

Teleoperation Enhancement for Autonomous Vehicles Using Estimation Based Predictive Display*

Gaurav Sharma and Rajesh Rajamani

Abstract— Teleoperation is increasingly used in the operation of delivery robots and is beginning to be utilized for certain autonomous vehicle intervention applications. This paper addresses the challenges in teleoperation of an autonomous vehicle due to latencies in wireless communication between the remote vehicle and the teleoperator station. Camera video images and Lidar data are typically delayed during wireless transmission but are critical for proper display of the remote vehicle's real-time road environment to the teleoperator. Data collected with experiments in this project show that a 0.5 second delay in real-time display makes it extremely difficult for the teleoperator to control the remote vehicle. This problem is addressed in the paper by using a predictive display (PD) system which provides intermediate updates of the remote vehicle's environment while waiting for actual camera images. The predictive display utilizes estimated positions of the ego vehicle and of other vehicles on the road computed using model-based extended Kalman filters. A crucial result presented in the paper is that vehicle motion models need to be inertial rather than relative and so tracking of other vehicles requires accurate localization of the ego vehicle itself. An experimental study using 5 human teleoperators is conducted to compare teleoperation performance with and without predictive display. A 0.5 second time-delay in camera images makes it impossible to control the vehicle to stay in its lane on curved roads, but the use of the developed predictive display system enables safe remote vehicle control with almost as accurate a performance as the delay-free case.

Key Words: Autonomous vehicles, teleoperation, predictive display, trajectory estimation

I. INTRODUCTION

While there is significant on-going research in the US related to autonomous vehicles (AVs) that will operate entirely without the need for driver intervention, it is not expected that the first fully self-driving vehicles will be introduced into the market until at least 2028, even according to the most optimistic estimates [1]. Current automated driving systems can operate autonomously a majority of the time in many traffic and environment scenarios. However, human intervention is occasionally needed, and backup safety drivers are now almost always present as a part of test vehicles. Some examples of situations where human intervention may be needed are the presence of snow cover on the road making the lane markers invisible, active snow/rain precipitation, the presence of construction zones on road, and the failure of critical sensors, actuators or other components on the vehicle.

Enabling autonomous vehicles (AVs) to operate without requiring a backup in-vehicle safety driver is essential for scaling the deployment of the technology. Teleoperation has long been a solution in space robotics where a human teleoperator takes over when autonomous operation is prevented. In the case of AVs, a remote teleoperator can likewise step in to get the car past whatever hazard might be too hard for the vehicle to handle by itself in an autonomous fashion.

A German startup company Vay [2] is currently using a human teleoperator to remotely drive vehicles via computer to a location where a human customer wants to be picked up. Then the customer drives to their destination, after which the remote operator again drives the car away. Likewise, the company Halo is experimenting with a system in Las Vegas where a rental car is driven remotely by a human and dropped off for the customer to drive. When the customer is done, the remote operators take it to its next destination [3]. Applications of such autonomous vehicles with occasional teleoperation when needed can include valet parking, taxi fleets, and ride-sharing services similar to Uber or Lyft.

Teleoperation performance requires strong wireless connectivity that depends on network conditions such as latency, bandwidth, packet loss and reliability. Due to huge data streams originating from Lidar and camera sensors on the vehicle and also the need for reliably communicating critical real-time inputs from the tele-driver to the vehicle every few milli-seconds, addressing bandwidth and latency limitations is critical for this application.

Human drivers can solve delay-free tasks with cognitive challenges faster than automation systems, but this relationship reverses with the presence of time delays. Delayed perception during teleoperated driving significantly increases the effort of the human, and the operator has to take care of the task and the monitoring of the environment with more intensive effort [4,5]. This paper proposes to develop an estimation based predictive display (PD) system that estimates trajectories of the ego vehicle and other vehicles on the road to perform realistic intermediate updates of the remote environment to compensate for delayed camera data.

Previous researchers have studied the display of predicted ego-vehicle position on the teleoperator screen using a semitransparent vehicle [6], a rectangular frame and tracks [7] or a pointing line [8]. However, they just assumed

* Research supported in part by the University of Minnesota InterS&Ections Program and by NSF Grant CNS 2321531.

Rajesh Rajamani and Gaurav Sharma are with the Department of Mechanical Engineering, University of Minnesota, Twin Cities, Minneapolis,

MN 55455 USA (Email: rajamani@umn.edu; TEL: 612-626-7961; sharm936@umn.edu)

the non-delayed future position of the ego-vehicle was known and did not do position estimation.

Others have used open-loop predictions of future motion to predict ego vehicle position, such as clothoids trajectories [9,10] and predictors [11]. Finally, researchers have also used zooming and sliding (or image transformation) based on real-time throttle/brake/steering inputs from the teleoperator [12,13]. However, none of the authors described above have used model-based estimation algorithms for prediction of ego-vehicle position. Further, none of them have included predictions of positions of other nearby vehicles on the road. Predictions of trajectories of nearby vehicles is critical to ensure safe driving by the teleoperator.

The primary contributions of this paper are as follows:

- 1) The paper develops a teleoperation platform using MATLAB's Automated Driving Toolbox to provide a co-simulation environment for experimental human-in-the-loop teleoperation studies.
- 2) This paper presents for the first time an estimation-based PD system which uses both ego state estimation and estimation of other vehicle trajectories for teleoperation enhancement. Hence, the trajectories of both ego and non-ego actors are used to augment PD.
- 3) This paper describes the inaccuracies of using relative vehicle motion models and the importance of using inertial models for vehicle tracking.
- 4) This paper presents an experimental human-subjects study to evaluate the effectiveness of using estimation-based PD for teleoperation enhancement and compensation of the degradation caused by delay.

The outline of the rest of the paper is as follows. In section II, the MATLAB based teleoperation simulator is described. Section III describes the estimation-based PD system where both state estimation and vehicle tracking are described along with the inaccuracies of relative motion models and the need for an inertial motion model. Section IV and V describes and discusses the results for image comparison analysis, lane offset analysis for various time delay and from a human-subjects study and proves the efficacy of using PD. Section VI contains the conclusions.

II. TELEOPERATION SIMULATOR DESIGN

Designing a teleoperation platform requires a virtual driving environment, control input devices including throttle, brakes and steering wheel, accurate vehicle dynamics, delay control units and realistic visual feedback of the remote vehicle environment to the human user. This section details the various aspects of the teleoperator station design which has been used as the testing environment for human-in-the-loop teleoperation simulations. The MATLAB Automated Driving Toolbox has been used for this purpose. The toolbox provides a co-simulation environment which uses Simulink to model the driving algorithms and an Unreal Engine to create the virtual driving environment which is a 3D creation tool for photorealistic visualization. The driving scenario for the simulations has been implemented using MATLAB's Driving Scenario Designer. The teleoperation station used in the current work is shown in Fig. 1 along with the computer and



Fig. 1. Teleoperation Station

monitor. The computer has 24 cores with an Intel i9 processor, 64 GB RAM and a NVIDIA RTX 4090 24 GB graphics card. The monitor is a Samsung 49" Odyssey G29.

A. Control Input Device and Vehicle Dynamics

The teleoperator provides the control inputs i.e., steering angle, throttle and brake commands using the Logitech G29 racing wheel along with external throttle and brake pedals. The racing wheel provides dual-motor force feedback which accurately simulates the force effects and like real steering wheels, it has a 900-degree lock-to-lock rotation. The brake pedal, which is nonlinear, mimics the characteristics of a pressure sensitive brake system. The Joystick Input block has been used as an interface between the control commands provided by the user and the Simulink environment.

To simulate real-world driving, it is very important to use an accurate vehicle dynamics algorithm. A Simulink/MATLAB based 34-DOF vehicle dynamic simulator including nonlinear tire force models has been used to simulate the actual vehicle dynamics for the ego vehicle while utilizing the control inputs provided by the user.

B. Delay Control Unit

Wireless latency affects teleoperation due to delay in transmission of the remote vehicle environment over wireless communication network. In particular, camera images and Lidar data which have large sizes are more likely to be delayed during wireless transmission. To simulate this latency, the delay block in Simulink has been used to delay the display to the teleoperator. This block delays the display signal based on the specified delay length. Moreover, it is also possible to input a variable delay length based on the local wireless network characteristics. For the present simulations, a constant delay has been used.

C. Simulink-Unreal Engine Co-simulation Environment

The co-simulation environment in the MATLAB Automated Driving Toolbox uses Simulink to simulate the positions and orientation of various actors and 3D photorealism to visualize the various actors in the 3D simulation environment which acts as the visual display for

sensor measurements are needed to compensate for the drift due to IMU. Examples of position sensors include LIDAR, camera, and GPS. Although LIDAR and camera based odometry methods can be highly accurate but their use is infeasible in teleoperation due to increased data size which consumes large bandwidth thus adding delay in state estimation. On the other hand, due to low data size, GPS is a feasible option for teleoperation which will avoid latency. However, regular GPS suffers from low accuracy (± 1.5 m) but when used with RTK corrections the accuracy can increase to the order of 1-10 cm. The Mn-CORS network operated by MnDOT and the Minnesota State Government provides RTK-corrected GPS with accuracy of 10 cm throughout the state of Minnesota. Hence, in this work GPS measurements corrected using the Mn-CORS type networks have been used.

The noisy and biased measured IMU readings are related to the true signals as follows,

$$a_x = a_{x,t} + a_{x,bt} + a_{x,n} \quad (4)$$

$$a_y = a_{y,t} + a_{y,bt} + a_{y,n} \quad (5)$$

$$\omega_E = \omega_{E,t} + \omega_{E,bt} + \omega_{E,n} \quad (6)$$

where, $a_{x,t}$, $a_{y,t}$ and $\omega_{E,t}$ are the true readings, $a_{x,bt}$, $a_{y,bt}$ and $\omega_{E,bt}$ are the constant accelerometer and gyro bias respectively and $a_{x,n}$, $a_{y,n}$ and $\omega_{E,n}$ are as follows,

$$\begin{bmatrix} a_{x,n} \\ a_{y,n} \\ \omega_n \end{bmatrix} \sim \begin{bmatrix} N(0, \sigma_{a_{x,n}}) \\ N(0, \sigma_{a_y}) \\ N(0, \sigma_{\omega_n}) \end{bmatrix} \quad (7)$$

where, $N(0, \sigma_{a_{x,n}})$ indicates white noise with zero mean and standard deviation of $\sigma_{a_{x,n}}$.

The measurement equation for the GPS is as follows,

$$y = [x_E \ y_E]^T + v \quad (8)$$

where, $v \sim N(0, R_{gps})$ is the measurement noise with error covariance matrix R_{gps} . Although the state is observable using the measurement in (8), however error analysis for Vehicle Tracking (described in the next subsection) reveals that yaw angle of the ego vehicle must be estimated with high accuracy. Hence, it is very important to use measurements of yaw angle. In literature [15], various methods have been described to compute heading which includes, using magnetic field, angular velocity of earth, vision, dual antenna GNSS, velocity and acceleration heading. The use of magnetic field to compute heading accurately is infeasible in AVs due to the presence of many local magnetic materials. Using angular velocity of earth requires costly sensors and using vision will add latency to state estimation. A dual antenna GPS provides accurate measurements for heading; however, this method requires a large baseline. Computing heading using velocities from GPS is a viable option as it is low cost and suitable for teleoperation. Acceleration based methods for computing heading require the differentiation of velocities obtained from GPS and are prone to errors at low acceleration. Hence, velocities computed using GPS have been used to measure the

yaw angle.

The velocity of the ego vehicle in the inertial frame ($\dot{x}_E^I$ and $\dot{y}_E^I$) are related to those in the ego frame as follows,

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = R_E \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} c_E v_x - s_E v_y \\ c_E v_y + s_E v_x \end{bmatrix} \quad (9)$$

where, v_x and v_y is the velocity of ego vehicle along the ego frame axis, c_E and s_E are $\cos(\psi_E)$ and $\sin(\psi_E)$ respectively, and R_E is the rotation matrix of ego frame w.r.t. inertial frame given by,

$$R_E = \begin{bmatrix} c_E & -s_E \\ s_E & c_E \end{bmatrix} \quad (10)$$

If $v_x \gg v_y$, then (9) reduces to,

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} c_E v_x \\ s_E v_x \end{bmatrix} \quad (11)$$

Hence,

$$\psi_E = \tan^{-1} \left(\frac{\dot{y}_E}{\dot{x}_E} \right) \quad (12)$$

Both $\dot{x}_E$ and $\dot{y}_E$ can be measured from GPS with sufficient accuracy [16] and hence can be used to measure the yaw angle. However, the measurements are prone to error when the lateral velocity is high (i.e., high slip angle).

Given the measurement of position and yaw angle along with the IMU inputs, an Extended Kalman Filter (EKF) has been used for state estimation. The prediction equations for the EKF are as follows,

$$\bar{x}_{k+1}^- = f(\bar{x}_k^+, u_k) \quad (13)$$

$$P_{k+1}^- = F_k P_k^+ F_k^T + Q_k \quad (14)$$

where, $F_k = I_5 + \Delta t A_k$, $A_k = \frac{\partial f}{\partial x}(\bar{x}_k^+, u_k)$ and P_k and Q_k are the state covariance matrix and process noise covariance matrix respectively. The correction equations for the EKF are as follows,

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1} \quad (15)$$

$$\bar{x}_{k+1}^+ = \bar{x}_{k+1}^- + K_{k+1} (y_{k+1} - H_{k+1} \bar{x}_{k+1}^-) \quad (16)$$

$$P_{k+1}^+ = (I - K_{k+1} H_{k+1}) P_{k+1}^- \quad (17)$$

where, $H_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ and R_{k+1} is the measurement noise covariance matrix. Due to the low size of IMU and GPS data [17], it has been assumed that there is no latency in their transmission to the teleoperator and that state estimation is being done on the teleoperator side.

B. Vehicle Tracking

The relative orientation between the ego vehicle and tracked vehicle is shown in Fig. 4. The tracked vehicle frame (O_C, x_C, y_C) is located at the CoM of the tracked vehicle $O_C(x_C, y_C)$. The relative position of the tracked vehicle w.r.t. the ego vehicle in the Inertial frame (r_{EC}) is given as follows,

$$r_{EC} = O_C - O_E = \begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} x_C - x_E \\ y_C - y_E \end{bmatrix} \quad (18)$$

Let the state vector of the tracked vehicle be,

$$X_v = [x_C \ y_C \ V_C \ A_C \ \psi_C \ \beta_C]^T = [x_1 \ \dots \ x_6]^T \quad (19)$$

where V_C is the velocity, A_C is the acceleration, ψ_C is the yaw

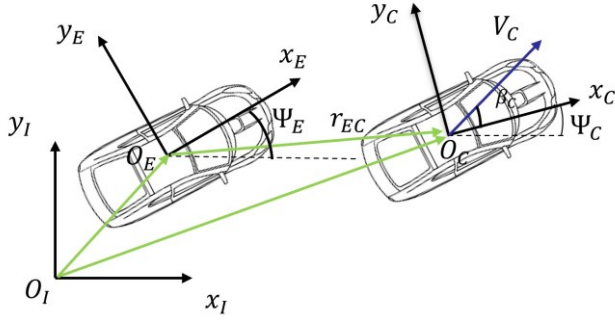


Fig. 4. Relative orientations of vehicles w.r.t. inertial frame

angle and β_C is the slide slip angle of the tracked vehicle.

The state dynamics of the tracked vehicle is as follows,

$$\dot{X}_v = \begin{bmatrix} x_3 \cos(x_5 + x_6) \\ x_3 \sin(x_5 + x_6) \\ x_4 \\ 0 \\ x_3 \sin(x_6) / l_r \\ 0 \end{bmatrix} = f_v(X_v) \quad (20)$$

where l_r is the distance of the rear wheels from the CoM of the tracked vehicle. The given vehicle tracking model is suitable to track a vehicle on both straight and curved road with constant acceleration and side slip angle [15]. The radar provides the position of the tracked vehicle w.r.t. the ego vehicle given by,

$$y_v = (R_E)^T r_{EC} + w \quad (21)$$

where, $w \sim N(0, \sigma_w)$. The radar measurements are w.r.t. the ego frame but the vehicle tracking model is w.r.t. the inertial frame. The radar measurements can be converted to the inertial frame using the ego state estimates and then can be used with the vehicle tracking model. The reasons for using this approach will be detailed in the next sub-section. The new measurement model is as follows,

$$y_{v_n} = \hat{R}_E y_v + \hat{O}_E = \hat{R}_E (R_E)^T r_{EC} + \hat{O}_E + \hat{R}_E w \quad (22)$$

where, $\hat{R}_E$ is the rotation matrix of the estimated ego frame w.r.t. the inertial frame formed using estimated yaw angle $\hat{\psi}_E$.

The product of $\hat{R}_E$ and R_E is given as follows,

$$\hat{R}_E (R_E)^T = \begin{bmatrix} c_{e_\psi} & -s_{e_\psi} \\ s_{e_\psi} & c_{e_\psi} \end{bmatrix} \quad (23)$$

where, $e_\psi = \hat{\psi}_E - \psi_E$ is the error in estimated yaw angle. The error in the measurement is given by,

$$\begin{aligned} e_y &= [e_y(1) \ e_y(2)]^T = y_{v_n} - O_C \\ &= \hat{R}_E (R_E)^T (O_C - O_E) - (O_C - \hat{O}_E) + \hat{R}_E w \end{aligned} \quad (24)$$

where $O_C - \hat{O}_E$ is the relative position of tracked vehicle w.r.t. the estimated ego frame given by,

$$O_C - \hat{O}_E = \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \end{bmatrix} = \begin{bmatrix} x_C - \hat{x}_E \\ y_C - \hat{y}_E \end{bmatrix} \quad (25)$$

where, $\hat{x}_E$ and $\hat{y}_E$ are the estimated ego position. On further solving (24),

$$\begin{aligned} e_y &= \hat{R}_E (R_E)^T \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \end{bmatrix} - \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \end{bmatrix} + \hat{R}_E w \\ &= \begin{bmatrix} c_{e_\psi} \hat{r}_x - s_{e_\psi} \hat{r}_y - \hat{r}_x \\ c_{e_\psi} \hat{r}_y + s_{e_\psi} \hat{r}_x - \hat{r}_y \end{bmatrix} + \hat{R}_E w \end{aligned} \quad (26)$$

Using small angle approximation, (26) can be further simplified as follows,

$$\begin{aligned} e_y &\approx \begin{bmatrix} \hat{r}_x - e_\psi \hat{r}_y - \hat{r}_x \\ \hat{r}_y + e_\psi \hat{r}_x - \hat{r}_y \end{bmatrix} + \hat{R}_E w \\ &= \begin{bmatrix} \hat{x}_E - x_E - e_\psi \hat{r}_y \\ \hat{y}_E - y_E + e_\psi \hat{r}_x \end{bmatrix} + \hat{R}_E w \end{aligned} \quad (27)$$

From (27), it can be observed that the error in the measurement scales up with the relative position of the tracked vehicle w.r.t. ego vehicle and is linearly proportional to the error in estimated ego position and estimated yaw angle. Fig. 5 shows the plot of variation of the error in measurement w.r.t. increase in relative distance for various values of e_ψ . From this plot it can be easily observed that even a 1° error in the estimated yaw angle can give an error of 1.5 m for a

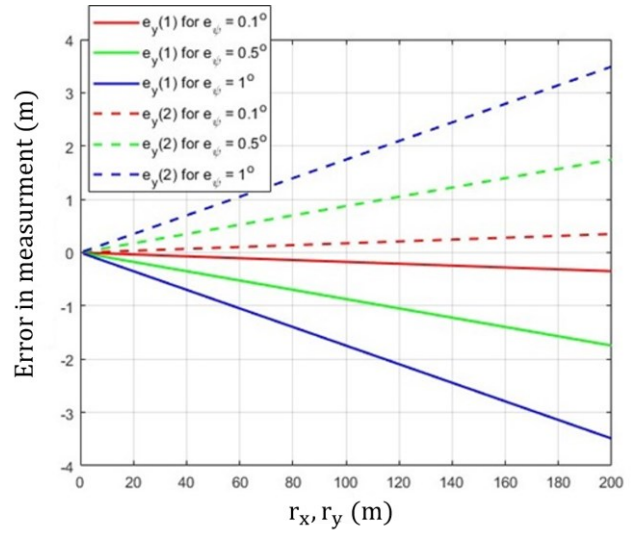


Fig. 5. The error in measurement Vs relative distance

relative distance of 100 m. Moreover, for vehicle tracking in the same lane, r_y is small compared to r_x introducing more error in measuring the lateral position of the tracked vehicle. For $r_x = 80$ m and $r_y = 5$ m, which is typical for lane following, it can be observed that even 1° error in yaw angle results in 1.39 m error in lateral measurement which is sufficient to misjudge the tracked vehicle as being in the adjacent lane. Hence it is very important to estimate the states of the ego vehicle such that there is least error in the measurement. Due to this reason, it was important to include heading measurements using GPS velocities in the previous sub-section.

For applications related to PD, it is very important to estimate the tracked vehicle accurately for displaying it to the teleoperator. Due to the increase in error in measurement with the relative distance, an upper bound was put on the relative distance. This upper bound is decided based on safe driving distance on highway which is around 80 m. If the tracked vehicle is within 80 m, vehicle tracking was performed. Otherwise, a delayed feed was displayed (without estimation). Given the vehicle tracking dynamics in (20) and the

measurement equations in (22), EKF as given in Eqs. (13)-(17) has been used to estimate the tracked vehicle variables, with f replaced with f_v and $H_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$.

Given the combined estimates of ego vehicle and tracked vehicle, the estimated vehicles were displayed to the teleoperator using the position and yaw angles of the vehicles. The complete flow chart of the PD system using state and vehicle tracking estimates is shown in Fig. 6.

It is important to know how the images are generated based

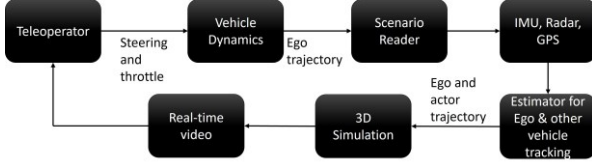


Fig. 6. Block diagram for estimation-based PD

on the estimates of ego vehicle and tracked vehicle. The Unreal Engine creates a colored mesh for the static environment (which includes the roads, trees, buildings and traffic lights) and loads the in-built colored mesh for the tracked vehicles. The tracked vehicles are placed in this environment based on the estimates of their position and yaw angle. A camera object is created on the ego vehicle based on the extrinsic and intrinsic matrix of the camera. The extrinsic matrix of the camera is computed based on the position and orientation of the ego vehicle while the intrinsic matrix is computed based on the pixel focal length and offset of principal points. Using the camera object, ray casting is done on the mesh of static scene and the tracked vehicles, and an image is generated to be shown to the teleoperator. A schematic of the complete PD algorithm in pseudo-code is shown in Algorithm 1. It is important to note that PD for real-time application requires a colored mesh which can be generated based on delayed point cloud and images but this task is beyond the scope of this paper and is an important research area in itself.

Algorithm 1: Estimation based predictive display

Require: Predictive Display $\leftarrow f(\text{Camera Intrinsic, IMU, GPS, Radar, } X_0^+, P_0, Q, R, X_{V0}^+, P_{V0}, Q_V, R_V)$

1. $X_0^+, P_0, Q, R, X_{V0}^+, P_{V0}, Q_V, R_V$
2. **while** The system runs **do**
3. ▷State Estimation:
4. $\hat{X}_{k+1}^- \leftarrow f(\hat{X}_k^+, \text{IMU})$
5. $P_{k+1}^- \leftarrow F_k P_k^+ F_k^T + Q$
6. $K_{k+1} \leftarrow P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R)^{-1}$
7. $\hat{X}_{k+1}^+ \leftarrow \hat{X}_{k+1}^- + K_{k+1} (\text{GPS} - H_{k+1} \hat{X}_{k+1}^-)$
8. $P_{k+1}^+ \leftarrow (I - K_{k+1} H_{k+1}) P_{k+1}^-$
9. ▷Vehicle Tracking:
10. $\hat{X}_{V,k+1}^- \leftarrow f_V(\hat{X}_k^+, \text{IMU})$
11. $P_{V,k+1}^- \leftarrow F_{V,k} P_{V,k}^+ F_{V,k}^T + Q_V$
12. $K_{V,k+1} \leftarrow P_{V,k+1}^- H_{V,k+1}^T (H_{V,k+1} P_{V,k+1}^- H_{V,k+1}^T + R_V)^{-1}$
13. $y_{V,n} \leftarrow \hat{R}_E(\hat{X}_{k+1}^+) \text{Radar} + \hat{O}_E(\hat{X}_{k+1}^+)$
14. $\hat{X}_{V,k+1}^+ \leftarrow \hat{X}_{V,k+1}^- + K_{V,k+1} (\text{GPS} - H_{V,k+1} \hat{X}_{V,k+1}^-)$
15. $P_{V,k+1}^+ \leftarrow (I - K_{V,k+1} H_{V,k+1}) P_{V,k+1}^-$
16. Predictive Display = UnrealEngine(Camera Intrinsic, $\hat{X}_{k+1}^+, \hat{X}_{V,k+1}^+$)
17. **end while**
18. **Result:** PD using state estimation of ego vehicle and other vehicles.

C. Models for Vehicle Tracking: Relative Vs Inertial

In previous literature, it has often been erroneously assumed that the state vectors of the tracked vehicle are relative to the ego vehicle. In other words, the state dynamics as given in (20), have been used by many authors to track vehicles [18-20], but with the states assumed to include relative distance variables, instead of inertial variables. Let the relative position of the tracked vehicle w.r.t. the ego vehicle be r'_{EC} , given by

$$\begin{aligned} r'_{EC} &= \begin{bmatrix} r'_x \\ r'_y \end{bmatrix} = (R_E)^T r_{EC} \\ &= \begin{bmatrix} c_E(x_C - x_E) + s_E(y_C - y_E) \\ -s_E(x_C - x_E) + c_E(y_C - y_E) \end{bmatrix} \end{aligned} \quad (28)$$

Consider a state vector of the relative motion model where the first two states are the relative longitudinal and lateral positions,

$$X_{vr} = [r'_x \ r'_y \ V_C \ A_C \ \psi_r \ \beta_C]^T = [x_1 \ \dots \ x_6]^T \quad (29)$$

Where, $\psi_r = \psi_C - \psi_E$ is the relative yaw angle.

The state dynamics of this relative model constructed similar to (20) is given by,

$$\dot{X}_{vr} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_3 \cos(x_5 + x_6) \\ x_3 \sin(x_5 + x_6) \\ x_4 \\ 0 \\ x_3 \sin(x_6) / l_r \\ 0 \end{bmatrix} = f_v(X_{vr}) \quad (30)$$

The output for the relative model is given by,

$$y_{vr} = r'_{EC} = C X_{vr} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} X_{vr} \quad (31)$$

The derivative of r'_{EC} w.r.t. time is as follows,

$$\begin{aligned} \dot{r}'_{EC} &= \begin{bmatrix} \dot{r}'_x \\ \dot{r}'_y \end{bmatrix} \\ &= \begin{bmatrix} c_E(\dot{x}_C - \dot{x}_E) + s_E(\dot{y}_C - \dot{y}_E) + \dot{\psi}_E r'_y \\ -s_E(\dot{x}_C - \dot{x}_E) + c_E(\dot{y}_C - \dot{y}_E) - \dot{\psi}_E r'_x \end{bmatrix} \end{aligned} \quad (32)$$

Assuming that the motion of the tracked vehicle in the inertial coordinates is correctly given by (20), then (32) can be simplified as follows,

$$\begin{aligned} \dot{r}'_{EC} &= \begin{bmatrix} c_E(V_C c_C - \dot{x}_E) + s_E(V_C s_C - \dot{y}_E) + \dot{\psi}_E r'_y \\ -s_E(V_C c_C - \dot{x}_E) + c_E(V_C s_C - \dot{y}_E) - \dot{\psi}_E r'_x \end{bmatrix} \\ &= \begin{bmatrix} V_C(c_E c_C + s_E s_C) - \dot{x}_E c_E - \dot{y}_E s_E + \dot{\psi}_E r'_y \\ V_C(s_C c_E - s_E c_C) + \dot{x}_E s_E + \dot{y}_E c_E - \dot{\psi}_E r'_x \end{bmatrix} \end{aligned} \quad (33)$$

where, $s_C = \sin(\psi_C + \beta_C)$ and $c_C = \cos(\psi_C + \beta_C)$. The above equation can be further simplified as follows,

$$\dot{r}'_{EC} = \begin{bmatrix} V_C \cos(\psi_C + \beta_C - \psi_E) - v_x + \dot{\psi}_E r'_y \\ V_C \sin(\psi_C + \beta_C - \psi_E) - v_y - \dot{\psi}_E r'_x \end{bmatrix} \quad (34)$$

where, $v_x = c_E \dot{x}_E + s_E \dot{y}_E$ and $v_y = c_E \dot{y}_E - s_E \dot{x}_E$. The error in the differential equation $e_r = [\dot{x}_1 \ \dot{x}_2]^T - \dot{r}'_{EC}$ is given by,

$$e_r = \begin{bmatrix} v_x - \dot{\psi}_E r'_y \\ v_y + \dot{\psi}_E r'_x \end{bmatrix} \quad (35)$$

It can be observed that $e_r = 0$, for the case when $v_x = v_y = \dot{\psi}_E = 0$, i.e. when the ego vehicle is stationary,

otherwise for a general vehicle motion $e_r \neq 0$, which means the two differential equations are not same indicating that the output is not compatible with the relative positions in the relative motion model. Moreover, the relative motion model does not consider the effect of the yaw rate of ego vehicle in the derivative of relative yaw angle ($\dot{x}_5$). Thus, for general vehicle motion such relative motion model will give inaccurate results and hence the observer design will also be inaccurate.

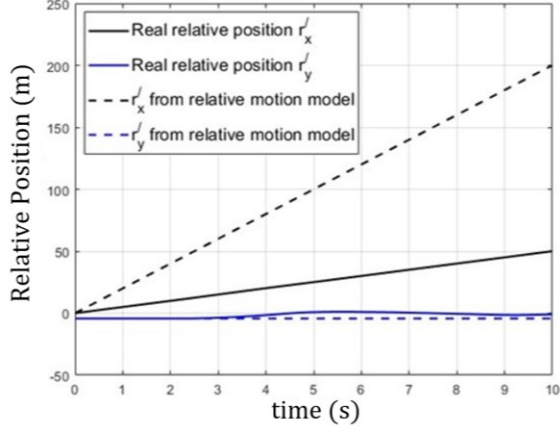


Fig. 7. Positions from relative motion model and measured outputs

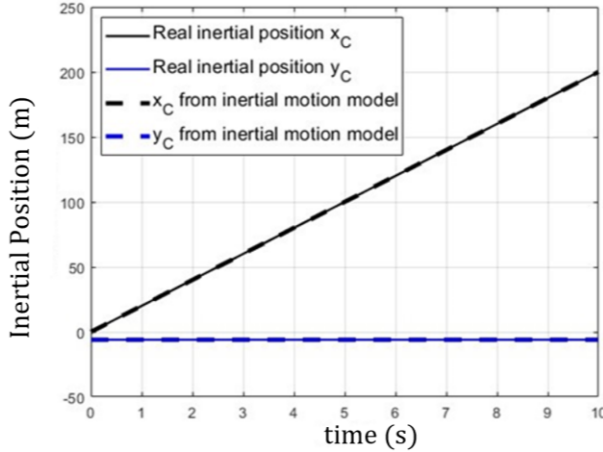


Fig. 8. Positions from inertial motion model and new outputs

One solution to address the inaccuracy in using the relative motion model is to use the inertial motion model as described in (19) and (20) and add the relative measurements obtained from the radar sensor to estimated ego state variables. The

new output for the inertial model is then given by,

$$y_{v_n} = R_E r'_{EC} + O_E = \begin{bmatrix} c_E & -s_E \\ s_E & c_E \end{bmatrix} \begin{bmatrix} r'_x \\ r'_y \end{bmatrix} + \begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (36)$$

It can be easily observed that if the motion of tracked vehicle in the inertial coordinates is given by (20), then the output given in (36) is consistent with the states in the inertial motion model. Thus, ego states estimates together with radar measurements can be used with the inertial motion model of the tracked vehicle to estimate its states as described in previous sub-section.

Consider the case when the ego vehicle is moving on a straight road with constant velocity of 6.07 m/s with another vehicle in the adjacent lane. Assume that the other vehicle moves with a constant velocity of 20 m/s on the straight road and the ego vehicle performs a lane change maneuver to the same lane as the target vehicle. The real relative position and the position obtained from the relative motion model is shown in Fig. 7. It can be easily observed that the real relative position is inconsistent with that of relative motion model, resulting in significant errors over time. However, if the ego states are used along with the inertial motion model, the position from the inertial motion model and inertial positions are consistent and there is negligible error as shown in Fig. 8. These results validate the inaccuracies in relative motion model and the importance of using inertial vehicle models along with ego state variables.

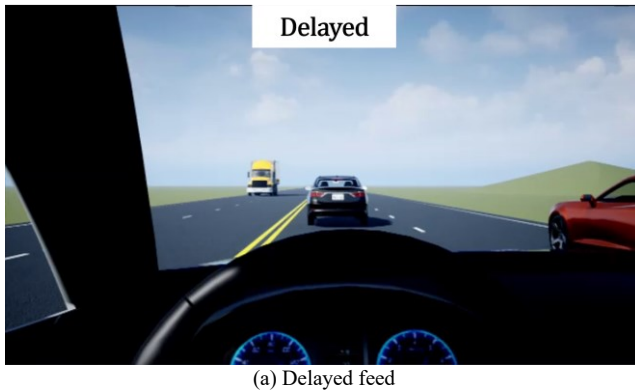
IV. RESULTS

To evaluate the performance of the estimation-based PD system, an experimental human subjects study was conducted and will be described in this section.

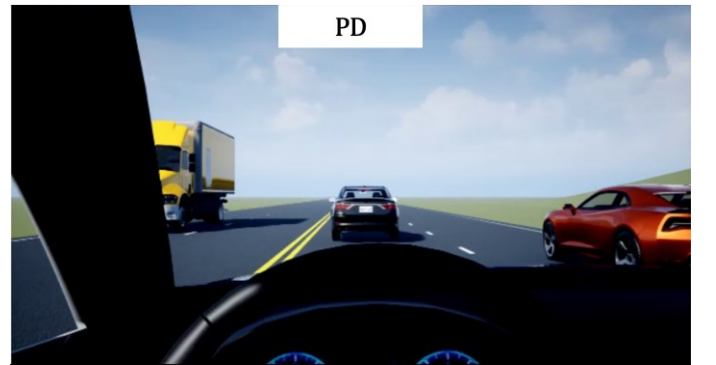
A. Image Comparison Analysis

First, a pixel-based image analysis has been used to evaluate the effectiveness of the estimation-based PD system as compared to the delayed display. Peak signal to noise ratio (PSNR), mean squared error (MSE) and structural similarity index measure (SSIM) have been used to compare both the delayed display and the PD enhancement with the un-delayed feed.

Given a reference image f and a test image g with size $M \times N$, where the location of pixel with coordinates (i, j) is given by f_{ij} , the MSE is given by,



(a) Delayed feed



(b) With PD

Fig. 9. Scenario for image comparison

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (37)$$

The lower the MSE the higher is the image quality. The PSNR between the reference image and test image is given as

$$PSNR(f, g) = 10 \log_{10}(255^2 / MSE(f, g)) \quad (38)$$

As PSNR increases the image quality also increases. The SSIM between the reference and test image is as given as follows,

$$SSIM(f, g) = \frac{(2\mu_f\mu_g + C_1)(2\sigma_{fg} + C_2)}{(\mu_f^2 + \mu_g^2 + C_1)(\sigma_f^2 + \sigma_g^2 + C_2)} \quad (39)$$

where, σ_{fg} , σ_f , σ_g , μ_f , μ_g are the cross-covariance, standard deviations and local means for images f and g . The SSIM ranges between 0 and 1. Zero SSIM means no correlation and an SSIM of one means $f = g$.

For the image comparison analysis, a straight road highway driving scenario was designed using MATLAB's Driving Scenario Designer and is shown in Fig. 9 which also shows the display feed with delay and with PD. In this specific comparison scenario only a constant throttle was provided to the ego vehicle. The scenario had three more vehicles one in the same lane, another in the adjacent lane and a truck in the opposite lane. All the vehicles had straight motion. A one second delay was present for the delayed feed and state estimation and vehicle tracking was used for the PD system. Both delayed display and PD enhanced display were compared with the un-delayed video display for the image comparison analysis.

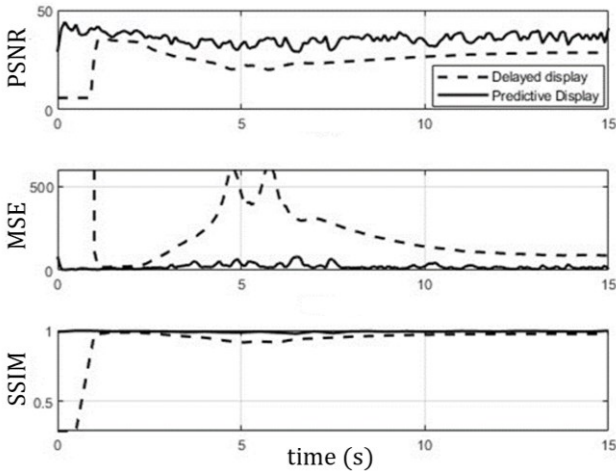


Fig. 10. Plots for PSNR, MSE and SSIM for delayed display and PD

TABLE I. IMAGE COMPARISON WITH AND WITHOUT PD

Metric	Scenario		% improvement using PD
	1 s delay	PD	
Average PSNR	25.28	35.8	41.59
Average MSE	189.51	21.56	88.62
Average SSIM	0.92	0.99	8

The plot of PSNR, MSE and SSIM for the delayed display and PD is shown in Fig. 10. Table I describes the average values of the Image comparison metrics.

B. Emergency Lane Change

An experiment was performed to analyze the effect of time delay on the steering control performance of the ego vehicle. Consider a scenario in which the ego vehicle follows another vehicle in the same lane but the other vehicle suddenly stops forcing the ego vehicle to change lanes. The whole scenario is depicted in Fig. 11 where the blue car is the ego vehicle, and the red car is the other vehicle.

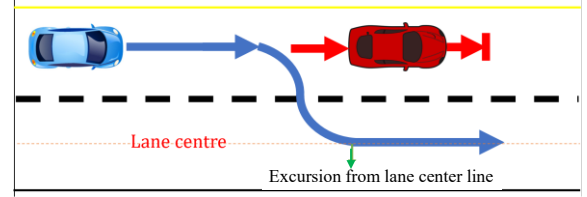


Fig. 11. Lane change scenario

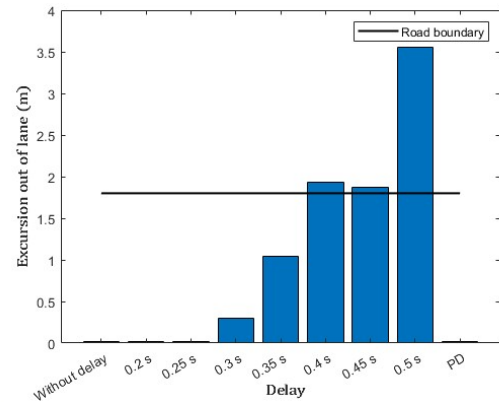


Fig. 12. Excursion from lane center for ego vehicle with various values of delay

This scenario was studied for the no-delay case, with various values of delay and with PD to evaluate the degradation caused due to delay and the effectiveness of PD. Previous work from the literature showed that even a 0.17 s delay can start causing degradation in performance, hence the scenario was analyzed for the following values of time delay: 0.2 s, 0.25 s, 0.3 s, 0.35 s, 0.4 s, 0.45 s and 0.5 s and compared to the no-delay (i.e. 0.01 sec sampling) case. The performance for this scenario was evaluated using a metric of excursion from the lane center, as shown in Fig. 11. The results for the excursion from lane center of the ego vehicle without delay, with various values of delay and with PD are shown in the bar chart given in Fig. 12. From this figure it is clear that the degradation in teleoperator's performance is quite high at 0.5 s delay and hence the larger human subjects study was designed based on this value of delay.

C. Human Subjects Study

To further evaluate the performance of the estimation-based PD system, a human subjects study has been conducted. In this study the data of five teleoperator participants was analyzed to evaluate the degradation caused due to delay and the effectiveness of the PD system. For the human subject

study, a curved road scenario was used. The curved road extends from 1400 m to 200 m in x direction and -200 m to 1600 m in y direction as shown in Fig. 13.

The curved road has 4 lanes of width 3.85 m each and have

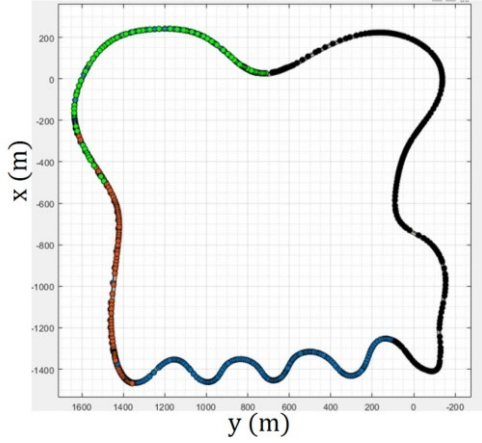


Fig. 13. Curved road scenario for human subjects study

been rendered in the teleoperation platform as shown in Fig. 14 and was displayed to the participants. The display included the cockpit of the ego vehicle along with the speed display (in miles per hour (mph)).



Fig. 14. Real-time display to the teleoperator during remote driving

Each participant was required to drive the vehicle in three tests. In the first test, the participants drove the vehicle without any delay, in the second test with a delay of 0.5 s and in the third test with estimation-based PD. In each test, the participant drove the vehicle for 10 mins. For the first few minutes, the participants had to drive without any vehicle in the lane, then they had to follow a red sedan, then sole driving on an extremely curved road, then follow a black SUV and then a green sports car. This scenario was common to all the three tests. Thus, a total of 15 experiments were conducted for the human subjects study. The participants were instructed to always try to keep the ego vehicle in the same lane, drive at a speed of 30 to 35 mph and if they encountered another vehicle, they had to follow it even if their speed had to be reduced but never leave the lane even if the other vehicle does.

The sensor specifications of the Ego vehicle are provided in Table II which describe the specifications of GPS, IMU,

camera and radar.

TABLE II. SENSOR SPECIFICATION OF EGO VEHICLE

Sensor Specification	Value
GPS position accuracy (m)	0.1
GPS velocity accuracy (m/s)	0.1
GPS rate (Hz)	100
Accelerometer initial bias (m/s^2)	0.0141
Accelerometer VRW (mg)	0.2
Gyroscope initial bias (deg/s)	0.0573
Gyroscope ARW ($\text{deg}/\sqrt{\text{Hz}}$)	0.21
IMU rate (Hz)	100
Radar accuracy (m)	0.1
Radar rate (Hz)	100
Camera field of view horizontal (deg)	56.72
Camera field of view vertical (deg)	87.66
Camera frame rate (FPS)	100

For estimation-based PD, the sensor data was transmitted to the teleoperator side and due to very small data size for these variables, it was assumed that there is no delay in the transmission. An EKF based state estimation and vehicle tracking was then done to modify the delayed feed for PD. Table III describes the state estimation error for each of the

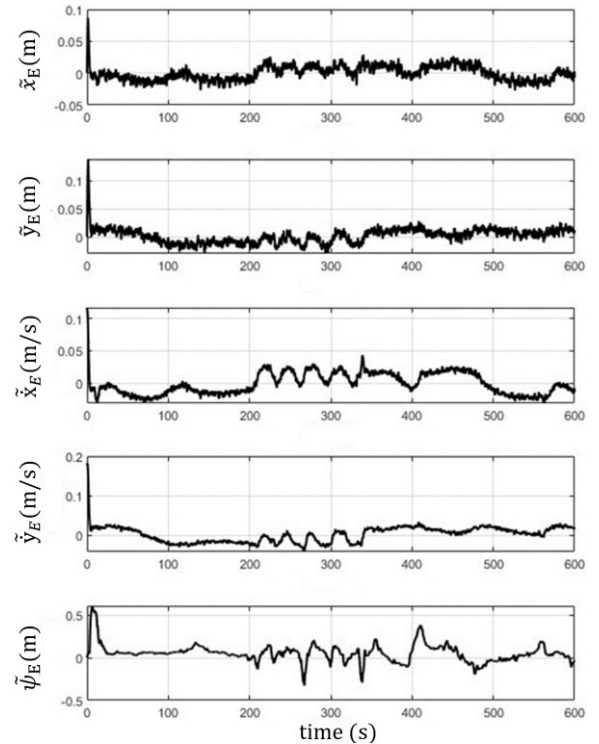


Fig. 15. Error in state estimation of ego vehicle

TABLE III. STATE ESTIMATION ERROR FOR EACH PARTICIPANT

Participant	$\tilde{x}$ (m)		$\tilde{y}$ (m)		$\tilde{\psi}$ (deg)	
	RMSE	Max error	RMSE	Max error	RMSE	Max error
1	0.012	0.087	0.013	0.139	0.115	0.607
2	0.012	0.087	0.014	0.139	0.119	0.539
3	0.012	0.087	0.014	0.139	0.167	0.825
4	0.011	0.087	0.013	0.139	0.108	0.4
5	0.011	0.087	0.013	0.139	0.100	0.472

TABLE IV. VEHICLE TRACKING ERROR FOR EACH PARTICIPANT

Participant	$\tilde{x}$ (m)		$\tilde{y}$ (m)		$\tilde{\psi}$ (deg)	
	RMSE	Max error	RMSE	Max error	RMSE	Max error
1	0.039	0.347	0.076	0.690	1.26	9.76
2	0.058	0.347	0.082	0.690	1.28	9.62
3	0.140	0.677	0.090	0.692	1.30	9.38
4	0.044	0.347	0.099	0.689	1.29	9.85
5	0.040	0.347	0.072	0.690	1.24	9.37

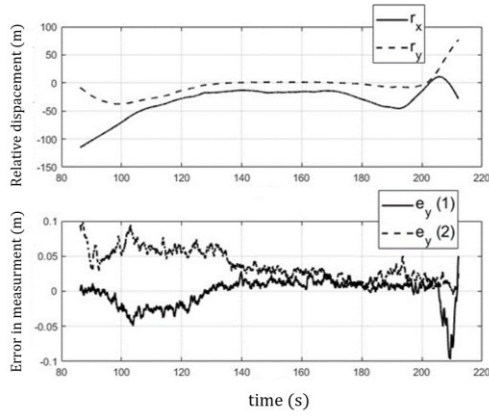


Fig. 16. Error in measurement for Red Sedan

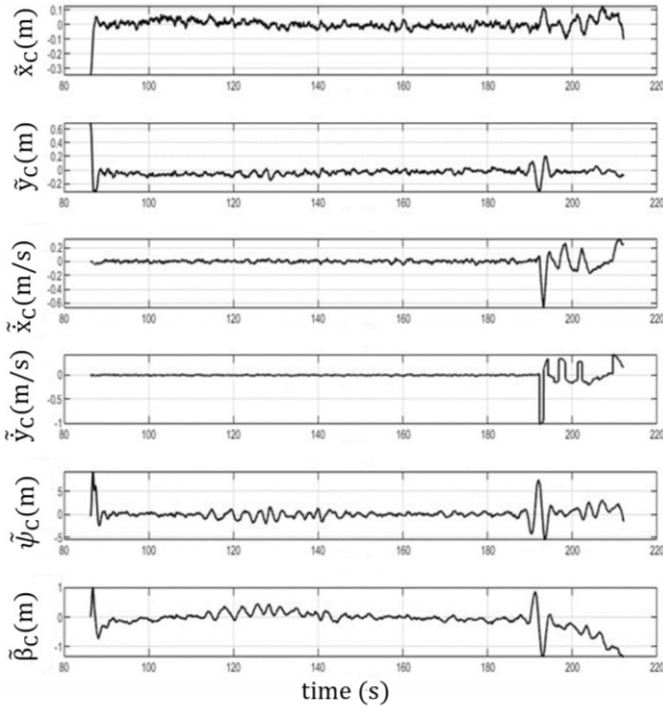


Fig. 17. Vehicle Tracking for Red Sedan

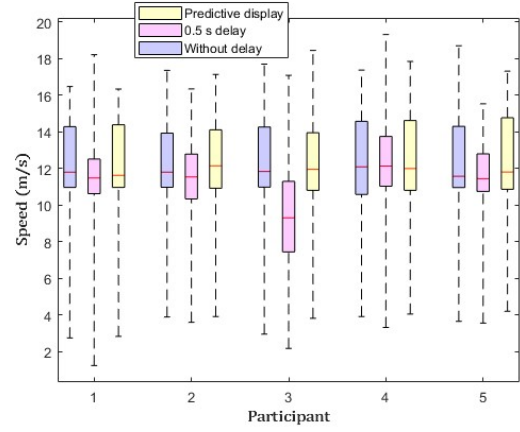


Fig. 18. Box plot for speed of each participant

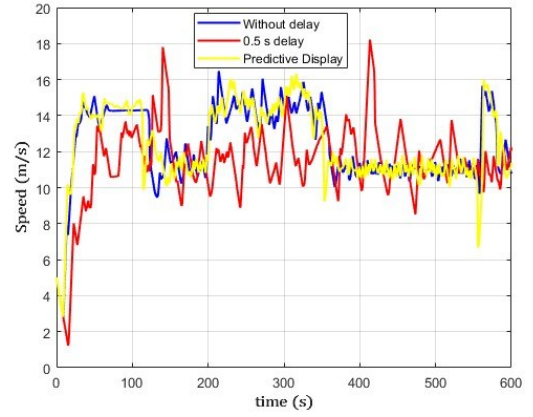


Fig. 19. Speed Vs time for one of the participants

participants. The state estimation error is $\tilde{e} = e - \hat{e}$, where e is the true value and $\hat{e}$ is the estimated value. The state estimation results for a sample participant are shown in Fig. 15. Table IV shows the vehicle tracking error for red sedan (actor 1) for each of the participants. The relative distance and error in measurement for the red sedan is shown in Fig. 16 and Fig. 17 shows the error in vehicle tracking for a sample participant.

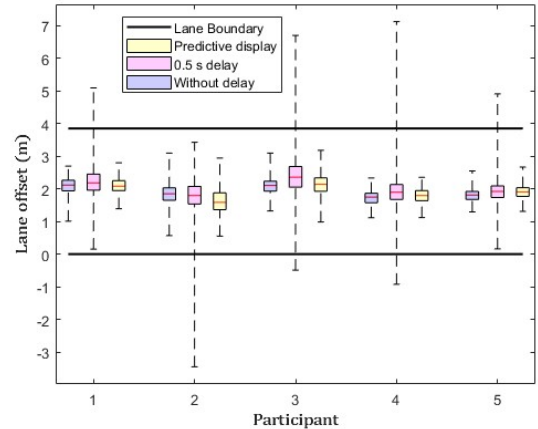


Fig. 20. Box plot for lane offset of each participant

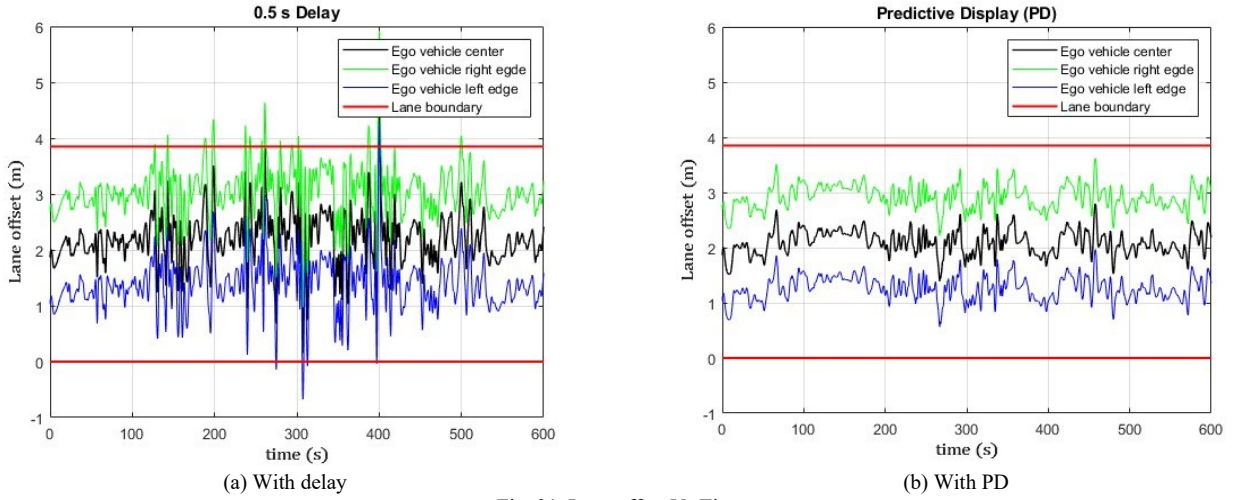


Fig. 21. Lane offset Vs Time

For evaluating longitudinal control performance, the speed of the ego vehicle for each participant has been computed. Fig. 18 shows the box plot for the speed of each participant over the whole experiment. In all box plots, the lower and upper boundaries of the whisker correspond to the minimum and maximum value of data respectively and the red line in the box corresponds to the median of the data. Fig. 19 shows the plot of the speed of a sample participant for all the three driving cases.

For evaluating the lateral control performance for the ego vehicle of each participant, the lane offsets have been evaluated for each participant. The box plot for the lane offset of each participant for all three cases is given in Fig. 20. The plot for the lane offset of the ego vehicle for both delayed and PD cases for a sample participant is shown in Fig. 21.

For vehicle following performance the relative distance from the tracked vehicle has been evaluated for each participant. The box plot for the relative distance from the red sedan has been shown in Fig. 22. The plot shows the participants performances for all three cases.

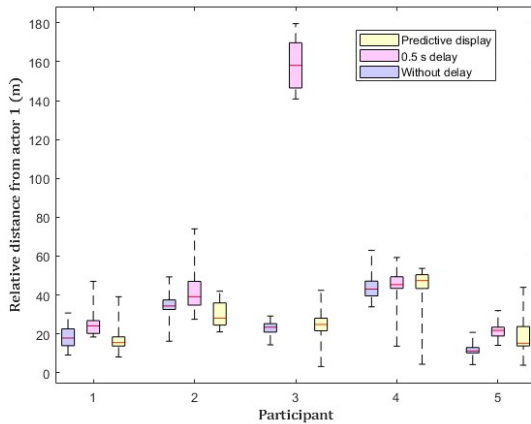


Fig. 22. Box plot for relative distances of each participant

The overall results for the human subjects study are summarized in Table V. The metrics are an average over all the five participants. The average speed and distance covered were used as metrics to evaluate the longitudinal

performance. The maximum distance outside the lane and the number of times the vehicle moved outside the lane were used for evaluating lateral performance. The relative distance to the tracked vehicle were used for evaluating the vehicle following performance. The table shows the data for all three cases and percent change due to delay and PD case as compared to the without delay case.

TABLE V. RESULTS FROM TELEOPERATION STATION STUDY

Metric	Average Values			% change due to delay	% change after PD
	Without delay	0.5 s delay	PD		
Average speed (m/s)	12.38	11.35	12.4	8.3	0.16
Distance covered (km)	7.43	6.75	7.44	9.15	0.13
Max distance outside lane (m)	0.07	3.2	0.09	4471	28.6
Number of times outside lane	0.6	9.6	0.8	1500	33.3
Relative distance Red Sedan (m)	26.8	58.77	27.44	119.3	2.38
Relative distance Black SUV (m)	24.16	98.41	28.1	307.3	16.3

V. DISCUSSION

A. Image Comparison Analysis

The image comparison metric as shown in Fig. 10 clearly shows that PSNR for PD is always higher than that of the delayed display, the MSE for PD is always less than that of the delayed display and the SSIM of PD is closer to 1 as compared to delayed display. The results indicate that image quality using PD is closer to the undelayed video as compared to the delayed display. Table I describes the average values of the Image comparison metrics. It can be clearly observed from the table that estimation-based PD is able to increase the average PSNR by 41.59 %, decrease the average MSE by 88.62 % and increase the average SSIM by 8 %. Hence, it can

be concluded from the Image comparison analysis that the use of state estimation and vehicle tracking in estimation-based PD is able to enhance the delayed video display.

B. Emergency Lane Change

The excursion from lane center performance as shown in Fig. 12 indicates that the ego vehicle is able to perform the emergency lane change for the no delay case and for delay values less than 0.3 s. In this test case the speed of the participant was less than 35 mph because of which a delay lower than 0.3 s did not hamper the performance of the teleoperator. But in prior work a degradation was observed even at 0.17 s because the speed of the ego vehicle was 55 mph which is quite high. In the current test case, the performance of the teleoperator starts degrading after 0.3 s delay where a slight excursion is seen. This degradation in performance peaks after 0.4 s when the ego vehicle starts moving out of road boundary and at 0.5 s the teleoperator is not able to control the vehicle properly as indicated by the large out of lane excursion. In fact, for the 0.5 s case, the ego vehicle goes completely off the road during the lane change. This clearly indicates that a 0.5 s delay can hamper teleoperation drastically and hence this value of delay was used for the larger human subjects study. The last bar in Fig. 12 shows the performance for the PD case. From the plot it is clear that using PD allows the teleoperator to keep the ego vehicle in lane and gives a performance close to that of the no-delay case. Thus, using PD effectively compensates for latency and provides performance much better than delayed cases.

C. Human Subjects Study

The error in state estimation for each participant, as shown in Table III, indicates the error in estimates for position and yaw angle. From the table it is clear that the estimator is able to achieve an accuracy of 1.2 cm (RMSE) using the GPS measurements which have an accuracy of 10 cm. Moreover, the use of heading angle for measuring the yaw angle results in more accurate estimation of the yaw angle giving an accuracy of 0.1 deg (RMSE). - This indicates that (12) is a valid approximation for measuring yaw angle when the slip angle is not high. The 0.1 deg accuracy of yaw angle ensures that the error in measurement for vehicle tracking will be less than 0.2 m for a relative distance of 80 m, thus allowing for accurate vehicle tracking. From Fig. 15, it is clear that the error in the position and velocity is of the order of 5 cm and 5 cm/s. The error in the yaw angle is less than 0.50 indicating that the filter is able to estimate the yaw angle accurately.

The error in vehicle tracking for each of the participants, as shown in Table IV, indicates that the error in the position is of the order of 5 cm (RMSE). Only for Participant 3 the error is more than 10 cm, this is because of the fact that the relative distance between the ego vehicle and tracked vehicle was quite high (as will be described later in this section) resulting in increased error in measurement thus hampering the accuracy of the estimator. Moreover, the error in estimate of yaw angle is less than 1.25 deg. Thus, the estimator is able to accurately estimate the position and yaw angle of the tracked vehicle - Only in cases when the relative distance is high the

performance of the estimator decreases. These results also show that the accuracy of state estimation for the ego vehicle is sufficient to perform accurate vehicle tracking.

From Fig.16 it can be observed that r_x is larger than r_y till 200s after which r_y starts increasing, this is because till 200s the red sedan is in the same lane as the ego vehicle and starts to change lane at around 196s. Due to this the error in measurement in y ($e_y(2)$) is larger as compared to the error in measurement in x ($e_y(1)$) till 200s but after the lane change maneuver error in measurement in x ($e_y(1)$) starts increasing. Since the error in the estimated yaw angle and position is less than 2.50 and 0.05 m, the error in the measurement is less than 0.1 m. From Fig. 17, it can be observed that the estimator is able to track the position of the target vehicle accurately along with the velocity, acceleration, yaw angle and the side slip angle of the vehicle. However, when the vehicle changes lane (at around 196 s) the assumption of constant acceleration is not valid and hence the accuracy of acceleration estimation decreases which affects other estimates also.

The box plot for the speed of each participant given in Fig. 18, is indicative of the longitudinal control performance of the ego vehicle. The higher the speed of the ego vehicle for the ego vehicle, the higher will be the longitudinal performance of the participant. From the plot we can clearly observe that the median speed for each of the participant for the without delay case and PD case is greater than that of delayed case. The decrease in speed of the vehicle due to time delay is also observed in previous works and this study proves it even more. Because of the presence of time delay there is a lag in the visual feed of the teleporter, resulting in increased efforts to control the vehicle thus compromising the speed. However due to accurate ego vehicle state estimation the PD is able to improve the speed of the teleoperator such that it is better than delayed case and is closer to the without delay case. To analyze this further, the plot of the speed of a sample participant for all the three cases as shown in Fig. 19 were examined. From the plot it is clear that the speed of the participant for the delayed case is less than that of the without delay and PD case. Moreover, high magnitude of variation in speeds indicates difficulty in driving the vehicle in the delayed case. The speed of the participant for the PD case is comparable to that of the without delay case and is higher as compared to delayed case indicating an improved performance in longitudinal control. From Table V, it can be observed that both the average speed and distance covered with 0.5 s delay decreases drastically but when PD is used the performance is much closer to the without delay case and there is an increase of around 8 % in these metrics when PD is used as compared to delayed case.

The lateral offset of the center of the ego vehicle for each participant as shown in Fig. 20, indicates that the vehicle center was always inside the lane for the without delay and PD case. However, it went outside the lane boundaries for all the participants when there was a delay of 0.5 s. The results clearly indicate that the delay hampers the lateral control performance as more control effort is need to stabilize the ego vehicle. To analyze this further, the plot of the lateral offset as shown in Fig. 21 (which also shows the right and left edge

of the ego vehicle) was examined. From the plot it is clear that the ego vehicle goes outside the lane many times for the delayed case, moreover at around 400 s the whole vehicle is outside the lane. But for the PD system the lane keeping performance is much better than the delayed case as the ego vehicle is always within the lane boundaries. Table V indicates that the maximum times the vehicle goes out of lane is much less for PD as compared to delayed case. Moreover, the frequency of vehicles moving outside the lane is 1467 % more in the delayed case as compared to the PD case.

The box plot for the relative distance to actor 1, as shown in Fig. 22, indicates that the relative distance between the tracked and ego vehicles is always more in the delayed case as compared to the no-delay and PD cases. This indicates that the vehicle following performance degrades when there is a delay but improves when PD is used. Furthermore, the average relative distance with the two tracked vehicles is significantly less in PD as compared to the delayed case and much closer to the without delay case. Also, PD decreases the relative distance by 117 % with the red sedan and 290 % for the black SUV as compared to the delayed case.

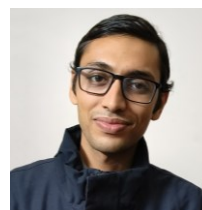
VI. CONCLUSION

In this paper, an estimation-based predictive display (PD) system was designed to improve teleoperation performance with autonomous vehicles. The teleoperation application suffers from a problem of latency in transmitting images to the teleoperation station which can degrade the teleoperator's performance. A MATLAB-based human in-loop teleoperation environment was developed to evaluate the degradation caused due to delay and the effectiveness of PD in compensating for the delay. State estimation and vehicle tracking were used by the PD system in predicting the position of ego and non-ego vehicles. A novel vehicle tracking algorithm was developed using an inertial motion model and ego state estimates. Error analysis was performed to further analyze the new vehicle tracking algorithm. Image comparison analysis compared delayed and modified displays (based on the developed PD algorithm) with the original undelayed display, and results indicated that PD increased the performance of the display using metrics of PSNR, MSE, and SSIM. A human subjects experimental study demonstrated the determinantal effect of even a 0.5 s delay and improvements obtained by PD in longitudinal control, lateral control, and vehicle following. Although this study proves the efficacy of PD, it used a 0.01 second update rate for measurements of GPS, IMU and radar. There is a need to analyze the performance of PD at lower update frequencies which will be a part of future research.

REFERENCES

- [1] S. Neumeier, N. Gay, C. Dannheim and C. Facchi, "On the Way to Autonomous Vehicles Teleoperated Driving," AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium, Dortmund, Germany, 2018, pp. 1-6.
- [2] Startup Vay's Autonomy Workaround: Teledrivers to Operate Cars from Remote Location, URL:

- <https://www.caranddriver.com/news/a37648114/vay-autonomous-teledriver-startup/>
- [3] This Driverless Car-Sharing Service uses Remote Human "Pilots", not AI, URL: <https://www.fastcompany.com/90653650/halo-driverless-car-sharing-service>
- [4] Gnatzig, S., Chucholowski, F., Tang, T., and Lienkamp, M. (2013). A system design for teleoperated road vehicles. ICINCO (2), 231–238.
- [5] Jessie Y. C. Chen, Ellen C. Haas, and Michael J. Barnes. Human performance issues and user interface design for teleoperated robots. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2007.
- [6] James Davis, Christopher Smyth, and Kaleb McDowell. The effects of time lag on driving performance and possible mitigation. IEEE Transactions on Robotics, 26(3):590–593, 2010.
- [7] Frederic Chucholowski. Evaluation of display methods for teleoperation of road vehicles. Journal of Unmanned System Technology, 3:80–85, 02 2016. doi: 10.21535/just.v3i3.38.
- [8] Sato et al. Implementation and evaluation of latency visualization method for teleoperated vehicle. In 2021 IEEE Intelligent Vehicles Symposium (IV), pages 1–7, 2021.
- [9] Frederic Chucholowski. Eine vorausschauende Anzeige zur Teleoperation von Straßenfahrzeugen. PhD thesis, 03 2016.
- [10] Gaetano Graf, Hao Xu, Dmitriy Schitz, and Xiao Xu. Improving the prediction accuracy of predictive displays for teleoperated autonomous vehicles. In 2020 6th International Conference on Control, Automation and Robotics (ICCAR), pages 440–445, 2020.
- [11] Yingshi Zheng, Mark J. Brudnak, Paramsothy Jayakumar, Jeffrey L. Stein, and Tulga Ersal. Evaluation of a predictor-based framework in high-speed teleoperated military ugv's. IEEE Transactions on Human-Machine Systems, 50(6):561–572, 2020.
- [12] Henrikke Dybvik, Martin Løland, Achim Gerstenberg, Kristoffer Bjørnerud Slattsveen, and Martin Steinert. A low-cost predictive display for teleoperation: Investigating effects on human performance and workload. International Journal of Human-Computer Studies, 145: 102536, 2021.
- [13] MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation. J Intell Robot Syst, 106, 2022.
- [14] Frank, L. H., Casali, J. G., & Wierwille, W. W. (1988). Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator. Human Factors, 30(2), 201-217.
- [15] Gade, K. (2016). The Seven Ways to Find Heading. The Journal of Navigation, 69(5), 955-970.
- [16] Ji, L., Sun, R., Cheng, Q. *et al.* Evaluation of the performance of GNSS-based velocity estimation algorithms. *Satell Navig* 3, 18 (2022).
- [17] den Ouden J, Ho V, van der Smagt T, Kakes G, Rommel S, Passchier I, Juza J and Tafur Monroy I (2022) Design and Evaluation of Remote Driving Architecture on 4G and 5G Mobile Networks. *Front. Future Transp.* 2:801567. doi: 10.3389/ffutr.2021.801567
- [18] H. Alai, A. Zemouche, et al, "On Challenges in Coordinate Transformation for Using a High-Gain Multi-Output Nonlinear Observer," Proceedings of the 2023 American Control Conference (ACC), San Diego, CA, USA, 2023, pp. 1024-1029.
- [19] W. Jeon, A. Zemouche and R. Rajamani, "Tracking of Vehicle Motion on Highways and Urban Roads Using a Nonlinear Observer," in IEEE/ASME Transactions on Mechatronics, vol. 24, no. 2, pp. 644-655, April 2019, doi: 10.1109/TMECH.2019.2892700.
- [20] Rajamani, R., Jeon, W., Movahedi, H. and Zemouche, A., 2020. On the need for switched-gain observers for non-monotonic nonlinear systems. Automatica, 114, p.108814.



Gaurav Sharma obtained his B.Tech. degree in mechanical engineering from the Birla Institute of Technology and Sciences, Pilani in 2021. He is currently pursuing his Ph.D. degree at the University of Minnesota, Twin Cities, MN, USA. His research interests are in state estimation, and control, including applications in intelligent transportation and robotic systems.



Rajesh Rajamani (*Fellow, IEEE*) obtained his M.S. and Ph.D. degrees from the University of California at Berkeley in 1991 and 1993 respectively and his B.Tech degree from the Indian Institute of Technology at Madras in 1989. Dr. Rajamani is currently the Benjamin Y.H. Liu-TSI Endowed Professor of Mechanical Engineering and Associate Director of the Minnesota Robotics Institute at the University of Minnesota. His active research interests include sensing and estimation

for smart mechanical systems.

Dr. Rajamani has co-authored over 185 journal papers and is a co-inventor on 14 granted patents. He is the author of the popular book “Vehicle Dynamics and Control” published by Springer. Dr. Rajamani has served as Chair of the IEEE Technical Committee on Automotive Control and is currently Senior Editor of the *IEEE Transactions on Intelligent Transportation Systems*. He is a Fellow of IEEE and ASME and has been a recipient of the CAREER award from the National Science Foundation, the Ralph Teetor Award from SAE, the Charles Stark Draper Award from ASME, the O. Hugo Schuck Award from the American Automatic Control Council, and a number of best paper awards from conferences and journals. Several inventions from his laboratory have been commercialized through start-up ventures co-founded by industry executives. One of these companies, Innotronics, was recently recognized among the 35 Best University Start-Ups of 2016 by the US National Council of Entrepreneurial Tech Transfer.