive control and teleoperation buffering methods. Furthermore, planned user studies will assess the impact of the predictive display on operator experience and task success rates.

To further strengthen the validation of the proposed framework, we recognise the importance of benchmarking its performance against other established delay-compensation strategies. While the present study focuses on validating the AS-EKF within a controlled simulation–real hybrid setup, the modular architecture of our framework has been deliberately designed to accommodate a wide range of compensation algorithms. This includes adaptive controllers with parameter convergence, explicit predictor-based control, motion prediction methods, and reinforcement learning–driven adaptive PD schemes.

In future evaluations, we plan to conduct comparative experiments using these techniques under equivalent delay and task conditions to quantify performance differences in terms of position accuracy, latency reduction, and computational efficiency. Additionally, teleoperation buffering and adaptive control approaches will be integrated into the same simulation environment for direct comparison with AS-EKF. These benchmarking studies will provide a clearer understanding of the framework's adaptability, robustness, and relative advantages in latency-tolerant teleoperation.

5. Conclusions

Telepresence systems have long supported remote presence and interaction, yet their effective deployment in dynamic, time-delayed environments remains an ongoing research challenge. This paper presented an adaptable and simulation-integrated framework for differential-drive telepresence robot navigation that compensates for communication delays through an AS-EKF–based predictive display. The framework's hybrid design, combining real-world and simulated experimentation, demonstrates its capacity to reproduce latency conditions safely, reduce development time, and enable controlled evaluation of robot control strategies.

The experiments carried out in both the real Beam Plus robot and its simulated counterpart in Gazebo/RViz confirmed the framework's robustness. The AS-EKF consistently achieved over 70% error reduction compared with a standard EKF, validating its capability to compensate for both certain and uncertain time delays. Importantly, the similar performance trends observed in simulation and real environments indicate that the proposed setup offers a high-fidelity representation of real-world robot behaviour.

Within the scope of this work, we focused on a single robot platform and structured indoor settings. The framework was intentionally developed with ROS-standard modular interfaces and parameterised configurations. This enables straightforward adaptation to other mobile robot platforms — including wheeled, tracked, and omnidirectional designs — and to diverse teleoperation tasks. Ongoing work is extending the framework to these new robot classes to evaluate adaptability and generalisation. We outline two significant directions for future research and validation:

- **Comparative benchmarking:** We will incorporate additional delay compensation strategies such as adaptive control, predictor-based approaches, motion prediction, and reinforcement-learning-driven adaptive PD controllers. Comparative experiments will quantify improvements in prediction accuracy, latency compensation, and computational efficiency under equivalent network conditions.
- **Human-in-the-loop evaluation:** Future studies will include user experiments designed to assess the predictive display's impact on operator performance, situational awareness, and cognitive workload. We will employ standardised measures such as the NASA-TLX and task-completion metrics to compare navigation efficiency with and without predictive displays. Operator interviews and feedback sessions will further inform usability refinements.

Together, these extensions will transform the current proof-of-concept into a validated, generalisable telepresence control platform that not only supports technical robustness but also provides empirical evidence of human-operator benefits. Additional research will explore integrating visual SLAM for autonomous exploration, allowing the framework to evolve from known-path navigation towards adaptive, self-guided operation in unknown environments.

CRediT authorship contribution statement

Barnali Das: Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gordon Dobie:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Barnali Das reports financial support was provided by University of Strathclyde Department of Electronic and Electrical Engineering. Barnali Das reports financial support was provided by Pressure Profile Systems. Barnali Das reports financial support was provided by University of Sunderland. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We appreciate the support for the Ph.D. studentship from the University of Strathclyde, Glasgow, UK, and PPS UK Ltd.

Data availability

Data will be made available on request.

References

[1] K.M. Tsui, M. Desai, H.A. Yanco, C. Uhlik, Exploring use cases for telepresence robots, in: Proceedings of the 6th International Conference on Human-Robot Interaction, ACM, 2011, pp. 11–18.
[2] R. Held, N. Durlach, Telepresence, time delay and adaptation, Pict. Commun. Virtual Real Environ. (1991) 232–246.
[3] B. Das, G. Dobie, Delay compensated state estimation for Telepresence robot navigation, Robot. Auton. Syst. 146 (2021) 103890, http://dx.doi.org/10.1016/j.robot.2021.103890, URL https://www.sciencedirect.com/science/article/pii/S0921889021001755.
[4] H. Dybvik, M. Løland, A. Gerstenberg, K.B. Slåttsveen, M. Steinert, A low-cost predictive display for teleoperation: Investigating effects on human performance and workload, Int. J. Hum.-Comput. Stud. 145 (2021) 102536.
[5] B. Das, G. Dobie, S.G. Pierce, AS-EKF: a delay aware state estimation technique for telepresence robot navigation, in: IEEE International Conference on Robotic Computing, IRC, 2019.
[6] J.Y.C. Chen, E.C. Haas, M.J. Barnes, Human performance issues and user interface design for teleoperated robots, IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 37 (6) (2007) 1231–1245, http://dx.doi.org/10.1109/TSMCC.2007.905819.
[7] A. Matheson, B. Donmez, F. Rehmatullah, P. Jasiobedzki, H.-K. Ng, V. Panwar, M. Li, The effects of predictive displays on performance in driving tasks with multi-second latency: Aiding tele-operation of lunar rovers, Proc. Hum. Factors Ergon. Soc. Annu. Meet. 57 (1) (2013) 21–25, http://dx.doi.org/10.1177/1541931213571007.
[8] A. Bejczy, W. Kim, S. Venema, The phantom robot: predictive displays for teleoperation with time delay, in: Proceedings., IEEE International Conference on Robotics and Automation, Vol. 1, 1990, pp. 546–551, http://dx.doi.org/10.1109/ROBOT.1990.126037.
[9] A. Rachmielowski, N. Birkbeck, M. Jägersand, Performance evaluation of monocular predictive display, in: 2010 IEEE International Conference on Robotics and Automation, IEEE, 2010, pp. 5309–5314.
[10] J. Jin, L. Petrich, S. He, M. Dehghan, M. Jagersand, Long range teleoperation for fine manipulation tasks under time-delay network conditions, 2019, arXiv preprint arXiv:1903.09189.
[11] T. Burkert, J. Leupold, G. Passig, A photorealistic predictive display, Presence 13 (1) (2004) 22–43.
[12] K. Yerex, D. Cobzas, M. Jagersand, Predictive display models for tele-manipulation from uncalibrated camera-capture of scene geometry and appearance, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Vol. 2, IEEE, 2003, pp. 2812–2817.
[13] D. Cobzas, M. Jagersand, H. Zhang, A panoramic model for remote robot environment mapping and predictive display, Int. J. Robot. Autom. 20 (1) (2005) 25–33.
[14] H. Hu, C. Quintero, H. Sun, M. Jagersand, On-line reconstruction based predictive display in unknown environment, in: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 4446–4451.
[15] D. Lovi, N. Birkbeck, A.H. Herdocia, A. Rachmielowski, M. Jägersand, D. Cobzaş, Predictive display for mobile manipulators in unknown environments using online vision-based monocular modeling and localization, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2010, pp. 5792–5798.
[16] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, pp. 225–234, http://dx.doi.org/10.1109/ISMAR.2007.4538852.
[17] D. Lovi, N. Birkbeck, D. Cobzas, M. Jagersand, Incremental free-space carving for real-time 3d reconstruction, in: Fifth International Symposium on 3D Data Processing Visualization and Transmission, 3DPVT, 2010.
[18] A. Kelly, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, E. Capstick, Real-time photorealistic virtualized reality interface for remote mobile robot control, Int. J. Robot. Res. 30 (3) (2011) 384–404.
[19] S. Livatino, D.C. Guastella, G. Muscato, V. Rinaldi, L. Cantelli, C.D. Melita, A. Caniglia, R. Mazza, G. Padula, Intuitive robot teleoperation through multi-sensor informed mixed reality visual aids, IEEE Access 9 (2021) 25795–25808, http://dx.doi.org/10.1109/ACCESS.2021.3057808.
[20] K. Schmid, T. Tomic, F. Ruess, H. Hirschmüller, M. Suppa, Stereo vision based indoor/outdoor navigation for flying robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 3955–3962.

[21] H. Hu, C. Perez, H.-X. Sun, M. Jagersand, Performance of predictive display teleoperation under different delays with different degree of freedoms, in: 2016 International Conference on Information System and Artificial Intelligence, ISAI, IEEE, 2016, pp. 380–384.

[22] J. Orlosky, K. Theofilis, K. Kiyokawa, Y. Nagai, Effects of throughput delay on perception of robot teleoperation and head control precision in remote monitoring tasks, PRESENCE: Virtual Augment. Real. 27 (2) (2020) 226–241.

[23] Q. Zhang, L. Yang, Z. Huang, X. Song, A delay compensation framework based on eye-movement for teleoperated ground vehicles, IEEE Trans. Veh. Technol. (2024) 1–11, http://dx.doi.org/10.1109/TVT.2024.3459865.

[24] Y. Li, Y. Yin, S. Zhang, Adaptive control of delayed teleoperation systems with parameter convergence, Math. Probl. Eng. 2018 (1) (2018) 1046419, http://dx.doi.org/10.1155/2018/1046419.

[25] K. Abidi, Y. Yildiz, B. Korpe, Explicit time-delay compensation in teleoperation: An adaptive control approach, Internat. J. Robust Nonlinear Control 26 (15) (2016) 3388–3403.

[26] E. Slawiñski, V. Mut, L. Salinas, S. García, Teleoperation of a mobile robot with time-varying delay and force feedback, Robotica 30 (1) (2012) 67–77, http://dx.doi.org/10.1017/S0263574711000427.

[27] S. Shen, A. Song, T. Li, H. Li, Time delay compensation for nonlinear bilateral teleoperation: A motion prediction approach, Trans. Inst. Meas. Control 41 (16) (2019) 4488–4498.

[28] L. McCutcheon, S. Fallah, Adaptive PD control using deep reinforcement learning for local-remote teleoperation with stochastic time delays, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2023, pp. 7046–7053.

[29] J. Prakash, M. Vignati, D. Vignarca, E. Sabbioni, F. Cheli, Predictive display with perspective projection of surroundings in vehicle teleoperation to account time-delays, IEEE Trans. Intell. Transp. Syst. 24 (9) (2023) 9084–9097.

[30] Y. Chen, L. Sun, M. Benallegue, R. Cisneros-Limón, R.P. Singh, K. Kaneko, A. Tanguy, G. Caron, K. Suzuki, A. Kheddar, et al., Enhanced visual feedback with decoupled viewpoint control in immersive humanoid robot teleoperation using SLAM, in: 2022 IEEE-RAS 21st International Conference on Humanoid Robots, Humanoids, IEEE, 2022, pp. 306–313.

[31] Beam, 2024, https://suitabletech.com/support/welcome. (Accessed 09 June 2024).

[32] J. Borenstein, L. Feng, UMBmark: A benchmark test for measuring odometry errors in mobile robots, in: SPIE Mobile Robots X, Vol. 2591, 1995, pp. 113–125.

[33] Vicon motion systems, 2024, https://www.vicon.com/about-us/what-is-motion-capture/. (Accessed 09 June 2024).

[34] Gazebo, 2024, https://gazebosim.org/home. (Accessed 06 June 2024).

[35] RViz, 2024, http://wiki.ros.org/rviz. (Accessed 06 June 2024).

[36] R. Summan, S. Pierce, C. Macleod, G. Dobie, T. Gears, W. Lester, P. Pritchett, P. Smyth, Spatial calibration of large volume photogrammetry based metrology systems, Meas. 68 (2015) 189–200, http://dx.doi.org/10.1016/j.measurement.2015.02.054.