This is the post-print, accepted manuscript in the *AIAA Journal of Aerospace Information Systems*: "Time delay mitigation in aerial telerobotic operations using heterogeneous stereo vision systems," N. Sakib, K. C. Gahan, and C. A. Woolsey. *AIAA. J. Aerospace Information Systems* **20(9), pp. 526-545, 2023. [DOI: 10.2514/1.I011204]**

Time Delay Mitigation in Aerial Telerobotic Operations Using Heterogeneous Stereo-Vision Systems

Nazmus Sakib*, Kenneth C. Gahan† and Craig A. Woolsey‡ §
Virginia Tech, Blacksburg, VA, 24061

This paper investigates the use of a heterogeneous stereo-vision system to mitigate the effects of time delays in a drone-based visual interface presented to a human operator. Time delays in the display for a telerobotic interface refer to the time difference between the operator's input action and the corresponding visible outcome. In human/machine interfaces, time delays can arise due to computation, telecommunication, and mechanical limitations. These delays can degrade the performance of the human/machine system. A heterogeneous stereo-vision predictive algorithm is presented that can reduce the negative effects of time delays in the operator's display. The heterogeneous stereo-vision system consists of an omnidirectional camera and a pan-tilt-zoom camera. Two predictive display setups were developed that modify the delayed video imagery that would otherwise be presented to the operator in a way that provides an almost immediate visual response to the operator's control actions. The usability of the system is determined through human performance testing with and without the predictive algorithms. The results indicate that the predictive algorithm allows more efficient, accurate, and user-friendly operation.

I. Introduction

In the near future, the integration of robot sensing, cognition, and learning in human operations will redefine the workforce. The synergy between human operators and semi-autonomous robotic systems will provide increased capability, efficiency, and safety by allowing humans to perform specialized tasks remotely. This study examines the challenges to uncrewed aerial vehicle (UAV) based telerobotic operations inspired by aerial robotic bridge inspection. Regular and thorough bridge inspections are crucial to ensure the safety of the highway infrastructure. The deadly collapse of the 51-year-old Morandi Bridge in Genoa, Italy (2018) [1] or the interstate I-40 closure in Memphis, Tennessee (2021) due to "a significant fracture in one of two 900-foot horizontal steel beams" [2] illustrate the importance of regular inspection and maintenance.

An autonomous ground- or air-based telerobotic system consists of motion measurement sensors for navigation and image sensors like LiDAR or cameras for task completion. It may also be necessary to have powerful onboard computers to perform various computer vision tasks. The effectiveness of ground- or air-based autonomous robots has been demonstrated in numerous civil engineering applications like construction safety and progress monitoring [3–5], geotechnical engineering [6, 7], and post-disaster reconnaissance [8, 9]. With advances in UAV technologies, their use in telerobotic operations is becoming more common. Chen et al. [10], for example, demonstrated how large fractures and defects in roadways can be found in non-contact inspections using UAVs. Reagan et al. [11] concluded that a fully developed UAV-based surveillance system might enhance the efficiency and frequency of inspection, enabling more timely responses to bridge defects. The authors of [12–15] describe several methods developed for computer vision-based detection of cracks in concrete and steel, corrosion, and spalling using artificial intelligence (AI). These

*Graduate Student, Aerospace and Ocean Engineering, Virginia Tech, Email: nazmus.sakib@vt.edu.
†Graduate Student, Aerospace and Ocean Engineering, Virginia Tech, Email: kcgahan21@vt.edu.
‡Professor, Aerospace and Ocean Engineering, Virginia Tech, AIAA Member, Email: cwoolsey@vt.edu.
§Accepted for publication in the Journal of Aerospace Information Systems on April 20, 2023. Published online on June 10, 2023. DOI: https://doi.org/10.2514/1.I011204.

algorithms can be implemented on a UAV, providing augmented reality (AR) cues to a human operator/inspector concerning possible defects.

One challenge in the use of aerial telerobotics for first-person view (FPV) inspection, where AI provides AR cues, is the issue of time delay. According to de Vries [16], some of the sources of delays in telerobotic operations include: signal transport, data encryption, data compression, error correction, and computation. A summary of human performance issues associated with teleoperation time delays can be found in the article by Chen et al. [17] where the authors defined *time delay* to be the time difference between an input action and the corresponding visible response. They showed that if the delay is more than $170\,\mathrm{ms}$ then an operator's performance in tracking and telemanipulation tasks degrades. Excessive delay can also create over-actuation, field-of-view (FoV) reduction, and motion sickness. If the delay is large ($1\,\mathrm{s}$ and above), Sheridan and Ferrell [18] showed that operators use a "move and wait" strategy, where they actuate the system slowly and wait to see the outcome of their actions, instead of continuously manipulating the robot. After observation, the operators provide corrective inputs that bring the system closer to the goal state. The task may ultimately be completed by repeating this iterative process, but the approach is time-consuming and frustrating to a human operator. For a UAV with limited endurance, excessive delays can make telerobotic operations impractical.

Predictive displays can counter delay-induced effects as described in [17, 18]. Sheridan [19] considers a predictive display to have a computer-generated visual indicator that displays the motion of the telerobotic system. The indicator is then projected ahead in time to offer the operator an instant understanding of what might occur to the system given its current state and operator inputs.

Motivated by the problem of time delays in the visual telerobotic operation of an aerial robot, this study aims to:

- explain how time delays affect telerobotic bridge inspection operations.
- develop computer vision-based predictive display algorithms for delay mitigation.
- quantify user satisfaction with the proposed solution through human subject testing.
- create a framework for real-time implementation of the algorithm.
- determine a model for the human operator to determine the delay tolerance of the human/machine system (HMS).
- use the human operator model along with the machine model to predict task performance in the presence of delay.

A heterogeneous stereo-vision setup was used by Kang et al. [20] to develop a peripheral-central vision system for small UAV tracking incorporating a pan-tilt-zoom (PTZ) camera for "central vision" and an omnidirectional camera (with fisheye lenses) for "peripheral vision." The UAV-based heterogeneous vision platform developed for the purpose of this study is similar to the platform developed by Kang et al. The rationale for a heterogeneous camera system is to provide high-quality imagery from the PTZ camera to the inspector, and possibly to an AI computer vision algorithm that has been trained to detect defects [21], while the lower quality imagery from the wide FoV omnidirectional camera provides supplemental information to improve the inspector's situation awareness, either directly or indirectly with cues produced using computer vision. The UAV motion can be managed autonomously, as shown by Shanthakumar et al. [22], while allowing the human inspector to guide the motion or even take manual control when needed. Delays in the visual interface, however, make it more difficult for an inspector to perform both the inspection and operational supervision duties.

The paper is organized as follows. Section II describes related work on heterogeneous stereo computer vision, human operator modeling and human subjects testing, and predictive display technology and identifies some knowledge gaps addressed in this paper. Section III describes the theory related to the proposed predictive algorithm, including overviews of delay and prediction, perspective projection, and temporal and spatial prediction. Section IV describes the hardware and software implementation. Section V provides details about the experimental setup and the study design. Section VI describes the results of the experimental investigation. Section VII describes some issues that require further investigation and Section VIII provides conclusions from the effort.

II. Related Work

This paper describes the development of image processing methods that enable a predictive display capable of mitigating large delays in the visual interface of a telerobotic system. To support the predictive display system development, the negative effects of delays are characterized using human operator modeling and classical stability analysis. Finally, human subject testing results obtained using a real-time implementation demonstrate the effectiveness of the proposed predictive display system.

A. Related work: Heterogeneous stereo-vision systems

Computer vision uses software to produce information from visual sensors like cameras or LiDARs. This information could be the pixel locations of features that have been robustly detected using algorithms like SIFT [23], LIFT [24], ORB [25], etc. If two cameras have overlapping FoVs, this stereo-vision system enables depth perception, which is essential for applications such as localization and mapping. Typically, the two cameras are identical, with a fixed baseline and parallel bore sights. If one camera is a PTZ camera and the other is omnidirectional, then this *heterogeneous stereo-vision* problem becomes more challenging, but it is still feasible. For example, Kang et al. [26] use heterogeneous stereo-vision to determine the class and pose of a threat aircraft as well as the range to this threat. One difficulty with using real-time heterogeneous stereo-vision systems, however, is that feature detection and matching techniques like ORB, SIFT, LIFT, etc. assume constant illumination and pixel density between two images being compared. As these assumptions are violated in a heterogeneous stereo-vision system, it is difficult to match feature points accurately across the two FoVs. Some other constraints are required, like known camera positions and orientations. Very accurate calibrations are also required among heterogeneous sensors to reduce warping, matching, and transformation errors. The stereo camera calibration algorithm of Rathnayaka et al. [27] uses a customized checkerboard pattern and performs camera calibration automatically when the two cameras are fixed relative to each other. Alternatively, one may perform manual calibration, provided there is no relative motion between the heterogeneous sensors. This *fixed relative motion* constraint replaces the brightness and pixel density constraints mentioned above.

B. Related work: Human subjects testing

A human inspector can be expected to control the gimbal-camera system while performing structural inspection tasks but the human may also serve a supervisory role in monitoring the UAV's autonomous navigation around the structure. If a region of interest appears on the display, the human can focus on the inspection task, taking control of the gimbal-camera system while the UAV hovers or moves slowly along the structure.

The inspection scenario described above can be simulated using precision tracking and target acquisition tasks like those used in aircraft handling quality experiments. The precision tracking task can be used to capture human performance data simulating the case of looking for defects in a region of interest. The target acquisition task can simulate aiming the camera, or a cursor, at a particular point of interest.

To assess the performance of human subjects a proper subjective evaluation scale must be chosen. Different scientific societies use a variety of subjective evaluation techniques. The system usability scale (SUS) [28] is used by the human-computer interaction community. NASA created the task-load index [29] for subjective evaluations. Here, the Cooper-Harper handling qualities rating (C-H HQR) [30] scale is used to evaluate human/machine system (HMS) performance. Aircraft test pilots and flight test engineers frequently use the C-H HQR scale to assess an aircraft's task-specific performance. As noted in [31], the C-H HQR scale was used as a foundation for the Pilot Quality Rating (PQR) scale to rate the performance of a highly automated task, such as the automated landing of a vehicle on an aircraft carrier. Although the scale is used widely in aircraft handling qualities assessment by pilots, it can also be used for other applications. One major reason for favoring the C-H HQR scale is because of its ability to capture the effects of time delays present in a system by comparing the performance deterioration due to the delay with performance for a similar non-delayed task. This scale was used by Jennings et al. [32] to calculate a pilot's maximum delay tolerance in a helicopter simulation. Their findings demonstrated delays as small as 134 ms increased the variability of position error and that handling qualities degraded as the delays increased. Because of its ability to capture the effects of time delays subjectively and because of its familiarity with aerospace and aviation communities, the C-H HQR scale was chosen to be the subjective measurement tool for understanding human/machine performance.

According to [33], there are two types of pilots in aircraft testing: a high-gain pilot and a low-gain pilot. Compared with a high-gain pilot, a low-gain pilot often injects inputs that are smoother and smaller in amplitude. Since high-gain pilots are more likely to experience non-linear responses in handling qualities than low-gain pilots, they are excellent at highlighting potential shortcomings in an aircraft's handling qualities. The participants in this study can also be divided into low-gain and high-gain participants. A good experimental setup is one in which the system is rated similarly by participants with high or low gains. To enable assessment using the C-H HQR scale, bounds for *desired* and *adequate* performance must be established; see Figure 8 in Section V. More discussion about the types of bounds and the experimental setup is presented in Section V.A.

C. Related work: Modeling the human operator

For the work presented here, the HMS under evaluation is the inspector and camera gimbal. Determining a model for the human in an HMS is valuable for predicting and improving overall system performance. As early as World War II, efforts were made to mathematically describe HMSs [34] motivated by improving the performance of warfighting HMSs: aircraft, artillery, etc. Some of the early pioneers of human/machine research were Tustin [35] who identified that humans adapt their behavior when operating manual control systems and Russell [36] who expanded on Tustin's work and conducted some of the first experiments to identify a transfer function for the human operator. Perhaps the most well-known researchers of human transfer function modeling are McRuer and Krendel [37] whose work to characterize the human as a quasi-linear element in a closed-loop control system has been the foundation of piloted aircraft control design [38] for decades. Although more sophisticated models of human behavior exist, such as Baron and Kleinman's [39] "optimal controller" model, the structure proposed by McRuer and Krendel is adequate for predicting HMS performance in the scenarios considered here, as will be shown.

D. Related work: Predictive display

Telerobotic time delays can potentially be mitigated using predictive controllers which are robust to time delays. Sirouspour and Shahdi [40] constructed a linear quadratic Gaussian (LQG) controller in discrete time for teleoperation with communication time delay. The authors reduced the sampling rate intentionally, in their example, to avoid potential numerical issues due to increased computational load as the delay increased. The lowering of the sampling rate limits their method's applicability as it adversely affects the closed-loop reaction and teleoperation stability. To ensure performance and stability when dealing with packet loss and time delay, Desoer and Vidyasagar [41] introduced passivity-based bilateral teleoperation, where the master-slave teleoperation system is represented using linear models. Nuño et al. [42] reviewed work on passivity-based controllers for non-linear bilateral teleoperation. Using autoregressive models to predict future control inputs from available past inputs, Lu et al. [43] and Mirfakhrai and Payandeh [44] developed predictive techniques to model and mitigate the random time delays due to internet-based teleoperation with fuzzy adaptive control methods. While these control-based approaches are robust to small time delays and improve stability, there are some drawbacks. For example, these controllers can provide delay mitigation up to only a small amount of delay before becoming unstable. Additionally, models for the system and the operator are required, which may be difficult to obtain. Such approaches are also not favorable if direct human involvement is necessary for task completion as in telerobotic surgery and infrastructure inspection. For these situations, open-loop reference control with the human acting as the loop closure operator generally performs better. Open-loop systems, as in the predictive display system presented here, can tolerate large delays and require no prior knowledge of the system or operator dynamics.

There are a few open-loop display-based predictors developed to mitigate telerobotic time delays. Using a single camera predictive display, Brudnak [45] created an algorithm for delay mitigation in teleoperated ground vehicles. The effectiveness of the algorithm was demonstrated by simulating ground vehicle motion on a road. The imagery from a virtual camera attached to the vehicle was manipulated to create the predictive display. It was found that with the predictive display active, and with $500\ \mathrm{ms}$ of delay, drivers were able to move 29% faster and had 35% less path deviation than in the case where no prediction was provided. Except for the case of purely forward movement, all vehicle motions and maneuvers resulted in the predicted image frame falling outside of the currently displayed image frame. As a result, the predicted image contained empty regions which reduced the information available to the operator. (See Figs. 29-30 in [45].) Another work by Jung et al. [46] tried to alleviate delay-induced simulator sickness effects due to wearing head-mounted displays (HMDs) by developing a predictive display algorithm to compensate for the bidirectional communication and operation delay. The authors constructed a robot-camera system that simulates the human head-neck motion using kinematic models of the human neck. The delayed imagery from the robot-mounted camera was modified according to the current head-neck orientation of the user. To ensure that no part of the predicted frame fell outside of the image frame, the authors intentionally reduced the size of the predicted image. Again, the result was a reduction in the information available to the operator. (See Figs. 4-5 in [46].) The margin of reduction varied with the amount and the intensity of rotation. For a slow and small angle rotation, the predicted image frame was almost equal to the actual image frame. For faster, larger rotations, the predicted image frame was proportionally reduced. The algorithm was tested with delays varying within the range of $70 - 230\ \mathrm{ms}$.

Perhaps the most closely related work concerning predictive displays for UAV operations was done by Cox and Wong [47]. They predicted the vehicle's state change due to commands that were yet to reach the vehicle and then modified the imagery displayed to the pilot based on the predicted states. This modification allowed the pilot to immediately see the effect of the control inputs. The study implemented the concept on a prototype system in real-time flight tests and

presented qualitative results for 500 ms and 1000 ms of delay. The authors concluded that the predictive compensation allowed for smooth control of the vehicle compared to the case without delay compensation and achieved satisfactory manual control of teleoperated UAVs with time delays. As in the work of Brudnak, Cox et al. also experienced the issue where the predicted image frame fell outside the currently displayed image frame. (See Fig. 8 in [47].)

E. Gaps in current knowledge

Since heterogeneous vision systems violate the constant illumination and pixel density assumptions, as discussed in Section II.A, there are no automated methods to perform feature detection and matching that might be used by powerful computer vision and AI-based techniques. As such, all the predictive display methods described in Section III.B use monocular cameras. But using a single camera means that in most cases the modified predicted frame will fall outside the currently displayed image frame, since imagery from the predicted region has not yet been acquired by the camera. The respective authors of [45–47] chose either to ignore the issue or to reduce the FoV of the modified image. Moreover, these studies do not focus on the subjective human-robot interaction aspects of the experiment but instead focus on comparing objective data such as *path deviation*, *average vehicle velocity*, etc. to show the effectiveness of the predictive display algorithm.

The system presented in this paper uses heterogeneous stereo cameras together with a novel predictive display algorithm that replaces empty regions in the predicted frame with imagery from the second camera. The effectiveness of the algorithm is demonstrated using human subject testing that captures the objective human performance data as well as the subjective human perception of the system. In addition, the paper describes the negative effects of time delays in human/machine systems from the perspective of classical stability theory. The work described here expands on prior work [48], where a similar heterogeneous system was used but where the omnidirectional camera remained fixed while the PTZ camera moved with the gimbal. As mentioned in Section II.A, some complications arise if there is relative motion between the heterogeneous cameras. A detailed discussion is presented in Section III.B. In the implementation described in this paper, the two cameras were constrained to have no relative motion.

The primary contributions of this paper are:

- a novel predictive display algorithm using a heterogeneous stereo-vision system to mitigate display delays for real-time UAV-based inspection tasks.
- improvement of the predictive display algorithm to perform efficient and accurate predictions on low frame rate video imagery.
- an analysis of subjective user ratings of the predictive display to show what a typical user would feel if such systems are available for delay mitigation.
- a demonstration that classical stability concepts developed for piloted aircraft can be applied to HMSs such as an operator-gimbal system.

III. Methodology

This section describes the underlying theory for the proposed predictive algorithm.

A. Delay and Prediction Overview

In the motivating application of telerobotic bridge inspection, an inspector operates the robot and its camera using an FPV display to monitor a live video feed coming from the PTZ camera. To aid the inspection process, the presented video imagery may include augmented reality (AR) cues that indicate features of interest such as bars, beams, and gusset plates as well as defects in these features such as cracks or corrosion [49–53]. Due to various delays, there is a difference between the time when an image is acquired and the time when it appears on the inspector's screen. These delays may hamper real-time structural inspections by making it more difficult to control the camera; delays can cause significant overshoots or undershoots. If the delays are not mitigated, then they may cause aversion to proposed technologies like AI-enhanced aerial telerobotic inspection.

To better appreciate the propagation of delays, consider the information flow depicted in Fig 1: an image is displayed on a screen, and at time t a user commands the PTZ camera to move. That is, a reference command is sent to the gimbal which controls the orientation of the PTZ camera. The command can be sent to the gimbal using a joystick, for example, which has some internal sensors with inherent delay (Δ_{joy}). The gimbal then receives this input command over a wireless network with a communication delay (Δ_{net}). After the gimbal receives the input, its actuators start moving after some additional delay time (Δ_{act}), ultimately reaching the commanded orientation where the camera captures imagery

along the new line of sight. The image is then sent to a ground station through a wireless network, incurring another network delay (Δ_{net}), before reaching the image processing computer. The computer augments the image with cues, such as pointers to possible defects or georectified annotations from earlier inspections, adding some computational delay (Δ_{comp}). The computer then sends the augmented image to the user's display. The outcome of a commanded input at time t is displayed on the inspector's screen at time $t + \tau$, where the total delay $\tau = \Delta_{\text{joy}} + 2\Delta_{\text{net}} + \Delta_{\text{act}} + \Delta_{\text{comp}}$; see Fig. 1.

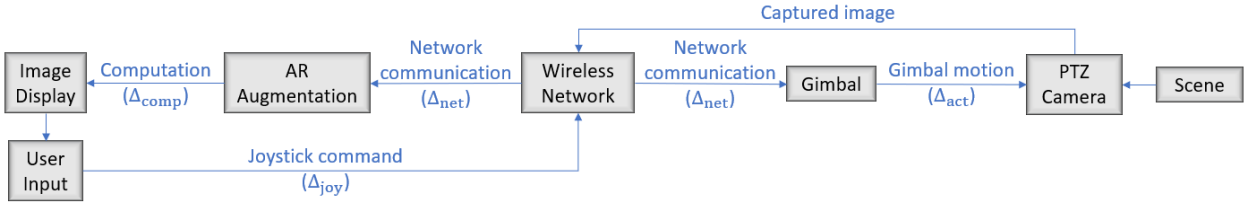


Fig. 1 Delays involved in each step

An overview of the prediction algorithm is depicted in the timeline shown in Fig. 2 where a notional sequence of raw and modified images from the omnidirectional and PTZ cameras are displayed. The images in the far left column, labeled Fig. 2a, show the view from the omnidirectional and PTZ cameras attached to a UAV hovering in place with respect to a scene. The location of the currently displayed PTZ camera frame is shown by the black rectangle in the omnidirectional camera image. The yellow rectangles represent image augmentations, obtained by running some other computer vision algorithms that introduce the computational delay, Δ_{comp}. So long as the UAV or the gimbal does not move, the images from the omnidirectional and the PTZ cameras remain fixed. When the user commands an input at time t in order to move the gimbal or the UAV, the currently displayed PTZ frame is predicted almost instantly to display the change due to this commanded input. Without this prediction, the user would have to wait until the delay time τ to pass to see the change. This prediction is represented in Fig. 2b. Because of the nature of the movement of the gimbal or the UAV, the predicted PTZ frame would fall outside the camera's current view as indicated by the white empty spaces in Fig 2b. This missing portion in the predicted image is replaced by the image from the omnidirectional camera. By using the information available from another wide FoV camera, no portion of the predicted image remains empty, as would occur with a single camera [45–47]. As the omnidirectional camera is mounted in a different location than the PTZ camera, the omnidirectional image must undergo a perspective transformation to match the viewpoint of the PTZ camera. An example of this *heterogeneous-stitch* image is depicted in Fig 2c. However, stitching the omnidirectional camera image together with the PTZ image is only one method of displaying the predicted image to the user. The images can also be displayed in a *split-screen* format. For example, the transformed omnidirectional image can be displayed as an inset image within the original delayed PTZ image; see Sec. IV.B. Let the time required for the predictive algorithm, after performing all the calculations and predictions, be Δ_{alg}. In practice, this time increment is at least an order of magnitude smaller than the delay in the system. In the end, the predicted image is displayed on the inspector's screen almost instantly, at time $t + \Delta_{\text{alg}}$ instead of $t + \tau$.

B. Perspective Projection

The theory and the notation are similar to [54], but here two different camera frames are considered: the omnidirectional camera frame $\{\mathcal{F}_{\text{o}}\}$ and the PTZ camera frame $\{\mathcal{F}_{\text{p}}\}$, as shown in Fig. 3. Additionally, $\{\mathcal{F}_{\text{w}}\}$ is the world coordinate frame attached to a plane. Let $\boldsymbol{\mathcal{P}}$ denote a point of interest, with Cartesian world coordinates $\boldsymbol{\mathcal{P}} = [\mathcal{X}, \mathcal{Y}, 0]^{\text{T}}$. Suppose that both cameras observe this common 3D feature point. This world coordinate point can also be represented in the frames $\{\mathcal{F}_{\text{o}}\}$ and $\{\mathcal{F}_{\text{p}}\}$ and are denoted as ${}^{\text{o}}\boldsymbol{\mathcal{P}}$ and ${}^{\text{p}}\boldsymbol{\mathcal{P}}$, respectively. The symbols ${}^{\text{p}}\boldsymbol{t}_{\text{o}}$ and ${}^{\text{p}}\boldsymbol{R}_{\text{o}}$, respectively, indicate the translation and rotation of the omnidirectional camera frame with respect to the PTZ camera frame. The perpendicular distances from the object plane, $\{\mathcal{F}_{\text{w}}\}$, to the corresponding camera frames are d_{o} and d_{p}, respectively. The normal to the object plane, $\boldsymbol{n}$, can also be expressed in the $\{\mathcal{F}_{\text{o}}\}$ and $\{\mathcal{F}_{\text{p}}\}$ frame coordinates as ${}^{\text{o}}\boldsymbol{n} = {}^{\text{o}}\boldsymbol{R}\boldsymbol{n}$ and ${}^{\text{p}}\boldsymbol{n} = {}^{\text{p}}\boldsymbol{R}\boldsymbol{n}$, respectively, where ${}^{\text{o}}\boldsymbol{R}$ and ${}^{\text{p}}\boldsymbol{R}$ are the rotation matrices of $\{\mathcal{F}_{\text{o}}\}$ and $\{\mathcal{F}_{\text{p}}\}$ frames used to convert vectors in the $\{\mathcal{F}_{\text{w}}\}$ frame to those respective frames.

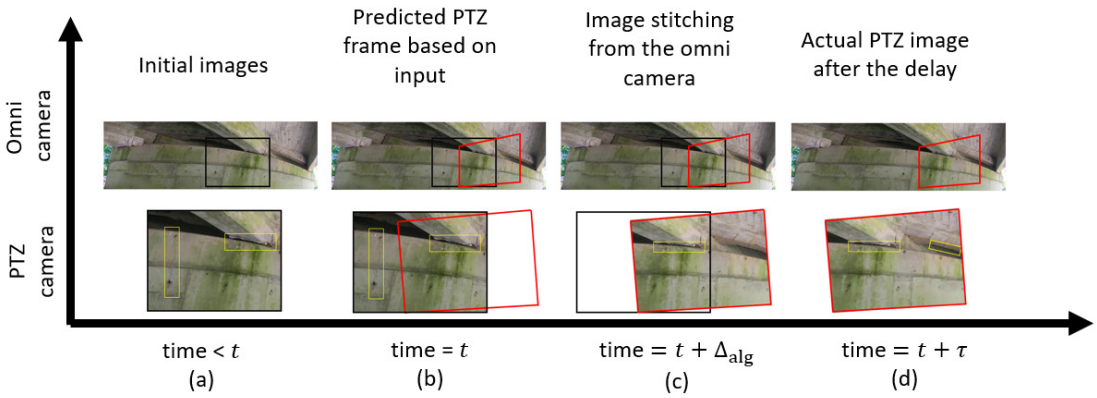


Fig. 2 Prediction timeline

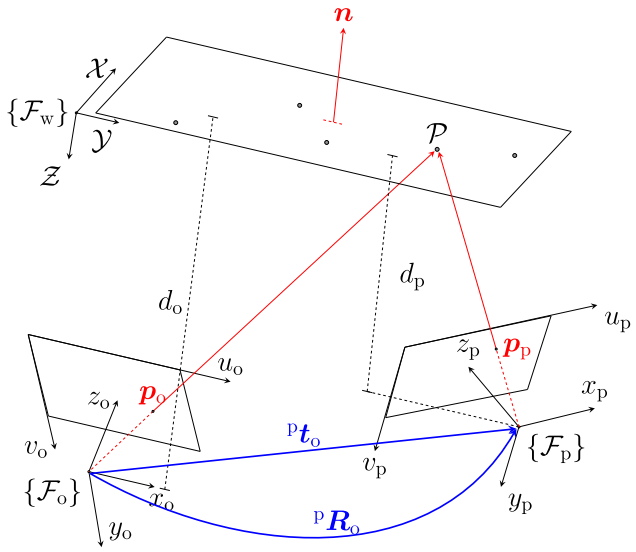


Fig. 3 Omnidirectional and PTZ camera frames with involved notations

1. PTZ Camera Model and Calibration.

PTZ cameras can be modeled by a simple pinhole model described in [55]. A point in the object plane in world coordinates can be represented in the homogeneous form by appending a "1" to the coordinate vector: $\tilde{\boldsymbol{\mathcal{P}}} = [\mathcal{X}, \mathcal{Y}, 0, 1]^{\mathrm{T}}$. A 3D feature point with world coordinates $\boldsymbol{\mathcal{P}}$ is projected by the PTZ camera onto an image plane at depth f_{p} from the PTZ camera origin. If $\boldsymbol{p}_{\mathrm{p}} = [u_{\mathrm{p}}, v_{\mathrm{p}}]^{\mathrm{T}}$ is the projection of $\boldsymbol{\mathcal{P}}$ onto the image plane, in pixels, and $\tilde{\boldsymbol{p}}_{\mathrm{p}} = [u_{\mathrm{p}}, v_{\mathrm{p}}, 1]^{\mathrm{T}}$ is the homogeneous transformation of $\boldsymbol{p}_{\mathrm{p}}$, then the relationship between the two is given by:

$$\lambda \tilde{\boldsymbol{p}}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{p}}[{}^{\mathrm{p}}\boldsymbol{R},\, {}^{\mathrm{p}}\boldsymbol{t}]\tilde{\boldsymbol{\mathcal{P}}} \quad \text{with } \boldsymbol{K}_{\mathrm{p}} = \begin{bmatrix} f_{\mathrm{p}}/Sx_{\mathrm{p}} & \gamma & u_{\mathrm{p0}} \\ 0 & f_{\mathrm{p}}/Sy_{\mathrm{p}} & v_{\mathrm{p0}} \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where λ is a dimensionless scale factor that captures the depth information of the object plane with respect to the camera frame $\{\mathcal{F}_{\mathrm{p}}\}$. The matrix $[{}^{\mathrm{p}}\boldsymbol{R},\, {}^{\mathrm{p}}\boldsymbol{t}]$ is called the extrinsic parameter matrix for the PTZ camera, consisting of the rotation and the translation of the object plane in the world coordinate system with respect to the PTZ camera coordinates. The matrix $\boldsymbol{K}_{\mathrm{p}}$ is the camera intrinsic parameter matrix with $(u_{\mathrm{p0}}, v_{\mathrm{p0}})$ as the pixel coordinate of the *principal point* – the point on the image plane onto which the perspective center is projected. In Fig. 3 the coordinates of the principal point of the PTZ camera are $(u_{\mathrm{p0}}, v_{\mathrm{p0}}) = (u_{\mathrm{p}}/2, v_{\mathrm{p}}/2)$. The symbols Sx_{p} and Sy_{p} represent the metric-to-pixel conversion factors, also known as the "pixel size", in the u_{p} and v_{p} axes. The focal length of the camera is f_{p} and γ is the parameter

that describes the skew of the two image axes. However, modern lens-based cameras distort the light rays causing the actual pixel location of a given point to be different than the location indicated by the pinhole camera model. According to [56], the most commonly encountered distortion is *radial lens distortion* that causes the actual image point to be displaced radially in the image plane. The radial distortion of any pixel point is given by Eqn. (2):

$$\begin{bmatrix} \delta u_{\mathrm{p}}^{(r)} \\ \delta v_{\mathrm{p}}^{(r)} \end{bmatrix} = \begin{bmatrix} \lambda u_{\mathrm{p}}(k_1 r_{\mathrm{p}}^2 + k_2 r_{\mathrm{p}}^4 + ...) \\ \lambda v_{\mathrm{p}}(k_1 r_{\mathrm{p}}^2 + k_2 r_{\mathrm{p}}^4 + ...) \end{bmatrix} \tag{2}$$

where $k_1, k_2, ...$ are the radial distortion coefficients and $r_{\mathrm{p}} = \lambda\sqrt{u_{\mathrm{p}}^2 + v_{\mathrm{p}}^2}$. Typically, a larger number of coefficients produce better approximations of radial distortion but it takes more time and effort to determine the coefficient values.

Additionally, the centers of curvature of lens surfaces are not always strictly collinear. This introduces another type of distortion, called *decentering distortion* or *tangential distortion*, which has both a radial and a tangential component. The tangential distortion is modeled by Eqn. (3):

$$\begin{bmatrix} \delta u_{\mathrm{p}}^{(t)} \\ \delta v_{\mathrm{p}}^{(t)} \end{bmatrix} = \begin{bmatrix} 2\lambda^2 p_1 u_{\mathrm{p}} v_{\mathrm{p}} + p_2(r_{\mathrm{p}}^2 + 2\lambda^2 u_{\mathrm{p}}^2) \\ 2\lambda^2 p_2 u_{\mathrm{p}} v_{\mathrm{p}} + p_1(r_{\mathrm{p}}^2 + 2\lambda^2 v_{\mathrm{p}}^2) \end{bmatrix} \tag{3}$$

where p_1 and p_2 are the coefficients for tangential distortion.

Thus a proper model for accurately calibrating real-world cameras can be obtained by combining the pinhole camera model defined by Eqn. (1) with the distortion models defined by Eqns. (2) and (3) as:

$$\begin{bmatrix} u_{\mathrm{p}}^{(u)} \\ v_{\mathrm{p}}^{(u)} \end{bmatrix} = \begin{bmatrix} \lambda u_{\mathrm{p}} + \delta u_{\mathrm{p}}^{(r)} + \delta u_{\mathrm{p}}^{(t)} \\ \lambda v_{\mathrm{p}} + \delta v_{\mathrm{p}}^{(r)} + \delta v_{\mathrm{p}}^{(t)} \end{bmatrix} \tag{4}$$

All of the quantities of the matrix $\boldsymbol{K}_{\mathrm{p}}$ along with the distortion coefficients can be obtained by following the calibration technique outlined in [55, 56]. `OpenCV` and `MATLAB` use the techniques outlined in [55] and [56], respectively, for camera calibration with two tangential distortion coefficients p_1 and p_2 and up to a maximum of three radial distortion coefficients k_1, k_2 and k_3.

2. Omnidirectional Camera Model and Calibration.

Omnidirectional cameras have much larger FoVs and significant distortion because of the fisheye lenses they use. For viewing, the images must be undistorted into a simple pinhole projection model. The omnidirectional camera calibration and undistortion can be done using common `OpenCV` functions that implement the algorithm described in [57], which is based on the following equation:

$$^{\mathrm{o}}\boldsymbol{\mathcal{P}} = [^{\mathrm{o}}\boldsymbol{R},\ ^{\mathrm{o}}\boldsymbol{t}]\tilde{\boldsymbol{\mathcal{P}}} \tag{5}$$

The matrix $[^{\mathrm{o}}\boldsymbol{R},\ ^{\mathrm{o}}\boldsymbol{t}]$ is called the extrinsic parameter matrix for the omnidirectional camera, consisting of the rotation and the translation of the object plane in the world coordinate frame with respect to the omnidirectional camera coordinates. The pinhole projection coordinates, $[a_{\mathrm{o}}, b_{\mathrm{o}}]^{\mathrm{T}}$, are then mapped to the distorted coordinates using the following procedure:

If $^{\mathrm{o}}\boldsymbol{\mathcal{P}} = [x_{\mathrm{o}}, y_{\mathrm{o}}, z_{\mathrm{o}}]^{\mathrm{T}}$ then:

$$\begin{aligned} a_{\mathrm{o}} = x_{\mathrm{o}}/z_{\mathrm{o}} \text{ and } b_{\mathrm{o}} &= y_{\mathrm{o}}/z_{\mathrm{o}} \\ \Rightarrow \quad r_{\mathrm{o}}^2 &= a_{\mathrm{o}}^2 + b_{\mathrm{o}}^2 \\ \Rightarrow \quad \theta &= \tan^{-1}(r_{\mathrm{o}}) \\ \Rightarrow \quad \theta_{\mathrm{d}} &= \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8 + ...) \quad [\theta_{\mathrm{d}} = \text{Fisheye lens distortion}] \end{aligned}$$

where $k_1, k_2, k_3, ...$ are the fisheye lens distortion coefficients which are not to be confused with the radial distortion coefficients described in Eqn. (2). Then the distorted point coordinates are:

$$x_{\mathrm{d}} = \frac{\theta_{\mathrm{d}} a_{\mathrm{o}}}{r_{\mathrm{o}}} \qquad \text{and} \qquad y_{\mathrm{d}} = \frac{\theta_{\mathrm{d}} b_{\mathrm{o}}}{r_{\mathrm{o}}} \tag{6}$$

Finally, conversion to pixel coordinates gives:

$$\eta\tilde{\boldsymbol{p}}_{\mathrm{o}} = \boldsymbol{K}_{\mathrm{o}}[x_{\mathrm{d}}, y_{\mathrm{d}}, 1]^{\mathrm{T}} \tag{7}$$

The matrix $\boldsymbol{K}_{\mathrm{o}}$ takes the same form as $\boldsymbol{K}_{\mathrm{p}}$ and can be obtained during the omnidirectional camera calibration along with the fisheye distortion coefficients $k_1, k_2, ...$ There is no notion of radial or tangential distortion for fisheye lenses. To capture distortions more accurately, a higher number of coefficients can be used. For example, `OpenCV` uses four coefficients $k_1, ..., k_4$ when calibrating fisheye lenses. Using Eqns. (5)-(7), the omnidirectional camera frame can be calibrated and undistorted as shown in Fig. 4. The arbitrary scale factor η captures the depth information of the object plane with respect to the camera frame $\{\mathcal{F}_{\mathrm{o}}\}$. In Fig. 4 the edges of the undistorted image have been cropped to avoid division by $z_{\mathrm{o}} = 0$ for the pixel points near the edges; otherwise, one would have $a_{\mathrm{o}}, b_{\mathrm{o}} \to \infty$. It can also be seen in the undistorted image of Fig. 4 that near the edges, the undistortion is not perfect and results in the curving of straight lines.

Fig. 4 Original omnidirectional camera image (left) and undistorted omnidirectional camera image (right). Comparing the vertical red line in the undistorted image with the vertical door frame shows imperfect undistortion near the edges.

3. Transforming Between the Frames $\{\mathcal{F}_{\mathrm{o}}\}$ and $\{\mathcal{F}_{\mathrm{p}}\}$.

A *homography* matrix is a linear transformation that converts pixel or metric points from one frame to another. According to [54], two types of homographies relate one camera frame to another. The *Euclidean Homography* matrix $\boldsymbol{H}$ transforms 3D points in the euclidean space whereas the *projective homography* matrix $\boldsymbol{G}$ transforms pixel coordinates in the image space of one camera frame to another. The euclidean homography matrix can be obtained from a known camera displacement using:

$$\boldsymbol{H} = {}^{\mathrm{o}}\boldsymbol{R}_{\mathrm{p}} + \frac{{}^{\mathrm{o}}\boldsymbol{t}_{\mathrm{p}}\, {}^{\mathrm{o}}\boldsymbol{n}^{\mathrm{T}}}{d_{\mathrm{o}}} \tag{8}$$

Then the projective homography $\boldsymbol{G}$ can be obtained from the euclidean homography $\boldsymbol{H}$ using:

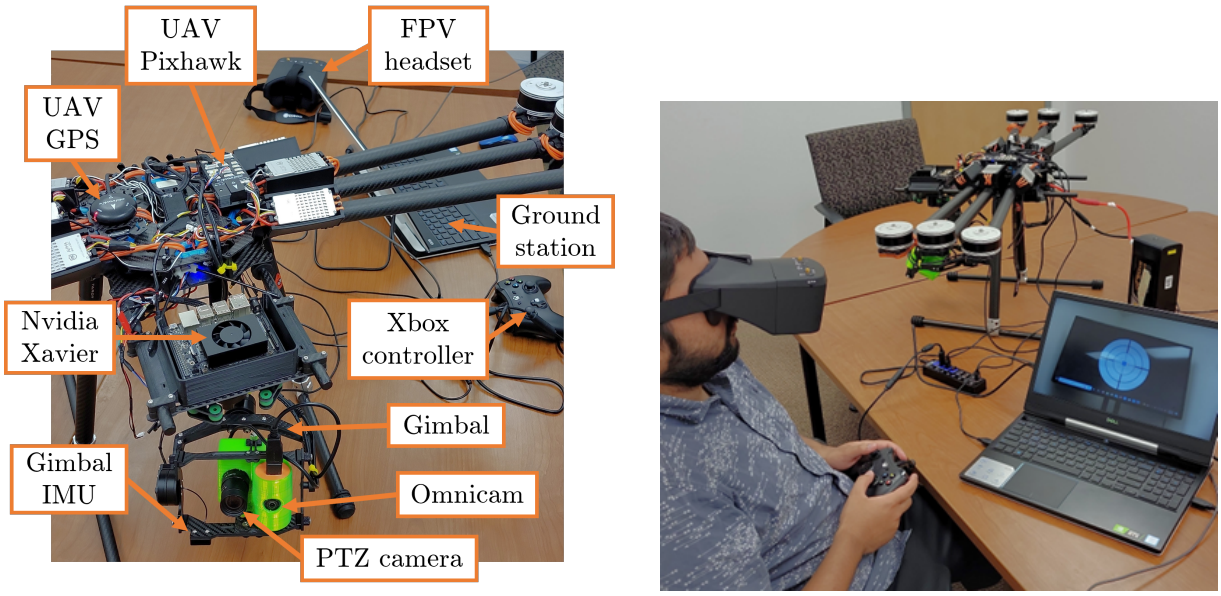
$$\boldsymbol{G} = \boldsymbol{K}_{\mathrm{p}}\boldsymbol{H}\boldsymbol{K}_{\mathrm{o}}^{-1} \tag{9}$$

Equation (9) takes vectors of pixel points in the undistorted omnidirectional camera image and gives the corresponding vectors of pixel points in the PTZ camera image. It is seen from Eqn. (8) that four quantities are required to calculate the homography matrix $\boldsymbol{H}$. In the application considered here, the translation vector, ${}^{\mathrm{o}}\boldsymbol{t}_{\mathrm{p}}$, between the frames is constant because the cameras are mounted on the same rigid body. If the omnidirectional camera remains fixed while the PTZ camera rotates with the gimbal, as in the case considered by Sakib et al. [48], then the value of the rotation matrix ${}^{\mathrm{o}}\boldsymbol{R}_{\mathrm{p}}$ can be obtained from the gimbal's inertial measurement unit (IMU). However, the perpendicular distance d_{o} from the

camera frame to the scene plane and the orientation of the plane ${}^{\mathrm{o}}\boldsymbol{n}$ are not readily available for an arbitrary scene and are quite difficult to obtain using computer vision methods. Because of such difficulties Eqn. (8) is rarely used in real-world computer vision algorithms. Instead, these algorithms use pixel-pixel correspondence by making use of feature detection techniques like LIFT, SURF, ORB, etc. The equation for finding the projective homography matrix is

$$\eta\tilde{\boldsymbol{p}}_{\mathrm{o}} = \lambda\,{}^{\mathrm{o}}\boldsymbol{G}_{\mathrm{p}}\tilde{\boldsymbol{p}}_{\mathrm{p}} \tag{10}$$

where ${}^{\mathrm{o}}\boldsymbol{G}_{\mathrm{p}}$ is the projective homography matrix that transforms pixel coordinates in the PTZ camera image to pixel coordinates in the omnidirectional camera image. It can be obtained by performing least square minimization of the detected feature points with robust outlier rejection techniques like random sample consensus (RANSAC) [58]. Details about determining the homography matrix using least square minimization can be found in computer vision textbooks [59, 60]. Even though Eqn. (10) is widely used in the computer vision community for determining the homography matrix, it could not be used in [48] because of the relative motion between the two camera frames, which results in a time-varying homography matrix. A recently developed algorithm called SuperGlue [61] is fast enough to detect features in real time but requires "constant illumination." This requirement implies that one should use similar cameras; the undistorted and cropped images from the omnidirectional camera do not exhibit the same illumination level as the PTZ camera; see Fig. 7b. As a result, these state-of-the-art AI-based feature detection algorithms produce inaccurate matches. To use Eqn. (10), a physical constraint is introduced: there is no relative motion between the two camera frames. Enforcing this constraint means that the homography can be estimated manually and will remain unchanged as long as the setup does not change. Figure 5a shows the setup used in this paper which depicts two different camera frames fixed relative to each other.



(a) Annotated system setup **(b) Laboratory testing**

Fig. 5 Overall system setup and an example telerobotic task performed by a participant. Here, both cameras are rigidly mounted to the gimbal, allowing a constant homography matrix.

C. Human Transfer Function

According to the references cited in Section II.C, to find the human transfer function a human operator can perform a variety of tasks such as tracking a step change in a command signal or tracking a desired path. One of the goals of this paper is to show that classical stability concepts can predict the maximum delay tolerance of a human operator using an aerial telerobotic system. To do this, it is important to accurately identify the human transfer function.

The first step in determining the human transfer function is to determine the transfer function of the system. Once the system's transfer function is identified, the response of the HMS can be evaluated to obtain the human transfer function. In doing so, it is assumed that both the human and the machine are linear elements and the overall system

response is a linear combination of each element with different time delays, as shown in Fig. 6. In this paper, *human* refers to the inspector performing telerobotic tasks, and *machine* refers to the gimbal-camera system. The term *gimbal* is used throughout the paper to refer to the combined gimbal-camera system.

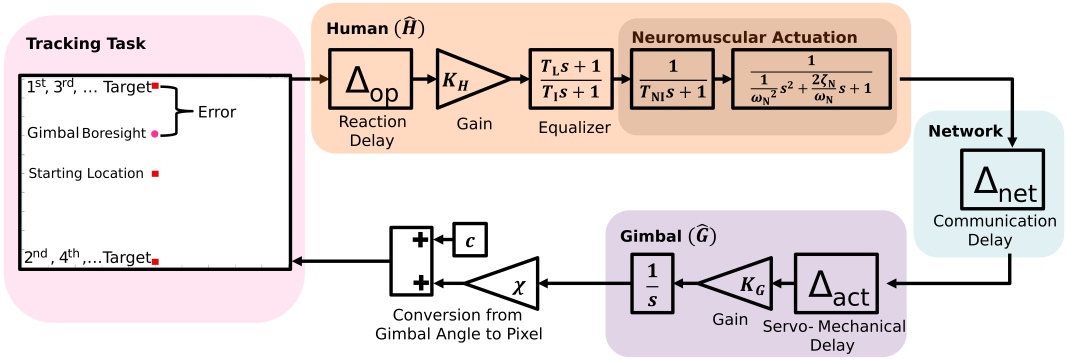


Fig. 6 Block diagram for the inspector-gimbal human/machine system, depicting the dynamic tracking task used to find the human transfer function.

To determine the gimbal's transfer function, the gimbal is excited in both the pitch and yaw axes using a sinusoidal angular rate command of increasing frequency, a *frequency sweep*, given as:

$$\Omega_{\text{cmd}} = A \sin\left(2\pi\left(\omega_{\text{i}} t + \frac{\omega_{\text{f}} - \omega_{\text{i}}}{2T_D} t^2\right)\right) \tag{11}$$

where Ω_{cmd} is the angular rate command sent to the gimbal, A is the amplitude of the angular rate command, T_D is the duration of the sweep, and ω_{i} and ω_{f} are the minimum and maximum frequency of the sweep, respectively. This particular frequency sweep is known as a linear sine sweep. The gimbal's IMU angles can then be compared with the corresponding commanded angular rates using `MATLAB` to determine the linear transfer function of the gimbal. While obtaining the transfer function of the gimbal-camera system it is assumed that the gimbal's internal controller is calibrated properly to reduce any IMU bias and drift. Thus the transfer function obtained here is the overall transfer function of the gimbal-camera system including the internal gimbal controller which maps the commanded angular rates sent to the gimbal and the corresponding gimbal IMU angles measured. The commanded signal and the measured and simulated responses are shown later, in Section VI.A, where it is found that the gimbal dynamics are well modeled by a transfer function from a given gimbal angular rate to the corresponding gimbal IMU angle that comprises an integrator, a gain, and an actuator delay:

$$\hat{G}(s) = e^{-\Delta_{\text{act}} s} \, K_G \, \frac{1}{s} \tag{12}$$

A properly designed target acquisition task can induce an operator to perform a manual frequency sweep in response to a dynamic target. This can be accomplished by presenting the operator with a target that alternates between the top and bottom of a screen at random but decreasing intervals. The operator's goal is to smoothly and quickly move the camera boresight using a handheld controller to capture the target on the screen, as shown on the left-hand side of Fig. 6. The target pixel location on the screen and the boresight pixel location from human movement can then be compared to obtain the response of the human operator. Of the several linear human models available, the McRuer-Krendel precision pilot model (PPM) is chosen for this study because according to [38] the PPM is valid over a broader range of frequencies than other models. The most common formulation of the PPM, which is used in the present work, is

$$\hat{H}(s) = \underbrace{e^{-\Delta_{\text{op}} s}}_{\text{Reaction Delay}} \underbrace{K_H}_{\text{Gain}} \underbrace{\left(\frac{T_{\text{L}} s + 1}{T_{\text{I}} s + 1}\right)}_{\text{Equalizer}} \underbrace{\left[\frac{1}{(T_{\text{NI}} s + 1)\left(\frac{1}{\omega_{\text{N}}^2} s^2 + \frac{2\zeta_{\text{N}}}{\omega_{\text{N}}} s + 1\right)}\right]}_{\text{Neuromuscular Actuation}} \tag{13}$$

where $\hat{H}(s)$ is the human transfer function and Δ_{op} is the reaction time of the human operator, which is treated as a pure time delay. The parameters T_{L}, T_{I}, and T_{NI} are the equalizer lead, equalizer lag, and neuromuscular lag time constants, respectively. Lastly, ω_{N} and ζ_{N} are the neuromuscular natural frequency and damping ratio, respectively. As mentioned in Section II.C, although the model is called the PPM, it is applicable to HMS operations other than a pilot and aircraft.

D. Temporal and Spatial Prediction

Section III.A gives an overview of how prediction works in the "temporal domain," that is, predicting the behavior of the gimbal-camera system over some delay time. This ability allows the algorithms to mitigate large network- and internet-based delays. Additionally, the algorithm can also predict over the image space – the "spatial domain." This enables the algorithms to predict low frame-rate imagery at a higher rate. As mentioned in Section II.A, computer vision algorithms are computationally expensive; their run times can be on the order of one or two seconds. This means that a single image from a set of video images can take multiple seconds to process which is not feasible for a real-time application like a structural inspection. Moreover, transmitting high-quality video imagery over a limited-capacity network can result in a lower video frame rate. As discussed in the following two subsections, the predictive algorithm can take low frame rate imagery and predict approximate image frames at a higher rate enabling real-time application. The prediction process relies on the assumption that the acquisition times for all the sensor and image data are known accurately. The known acquisition time is used to predict, modify or convert the corresponding sensor or image data from one time step to another. In principle, having accurately time-stamped data histories allows the algorithm to predict over infinitely large time delays. In practice, hardware limitations such as camera FoV, computer storage, or computer processing power limit the prediction horizon.

1. Temporal Prediction

Here, the objective is to predict the stereo camera frames based on past inputs that are delayed by some time τ_1. This requires predicting the gimbal IMU angles based on user inputs to the gimbal at the current time. Let $\phi_{\text{pred}}(t_k)$, $\theta_{\text{pred}}(t_k)$ and $\psi_{\text{pred}}(t_k)$ be the predicted roll, pitch, and yaw angles reported by the gimbal IMU at time t_k. These angles represent the orientation of the gimbal at t_k had there been no delays. Information available at time t_k includes the actual gimbal IMU angles $\phi_{\text{act}}(t_k - \tau_1)$, $\theta_{\text{act}}(t_k - \tau_1)$ and $\psi_{\text{act}}(t_k - \tau_1)$ which are delayed by τ_1 seconds. The gimbal angular rates $\dot{\phi}(t_k)$, $\dot{\theta}(t_k)$ and $\dot{\psi}(t_k)$ commanded at the current time step are also available. The current gimbal angles can be predicted as follows:

$$\begin{bmatrix} \phi_{\text{pred}}(t_k) \\ \theta_{\text{pred}}(t_k) \\ \psi_{\text{pred}}(t_k) \end{bmatrix} = \begin{bmatrix} \phi_{\text{act}}(t_k - \tau_1) \\ \theta_{\text{act}}(t_k - \tau_1) \\ \psi_{\text{act}}(t_k - \tau_1) \end{bmatrix} + \begin{bmatrix} \hat{\phi}(t_k) - \hat{\phi}(t_k - \tau_1) \\ \hat{\theta}(t_k) - \hat{\theta}(t_k - \tau_1) \\ \hat{\psi}(t_k) - \hat{\psi}(t_k - \tau_1) \end{bmatrix} \tag{14}$$

where $\hat{\phi}(t_k), \hat{\theta}(t_k),$ and $\hat{\psi}(t_k)$ are the estimated current angles and $\hat{\phi}(t_k - \tau_1), \hat{\theta}(t_k - \tau_1),$ and $\hat{\psi}(t_k - \tau_1)$ are the estimated delayed angles, respectively, based on the current angular rate commands obtained by taking the inverse Laplace transformation of the gimbal transfer function given in Eqn. (12). Let $\{Im\}_{\text{p}}(t_k - \tau_1)$ and $\{Im\}_{\text{o}}(t_k - \tau_1)$ be the delayed PTZ and omnidirectional camera images available at time t_k. The times $t_k, t_{k+1}, ...$ are the time steps for the predictive algorithm, which are more closely spaced than the data acquisition time steps (imagery and other sensor data), since the algorithm is assumed to run at a faster rate. Using Eqn. (14) the predicted and the actual IMU angles at the current time t_k can be obtained and the rotation matrix between them can be found. Here, it is assumed that the PTZ camera frame and the gimbal frame coincide. This important assumption enables the use of Eqn. (8) which has two unknowns: the normal vector ${}^{\text{p}}\boldsymbol{n}$ of the world plane in PTZ coordinates and the perpendicular distance d_{p} to the world plane. Because it is assumed that the two frames coincide, the translation is the zero vector: ${}^{\text{p}}\boldsymbol{t}_{\text{p}} = \boldsymbol{0}$. Eqn. (8) therefore gives

$${}^{t_k}\boldsymbol{H}_{t_k - \tau_1} = {}^{t_k}\boldsymbol{R}_{t_k - \tau_1} \tag{15}$$

The projective homography for the PTZ and the omnidirectional images becomes

$${}^{t_k}\boldsymbol{G}_{t_k - \tau_1} = \boldsymbol{K}_{\text{p}}\, {}^{t_k}\boldsymbol{R}_{t_k - \tau_1} \boldsymbol{K}_{\text{p}}^{-1} \tag{16}$$

Equation (16) takes pixel points of the (delayed) PTZ image $\{Im\}_{\text{p}}(t_k - \tau_1)$ and transforms them into predicted pixel points $\{Im\}_{\text{pp}}(t_k)$ Finally, the pixel location of the predicted PTZ image frame in the delayed omnidirectional camera image can be obtained using Eqn. (10) which denotes the predicted PTZ image at time t_k in the omnidirectional image of time $t_k - \tau_2$, defined as $\{Im\}_{\text{op}}(t_k)$.

2. Spatial Prediction

Here, the goal is to predict the perspective orientation of the currently available images, which are delayed by time τ_1, at a faster rate than the data acquisition rate. By assumption, the update rate of the algorithm is faster than the update

rate of the sensor and image acquisition data. This means the actual images and the corresponding gimbal IMU angles available for manipulation are going to be the same for many of the algorithmic time steps, $t_k, t_{k+1}, t_{k+2}, ...$, until new sensor and image data are available. Let t_k be the time at which a new PTZ image is obtained which is delayed by τ_1, denoted $\{Im\}_{\mathrm{p}}(t_k - \tau_1)$. Then, at the next algorithmic time step t_{k+1} the projective homography from Eqn. (16) becomes

$$ {}^{t_{k+1}}\boldsymbol{G}_{t_k-\tau_1} = \boldsymbol{K}_{\mathrm{p}}\, {}^{t_{k+1}}\boldsymbol{R}_{t_k-\tau_1} \boldsymbol{K}_{\mathrm{p}}^{-1} \tag{17} $$

Again, by using Eqn. (10) the predicted PTZ image at time t_{k+1} in the omnidirectional image of time $t_k - \tau_1$, defined as $\{Im\}_{\mathrm{op}}(t_{k+1} - \tau_1)$, can be obtained. This process can be repeated for the time steps $t_{k+2}, t_{k+3}, ...$ until new image data is available and the steps restart from t_k.

IV. Algorithm Implementation

A. System Setup

As mentioned in Section II.E, all the predictive display algorithms developed using a single camera setup suffer from the likely possibility that the predicted frame will fall outside the currently available image frame. This creates empty regions in the output image. Here, this problem is overcome by using a second, omnidirectional camera to replace the empty regions in the PTZ image with imagery obtained from the wider FoV camera. There are at least two possible methods for displaying the final predicted image to an operator: (i) a heterogeneous-stitch image or (ii) a split-screen image. The idea of a heterogeneous-stitch image was described in detail in Section III.A. For the split-screen setup, the images coming from the two different cameras are kept separate and are displayed to the operator simultaneously. In this case, the delayed PTZ image is the primary image that is presented to the operator without modification. The image from the omnidirectional camera is undistorted, cropped, and overlaid onto the primary PTZ image in a corner of the screen, for example. The predictive algorithm modifies this secondary image. When the operator issues a command, they may choose to focus on this secondary image to accomplish their task. The split-screen setup requires less computation since image stitching is not needed.

Table 1 Camera specifications

Parameters	Omnidirectional Camera	PTZ Camera
Model	Insta 360 Air	Webcamera USB
Focal Length (mm)	1.0	$5 - 50$
Sensor Size (mm)	3.3×3.3	5.42×3.07
Pixel Size (μm)	2.2×2.2	2.82×2.82
Resolution (px)	1504×1504	1920×1080
Mass (g)	26.5	330

Different camera models and parameters used in the experiments are shown in Table 1. A picture of the setup mounted on a UAV (stowed for transport) is shown in Fig. 5a. Table 2 contains the hardware specifications for the ground station, a Linux machine used for data acquisition and algorithm implementation. The heterogeneous stereo-vision predictive algorithm is implemented in real-time using ROS (Robot Operating System) which is a collection of tools, libraries, and conventions that support robotic hardware and software integration. In ROS all the sensor messages are marked with accurate timestamps which satisfies the assumption of known data acquisition times made in Section III.D.

To validate the effectiveness of the predictive algorithm, data was collected by performing human subject testing where the ground station received the ROS messages generated by the cameras and the gimbal through wired USB connections. The average transmission delay in these wired connection paths was very small (about $1 - 3$ ms). Thus, to demonstrate the predictive algorithm's capabilities in scenarios where delays may be more significant, an artificial network communication delay Δ_{net} was injected into the code. The gimbal used for the experiments was the Infinity MR-S2 three-axis gimbal manufactured by HD Air Studio [62] which uses the SimpleBGC gimbal controller [63]. A generic Microsoft Xbox controller interfaced with the gimbal using the standard `joystick_driver` package provided

by ROS. Commanded joystick inputs were first converted into angular rate commands using simple linear interpolations. Letting $\dot{\theta}_{\max}$ denote the maximum allowable gimbal slew rate and letting δ_{joy} denote the normalized maximum stick deflection value of the Xbox controller, the commanded gimbal pitch rate at time t_k was $\dot{\theta}(t_k) = \dot{\theta}_{\max} \cdot \delta_{\text{joy}}(t_k)$. Typically, $\delta_{\text{joy}} \in [-1, 1]$ which after scaling gives $\dot{\theta} \in [-\dot{\theta}_{\max}, \dot{\theta}_{\max}]$. The maximum angular velocity of the gimbal for the human subject tests was $\dot{\theta}_{\max} = 5 \text{ deg/s}$, which was empirically found to provide the best open-loop reference control of the gimbal by an operator. Similar conversions were done for the yaw axis. The roll axis was not perturbed since it is desirable to keep the horizon level for telerobotic inspection operations. Panning (or yawing) and tilting (or pitching) were the usual methods by which the camera achieved the desired orientation and also kept the horizon level. The commanded angular velocities were then delayed by a fixed time Δ_{net_1} using a C++ script to simulate the transmission delay that exists between the ground station and the gimbal in an actual telerobotic operation. Thus, if a command was given at time t it was received by the gimbal at time $t + \Delta_{\text{joy}} + \Delta_{\text{net}_1}$, though the timestamp of the gimbal messages was not modified. That is, the commanded timestamp was t, not $t + \Delta_{\text{joy}} + \Delta_{\text{net}_1}$. After receiving this delayed command, the gimbal began to move following its actuator delay Δ_{act} which was known a priori using the methods described in Section III.C. The movement of the gimbal and the corresponding changes seen in the images were instantaneous as the wired connections have very little communication delay. Thus another C++ script was run to add more artificial time delay Δ_{net_2} to simulate the network delay from the sensor to the ground station. As before, the message timestamps do not reflect this delay so that, according to the timestamps, the changes seen in the gimbal angles and the images at time $t + \Delta_{\text{joy}} + \Delta_{\text{net}_1} + \Delta_{\text{act}} + \Delta_{\text{net}_2}$ due to a commanded input at time t were actually obtained at time $t + \Delta_{\text{joy}} + \Delta_{\text{net}_1} + \Delta_{\text{act}}$. It is reasonable to assume that the same network is used for sending and receiving data which means the network delays between the ground station and the sensor are equal in both directions: $\Delta_{\text{net}_1} = \Delta_{\text{net}_2} = \Delta_{\text{net}}$. The same values were therefore used in the C++ scripts for injecting the delays.

The entire experiment was carried out on the ground, with wired connections among components and where disturbances and noise were minimal. This ground-based experiment mimics an inspection task carried out while the UAV is hovering. The network delays Δ_{net} were injected artificially into the system using C++ scripts which made use of the `TimeSequencer()` function in the `message_filters` package in ROS. This function guaranteed that ROS messages would be called in the temporal order in which they were received. When a specific delay is specified the `TimeSequencer()` filters out and stores the messages based on their timestamps until the messages are out of date by at least the specified delay amount. This feature enabled the creation of a simulated environment where the images displayed on the screen were delayed by at least the specified amount. It had been observed that the ROS `image_transport` package, responsible for publishing or subscribing to image messages, published both the PTZ and omnidirectional images at 30 Hz. Each of these image messages includes extensive data containing pixel intensities of three different color channels (red, green, and blue) and one alpha channel (a component that represents the degree of transparency or opacity of a color).

The PTZ camera generates image frames with a resolution of $1920 \times 1080 \times 4 = 8.294$ Megapixels. As each pixel requires 32 bits of data, approximately 31.64 Mbits of data transfer was required per PTZ camera image. Similarly, the omnidirectional camera sent $3008 \times 1504 \times 4 \times 32 \approx 69.04$ Mbits of data per frame. Since all the scripts for image acquisition, delay injection, image processing, and image prediction were run on a single ground station, it created a hardware bottleneck for the algorithm limiting the maximum algorithmic run-time frequency to 10 Hz. That is, all sensor and image data was acquired and the final output was displayed by the algorithm to the operator at no greater than 10 frames-per-second (FPS).

Table 2 Ground station specifications

Parameters	Specifications
CPU	Intel Core i7-8750H
GPU	GeForce RTX 2060
Memory (GB)	16
Operating System	Ubuntu 20.04

B. Predictive algorithm

The real-time implementation of the predictive stitching algorithm can be described in the following 8 steps. For the split-screen setup, which is faster and simpler than the heterogeneous-stitch process, only the first four steps must be performed.

Step 1. The gimbal IMU angles are obtained as unit quaternions, converted to Euler angles, and stored in a vector along with their respective timestamps. The most recent of these IMU angles are assigned as the pitch and yaw angles of the current and the predicted frames at time step t_k. (This initialization of the predicted frame orientation is temporary; it is updated in subsequent steps.)

Step 2. Using Eqn. (14) the commanded angular rates are integrated to obtain the predicted IMU angles, ($\theta(t_k)$ and $\psi(t_k)$). The actual gimbal angles required by the equation, which are delayed by τ, are the most recently stored IMU angles in Step 1.

Step 3. The algorithm then searches through the stored IMU angle vectors using the timestamps in the current images to find the gimbal angles ($\theta(t_k - \tau)$ and $\psi(t_k - \tau)$) corresponding to the currently displayed PTZ and omnidirectional image frames $\{Im\}_{\mathrm{p}}(t_k - \tau)$ and $\{Im\}_{\mathrm{o}}(t_k - \tau)$, respectively. Using Eqns. (15) and (16), the homography matrix ${}^{t_k}\boldsymbol{G}_{t_k-\tau}$ from the current frame to the predicted frame is obtained.

Step 4. Using Eqn. (10) the pixel location of the predicted PTZ image $\{Im\}_{\mathrm{p}}(t_k)$ in the delayed omnidirectional image $\{Im\}_{\mathrm{o}}(t_k - \tau)$ is obtained. This is approximately the same image as $\{Im\}_{\mathrm{p}}(t_k)$ but obtained by cropping and warping the available, delayed omnidirectional camera image. This image is denoted $\{Im\}_{\mathrm{op}}(t_k)$.

Step 4a. For the split-screen case, $\{Im\}_{\mathrm{op}}(t_k)$ from Step 4 is resized and displayed; see Fig. 7a.

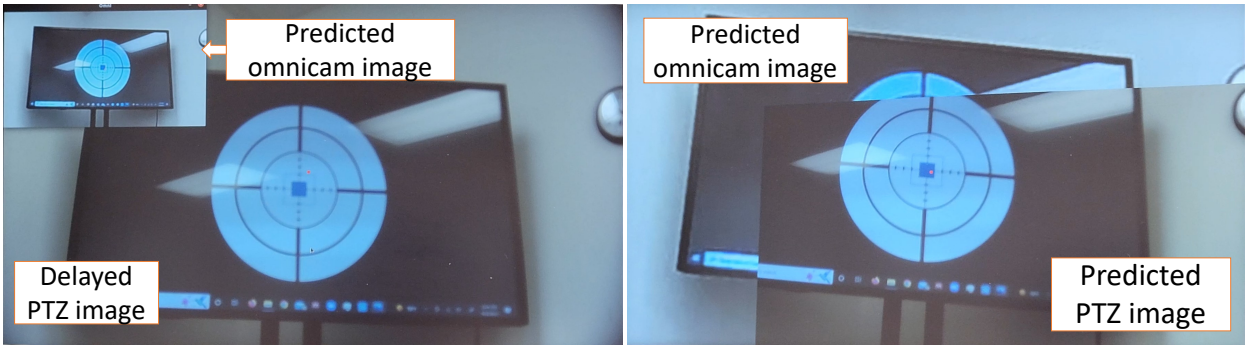
Step 5. The homography matrix ${}^{t_k}\boldsymbol{G}_{t_k-\tau}$ from Step 3 is used to warp the currently displayed (delayed) PTZ image $\{Im\}_{\mathrm{p}}(t_k - \tau)$ to the predicted PTZ camera image in the PTZ camera frame using the `OpenCV` function `cv::warpPerspective()`. This predicted image $\{Im\}_{\mathrm{p}}(t_k)$ contains empty regions which must be filled.

Step 6. Using Eqn. (10), the pixel coordinates of the FoV of the currently displayed (delayed) PTZ image $\{Im\}_{\mathrm{p}}(t_k - \tau)$ contained within the FoV of the currently displayed omnidirectional camera image $\{Im\}_{\mathrm{o}}(t_k - \tau)$ are found. The resultant image, denoted $\{Im\}_{\mathrm{op}}(t_k - \tau)$, is similar to the currently displayed (delayed) PTZ image but obtained from the currently displayed (delayed) omnidirectional camera image.

Step 7. The image $\{Im\}_{\mathrm{op}}(t_k - \tau)$ obtained in Step 6 is subtracted from $\{Im\}_{\mathrm{op}}(t_k)$, obtained in Step 4, using the `OpenCV` function `cv::subtract()`. This subtracted image contains the empty portion of the predicted PTZ image $\{Im\}_{\mathrm{p}}(t_k)$ in Step 5.

Step 8. The subtracted image in Step 7 is added to the predicted PTZ image $\{Im\}_{\mathrm{p}}(t_k)$ in Step 5 using the `OpenCV` function `cv::add()` to complete the stitching. This image stitches regions from the small FoV PTZ camera together with regions from the wide FoV omnidirectional camera.

Step 8a. The heterogeneous-stitch image, with the same dimensions as a normal PTZ camera image, is displayed without further processing; see Fig. 7b.



(a) **(b)**

Fig. 7 (a) Screenshot of the split-screen display. (b) Screenshot of the stitched image display.

V. System Usability Determination

A. Study Design

A human performance study was designed to test the effectiveness of the proposed predictive display system in mitigating display delays in the FPV environment. The study was designed to reflect common maneuvers that a typical bridge inspector might perform while using the gimbal-camera system. Participants wore an Eachine EV800D FPV headset and completed several tasks under four different test cases:

(i) Case 1: 2 s delay at 10 FPS image acquisition and 10 FPS predictive display output.
(ii) Case 2: 2 s delay at 1 FPS image acquisition and 10 FPS predictive display output.
(iii) Case 3: 6 s delay at 10 FPS image acquisition and 10 FPS predictive display output.
(iv) Case 4: 6 s delay at 1 FPS image acquisition and 10 FPS predictive display output.

The delays in the four cases above concern the algorithm's ability to predict correctly over different delays (termed *temporal prediction* in Section III.D) whereas the different image acquisition FPS values concern the algorithm's ability to predict over different image acquisition frame rates at an identical output frame rate for all cases (termed *spatial prediction* in Section III.D). The reason for testing with 2 s and 6 s of total delays is because 2 s of delay was close to the maximum unmitigated delay that can be tolerated within the human/machine system, as discussed later in Section VI.B, and 6 s of delay is close to the worst-case delay that can be accommodated with this stereo camera setup, found through experimental testing. The *worst-case delay* is defined as the largest delay that must be accommodated by the predictive display algorithm, corresponding to the operator's worst-case input, i.e., a sustained maximum gimbal slew rate command. For any greater time delay, the operator could feasibly command the PTZ camera to image a region for which no backfill imagery from the omnidirectional camera is available. For reference, the empty white region in the PTZ camera FoV in Fig. 2b, indicated by the red rectangle, is the region for which no backfill PTZ imagery is available. Due to a hardware bottleneck, the predictive algorithm was able to run at a maximum rate of 10 Hz corresponding to an output display frame rate of 10 FPS independent from the image acquisition frame rate.

In each of cases (i)-(iv), a precision tracking task (a) was performed followed by a target acquisition task (b). The Cooper-Harper rating scale was used to obtain a subjective evaluation of the system. To use the C-H HQR scale, an operator must judge whether the system being used allows the adequate performance of the prescribed task and must assess the workload required to do so. Participants were required to complete the tasks with as much precision as possible in the spatial domain. Bounds around the desired path denoting desired and acceptable tracking forced high-gain operators to proceed carefully in order to emphasize tracking accuracy. Minimum completion times for desired and adequate performance forced low-gain operators to move quickly in order to complete the task within an acceptable time.

In the precision tracking task, the participants were asked to track a stepped line from left to right, as shown in Fig. 9a. The thick blue line was the *desired* bound whereas the thin blue line was the *adequate* bound in the spatial domain. The center of the displayed image was augmented with a red dot, indicating the PTZ boresight, which the operator controlled in order to track the line as closely as possible. To achieve the desired performance, participants had to ensure that the red dot (or cursor) at least touched the thick blue line while completing the tracking task. For the tracking task, a limit of five excursions was set, meaning if the red dot exceeded the desired or adequate bounds more than five times in a given experiment, the experiment would end with a declaration that desired performance was not attained. If the participants failed to reposition the cursor within the bounds after an excursion – for example, if they continued to move the cursor toward the goal without trying to reposition it within the boundaries – then the performance ratings were automatically dropped to the next tier in terms of the spatial bounds. The temporal bounds were specified as a 40 s maximum completion time for desired performance and a 70 s maximum completion time for adequate performance. Additionally, an absolute maximum time of 90 s was established; failure to complete the task within this time limit meant the system was automatically assigned the worst rating of 10 at the tested delay and frame rate. This cutoff time limit was implemented to limit the total time required to complete the tasks. Otherwise, there was a possibility that participants might take a very slow approach to finishing the tasks and then assign ratings that are uncharacteristic of the system's performance. This limit also reflects the reality that an inspector has limited time to complete an inspection. This cutoff time limit criterion is a modification of the actual C-H HQR criterion, in which a rating of 10 indicates that "control will be lost during some portion of the required operation" – see Fig. 8. Since the notion of "loss of control" is unclear in the case of a slow-moving gimbal-camera system the definition was modified. The same spatial and temporal bounds for *desired* and *adequate* performance were used for all test cases, with or without prediction.

In the target acquisition task the participants were asked to acquire a target by placing the red boresight indicator

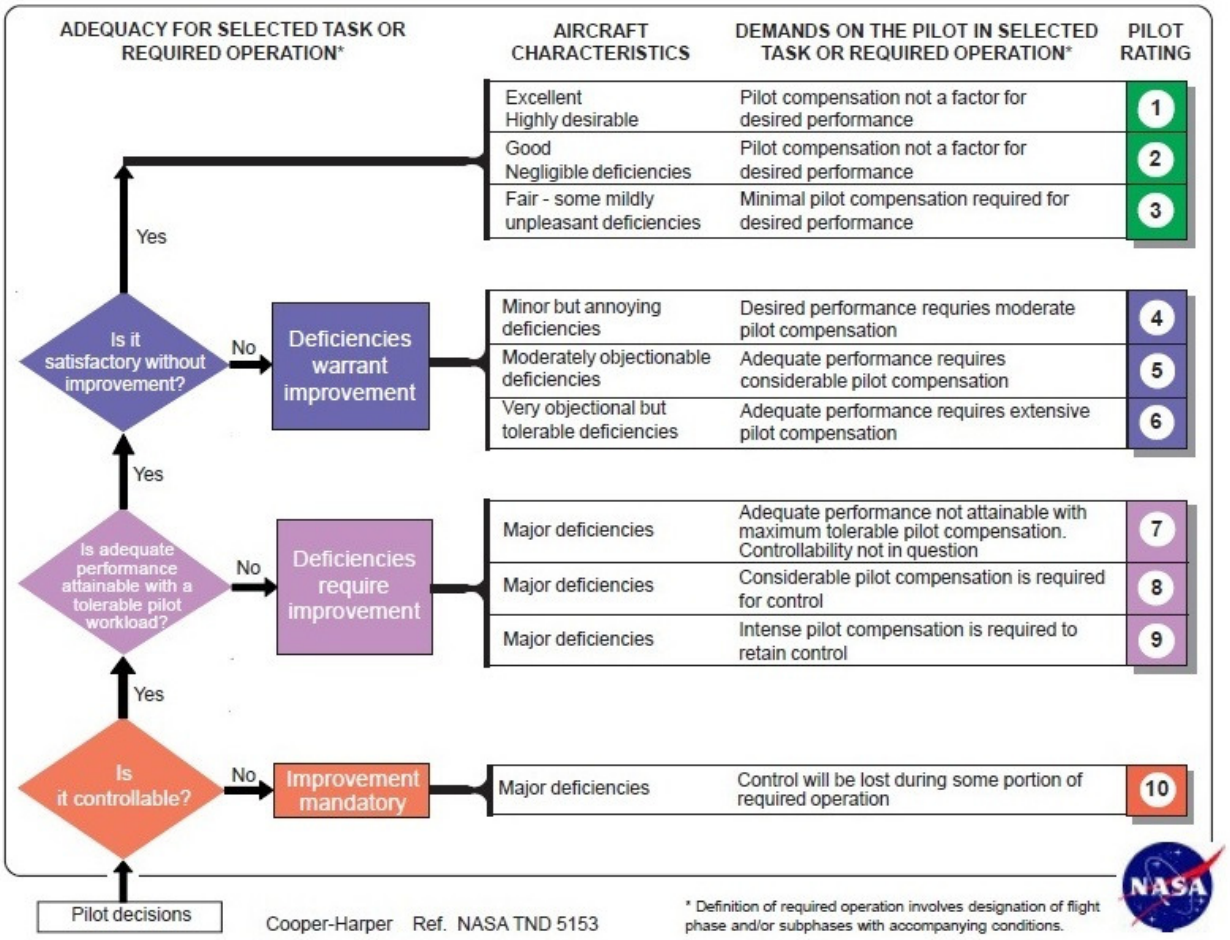
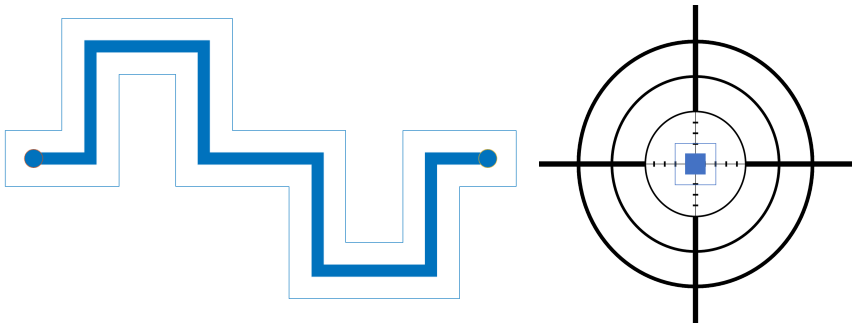


Fig. 8 Cooper-Harper Handling Qualities Rating (HQR) scale. [30]

in the image at the center of a bullseye, starting from a given initial position as fast as possible; see Fig. 9b. Just like before, the thick and the thin blue squares in the image indicate desired and adequate performance in the spatial domain, respectively. The temporal bounds were set to 12 s for desired performance, 15 s for adequate performance, and 30 s for the cutoff time. Again, as in the tracking task, the spatial and temporal bounds remained unchanged for all the different test cases with or without prediction. To claim the system exhibits desired performance in an assigned rating, a participant must have achieved desired performance with respect to both the temporal and spatial specifications.

To remove any potential bias in the ratings, the participants were not told the amount of delay or the frame rate being applied. Each participant had to perform 12 tracking tasks (a), one for each of the four cases (i)-(iv) with and without the predictive display algorithm running. The predictive display was again split into two parts based on the type of output used – split-screen vs heterogeneous-stitch. Participants then performed 12 target acquisition tasks (b), categorized in the same way as the tracking tasks. Data obtained from each experiment included the participant-assigned handling qualities ratings and the time required to complete each task. Additionally, in the tracking task, the gimbal's IMU angles were recorded to plot the cursor path for each trial; see Fig. 15. These cursor position histories were used to objectively assess the task accuracy under different setups. The angles in deg were converted to pixels using proper 3D to 2D camera calibration matrices, taking into account the rotation and translation of the camera with respect to the screen where the task was displayed. The experimental test plan also allowed for the assessment of motion sickness symptoms in participants, induced by wearing an FPV headset. However, none of the participants reported feeling any motion-sickness effects.



(a) (b)

Fig. 9 Images used for the (a) precision tracking and (b) target acquisition tasks.

B. Study Population

Seven participants, A through G, of varying ages, experience, and skill levels took part in the study. Using participants with widely varying skills allows one to assess whether experience level influences the objective and subjective evaluation of system performance. Such an assessment is important if a technology is to be broadly acceptable within a prospective user community, such as the community of bridge inspectors. C-H HQR data from participant A was used to establish a baseline rating for the system and to set the desired and adequate spatial and temporal bounds. Participant A was a trained US Air Force test professional with prior experience using the C-H HQR scale who self-identified as a moderate-high gain participant.

Since no participant other than participant A was proficient at rating the system using the C-H HQR scale, all were trained using example cases. The baseline ratings for the example cases were chosen by participant A. Each participant was given sufficient time to become familiar with the system and the controller. Then each was asked to perform some tracking and targeting tasks for several different example cases spanning the range of system usability (good, bad, and average). The baseline rating for each of those cases was then revealed to the participants so that they could calibrate their notions of system usability with the C-H HQR assessments by participant A. After the participants were shown five cases of good, bad, and average handling quality ratings, they were tested on three different example cases and asked to rate the system after completing a task. Their ratings were then matched with the baseline rating obtained from participant A. If the participant's ratings fell within one unit of the baseline rating, they were considered proficient in rating the system. All participants showed proficiency in rating the system using the C-H HQR scale in the tests.

VI. Results

This section provides a discussion of the results of the experimental investigation. A publicly accessible `GitHub` repository [64] provides all the `ROS`-based `C++` scripts for doing the predictive display algorithm. This repository contains all the scripts necessary to perform heterogeneous camera calibration, stitched screen display, split-screen display, and delay injection. Additionally, a publicly accessible `Google drive` [65] contains example videos of the dynamic tracking task used in identifying the human transfer function and of the target acquisition and tracking tasks obtained during the human subject testing.

A. Obtaining the Human Transfer Function

Following the method provided in Section III.C, the gimbal transfer function was found by performing a frequency sweep using Eqn. (11) with $A = 10$ deg/s, $\omega_i = 0.1$ Hz, $\omega_f = 4$ Hz, and $T_D = 30$ s. The maximum frequency value, $\omega_f = 4$ Hz, was chosen based on the observation that the quickest joystick actuation by a thumb was about 2 Hz. The gimbal dynamics were identified from zero frequency up to twice the frequency at which a human operator might be expected to operate it to ensure the modeling results would be valid for the intended use.

Figure 10 shows the frequency sweep, the measured gimbal response, and the simulated response using the transfer function model

$$\hat{G}(s) = e^{-0.09s} 0.98 \frac{1}{s} \tag{18}$$

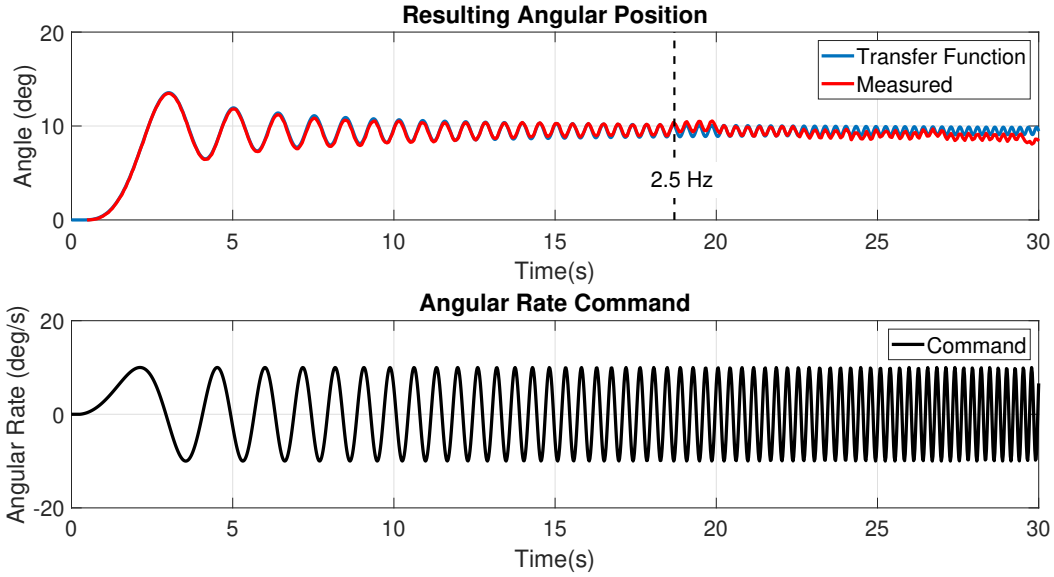


Fig. 10 Pitch response of the gimbal and the model $\hat{G}(s)$ to the frequency sweep used for model identification.

which is in the form of Eqn. (12). Thus, the gimbal was found to have a servo-mechanical delay, $\Delta_{\text{act}} = 90$ ms, and a proportional gain, $K_G = 0.98$. The gimbal transfer function $\hat{G}(s)$ was validated against measured data in both the pitch and yaw axes at $A \in \{10, 20\}$ deg/s and $\omega_f \in \{4, 8\}$ Hz and was found to be consistent. It can be seen in Fig. 10 that the measured response deviates momentarily from the transfer function at 2.5 Hz. This was observed during testing and is likely the result of the unmodeled dynamics of the gimbal. However, as discussed, since this is below the operating frequency of the human operator it was determined to not be an issue.

To find the human transfer function an operator performed several dynamic tracking tasks as described in Section II.C and depicted in Fig. 11. The target alternated between the top and bottom of a screen and dwelled at that location based on the following equations:

$$\begin{aligned} D_{\text{dwell}} &= D_{\min} + (D_{\max} - D_{\min})r \\ d_{\text{slope}} &= \frac{\phi_{t_\text{i}} - \phi_{t_\text{i}}}{t_\text{i} - t_\text{f}} \\ D_{\text{scaled}} &= D_{\text{dwell}}(1 + d_{\text{slope}}(t - t_\text{i})) \end{aligned} \tag{19}$$

where D_{dwell} is the unscaled dwell time, $D_{\min}$ is the minimum dwell duration, $D_{\max}$ is the maximum dwell duration, and $r \in [0, 1]$ is a uniformly distributed random number. Additionally, d_{slope} defines the linear decrease in dwell time with ϕ_{t_i} representing the percent of dwell time at the start of the task, ϕ_{t_f} the percent of dwell time at the end of the task, and t_i and t_f the start and end times of the task, respectively. Lastly, D_{scaled} is the resulting dwell time of the target as presented to the operator. For the task shown in Fig. 11, the values used were $D_{\min} = 5$ s, $D_{\max} = 6$ s, $\phi_{t_\text{i}} = 1$, $\phi_{t_\text{f}} = 0.1$, $t_\text{i} = 0$ s , and $t_\text{f} = 110$ s. An initial pause of 5 s was used before the first movement of the target. This process of randomizing the dwell time of the target was accomplished so as to prevent the operator from "anticipating" the target's movement by learning a pattern.

To establish the conversion from the gimbal angle to the pixel value on the screen, it was found that the deflection of the gimbal was less than 6 deg to reach the limits of the screen. Thus the classical "small angle assumption" was applied and the linear relationship between the gimbal angle in pitch and yaw to the pixel value χ was found to be 38.42 px/deg. The bias term c shown in Fig. 6 was 546 px which was the vertical center of the target screen and the starting location for the tracking task.

For each task, the operator was directed to smoothly and quickly move the boresight of the gimbal to track the target. To simplify the transfer function fitting process, the operator was asked to not make additional "fine-tuning" adjustments if the boresight was near the target. As is shown in Fig. 11, the operator had a tendency to overshoot the target slightly. Since human responses are complex and inconsistent, five trials were conducted and the results were evaluated to find the human transfer function that most closely matched the measured response.

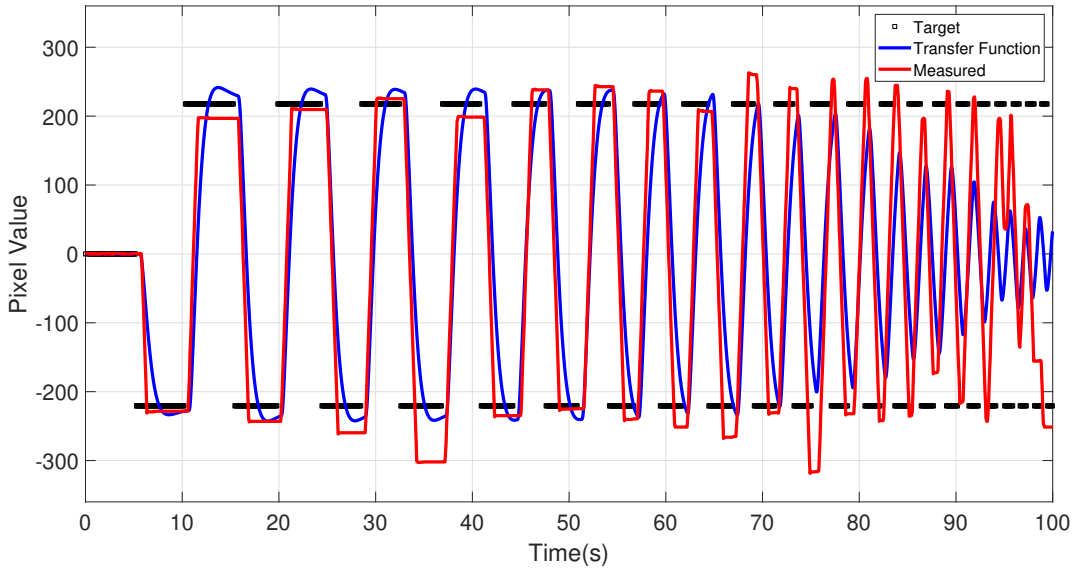


Fig. 11 Pixel motion response of the human/machine system and the model $\hat{G}(s)$ to the dynamic tracking task used for model identification.

To find the human transfer function, nominal constant values from [38] were chosen initially. These values are given in Table 3. The constants were then adjusted to achieve the HMS response that matched the set of tracking task trials. The neuromuscular terms (T_{NI}, ω_{N}, and ζ_{N}) were left at the nominal value. The reaction delay time constant τ_H was found by examining the average reaction time to the moving target which was 0.3 s. The gain, equalizer lead, and equalizer lag terms were iterated upon to find values that most closely match the rise time and overshoots seen in the measured tracking tasks. As can be seen in Fig. 11, the overall HMS transfer function matches the operator up to a target switching frequency of about 1 Hz (2 rad/s), which captures the desired operating speed without erratic movement.

Table 3 Nominal constant values in Precision Pilot Model [38] and values used in this research

Constant	Units	Nominal Value	Value Used
Δ_{op}	s	0.1-0.5	0.3
K_H	-	machine/task dependent	0.025
T_{L}	s	0.2-6.1	3
T_{I}	s	0.1-3.7	3.7
T_{NI}	s	≈ 0.1 s	0.1
ω_{N}	rad/s	≈ 20	20
ζ_{N}	-	≈ 0.7	0.7

Summarizing, the human transfer function was determined to be

$$\hat{H}(s) = \underbrace{e^{-0.3s}}_{\text{Reaction Delay}} \cdot \underbrace{0.025}_{\text{Gain}} \cdot \underbrace{\left(\frac{3s+1}{3.7s+1}\right)}_{\text{Equalizer}} \cdot \underbrace{\left[\frac{1}{(0.1s+1)\left(\frac{1}{(20)^2}s^2 + \frac{2\times 0.7}{20}s + 1\right)}\right]}_{\text{Neuromuscular Actuation}} \tag{20}$$

The complete HMS model was used for performance prediction as described in the following section.

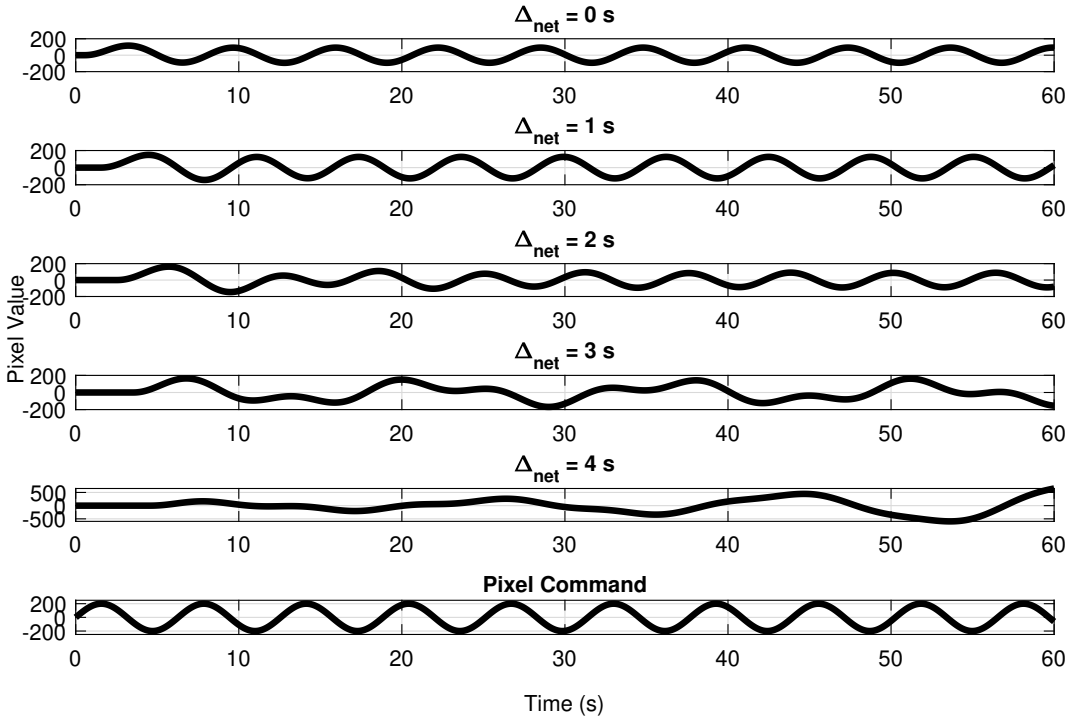


Fig. 12 Response of the inspector-gimbal HMS to a sinusoidal input for network delays of 0, 1, 2, 3, and 4 seconds.

B. HMS Performance Predicition

With the gimbal transfer function $\hat{G}(s)$ and the human transfer function $\hat{H}(s)$ determined, the overall HMS depicted in Fig. 6 can be modeled as

$$\hat{S}(s) = \hat{H}(s)e^{-\Delta_{\text{net}}s}\hat{G}(s)\chi \tag{21}$$

where Δ_{net} is the communication delay in the network between the Xbox controller and the gimbal. The system $\hat{S}(s)$ can be used to predict the closed-loop performance of the operator-gimbal HMS. Of primary concern in this research is the performance of the HMS in the presence of increasing time delay. As such, the following analysis focuses on the *delay margin* of $\hat{S}(s)$ which is defined as "the largest time delay such that, for any delay less than this value, the closed-loop stability is maintained" [66].

Following the dynamic tracking task experiments, it was desirable to reduce the maximum angular rate of the gimbal from $10\ \text{deg/s}$ to $5\ \text{deg/s}$. This allowed the operator to have more precise control of the gimbal's motion while still achieving adequately fast operation. This reduction was analogous to reducing the gimbal transfer function gain, K_G, from 0.98 to 0.49. Using the `MATLAB` command `allmargin()` with the system $\hat{S}(s)$ with $\Delta_{\text{net}} = 0$, the delay margin was $3.03\ \text{s}$. As can be seen in Fig. 12, the system exhibits a predictable response to a sinusoidal input up to a delay of $2\ \text{s}$. At $3\ \text{s}$ of delay the system response is stable but is no longer able to track the reference command. At $4\ \text{s}$ of delay the system is unstable. Based on a qualitative assessment of the system response, the maximum unmitigated delay that would be tolerable for this system is predicted to be about $2.3\ \text{s}$. This prediction aligns with the results of the human subject testing presented in Section VI.D.

C. Camera Calibration Results

Following the camera calibration methods outlined in `OpenCV` the important matrices obtained for camera calibration and frame transformation are given below:

$$\boldsymbol{K}_{\text{p}} = \begin{bmatrix} 2120.49 & 0.0 & 981.10 \\ 0.0 & 2119.06 & 535.16 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \boldsymbol{K}_{\text{o}} = \begin{bmatrix} 445.18 & 0.0 & 721.22 \\ 0.0 & 445.88 & 821.94 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, {}^{\text{p}}\boldsymbol{G}_{\text{o}} = \begin{bmatrix} 3.66 & .05 & -1999.16 \\ -.28 & 3.78 & -2018.67 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

D. Subjective Performance Evaluation

Figures 13 and 14 graphically display the subjective and objective data collected for the tracking and target acquisition tasks, respectively, for all four of the test cases labeled (i)–(iv) in Section V.A. In both figures, the graphs in the left column represent the C-H HQR ratings given by the seven participants and the graphs in the right column denote the total time taken while completing the tasks. The topmost row of graphs represents the situation in which no prediction algorithm was running. The middle row of graphs represents the situation in which the image prediction is displayed in a split-screen display, and the bottom row of graphs represents the case where predicted imagery is stitched with omnidirectional imagery. The solid markers in the C-H HQR graphs in the left column represented the mode of all the ratings, that is, the rating that was selected most often. The mode was used rather than the mean to produce an integer value consistent with the C-H HQR scale. Observing the subjective performance data in Figs. 13a, 13c, 13e, 14a, 14c, and 14e it is clear from the modes of the data that the participants preferred the heterogeneous-stitch display, followed by the split-screen display. The participants were unanimously dissatisfied with the system's performance when no predictive imagery was presented.

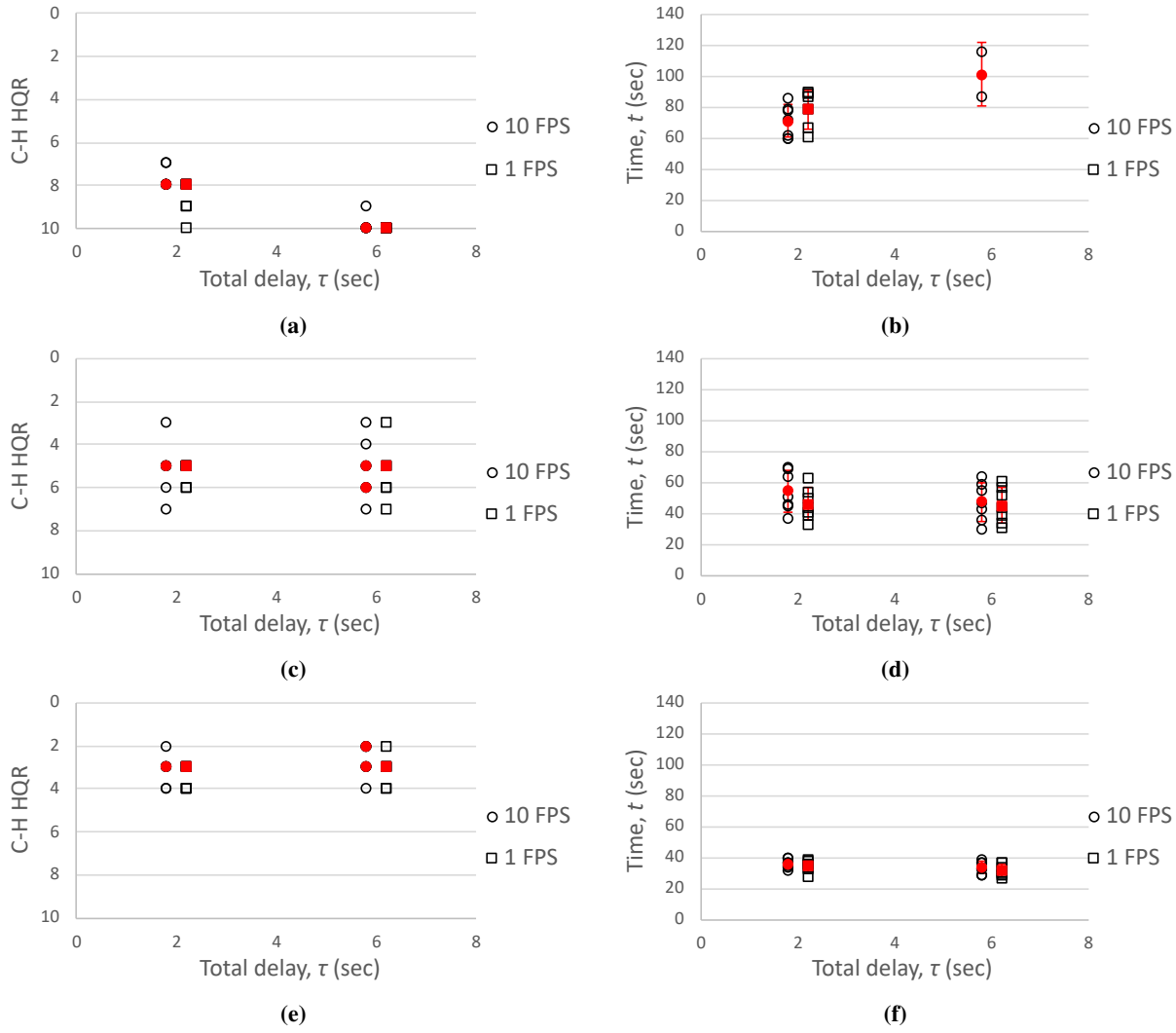


Fig. 13 Subjective ratings and completion times for the *tracking* task.

In Section VI.B, it was determined that without any active prediction to mitigate delays, a human operator would be able to tolerate about 2.3 s of delay. Figs. 14a and 13a indicate that most operators rated the system as 7 or 8 when the unmitigated delay was 2 s and 10 when the delay was 6 s. Referring to the rating descriptions in Fig. 8, a rating of 7 or 8 means the system is not adequate to complete the given tasks and considerable to intense operator compensation is

required to remain on track and complete the task within the time limit. As discussed in Section V, for the purpose of this study, a rating of 10 in the C-H HQR scale means that the system is not capable of completing the required tasks even with intense operator compensation within the cutoff time. Thus, the analysis of Section VI.B, based on HMS modeling, agrees with the subjective ratings of participants for the case where no predictive imagery is provided.

The split-screen setup yielded modal ratings of 5 and 6 for the tracking task with both 2 and 6 s delays. For the target acquisition tasks, the split-screen setup yielded modal ratings of 3 and 4 when the delay is 2 s and 5 when the delay is 6 s. Thus, in general, the performance of the split-screen setup can be classified as *adequate*, sometimes allowing *desired* performance at lower delays. The heterogeneous-stitch setup yielded modal ratings of 2 and 3 for 2 s and 6 s delays, respectively, for the target acquisition task. For the tracking task, it yielded ratings of 2 at 2 s of delay and $2-3$ at 6 s of delay. These results indicate the system performance using the heterogeneous-stitch setup is *desired*, that is, participants were able to achieve desired performance with this setup for all the tasks and test cases. Interestingly, dropping the incoming video frame rate from 10 FPS to 1 FPS, to simulate the case of imagery acquired at low FPS due to network or computational bottleneck, did not have any noticeable effect on the system's performance, according to the modes of the subjective ratings; see the ratings for the 10 FPS and 1 FPS cases in Figs. 13a and 14a. We hypothesize that the performance drop due to the large delays in the system was so high that participants did not notice the additional performance drop due to a lower frame rate.

Figures 14a, 13c and 13e show a bimodal character in the ratings. This distribution of ratings is unavoidable when using the C-H HQR scale and originates because of a human operator's subjective interpretation of the rating scale. For example, some participants may assign the system a rating of 3 for a particular task because they felt that minimal compensation was required to achieve desired performance while others might assign the system a rating of 4 for the same task because they felt the compensation required for the task was moderate. These multi-modal behaviors can be reduced by selecting participants who are experienced with the type of system being tested and with the C-H HQR rating process. Except for participant A, none of the other participants had any prior experience working with this system and only participants A, F, and G had prior experience using the C-H HQR scale. In any case, this bi-modal character did not affect the overall performance rating of the system. That is, the modal ratings did not span multiple performance categories (such as *desired* and *adequate*) for a given task.

The reason participants preferred the heterogeneous-stitch display over the split-screen display may be explained by the observations listed below. This list was generated by asking the participants to express their verbal opinions after all the data collection procedures were complete.

- The split-screen display had the predicted image displayed at one corner of the screen. This caused the participants to focus on a particular corner of the screen which took their focus away from the delayed primary display. It is possible that an inspector might miss important visual information if their focus is elsewhere in the screen.
- The split-screen display was smaller in size and more effort was required to accomplish the tasks in a small, lower-resolution display.
- Overlaying the PTZ image with the predicted omnidirectional split-screen image covered some regions in the PTZ image which might have contained points of interest.
- The primary PTZ image in the split-screen setup was delayed by some time. To accomplish the required tasks, the participants would rely on the predicted image during movement, which would be reflected on the primary image after the delay time had passed. This delayed movement in the background created minor distractions and the overall process felt unnatural to the participants.

E. Objective Performance Evaluation

The rightmost columns in Figs. 13 and 14 represent the total time taken in all test cases (i)–(iv) by all participants to complete the tracking task (a) and the target acquisition task (b), respectively. Similar to the discussion in Section VI.D, the top row contains completion times with no predictive display, the middle row displays completion times with the split-screen setup, and the bottom row denotes completion times for the stitched setup. The results discussed in this section are indicated in Figs. 13b, 13d, 13f, 14b, 14d, and 14f. The solid red markers in these figures represent the mean completion times for all the participants. As expected, the objective data mimic the subjective performance ratings, showing that the heterogeneous-stitch display is the most efficient of the tested setups followed by the split-screen setup and then the setup using no predictive display. The standard deviation in the data is found to be small for the stitched case and increases for the split-screen case and the no predictive display case. This is indicative of the fact that participants having different skill sets and expertise can perform almost at the same efficiency level when the predictive display is running. In Fig. 13b there are no data displayed for the case of a 6 s delay at 1 Hz and only two data points are

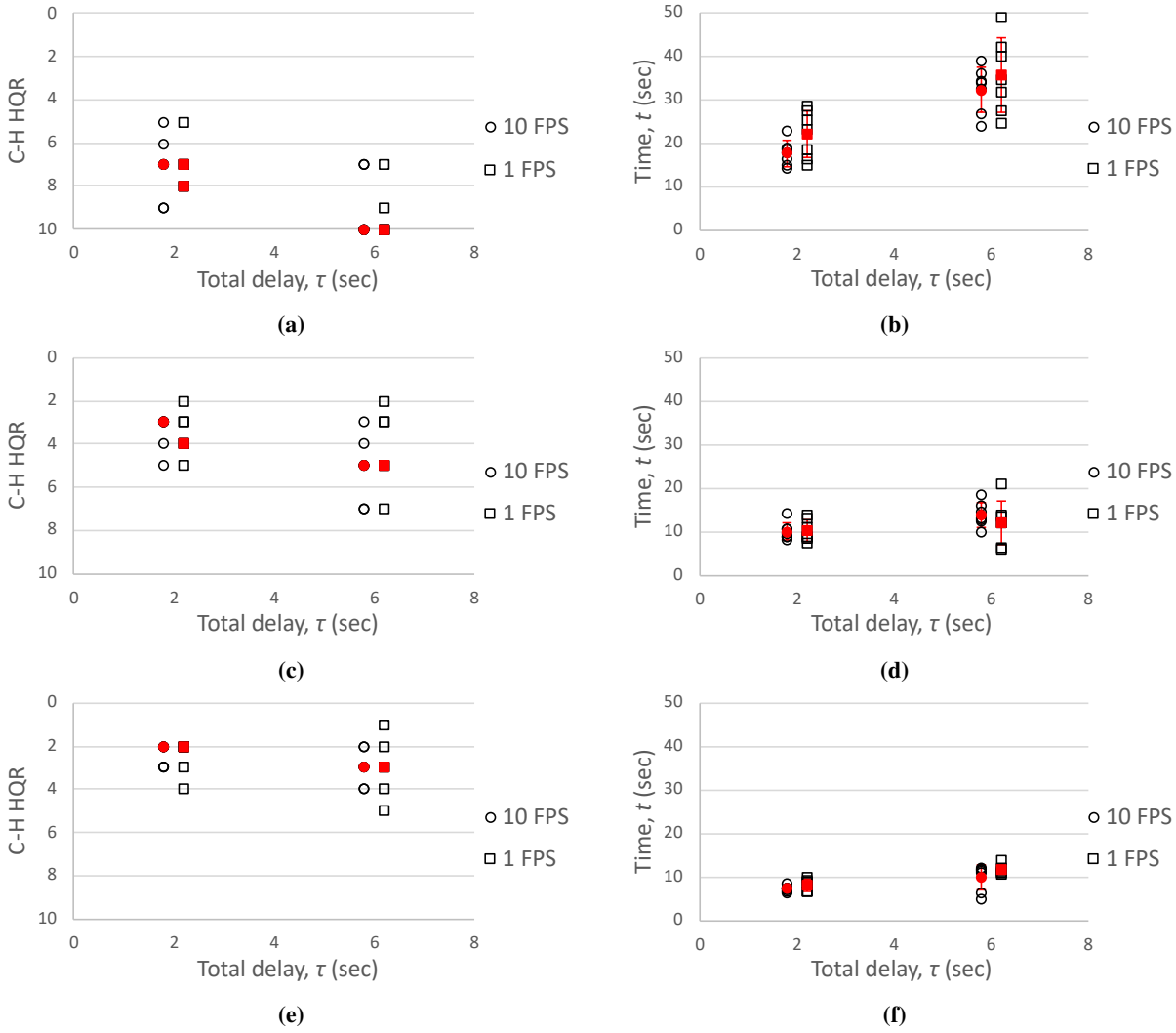


Fig. 14 Subjective ratings and completion times for the *target acquisition* task.

displayed for the case of a 6 s delay at 10 Hz as none of the participants were able to complete the tasks in the former case and only two participants were able to complete the task in the latter case.

For both the tracking and the target acquisition tasks the mean task completion time increased as the delays were increased when there was no predictor running. However, when the predictor was running, for both tasks, increasing the delay did not increase the mean completion times. This indicates that the task completion efficiency remains unchanged even if the delays change when the predictor is running. The efficiency also remains unaffected when increasing or decreasing the incoming video frame rate. To quantify efficiency, the following following formula is used:

$$\%\ \text{improvement} = \left(\frac{\text{Avg. time without predictors} - \text{Avg. time with predictors}}{\text{Avg. time without predictors}} \right) \times 100\%$$

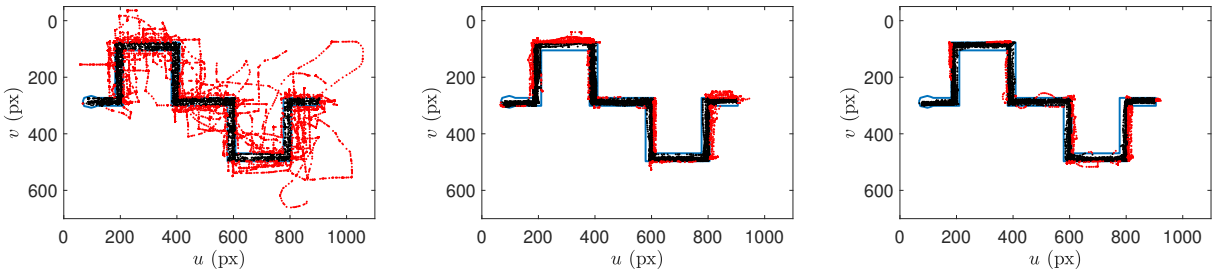
where Avg. time for the respective setups was found by averaging the mean completion times of the two FPS values (10 and 1) under the same delay. On average, for the tracking tasks, at 2 s of delay there is a 32.7% and 54.2% efficiency improvement with the split-screen and heterogeneous-stitch setups, respectively, compared to the cases with no prediction. The values are even higher – 54.2% and 67.6%, respectively – for the 6 s delay. Similarly, for the target acquisition tasks, with a 2 s delay there is a 48.7% and 60.9% efficiency improvement with the split-screen and heterogeneous-stitch setups, respectively, compared to the cases with no prediction. With a 6 s delay, the efficiency improvements are 61.6% and 68.1%, respectively.

Even though the subjective rating graphs did not show any clear effect due to lowering the image acquisition frame rate, the time domain graphs of Figs. 13a and 14a show that the mean completion time increases by a couple of seconds as the frame rate is reduced. But the time domain data for the split-screen and heterogeneous-stitch cases show that the decrease in video FPS has no noticeable effect on the task completion times. In fact, Figs. 13d and 13f actually show that the mean completion time decreases which is an improvement. When the predictor was running the output frame rate was a constant 10 FPS independent of the image acquisition frame rate. As a result, the participants were able to control the system, make corrections to the inputs, and make decisions faster which is the reason for seeing an improvement in the objective ratings when the predictor was running. The objective data proves that the predictor performs well in the "spatial domain" by predicting the low FPS incoming video imagery at a higher output display frame rate. This indicates that the incoming video frame rate can intentionally be lowered to save valuable network bandwidth or run computationally expensive computer vision algorithms and real-time telerobotic operation can still be continued with the predictor displaying at a higher frame rate.

F. Accuracy in the Tracking Task

The path taken by the participants from start to finish is plotted in Fig. 15 which gives a clear idea of the task completion accuracy without prediction, with a split-screen predictive display, and with a heterogeneous-stitch predictive display. In this figure, blue lines indicate the *desired* bounds. Red dots are the discrete data points lying outside of the desired bounds and black dots are the points lying inside the desired bounds during the completion of the tracking task. It is evident from Fig. 15 that the cases where there is no prediction yield the least accurate paths and that the heterogeneous-stitch display yields the most accurate paths, which agrees with the results described in previous sections. The percentage of points lying outside of the desired region is calculated using:

$$\% \text{ accuracy} = \frac{\text{Points inside}}{\text{Points inside} + \text{Points outside}} \times 100\%$$



(a) No prediction **(b) Split-screen** **(c) Heterogeneous-stitch**

Fig. 15 Path followed by participants in completing the tracking task under different test setups.

Using the formula above it is found that the setup with no prediction is 52.6% accurate, the split-screen setup is 81.3% accurate and the heterogeneous-stitch is 88.3% accurate. What this actually means is that when performing the tracking task, all the participants in all four test cases were able, on average, to stay within the desired bounds 52.6% of the total time without prediction, 81.3% of the time with the split-screen display, and 88.3% with the heterogeneous-stitch display.

G. Summary of Key Findings

The key findings from the preceding sections are summarized below:

- In Section VI.B, the maximum tolerable delay from the human transfer function was found to be 2.3 s, which was verified by the subjective Cooper-Harper ratings in Section VI.D.
- In Section VI.D, it was observed that participants preferred the predictive heterogeneous-stitch display over the predictive split-screen display and the display with no prediction. Some possible reasons were given for this preference.

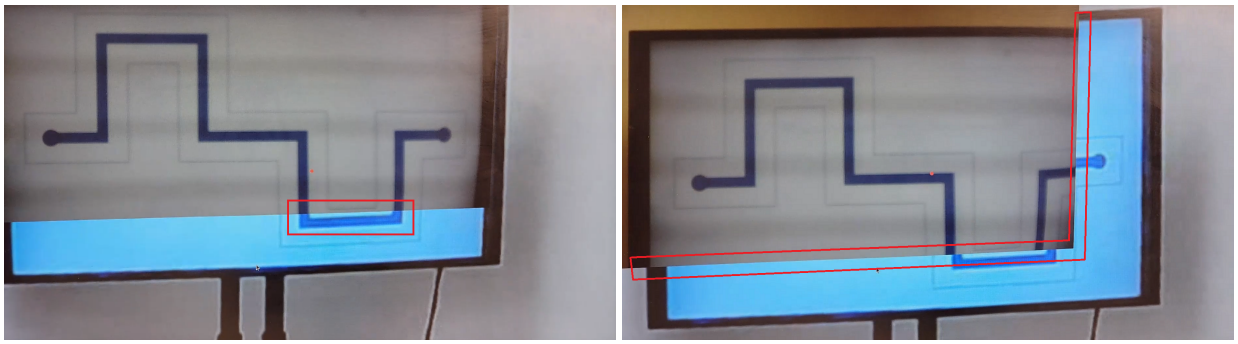
- In Section VI.E, it was found that stitched displays yielded the most efficient performance, followed by the split-screen display and the case without prediction. Observing the total task completion times for different participants it was seen that the standard deviations in the total task completion time reduced as the predictor was active. It is concluded from this finding that participants having different skill sets and expertise can still finish a task almost at the same time when delays are mitigated with the predictor.
- In Section VI.E it was found that the participants performed equally well or better when the predictor was running on low FPS incoming visual imagery.
- In Section VI.F, the heterogeneous-stitch and the split-screen setups have 35.7% and 28.7% accuracy improvements over the no prediction setup, respectively.

VII. Issues, Potential Solutions, and Future Work

This section describes some challenges that require further attention.

A. Incorrect Stitching

The image undistortion procedure described in Section III.B is only a polynomial approximation involving four to five coefficients depending on the type of camera. As the number of these coefficients increases, the approximation gets better but never becomes exact. The undistortion effects are more inaccurate the farther a point is from the image center. The only effective solution is to reduce the camera FoV by cropping out the inaccurate regions. For example, in Fig. 4, if the undistorted omnidirectional image is cropped at the red line, a better image will be obtained with a smaller FoV. Figures 16a and 16b show an example of good and bad stitching, respectively, during a tracking task. Good stitching takes place in the regions away from the corner while bad or incorrect stitching takes place in the regions near the corner. As the location of the PTZ image frame is near the omnidirectional image center, incorrect stitching happens entirely due to imperfect undistortions near the edges of the PTZ camera. However, if the delays are large or if the gimbal rotates fast then the location of the PTZ image could be near the edges of the omnidirectional image. In such cases, the stitch would have inaccuracies due to imperfect undistortions in both images. Since there is no image stitching done in the split-screen setup, the undistortion effects are less prominent.



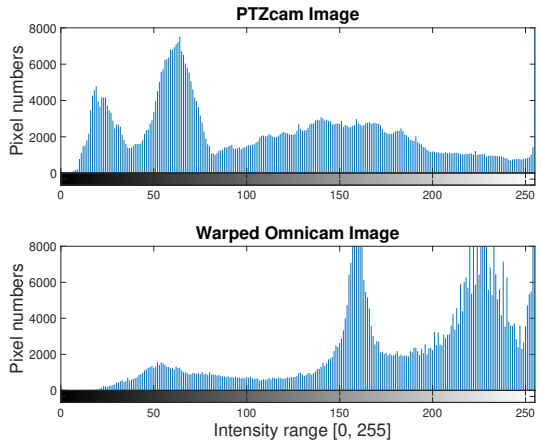
(a) Good stitch **(b) Bad stitch**

Fig. 16 Examples of good and bad stitching.

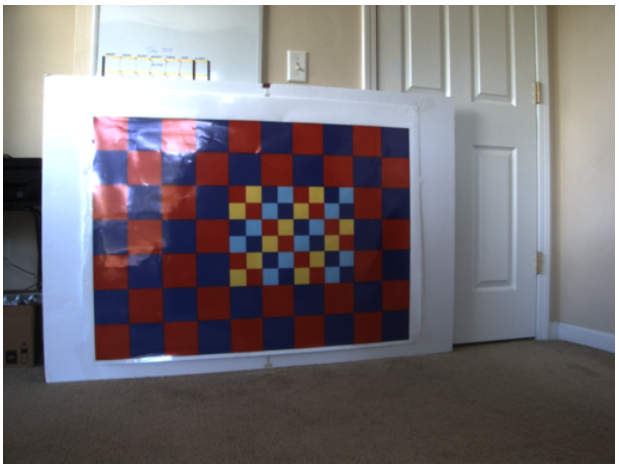
B. Different Image Intensities

As the wide FoV omnidirectional camera image gets cropped to match the small FoV PTZ camera, the omnidirectional image loses resolution and pixel intensity. These differences are depicted in Figs. 7b, 16a and 16b where two distinct regions are seen. The high-resolution, darker-intensity region comes from the PTZ camera, and the low-resolution, lighter-intensity region comes from the omnidirectional camera. There are computer vision techniques like *alpha blending* [67] that can blend the images near their borders to provide a smooth transition from one region to the next one. However, this method is computationally expensive and not implementable in real-time with the best existing hardware.

For real-time applications, it might be suitable to use a less expensive solution like histogram matching [68]. In *histogram matching*, histograms of the 8-bit pixel intensities in the range $[0, 255]$ are plotted for each color channel for each camera. Then they are normalized by the total number of pixels in the image to obtain the cumulative intensity



(a) Image histograms



(b) PTZ camera image

(c) Warped omnidirectional camera image

(d) Histogram matched omnidirectional image

Fig. 17 Example of histogram matching.

distribution function in the range $[0, 1]$ for each color channel for each camera. Then a mapping is used to map the pixel intensities of one image to obtain the corresponding pixel intensities of the other image. Fig. 17a gives an example of the image histograms for two different cameras for the red color channel. An intensity value of 0 means it is black and a value of 255 means it is white. As seen in Fig. 17a, a higher number of pixels of the PTZ image is near 0 which indicates that the image is darker in color; see Fig. 17b. On the other hand, more pixels in the omnidirectional camera image are near 255 indicating the image to be whiter in color; see Fig. 17c. Finally, using the mapping function the omnidirectional image pixels are mapped to PTZ image pixels; see Fig. 17d. This process is faster than the alpha blending method. However, this process can introduce some undesirable artifacts in the image where pixel intensities are high or where light is reflected. For example, the big red checkerboard squares in Fig. 17d appear to have some dark streaks as artifacts.

C. Adaptation to Non-hovering Flight

The predictive display scheme developed in this paper can be applicable directly to a hovering flight condition. The ground-based human subjects testing demonstrates the scenario when the UAV is stationary and the gimbal is in rotation. However, for a non-hovering flight, the simplification made to Eqn. (8), described by Eqn. (15), is no longer valid since there now exists a non-zero translation vector between two camera frames. As a result, additional terms like the normal vector $\boldsymbol{n}$ to the world plane must be computed. This could be done with the introduction of an additional sensor like a LiDAR or with complex stereo-vision algorithms. Proper system identification for the UAV

is also needed for accurately predicting the UAV behavior. For demonstration purposes, in this paper, all the image acquisition, manipulation, and prediction tasks have been performed using a single ground station which is not possible for in-flight testing. An onboard computer is necessary for image acquisition and UAV and gimbal control.

Prior work by Sakib et al. [48] involved image acquisition using the Nvidia Jetson Nano onboard computer. It was capable of acquiring the PTZ images at 15 Hz and omnidirectional images at 7 Hz. Additional testing with a more powerful onboard computer, the Nvidia AGX Xavier, showed that it is capable of acquiring both PTZ and omnidirectional images at 23 Hz. A bottleneck in wireless data transmission can arise as networks have limited bandwidth and high-quality image transmission requires a lot of bandwidth. From Section IV.A it was calculated that the PTZ camera used in this study must send approximately 31.64 Mbits and the omnidirectional camera must send 69.04 Mbits of data per frame. A state-of-the-art fifth generation (5G) router with a maximum bandwidth of 1800 Mbits can theoretically support both video streams up to 17 Hz. In addition, the ground station hardware can cause further bottlenecks in running the image processing and image manipulation algorithms required by the split-screen and the stitching cases. However, recent advances in graphical processing units (GPUs) can reduce this bottleneck. The success of the algorithm in a hovering flight and the complexities involved in non-hovering flights warrant future investigations with in-flight tests and actual network delays.

VIII. Conclusions

A heterogeneous stereo-vision system can support aerial telerobotic bridge inspection, but delays in the system can degrade usability, discouraging inspectors from adopting the technology. Two different approaches were used to show that delays degrade human/machine system performance: classical stability analysis and human subjects testing. For the classical stability analysis, a human transfer function was determined and the delay margin for that transfer function was obtained. Usability studies were performed using the Cooper-Harper Handling Qualities Rating scale that captured the system's subjective performance according to the human operators. Results from both approaches were in agreement showing that, for the system considered here, the maximum unmitigated delay that an operator can tolerate is about 2.3 s.

Two types of predictive displays were developed to address time delay in a telerobotic interface using a heterogeneous stereo-vision system: a split-screen predictive display and a heterogeneous-stitch predictive display. In the split-screen setup, the predicted omnidirectional image was shown as an inset in the PTZ image whereas, in the heterogeneous-stitch setup, the predicted omnidirectional image was cropped and stitched to the predicted PTZ image to avoid losing FoV. Usability studies showed that the prediction algorithm improves the users' experience of the system by decreasing the workload in performing a task, reducing the time to complete a task, and increasing the accuracy with which the task is performed. Of the two predictive display setups analyzed, the test participants preferred the heterogeneous-stitch predictive display. Ongoing work involves expanding and demonstrating the algorithm in non-hovering flights.

Acknowledgements

The authors gratefully acknowledge the participation of the operators who participated in the human subjects experiments, as well as the helpful assistance of Zakia Ahmed and Binay Rijal. The work in this paper was supported by the National Science Foundation (NSF) under Grant No. `IIS-1840044`.

References

[1] Morgese, M., Ansari, F., Domaneschi, M., and Cimellaro, G., "Post-collapse analysis of Morandi's Polcevera viaduct in Genoa Italy," *Journal of Civil and Structural Health Monitoring*, Vol. 10, 2020, pp. 69–85. https://doi.org/10.1007/s13349-019-00370-7.

[2] Brantley, M., "I-40 Bridge Fix May Take 'Considerable Amount of Time,' Senator Says. Crack 'significant,' Engineers Say," *Arkansas Times*, 2021. URL https://arktimes.com/arkansas-blog/2021/05/12/i-40-bridge-fix\-may-take-considerable-amount-of-time-senator-says.

[3] Bock, T., "The future of construction automation: Technological disruption and the upcoming ubiquity of robotics," *Automation in construction*, Vol. 59, 2015, pp. 113–121. https://doi.org/https://doi.org/10.1016/j.autcon.2015.07.022.

[4] Afsari, K., Halder, S., Ensafi, M., DeVito, S., and Serdakowski, J., "Fundamentals and Prospects of Four-Legged Robot Application in Construction Progress Monitoring," *57th Annual Associated Schools of Construction International Conference (ASC 2021)*, EPiC Series in Built Environment, Vol. 2, EasyChair, Stockport, UK, 2021, pp. 274–283. https://doi.org/10.29007/cdpd.

[5] Halder, S., Afsari, K., Serdakowski, J., DeVito, S., Ensafi, M., and Thabet, W., "Real-Time and Remote Construction Progress Monitoring with a Quadruped Robot Using Augmented Reality," *Buildings*, Vol. 12, No. 11, 2022. https://doi.org/10.3390/buildings12112027.

[6] Greenwood, W. W., Zhou, H., Zekkos, D., and Lynch, J. P., "Experiments Using a UAV-Deployed Impulsive Source for Multichannel Analysis of Surface Waves Testing," *Geotechnical Earthquake Engineering and Soil Dynamics V*, American Society of Civil Engineers (ASCE), Reston, VA, 2018, pp. 443–451. https://doi.org/10.1061/9780784481486.046.

[7] Zekkos, D., Lynch, J. D., Sahadewa, A., Hirose, M., and Ellis, D., "Proof-of-Concept Shear Wave Velocity Measurements Using an Unmanned Autonomous Aerial Vehicle," *Geo-Congress 2014 Technical Papers*, American Society of Civil Engineers (ASCE), Reston, VA, 2014, pp. 953–962. https://doi.org/10.1061/9780784413272.093.

[8] Zekkos, D., Manousakis, J., Greenwood, W., and Lynch, J., "Immediate UAV-enabled Infrastructure Reconnaissance following Recent Natural Disasters: Case Histories from Greece," *1st International Conference on Natural Hazards & Infrastructure*, Innovation Center for Natural Hazards & Infrastructure (ICONHIC), Athens, Greece, 2016.

[9] Greenwood, W., Lynch, J., and Zekkos, D., "Applications of UAVs in Civil Infrastructure," *Journal of Infrastructure Systems*, Vol. 25, No. 2, 2019. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000464.

[10] Chen, S., Rice, C., Boyle, C., and hauser, e., "Small Format Aerial Photography for Highway Bridge Monitoring," *Journal of Performance of Constructed Facilities*, Vol. 25, No. 2, 2011, pp. 105–112. https://doi.org/10.1061/(ASCE)CF.1943-5509.0000145.

[11] Reagan, D., Sabato, A., and Niezrecki, C., "Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges," *Structural Health Monitoring*, Vol. 17, No. 5, 2018, pp. 1056–1072. https://doi.org/10.1177/1475921717735326.

[12] Dong, C.-Z., and Catbas, F. N., "A review of computer vision–based structural health monitoring at local and global levels," *Structural Health Monitoring*, Vol. 20, No. 2, 2021, pp. 692–743. https://doi.org/10.1177/1475921720935585.

[13] Sony, S., Dunphy, K., Sadhu, A., and Capretz, M. A. M., "A systematic review of convolutional neural network-based structural condition assessment techniques," *Engineering Structures*, Vol. 226, 2021, p. 111347. https://doi.org/10.1016/j.engstruct.2020.111347.

[14] Flah, M., Vargas, I., Ben Chaabene, W., and Nehdi, M., "Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review," *Archives of Computational Methods in Engineering*, Vol. 28, 2020, pp. 2621–2643. https://doi.org/10.1007/s11831-020-09471-9.

[15] Spencer, B. F., Hoskere, V., and Narazaki, Y., "Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring," *Engineering*, Vol. 5, No. 2, 2019, pp. 199–222. https://doi.org/10.1016/j.eng.2018.11.030.

[16] de Vries, S. C., "UAVs and Control Delays," Tech. Rep. TNO-DV3 2005 A054, TNO Defence, Security and Safety, Soesterberg, Netherlands, 2005. URL https://www.researchgate.net/publication/235120515_UAVs_and_Control_Delays.

[17] Chen, J. Y. C., Haas, E. C., and Barnes, M. J., "Human Performance Issues and User Interface Design for Teleoperated Robots," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, Vol. 37, No. 6, 2007, pp. 1231–1245. https://doi.org/10.1109/tsmcc.2007.905819.

[18] Sheridan, T. B., and Ferrell, W. R., "Remote Manipulative Control with Transmission Delay," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-4, No. 1, 1963, pp. 25–29. https://doi.org/10.1109/THFE.1963.231283.

[19] Sheridan, T. B., "Space Teleoperation Through Time Delay: Review and Prognosis," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, 1993, pp. 592–606. https://doi.org/10.1109/70.258052.

[20] Kang, C., Chaudhry, H., Woolsey, C., and Kochersberger, K., "Development of a Peripheral-Central Vision System for Small UAS Tracking," *AIAA SciTech 2019 Forum*, 2019. https://doi.org/10.2514/6.2019-2074.

[21] Bianchi, E. L., Sakib, N., Woolsey, C., and Hebdon, M., "Bridge Inspection Component Registration for Damage Evolution," *Structural Health Monitoring*, Vol. 0, No. 0, 2022, p. 14759217221083647. https://doi.org/10.1177/14759217221083647.

[22] Shanthakumar, P., Yu, K., Singh, M., Orevillo, J., Bianchi, E., Hebdon, M., and Tokekar, P., "View Planning and Navigation Algorithms for Autonomous Bridge Inspection with UAVs," *Proceedings of the 2018 International Symposium on Experimental Robotics*, Vol. 11, 2020, pp. 201–210. https://doi.org/10.1007/978-3-030-33950-0_18.

[23] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110. https://doi.org/https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[24] Yi, K. M., Trulls, E., Lepetit, V., and Fua, P., "LIFT: Learned Invariant Feature Transform," *2016 European Conference on Computer Vision (ECCV 2016)*, Springer, New York, NY, USA, 2016, pp. 467–483. https://doi.org/10.1007/978-3-319-46466-4_28.

[25] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 2011, pp. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544.

[26] Kang, C., Davis, J., Woolsey, C. A., and Choi, S., "Sense and avoid based on visual pose estimation for small UAS," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 2017, pp. 3473–3478. https://doi.org/10.1109/IROS.2017.8206188.

[27] Rathnayaka, P., Baek, S.-H., and Park, S.-Y., "An Efficient Calibration Method for a Stereo Camera System with Heterogeneous Lenses Using an Embedded Checkerboard Pattern," *Journal of Sensors*, Vol. 2017, 2017, pp. 1–12. https://doi.org/10.1155/2017/6742615.

[28] Brooke, J., "SUS: A Retrospective," *Journal of Usability Studies*, Vol. 8, No. 2, 2013, p. 29–40.

[29] Hart, S. G., and Staveland, L. E., "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *Advances in Psychology*, Vol. 52, 1988, pp. 139–183. https://doi.org/https://doi.org/10.1016/S0166-4115(08)62386-9.

[30] Cooper, G. E., and Harper, R., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities," Tech. Rep. NASA-TN-D-5153, NASA Ames Research Center, Moffett Field, CA, United States, 1969.

[31] Costello, D. H., and Xu, H., "Relating Sensor Degradation to Vehicle Situational Awareness for Autonomous Air Vehicle Certification," *Journal of Aerospace Information Systems*, Vol. 18, No. 4, 2021, pp. 193–202. https://doi.org/10.2514/1.I010905.

[32] Jennings, S., Craig, G., Reid, L., and Kruk, R., "The Effect of Visual System Time Delay on Helicopter Control," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 44, SAGE Publishing, Newbury Park, CA, USA, 2000, pp. 69–72. https://doi.org/10.1177/154193120004401318.

[33] Mitchell, D. G., and Klyde, D. H., "This is pilot gain," *AIAA Scitech 2019 Forum*, 2019, pp. 1–17. https://doi.org/10.2514/6.2019-0562.

[34] Gray, W. D., *Integrated Models of Cognitive Systems*, Oxford University Press, 2007, pp. 29–44. https://doi.org/10.1093/acprof:oso/9780195189193.001.0001.

[35] Tustin, A., "The Nature of the Operator's Response in Manual Control and Its Implications for Controller Design," *Journal of the Institute of Electrical Engineers*, Vol. 94, 1947, pp. 190–202. https://doi.org/https://doi.org/10.1049/ji-2a.1947.0025.

[36] Russell, L., "Characteristics of the Human as a Linear Servo-Element," Master's thesis, Massachusetts Institute of Technology, 1951.

[37] McRuer, D. T., and Krendel, E. S., "The human operator as a servo system element," *Journal of the Franklin Institute*, Vol. 267, No. 5, 1959, pp. 381–403. https://doi.org/https://doi.org/10.1016/0016-0032(59)90091-2.

[38] McRuer, D. T., and Krendel, E. S., "Mathematical Models of Human Pilot Behavior," AGARDograph AGARD-AG-188, Advisory Group for Aerospace Research and Development, Jan. 1974.

[39] Baron, S., and Kleinman, D. L., "The Human as an Optimal Controller and Information Processor," *IEEE Transactions on Man-Machine Systems*, Vol. 10, No. 1, 1969, pp. 9–17. https://doi.org/10.1109/TMMS.1969.299875.

[40] Sirouspour, S., and Shahdi, A., "Model Predictive Control for Transparent Teleoperation Under Communication Time Delay," *IEEE Transactions on Robotics*, Vol. 22, No. 6, 2006, pp. 1131–1145. https://doi.org/10.1109/TRO.2006.882939.

[41] Desoer, C. A., and Vidyasagar, M., *Feedback systems: input-output properties*, SIAM, 2009, pp. 168–226.

[42] Nuño, E., Basañez, L., and Ortega, R., "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, Vol. 47, No. 3, 2011, pp. 485–495. https://doi.org/https://doi.org/10.1016/j.automatica.2011.01.004r.

[43] Lu, Z., Huang, P., and Liu, Z., "Predictive Approach for Sensorless Bimanual Teleoperation Under Random Time Delays With Adaptive Fuzzy Control," *IEEE Transactions on Industrial Electronics*, Vol. 65, No. 3, 2018, pp. 2439–2448. https://doi.org/10.1109/TIE.2017.2745445.

[44] Mirfakhrai, T., and Payandeh, S., "A delay prediction approach for teleoperation over the Internet," *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Vol. 2, New York, NY, USA, 2002, pp. 2178–2183 vol.2. https://doi.org/10.1109/ROBOT.2002.1014862.

[45] Brudnak, M. J., "Predictive Displays for High Latency Teleoperation," *NDIA Ground Vehicle Systems Engineering Technology Symposium*, National Defense Industrial Association (NDIA), Michigan Chapter, Sterling Heights, MI, USA, 2016. URL https://www.researchgate.net/publication/305904651_PREDICTIVE_DISPLAYS_FOR_HIGH_LATENCY_TELEOPERATION.

[46] Jung, Y., Han, K., and Bae, J., "A Tele-Operated Display With a Predictive Display Algorithm," *IEEE Access*, Vol. 7, 2019, pp. 154447–154456. https://doi.org/10.1109/ACCESS.2019.2948879.

[47] Cox, J., and Wong, K., "Predictive feedback augmentation for manual control of an unmanned aerial vehicle with latency," *International Journal of Micro Air Vehicles*, Vol. 11, 2019, p. 1756829319869645. https://doi.org/10.1177/1756829319869645.

[48] Sakib, N., Gresham, J., and Woolsey, C. A., "Usability Studies of a Predictive Heterogeneous Vision System in Mitigating the Effects of Visual Display Delay," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-0017.

[49] Dam, J. V., Krasner, A., and Gabbard, J. L., "Augmented Reality for Infrastructure Inspection with Semi-autonomous Aerial Systems: An Examination of User Performance, Workload, and System Trust," *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 2020, pp. 742–743. https://doi.org/10.1109/VRW50115.2020.00222.

[50] Van Dam, J., Krasner, A., and Gabbard, J. L., "Drone-based Augmented Reality Platform for Bridge Inspection: Effect of AR Cue Design on Visual Search Tasks," *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 2020, pp. 201–204. https://doi.org/10.1109/VRW50115.2020.00043.

[51] Bianchi, E., and Hebdon, M., "Corrosion Condition State Semantic Segmentation Dataset," Virginia Tech, Blacksburg, VA, USA, 2021. https://doi.org/10.7294/16624663.v2.

[52] Bianchi, E., and Hebdon, M., "Trained Model for the Semantic Segmentation of Structural Material," Virginia Tech, Blacksburg, VA, USA, 2021. https://doi.org/10.7294/16628620.v1.

[53] Bianchi, E., Abbott, A., Tokekar, P., and Hebdon, M., "COCO-Bridge: Structural Detail Data Set for Bridge Inspections," *Journal of Computing in Civil Engineering*, Vol. 35, 2021, p. 04021003. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000949.

[54] Malis, E., and Vargas, M., "Deeper understanding of the homography decomposition for vision-based control," Tech. Rep. RR-6303, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France, 2007.

[55] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, 2000, pp. 1330–1334. https://doi.org/10.1109/34.888718.

[56] Heikkila, J., and Silven, O., "A four-step camera calibration procedure with implicit image correction," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 1997, pp. 1106–1112. https://doi.org/10.1109/CVPR.1997.609468.

[57] Kannala, J., and Brandt, S., "A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, 2006, pp. 1335–40. https://doi.org/10.1109/TPAMI.2006.153.

[58] Fischler, M. A., and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM,*, Vol. 24, No. 6, 1981, p. 381–395. https://doi.org/10.1145/358669.358692.

[59] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer, 2011, pp. 487–498.

[60] Hartley, R., and Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2003, pp. 88–91.

[61] Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A., "SuperGlue: Learning Feature Matching with Graph Neural Networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, 2020. https://doi.org/10.1109/CVPR42600.2020.00499.

[62] "Infinity MR S2 Gimbal Manual," Web: HD Air Studio, 2023. URL https://hdairstudio.com/wp-content/uploads/2020/03/InfinityMR-S2-User-Manual-V.01.pdf.

[63] "SimpleBGC 32bit 3-Axis Software User Manual," Web: BaseCam Electronics, 2023. URL https://www.basecamelectronics.com/files/v3/SimpleBGC_32bit_manual_eng.pdf.

[64] Sakib, N., "PredictiveDisplay3.0," GitHub, 2022. URL https://github.com/Nazmus20/PredictiveDisplay3.0/.

[65] Sakib, N., "Videos for Time Delay Mitigation," Google Drive, 2022. URL https://drive.google.com/drive/folders/1f9gV5OK0pZHkXfprwHbA9QsX2AIyfRQ3?usp=sharing.

[66] Ju, P., and Zhang, H., "Achievable delay margin using LTI control for plants with unstable complex poles," *Science China Information Sciences*, Vol. 61, No. 9, 2018. https://doi.org/10.1007/s11432-017-9185-6.

[67] Salvi, M., and Vaidyanathan, K., "Multi-Layer Alpha Blending," *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Association for Computing Machinery (ACM), New York City, NY, 2014, p. 151–158. https://doi.org/10.1145/2556700.2556705.

[68] Shapira, D., Avidan, S., and Hel-Or, Y., "Multiple histogram matching," *2013 IEEE International Conference on Image Processing*, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, USA, 2013, pp. 2269–2273. https://doi.org/10.1109/ICIP.2013.6738468.