Predictive Display With Perspective Projection of Surroundings in Vehicle Teleoperation to Account Time-Delays

Jai Prakash®, Michele Vignati®, Member, IEEE, Daniele Vignarca®, Edoardo Sabbioni®, and Federico Cheli®, Member, IEEE

Abstract—Teleoperation provides the human operator with sophisticated perceptual and cognitive skills in an over-the-network control loop. It gives hope of addressing some challenges related to vehicular autonomy which is based on artificial intelligence by providing a fallback plan. Variable network time-delay in data transmission is the major problem in teleoperating a vehicle. On 4G network, variability of this delay is significant (70-150 ms ping). Due to this, both video streaming and driving commands encounter variable time-delay. This paper presents an approach to provide the human-operator with a forecasted video stream that replicates future perspective of vehicle’s field of view accounting for the delay present in the network. Regarding the image transformation, perspective projection technique is combined with correction given by Smith predictor in the control loop. This image transformation accounts current time-delay and tries to address both issues, time-delays as well as variability. For experiment sake, only frontward field of view is forecasted. Performance is evaluated by performing vehicle teleoperation with real vehicle on street edge-case maneuvers and later comparing the cross-track error with and without perspective projection. Results obtained show improvement in path following tasks.

Index Terms—Latency, perspective projection, predictive display, time-delay, vehicle teleoperation.

I. INTRODUCTION

TELEOPERATION indicates operation of a system from a distance. It means that there is no direct interference between human operator and teleoperated object. In vehicle teleoperation, teleoperated object is the vehicle (figure 1). The human operator controls the vehicle from the control station while the vehicle is on the road. Control station (figure 2) is a fixed control center that provides facilities for human interaction. It consists of display screens, speakers, and steering wheel/pedals control. Human-operator steers the steering joystick [1], [2], then the commands get actuated at the vehicle through a data communication protocol. Later, the commands get actuated at the vehicle through the actuators installed inside the vehicle.

The communication protocol can either be wired or wireless. To avail maximum benefits of teleoperation, this work aims wireless communication based on 4G LTE wireless broadband communication which is widely available across the world. This makes it a compelling choice to be used as data communication medium. With 4G, a vehicle can be operated miles away from its real location. Its potential applications could be remote last-mile delivery of rental/shared vehicles, avoiding driver presence in dangerous areas, human assistance in case of fallback of autonomous vehicles, and valet parking, etc. Because of many reasons, the driving experience of the human operator would not be exactly same as compared to driving the vehicle from inside of it. First, human-operator is not able to feel vehicle acceleration while sitting on a seat inside the control station. Second, although display screens try to emulate windshield view, display screens are usually smaller than windshield. Moreover, in cases where the camera is mounted outside the vehicle on its roof for better sensing, human operator observes different perspective as compared to perspective from inside of the vehicle. Altogether, visual input to human operator is different than visual input to a real vehicle driver. Third, is the absence of real haptic feedback. Haptic feedback are steering feedback force, pedal press force. Fourth, and most critical aspect in teleoperation, is time-delay or latency. Time-delay is defined as the time passing between the user’s input and its displayed response [3]. For human-in-the-loop control systems, time-delay has been considered to affect performance factors. The accuracy of control actions deteriorates because of operator’s inability to visualize or predict the outcome of his control actions. Humans can adapt to time-delays in control systems [4], however, this adaptation depends on human operator ability to predict the outcome of his control actions [5], and the extent of this adaptation is dependent upon the characteristics of the time-delay (e.g., magnitude and variability) [6].

Studies on human performance in virtual environments show that people are generally able to detect latency as low as 10–20 ms [7]. In a simulated driving task, the driver’s vehicle control is found to be significantly degraded with a latency of even 170 ms [8].
Time-delay components in teleoperation are camera frame capture delay (exposure delay), data communication delay, image-processing delay, human-operator response time, and vehicle actuation delay. Camera capture delay with a general purpose USB3.0 camera for 672x376 image frame is $\sim 70$ms [9]. It can be considered constant for constant illuminance of ambient light. Due to independent treatment of pixels (section II-B-II), image-processing delay is constant for constant image resolution. Human-operator response time is present in both teleoperation and real vehicle drive. Data communication delay is the bottleneck in time-delay cycle. During vehicle teleoperation, vehicle streams images to the control-station in the form of jpeg images and the control-station transmits driving commands to the vehicle as better described in the following. Bandwidth consumption during this data communication is around 1 MBps. Figure 3 shows the round-trip delay observed while performing vehicle teleoperation. Here the data corresponds to 1000 image frames and corresponding 1000 driving commands. This test is performed in a typical urban environment with the vehicle connected to 4G mobile communication and control-station connected to internet using wired LAN. Lastly, dynamics of vehicle actuation in non-aggressive driving can be considered fast and delay associated with it can be considered constant ($\sim 20$ms).

Several mechanisms have been demonstrated to counter the effects of time-delay including mathematical predictors/filters, predictive displays, command displays, and observer-based models. Mathematical filters/predictors are Kalman filter predictors and Smith predictors. Predictive and command displays try to counter the time-delay issue by providing immediate feedback to the operator through model representation. Command displays differ from predictive displays, as they require the remote system to be autonomous [10]. Smith predictor approach [11], [12], [13], [14], [15], [16] has the potential to mitigate the negative effect of the time delay. But it is highly dependent on the accuracy of the predictor model. Teng and Grant [17], uses online parameter estimation techniques for the predictor model, to adapt changes in predictor model in real-time.

Liu et al. [18], investigated two visual information display arrangement strategies: (i) presenting display frames to a remote driver as soon as the frames arrive, and (ii) smoothing the display by adding additional delay when necessary to the received frame to mitigate the delay variance. They demonstrated that in 2nd arrangement, mitigates the negative effect of variable delay incurred by LTE networks. In both strategies, operator is responsible to adapt to the delay and take maneuvering decisions by cognitively predicting vehicle position, which induces mental fatigue.

An observer-based framework [19], [20], [21], [22], in which a model-free approach is used to estimate the undelayed state of the teleoperator vehicle. Here, the closed-loop dynamics of the observer is based on a sliding surface. A blended prediction of the heading signal is considered by linearly combining the model-free prediction with a steering model. Zheng et al. [23] found that the blended architecture offers improvement when compared to the purely model-free realization. However, in human-in-the-loop experiments, view of the predicted position is generated directly from the simulation environment and not by the image processing of the delayed image.

Using command displays approach, Fennel et al. [24], proposed an offline path follower where the operator “draws” a desired 2D path by walking in a large-scale haptic interface while a guiding force is exerted, which ensures that the generated path can later be accurately followed by a path tracking controller running offline on a remote robot. However, in this strategy, operator is not actively controlling the task.

Chen [25], proposed a safety concept for teleoperated vehicles, called free corridor, with which the path of an emergency braking of the vehicle is calculated continuously and visualized as augmented reality in the received images at the operator workstation. With this concept, it is the task
of the human operator to continuously hold this path free while driving. In case of a communication loss, an emergency brake will be activated and the vehicle will brake along the before-visualized path. However, in estimating the free corridor, the delay associated with the driving command from control-station to vehicle is not considered. Which may cause a slight deviation in the predicted and actually realized vehicle trajectory.

In a previous work [26], the authors briefly introduce Perspective projection technique. Perspective projection (PP) or perspective transformation is a linear projection where 3D objects are projected on an image plane [27]. There the authors present post-processing results i.e., data are collected with the vehicle and processed offline to validate the algorithm. The approach was found to be able to generate the new perspective of the forward displaced camera, taking input as the image captured at the previous position of the camera.

In literature [11], [12], [13], [14], [15], [16], [19], [20], [21], [22], [23], [25], position of the vehicle is estimated using predictive techniques and then either predicted position is overlayed on the delayed image using coloured markers (figure 4a) or field of view (FOV) at the predicted position is generated directly from the simulation environment by placing a virtual camera at predicted position (which is infeasible in reality). This paper utilizes PP to generate the FOV at the predicted position in reality (figure 4b), which provides realistic perspective compared to using markers. The undelayed states are predicted by the Smith predictor in the control-station which employs a model of the vehicle. The predicted position corresponds to the position where the vehicle would receive driving commands back, corresponding to the input image it has sent before. For initial performance check of PP in real-life vehicle teleoperation, only front camera is used in experiments.

The core contributions of this paper are as follows:

- Perspective Projection approach to augment the delayed image (figure 4b) instead of using markers. It provides a better realistic perception to the human operator. The technique is detailed in section II-B-II.
- To allow 4G transmission of big depth-map data, a logarithmic encoding is used to utilize JPEG compression, which is explained in section II-A-III.
- Performance assessment through an experiment with five volunteers.

5G would decrease network latency, but latency would still be present in transmitting big data like images as compared to a simple ping. Apart from network latency, net latency consists of other factors such as camera exposure delay, data processing delay, and actuator delay (see section II-B-I). Availability of

5G would further enhance performance of proposed approach by reducing network latency.

The paper is organized as follows. In section II, the modified teleoperation control loop employing the Smith predictor is described. Then, the deployed architecture is presented with an explanation of its sub-elements. The sub-elements consist of the vehicle, sensing architecture, plant model for the Smith predictor, and PP. Section III presents the experiment procedure and shows the results of the experiment performing real-life vehicle teleoperation on street edge cases. Section IV carries a discussion on the respective results and discusses the benefits observed by employing PP. Section V concludes this work.

II. Method

The system model adopted for this vehicle teleoperation closely resembles Smith Predictor model [28] as shown in figure 5. In Smith predictor setup, the control input \((L_2)\) is passed through a predictor model \(P'\) of the vehicle, which then passes through a transfer function given by eq. 1.

\[
TF = 1 - e^{-\tau S}
\]  

where \(\tau\) is the time delay between output of the controller and the respective feedback it receives from the plant in the control loop. Historically designed for classical control, the purpose of the Smith Predictor is to bypass the time delay in the observation, and transform the system into a pure forward delayed system [17]. This is helpful for human-in-the-loop teleoperation, as it allows the human operator not to wait for feedback and provides the sense of controlling the vehicle in real-time.

In vehicle teleoperation, link \(L_5\) in figure 5a corresponds to delayed vehicle states i.e., vehicle pose in the surrounding environment. In particular, link \(L_5\) is the delayed image streaming transmitted by the vehicle and received by the control station. Link \(L_6\) contains the supplement contribution which can be added to link \(L_5\). This addition complements link \(L_5\) by the vehicle model action that took place in the delay time. In short, \(L_6\) is the prediction that could be added to link \(L_5\), to make it feel like undelayed feedback, link \(L_7\). To do so, time-delay and vehicle model should be known.
Link $L_5$ is discrete-time delayed image frame that has to be transformed to feel like an undelayed discrete time image frame, link $L_7$. Link $L_6$ is the forecasted position relative to the delayed position of the vehicle corresponding to the delays in the system.

It can be seen that when the vehicle model is accurate, i.e., $P' = P$, the system behaves like the one shown in figure 5b. Advantage of Smith predictor is that the delay is out of the control loop, which eliminates delay-induced instability and converts the system into just a delayed output system.

In this work, considering low vehicle teleoperation speeds, vehicle model considered is a single-track kinematic model which would be discussed later in section II-B-I-b.

As reported in figure 6, the system architecture for vehicle teleoperation is made of a control station that receives the image stream after downlink-delay ($\tau_1$), and subsequently the vehicle receives driving command after uplink-delay ($\tau_2$).

3) Data Compression: Since the images and depth maps are huge data to be sent through a 4G network, data compression is necessary. In fact, table II, shows the bandwidth required to communicate uncompressed image and depth map of resolution $376 \times 672$. The camera resolution is chosen as a trade-off between image quality, which is necessary to perceive the details of surrounding environment, and computational performance, which degrades as number of pixels increases.

Since the required bandwidth is much higher than the available one with a 4G network, data compression is necessary (maximum theoretical upload speed in 4G is 1MB/s). JPEG compression is then utilized for RGB image compression but the same can’t be used directly for depth-map which contains floating point data. To avail benefit of JPEG compression, depth-map elements have to be converted into ‘uint8’ datatype according to the eq. 2.

$$y = \frac{1}{A} \cdot \log \left\{ A (x - 1) + 0.01 \right\} + C$$

Here, $A = 0.0126194$ and $C = 364.92737$ are obtained to have a linear increase in depth resolution with depth (figure 9). Hence, less resolution is assigned to lower depths to assure more detail for closer objects. Above relation encodes all depths (x: 1-20 meters) in range (y: 0-255) of ‘uint8’ data.
type. In control station, inverse of eq. 2 is used to acquire depths in meters back again.

Exploiting this technique (eq. 2), JPEG compression can be utilized to further compress the depth map which now contains ‘uint8’ elements. Table III shows bandwidth required to communicate compressed image and depth map of resolution 376 × 672. Data compression makes it feasible to use a 4G network for wireless data communication.

### B. Control-Station

Control-station consists of a PC powered by Intel i7 processor, 24" display, Logitech G920 steering wheels, and pedals as shown in figure 2. Due to the constraint of live streaming, time available to process each image frame is \( \sim 33 \text{ms} \) (for 30fps). For faster processing, CPU parallel computing is utilized to perform the perspective transformation. Due to this processing time for each frame is \( \sim 15 \text{ms} \). The human-operator sees the transformed image and acts the steering wheel. Human operator can also see current speed, estimated path (based on current steering angle), and observed network latency for situational awareness.

Control-station tasks are predicting forecasted position corresponding to delay in the control loop, performing perspective projection to generate the new perspective, and capturing operator commands to transmit it back to the vehicle.

#### 1) Position Prediction:
Forecasted position is the position, where vehicle would receive driving commands back, respective to the input image frame it sent to the control-station (see figure 10). Control-station receives image frame, depth frame, vehicle speed, and acceleration. Forecasting time-window depends on time-delay present in data communication. Taking control-station as a reference point, downlink-delay (eq. 4) is the delay associated with image frame. Uplink-delay (eq. 5) is the delay associated with driving commands.

- **Downlink-delay:** It consists of image capture delay, depth-map computation delay, network downlink delay, and PP computation time. Image capture delay for ZED camera is \( \sim 70 \text{ms} \) [9]. At control-station, PP computation time is found to be \( 10 - 15 \text{ms} \). Rest of the delays are measured by timestamp subtraction. Vehicle and control-station PCs are synchronized with `chrony` application.

- **Uplink-delay estimation:** It consists of network uplink delay and vehicle steering control unit delay. Steering control unit delay is found to be small of \( \sim 20 \text{ms} \). Network uplink delay can’t be known in advance at control-station. But the control-station is aware of the previously observed network uplink delay by the vehicle. The control-station considers a high stochastic value (95\text{th} percentile) for uplink-delay estimation.

Making reference to figure 11, the uplink-delay measured during a vehicle teleoperation test is reported as an occurrence histogram. Analyzing the collected data of 9000 data-points of uplink-delay, it was found that Generalized extreme value (GEV) distribution fits accurately on uplink time-delay data. Also, Zheng et al. [23] found GEV distribution apt for representing distribution of time-delays in mobile communication. The probability distribution of generalized extreme value distribution is defined by eq. 3.

\[
f(t_2)_{\xi \neq 0, \mu, \sigma} = \frac{1}{\sigma} \left( 1 + \frac{\xi t_2 - \mu}{\sigma} \right)^{-\frac{1}{\xi} - 1} e^{-\left(1 + \frac{\xi t_2 - \mu}{\sigma}\right)^{-\frac{1}{\xi}}}  
\]  
(3)

For \( 1 + \xi \left( \frac{t_2 - \mu}{\sigma} \right) > 0 \), \( \xi \) the shape parameter, \( \mu \) the location parameter and \( \sigma > 0 \) the scale parameter. The shape parameter \( \xi \), is found positive. This means, the distribution has a lower bound \( \mu - \sigma \xi \) and heavy tail, based on the extreme value theory. Every second, control-station fits GEV distribution (using MATLAB `gevfit`) on latest uplink-delay data, computes its 95\text{th} and 99\text{th} percentile (using MATLAB `gevinv`), and send it to control-station.
Corde Lane et al. [31] found that a short variable lag could be more determinantal than a longer fixed one in a human-in-the-loop system. To eliminate the variability factor in the uplink-delay, hold and apply strategy is used. Consequently at the control station, for forecasting vehicle position, instead of considering most probable delay value of the uplink-delay distribution, 95\textsuperscript{th} percentile of distribution is considered. Considering higher percentile instead of most probable value for uplink delay, ensures prior reception of driving commands by vehicle compare to its desired actuation time (in 95\% instances). Besides containing the driving commands, the uplink msg consists of three time-info’s. First time-info corresponds to the control-station timestamp when the driving command was generated by the human operator. Second and third time-info’s consists of 95\% and 99\% percentile of delay distribution respectively. The vehicle is programmed to direct the commands to its actuators at the time ‘control-station timestamp + 95\% percentile of delay distribution’. Hold and apply strategy narrows the effective variability of uplink-delay. Purpose of computing 99\% percentile is to inform the vehicle to activate safety actions if the vehicle doesn’t receive any command even waiting till the time corresponds to 99\% percentile. In case of weak network conditions, the distribution fit may get widened. Which may result in a much higher value for 99\% percentile. A max limit of 200 ms is set for 99\% percentile. If vehicle doesn’t receive any command till max 200 ms from the time-stamp of last command, emergency stop manoeuvre would be activated by the vehicle. Figure 12 shows the trend of 95\% percentile and 99\% percentile of uplink-delay over a time window of 90s. Here the distribution fit is performed every second over the 50 uplink-delays observed in each second, as the control-station transmits command at 50 hz. It can be seen that variability of the 95\% percentile (used in hold and apply strategy) has been reduced to \sim10ms. Since the 99\% percentile is not directly involved in hold and apply strategy, its variability has less impact on the whole operation.

c) Trajectory integration to predict vehicle position: Considering low-speed vehicle teleoperation, single-track kinematic model is used to integrate vehicle trajectory. Gao et al. [32] found that at low lateral acceleration (<2m/s\textsuperscript{2}) and at non-small steering angles, kinematic vehicle model is comparable to non-linear coupled vehicle dynamics model. With trajectory integration, rear-axle center pose change is computed. Then, camera (fig 13) pose change is computed with the help of rear-axle center pose change.

Time-delay in wireless communication is variable. To make Smith predictor approach valid for variable time delays, instantaneous downlink delay value is considered instead of mean value of downlink delay. To account instantaneous value of downlink delay, a full integration of the vehicle predictor model from time \( t_1 - (\tau_1 + \tau_2) \) to \( t_1 \) is performed for every image frame received at the control station, see figure 10.

\[
\text{Downlink delay, } \tau_1 = t_1 - t_0 \quad (4)
\]
\[
\text{Uplink delay, } \tau_2 = t_2 - t_1 \quad (5)
\]
\[
\text{Total delay, } \tau = \tau_1 + \tau_2 \quad (6)
\]

At time instant \( t_1 \), when control-station receives an image frame, steering time-history of control-station (link \( L_2 \) in figure 5a) is known. Euler method, Runge-Kutta method or Exact trajectory integration method can be used for forward kinematic integration. Here, exact trajectory integration method is used with steering time-history of time-window equal to \( \tau \). Output of trajectory integration is the pose change of the camera relative to its input pose as shown in figure 13. Intermediate output of trajectory integration is the pose change of rear axle center with respect to its input pose (Input pose of rear axle always lies on the origin as given in eq 7). Image coordinate system is used to maintain relation between vehicle movement and image captured by camera. Considering initial conditions:

\[
X_0 = Z_0 = \Psi_0 = 0 \quad (7)
\]
\[
V_0 = \text{Speed (Received at time } t_1 - \tau \text{) } \quad (8)
\]
\[
a = \text{Acceleration (Received at time } t_1 - \tau \text{) } \quad (9)
\]

where, \( \tau \) is given by eq 6. To predict the curved path of the vehicle, the time history of the steering is needed.

Steering time-history is:

\[
\begin{bmatrix}
\delta_0 & d\tau_0 \\
\delta_1 & d\tau_1 \\
\vdots & \\
\delta_{n-1} & d\tau_{n-1}
\end{bmatrix}
\]

Here, first column contains the steer angles at front wheel and second column contains respective time intervals for which the steer is effective. Due to constant
sampling time of 0.020s, most of the time intervals are constant, except \( dt_0 \) and \( dt_{n-1} \). This is due to absence of synchronicity between steering angle sampling and image streaming frame rate.

Equation 10 calculates the velocity of rear axle center at each integration step.

\[
V_i = V_0 + a \sum_{n=0}^{i-1} dt_n
\]  
(10)

Given the steer, instantaneous radius of curvature \( (R) \) and angular velocity \( (\omega) \) is given by eq 11-12, where \( L \) is the vehicle wheelbase.

\[
R_i = \frac{L}{\tan \delta_i}
\]  
(11)
\[
\omega_i = \frac{V_i}{R_i}
\]  
(12)

Equation 13-15 gives the pose of rear axle center after \( i^{th} \) integrating step, according to reference frame shown in figure 13, where origin lies at the center of rear axle.

\[
\psi_{i+1} = \psi_i + \omega_i \cdot dt_i
\]  
(13)
\[
X_{i+1} = \begin{cases} 
X_i + (V_i \cdot dt_i) \cdot \sin \psi_i & \text{if } \omega = 0 \\
X_i - R(\cos \psi_{i+1} - \cos \psi_i) & \text{otherwise}
\end{cases}
\]  
(14)
\[
Z_{i+1} = \begin{cases} 
Z_i + (V_i \cdot dt_i) \cdot \cos \psi_i & \text{if } \omega = 0 \\
Z_i + R(\sin \psi_{i+1} - \sin \psi_i) & \text{otherwise}
\end{cases}
\]  
(15)

After performing integration over the steering time-history window, predicted pose \((X_n, Z_n, \psi_n)\) of rear-axe center is obtained. Camera pose change \((\Delta Z_{cam}, \Delta X_{cam}, \Delta \psi_{cam})\), i.e., relative pose of predicted camera position with respect to input camera position (figure 13) is obtained through eq. 16-18.

\[
\Delta Z_{cam} = Z_n - C \cdot (1 - \cos \psi_n)
\]  
(16)
\[
\Delta X_{cam} = X_n + C \cdot \sin \psi_n
\]  
(17)
\[
\Delta \psi_{cam} = \psi_n
\]  
(18)

Here, \( C \) is horizontal distance between camera and the rear axle center. Camera pose change is required because the link \( L_5 \) in figure (15a) is the image captured with the camera mounted on the vehicle. Camera pose change is the link \( L_6 \) in figure (15a).

2) Perspective Projection: Perspective projection refers to image transformation to obtain the new camera perspective after the camera has traversed. It corresponds to the 3D projection of the world on the image plane of the camera at the predicted position and its topology perceived by the camera at its predicted position as depicted in figure 13. Unlike typical zooming, it resizes each object ahead considering its distance from the camera. It also captures the effect of vehicle yaw motion on the new image formed, as shown in figure 13. Through this technique, human-operator sees a forecasted video stream that tries to replicate future perspective of vehicle FOV. It forecasts by accounting measured time-delay, vehicle speed, and steering time-history.

Object’s new perspective depends upon its relative coordinates from the camera. Nearer objects expand more than farther objects. While making a turn, some objects disappear and new objects appear. Object disappearance can be simulated well but appearance of a new object can not be simulated. Inpainting technique is used to fill those pixels which correspond to introduction of new objects. Algorithm proposed by Telea [33] is used for inpainting, which considers the color information of available neighboring pixels. To compute relative position of objects, depth-map is used as input for this transformation. Depth-map is an image channel that contains information relating to the distance of the surfaces of scene objects from the viewpoint. The output of this projection is the link \( L_7 \) in figure (15a). Process flow chart of perspective projection implementation is shown in figure (14). Errors corresponding to each block are discussed in Appendix A, where the prime contributor is the Point-cloud transformation block.

a) Inputs: Image, depth map, camera FOV angles (horizontal and vertical both), camera displacements \((\Delta Z_{cam}, \Delta X_{cam}, \Delta \psi_{cam})\) are the inputs to the perspective transformation algorithm. Input image is considered perfectly rectified from lens distortion. Depth-map has the same resolution as of the input image and every element in depth map corresponds exactly to the respective pixel in input RGB image. Ideally, point-cloud shall be used as an input to Perspective projection, but as point-cloud is high bandwidth consuming data, depth-map is used instead of point-cloud.

b) Conversion of Depth-map into Point-cloud: After receiving depth-map of old perspective, control-station converts it into point-cloud. Two approaches can be used for this conversion. First, using camera focal lengths in \( x \) and \( y \). Second, using the horizontal and vertical FOV of the depth-map image. Since, image and depth map are
free of lens distortion, FOV approach is used. Equations (19-20) are used to convert depth-map into point-cloud. $W$ and $H$ are width and height of the image in pixels. Say, a point in depth-map is at $x_d$-col and $y_d$-row (ranges are 1 to $W$ and 1 to $H$ respectively) has a depth value of $\hat{z}_p$ meters. First, shift the depth-map origin from top left corner to the center of the depth-map. This is done since image reference frame origin is in the top-left corner while the optical axis of the camera view passes through the middle of the image plane.

$$\hat{x}_d = x_d - (W/2 + 0.5)$$
$$\hat{y}_d = y_d - (H/2 + 0.5)$$

(19)

Equation 20 computes the Cartesian coordinates ($\hat{x}_p$, $\hat{y}_p$) of the pixel in meters.

$$\hat{x}_p = \hat{z}_p \left[ \frac{\tan(fo_H)}{W/2} \right]$$
$$\hat{y}_p = \hat{z}_p \left[ \frac{\tan(fo_V)}{H/2} \right]$$

(20)

$fo_H$ and $fo_V$ are the horizontal and vertical FOV of the depth-map image. Needless to say, that $z$ coordinate of the point is already known, as it is given in that particular depth map point. ($\hat{x}_p$, $\hat{y}_p$, $\hat{z}_p$) is the point in point-cloud corresponds to $(x_d, y_d)$ pixel in depth-map.

c) Point-cloud transformation: The above point-cloud has origin on the input position of the vehicle (refer figure 13, cam position input). It has to be transformed into the new reference frame corresponding to predicted position. Thus, a rotation and translation to transform the point-cloud from cam position input to cam position predicted (refer figure 13) are required.

Rotation is performed according to the following rotation matrix (eq 21),

$$R = \begin{bmatrix} \cos(\Delta \Psi_{cam}) & 0 & \sin(\Delta \Psi_{cam}) \\ 0 & 1 & 0 \\ -\sin(\Delta \Psi_{cam}) & 0 & \cos(\Delta \Psi_{cam}) \end{bmatrix}$$

(21)

Above matrix is based on coordinate system indicated in Figure 13. It means that a rotation is positive while turning right.

The translation vector, $D = \begin{bmatrix} \Delta X_{cam} \\ 0 \\ \Delta Z_{cam} \end{bmatrix}$

(22)

Previous two operations can be combined in a homogeneous transformation matrix (eq 23),

$$T = \begin{bmatrix} R^T & -R^T \cdot D \\ 0 & 1 \end{bmatrix}$$

(23)

In our case, camera is mounted pointing slightly downward, for better utilization of vertical FOV as shown in figure 15. The transformation matrix $T$ has then to be modified as per eq 24 and eq 25, accounting for the additional rotation about x-axis.

The rotation matrix, $R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$

(24)

New transformation matrix,

$$T_2 = \begin{bmatrix} R_2 & 0 \\ 0 & 1 \end{bmatrix} \cdot T_1 \cdot \begin{bmatrix} R_2^T & 0 \\ 0 & 1 \end{bmatrix}$$

(25)

New point coordinates ($\hat{x}_{pNeu}, \hat{y}_{pNeu}, \hat{z}_{pNeu}$) corresponding to the old coordinates ($\hat{x}_p$, $\hat{y}_p$, $\hat{z}_p$) are given by eq 26:

$$\begin{bmatrix} \hat{x}_{pNeu} \\ \hat{y}_{pNeu} \\ \hat{z}_{pNeu} \end{bmatrix} = T_2 \cdot \begin{bmatrix} \hat{x}_p \\ \hat{y}_p \\ \hat{z}_p \end{bmatrix}$$

(26)

New point coordinates ($\hat{x}_{pNeu}, \hat{y}_{pNeu}, \hat{z}_{pNeu}$) are in reference frame of camera of predicted position of the vehicle (refer figure 13).

d) Remapping of point-cloud to depth-map: Once the new Cartesian coordinates ($\hat{x}_{pNeu}, \hat{y}_{pNeu}, \hat{z}_{pNeu}$) of the pixel is computed, it is again converted back to depth-map, to get a new depth-map with eqs 27-28.

$$\hat{x}_{dNeu} = \hat{x}_{pNeu} \left( \frac{W}{2} \tan(\frac{fo_H}{\hat{z}_{pNeu}}) \right)$$
$$\hat{y}_{dNeu} = \hat{y}_{pNeu} \left( \frac{H}{2} \tan(\frac{fo_V}{\hat{z}_{pNeu}}) \right)$$

(27)

(28)

Above obtained ($\hat{x}_{dNeu}, \hat{y}_{dNeu}$) coordinates are new pixel location corresponds to the old pixel location ($\hat{x}_d, \hat{y}_d$). Right now it is centered with image center. Re-centering back to the top-left corner is performed as given by eq 29.

$$x_{dNeu} = \hat{x}_{dNeu} + (W/2 + 0.5)$$
$$y_{dNeu} = \hat{y}_{dNeu} + (H/2 + 0.5)$$

(29)

This is necessary because, in image pixel coordinate system, origin is at the top-left corner. Although in this example, only one pixel is discussed, the same has to be done for all the pixels. The output of this step is a map that contains information about the new location ($x_{dNeu}, y_{dNeu}$) of the old pixel ($x_d, y_d$) in the image.

e) Pixel scaling: Pixel scaling performs re-scaling of the objects in image according to their new distance from...
the camera, i.e., nearer objects would scale up more compared to farther objects. This scaling is performed at the pixel level. Scale factor for each pixel depends on old and new depth of that pixel. Scale factor for pixel \((x_{dNew}, y_{dNew})\) is given by eq 30.

\[
S = \frac{\hat{z}_p}{\hat{z}_{pNew}}
\]  

(30)

As scale factor is different for each pixel, span (or spread) of each pixel in new FOV would be different. Span of old pixel \((x_d, y_d)\) in new image over the rows is given by eq 31.

\[
i_{Rows} = \left[ \text{floor} \left( y_{dNew} - \frac{S - 1}{2} \right) : \text{ceil} \left( y_{dNew} + \frac{S - 1}{2} \right) \right]
\]

(31)

Span of old pixel \((x_d, y_d)\) in new image over the columns is given by eq 32.

\[
i_{Cols} = \left[ \text{floor} \left( x_{dNew} - \frac{S - 1}{2} \right) : \text{ceil} \left( x_{dNew} + \frac{S - 1}{2} \right) \right]
\]

(32)

f) Mapping RGB values for each pixel: Being known the old location and new location of each pixel, RGB information is carried from old image to the new image. Pseudo code for carrying RGB information for one old pixel is given by eq 33.

\[
\text{newImage}(i_{Rows}, i_{Cols}, 1:3) = \text{oldImage}(y_d, x_d, 1:3)
\]

(33)

Here \((x_d, y_d)\) are the inputs used in RHS in eq 19. Mapping of RGB pixels is responsible to place objects in previous FOV to new FOV, i.e., to place RGB pixels \((x_d, y_d)\) from figure 16a to \((x_{dNew}, y_{dNew})\) in figure 16b. To get the full new image, new locations of all the old pixels are calculated. Obviously, the pixels that fall out of frame need to be discarded. Important thing to keep in mind during this iteration is the order in which pixels need to be processed. As can be imagined, when camera moves forward, nearer objects scale up and try to blanket farther objects. Therefore, pixels that correspond to farther depths need to be processed before pixels that correspond to closer depths.

g) Performance check: To verify the performance of Perspective projection algorithm, an input RGB image (figure 16a) and corresponding depth-map is passed into it. Predicted position of vehicle is considered at \(\Delta X = 0.5m, \Delta Z = 2.1m, \Delta \Psi = 15^\circ\).

Figure 16b, shows output obtained after perspective projection of input image figure 16a. Output image can be compared with ground truth figure 16c. Ground truth corresponds to the perspective of the vehicle when the vehicle reached the predicted position. It can be noticed that sizes of farther objects didn’t change significantly, but sizes of closer objects changed significantly (e.g. the size off-road vehicle in image has scaled up significantly). Objects in the left part of the image are automatically discarded in the frame because they no longer belong to the new FOV. Right part of the image (figure 16b) is null because new objects can not be predicted by this approach (black pixels in figure 16b). Obtained output can be compared with effective view shown in figure 16c after vehicle motion. In general good matching is observed.

C. Network Communication

To teleoperate the vehicle, a LAN is generated connecting the vehicle with the control station. This is made thanks to Robot Operating System (ROS) network which is set up with university VPN over 4G communication using “Netgear MR1100” modem. The modem is LTE Category 16 which is compliant with 3GPP Release 12. So, it could avail LTE Advanced technology also, which is a major enhancement of the LTE standard.

D. GPS

To track vehicle trajectory, GPS AgGPS-332 receiver is mounted on vehicle. GPS measurements are RTK-corrected for 0.01m accuracy. To compare the trajectories traversed in the following section GPS tracked Geo-coordinates are used.

III. RESULTS

To assess the usefulness of the perspective projection technique, some on-street edge case maneuvers are performed in a controlled environment. Maneuvers considered for this assessment are 7-meter radius 80° left-angled turn (R7-80°), 5-meter radius 120° left-angled turn (R5-120°), and double lane change (as shown in figure 17). The test track is an asphalt rolled road with a center-line marked throughout with reflective wide tape.

In this experiment, five volunteers are invited to teleoperate the vehicle. Experiments are performed in 3 different modes: a) Driver in vehicle: Each volunteer drives one lap of the track at a constant speed of 10km/h, steering from inside of the vehicle.
b) Teleoperation with PP: In control-station, after performing 20 minutes of drive training on above maneuvers with PP, the volunteer performs three teleoperation laps at a constant speed of 10 km/h.

c) Teleoperation without PP: In control-station, after performing 20 minutes of drive training on above maneuvers without PP algorithm, the volunteer performs three teleoperation laps at constant speed of 10 km/h.

Short training time (of only 20 minutes) was considered intentionally to assess the performance benefit of PP in spontaneous scenarios. Considering longer training times may affect the assessment due to the fact that the human mind tries to correct, learn and memorize repetitive control actions.

To evaluate the performance benefits, trajectories are monitored with the help of two GPS antennas mounted at the front and rear of the vehicle rooftop. With the help of two GPS antennas vehicle heading also can be precisely tracked. While driving at low speeds in this experiment, general tendency of driver is to coincide forward-most point of the car with the center line of road. Forward-most point of the car is found at 0.8 m ahead of the front axle. Geographic coordinate of this forward-most point is estimated through linear extrapolation of the GPS coordinates from the two GPS antenna mounted on the vehicle rooftop. RMS deviation - To assess performance improvement, deviation is measured between this foremost point of the vehicle and closest point on center-line of the test track. Tests are performed at only one speed of 10 km/h for all maneuvers. Constant speed is maintained by a cruise control system installed inside the vehicle. Whereas, steering commands are generated by human inputs, transmitted from the control-station. During all teleoperation laps, the safety driver didn’t intervene.

RMS deviation is computed as given by eq 34.

$$RMSE, \varepsilon = \sqrt{\frac{\sum_{i=1}^{n} \text{deviation}^2}{n_{GPS}}}$$

where,

- deviation - is the minimum distance between forward-most point and the track
- $n_{GPS}$ - is the number of GPS readings observed in the specific maneuver

Figure 18 shows the input, output image of the Perspective projection node, and ground truth (a snap from Video1 of supplemental files). The ground truth is captured after 330ms of the input image. Here, the 330ms is approximating the round-trip delay considered by the PP node for the frame. Each frame of Video1 validates the performance, by comparing PP output with ground truth. Where the ground truth is the real image captured after the delay considered in PP. To analyze the role of this projection technique, some interesting areas are marked in the comparison. While performing the left-turning maneuver, objects start moving right in the FOV. Some objects on the right start disappearing from the FOV. E.g. Obj 1 marked is shifted towards right-bottom, and Obj 2 has completely disappeared from the FOV. This accurate transformation of the FOV, gives the human operator a feel of delay-free driving. Introduction of new objects on the left side of FOV can not be predicted (in case of just one camera facing forward), inpainting is used to fill the null pixels.

Figure 19a shows the actual track, the first two teleoperation laps with PP, the first two teleoperation laps without PP, and driver in vehicle lap for region-A. Alongside the trajectory traversed, x-axis is also mapped with experienced time delay. This figure is stretched in y-axis to emphasize the cornering region. Deviation (RMS 0.2 m) is minimum in the case when the driver is inside the vehicle, reason is clear that there is no network delay between the driver and the vehicle. In case of teleoperation without PP, deviation RMSE is $\sim 0.53$m. Performance improvement is observed in case of teleoperation with PP, where the deviation RMS is $\sim 0.4$m in PP laps.
It is apparent in the figure that vehicle starts turning earlier (at about $X = -12m$) in case of PP lap compared to the teleoperation lap without PP. This is because projection tries to generate a perspective that the vehicle would observe after traversing a distance, which corresponds to time delay in the network communication. Furthermore, observed latency is consistently higher in PP laps. This is because projection algorithm requires RGB image as well as the depth-map. This means more computation and a higher bandwidth requirement, and hence a longer delay in data transmission. The fact that performance is improved despite a slight increase in latency (by $\sim 30ms$) is intriguing.

Figure 19b shows similar comparison but for region-B. Here, time delay is mapped with Y-axis. The figure is stretched in x-axis to emphasize the cornering region. This is a high curvature region of the test-track. In first teleoperation without PP (purple) lap, the vehicle starts turning after traversing a distance of around 0.5m beyond the corner. To compensate this deviation, human operator steers more, due to which an oscillation is observed after the corner. This oscillation is induced due to the time-delay in the control loop between human and vehicle. In the second teleoperation without PP (green) lap, the human operator tried to steer rapidly to be close to the track. But again a small oscillation is observed due to time-delay in the system. In both the laps of teleoperation with PP (red and green), deviations are smaller than they were in laps without PP. And, the trajectories are oscillation free, which is the advantage of Smith predictor approach.

Figure 19c shows, trajectories traversed for the double lane change maneuver, region-C. Here, time delay is mapped with y-axis. This figure is also stretched in x-axis to emphasize the cornering regions. In the first teleoperation lap with PP (red), the deviation is quite low. In the second teleoperation lap with PP (green), the deviations are a little higher than the previous lap. This is because the latency in the network suddenly increased in that region, as is apparent in the latency trend of yellow plot (marked with yellow circles between Y-axis $[-5 \text{ to } 5m]$). Even in increased latency window, deviation is not degraded compared to laps without PP.

Figure 20, summarizes the whole experiment by presenting the CG deviation of teleoperation laps with respect to the CG trajectory of the driver in vehicle lap.

IV. DISCUSSION

As seen in the results section, trajectories traversed with teleoperation without perspective projection outcomes a significant deviation and oscillations in narrow turns. This deviation is due to time-delay present in the information flow. Even for a low vehicle speed of 10 km/h, a deviation of 0.5 meter is found for simple left-angled turn. This deviation may lead to

---

Fig. 19. Trajectories recorded in manoeuvres.

Fig. 20. RMS deviation respective to the Driver in vehicle lap.
unwanted incidents on street. Perspective projection technique doesn’t completely eliminate the deviations, but reduces the deviations and eliminates oscillations in the trajectory. Even the time-delay in the loop is more in case of perspective projection, trajectories were found to be stable and close to the reference as compared to that without the perspective projection. In all regions A-C, teleoperation with PP resulted in less deviations for all five drivers (figure 20). This indicates teleoperation with PP tries to bring human operator experience close to driver in vehicle case.

**Limitations:** As the vehicle speed increases (> 10 km/h), predictive image becomes blurry because of noise in depth-map. Also, it does not take into account the independent motion of objects in the frame. It considers objects to be still, and just the vehicle in motion. For small motion of objects and for small time-delay, this assumption is reasonable.

**Advantages:** Performing teleoperation with perspective projection resulted in a significant reduction in deviations while maneuvering for corner R7-80° and double lane change. In high curvature maneuver of corner R5-120° at 10 km/h, oscillations are eliminated as compared with teleoperation laps without perspective projection.

V. Conclusion

To mitigate the detrimental effect of time-delay in vehicle teleoperation, the perspective projection technique is used to transform the image streaming. Smith predictor approach is used to estimate the correction which can be added to the inputs (images) shown to the human operator. The plant model inside the Smith predictor is a single-track kinematic vehicle model, which is a reasonable simplification for low-speed vehicle teleoperation. Perspective projection technique merges the correction given by Smith predictor to the streaming images. It is able to generate the transformed image not only in straight driving conditions but also during cornering conditions. One of its inputs is the depth-map. In order to transmit depth-map to the control-station, it is first transformed into an 8-bit image using a mathematical relation that considers linear increasing resolution for depth measurements. Both RGB and depth images are transmitted to the control-station in JPEG. For estimation of uplink-delay, stochastic approach is used to estimate 95th percentile of the latest trend of uplink-delays in the network. The human-operator sees a perspective-projected image of the vehicle surrounding. This display tries to emulate vehicle’s perspective when the vehicle would receive driving commands. Due to this, human-operator is able to make driving decision bit in advance. Which results, in better control in following desired trajectory.

For approach validation, vehicle teleoperation is performed on some edge-case scenarios of street driving. Five volunteers performed 3 laps with PP and 3 laps without PP at 10 km/h. With PP, time-delay induced oscillations are found to be eliminated and laps are found to be closer to the Driver in vehicle laps, which improves the confidence of human operator.

Future work: Multi cameras and dynamic vehicle model (for position prediction) would also require a vehicle state estimator to transmit vehicle states in addition to the images.

APPENDIX A

ERRORS ASSOCIATED WITH PROCESS FLOW OF PP

1) Depth-Map to Point-Cloud Conversion: At control-station, depth-map decoding pertains to error. Depth decoding error is proportional to the depth itself as per figure 9b. E.g., the error for a depth of 1m is 1cm.

By comparing figures 21a and 21b, it has been found that buildings have moved a little closer. This is because of (uint8) integer overflow in encoding depths that are more than 20m. But accurate perspective is obtained for closer objects such as vehicles. During vehicle teleoperation, closer objects are of more interest than farther objects, this trade-off is acceptable to save network bandwidth.

2) Point-Cloud Transformation: Here, the factor of error is vehicle model inefficiency in integrating the trajectory for the delay time (~250ms). The kinematic vehicle model is predicting the vehicle pose with respect to the delayed pose accounting for delay and vehicle speed. Figure 22 presents the predicted pose (X, Y and Yaw) and corresponding
errors during TeleOp with PP-Lap 1 (animation in Video2 of supplemental files). The error is the difference between predicted pose and actual pose realized by the vehicle after the considered delay. Table IV presents the RMS error in lateral direction for different time sections. The percentage error is more during lane change maneuver, which caused due to unaccounted lateral slip by the kinematic vehicle model. The vehicle model is predicting more lateral movement compared to actual lateral movement in presence of under-steering behavior. Also, the relatively more percent error corresponds to the section (38-45s) where the steer changes its sign to be in the second lane. A dynamic single-track model can be used to reduce this error.

3) Point-Cloud to Depth-Map Conversion & Pixel Scaling: These steps are responsible to place objects back on the 2D-image plane of the predicted cam-position. Due to the fact that pixels are discrete in nature, error here is ±0.5 pixel.

4) RGB Mapping: This step is copying the color information of the objects; it doesn’t add additional error in the process flow.

| TABLE IV | LATERAL ERROR OBSERVED IN RESPECTIVE MANEUVER |
|----------|-----------------------------------------------|
| Time section | RMSE Y |
| 6-15s | 18cm |
| 19-24s | 14cm |
| 35-38s | 12cm |
| 38-45s | 16cm |

REFERENCES
[1] G. Graf, H. Palleis, and H. Hussmann, “A design space for advanced visual interfaces for teleoperated autonomous vehicles,” in Proc. Int. Conf. Adv. Vis. Interfaces. New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 1–3.
[2] S. Gnatzig, F. Chachułowski, T. Tang, and M. Lienkamp, “A system design for teleoperated road vehicles,” in Proc. 10th Int. Conf. Informat. Control, Autom. Robot., 2013, pp. 231–238.
[3] B. Watson, N. Walker, W. Ribarsky, and V. Spaulding, “Effects of variation in system responsiveness on user performance in virtual environments,” Hum. Factors, J. Hum. Factors Ergonom. Soc., vol. 40, no. 3, pp. 414–418, Sep. 1998.
[4] D. W. Cunningham, A. Chatziastros, M. von der Heyde, and H. H. Balthoff, “Driving in the future: Temporal visuomotor adaptation and generalization,” J. Vis., vol. 1, no. 2, p. 3, Nov. 2001.
[5] T. B. Sheridan, “Space teleoperation through time delay: Review and prognosis,” IEEE Trans. Robot. Autom., vol. 9, no. 5, pp. 592–606, Nov. 1993.
[6] J. Davis, C. Smyth, and K. McDowell, “The effects of time lag on driving performance and a possible mitigation,” IEEE Trans. Robot., vol. 26, no. 3, pp. 590–593, Jun. 2010.
[7] S. R. Ellis, K. Mania, B. D. Adelstein, and M. I. Hill, “Generalizability of latency detection in a variety of virtual environments,” in Proc. Human Factors Ergonom. Soc. Annu. Meeting, 2004, pp. 2632–2636.
[8] L. H. Frank, J. G. Casali, and W. W. Wierwille, “Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator,” Hum. Factors, vol. 30, no. 2, pp. 201–217, Apr. 1988.
[9] Stereo Labs. What is the Latency of the ZED camera? Accessed: Jan. 28, 2023. [Online]. Available: https://support.stereolabs.com/hc/en-us/articles/206918319-What-is-the-latency-of-the-ZED-camera
[10] J. Lane, C. Carignan, and D. Akin, “Time delay and communication bandwidth limitation on tethered control,” in Proc. SPIE, vol. 4195, pp. 405–419, Mar. 2001.
[11] Y.-D. Zheng, “Adaptive control for time-delay systems adopting Smith predictor models,” Kongzhi Lianhu Yu Yangyong/Control Theory Appl., vol. 38, no. 3, pp. 416–424, 2021.
[12] D.-H. Lee, J.-H. Jung, H.-N. Yoon, Y.-S. Park, and J.-M. Lee, “Simulation of time delay compensation controller for a mobile robot using the SMC and Smith predictor,” in Proc. Int. Conf. Intell. Autom. Syst. (Advances in Intelligent Systems and Computing), vol. 531, 2017, pp. 687–694.
[13] A. Kuzu, E. A. Baran, S. Bogosyan, M. Gokasan, and A. Sabanovic, “Improved bilateral teleoperation with proactive haptic sensing and transmission,” Proc. Inst. Mech. Eng., J. Syst. Control Eng., vol. 232, no. 1, pp. 79–91, Jan. 2018.
[14] T. E. Kim and J.-M. Lee, “Predictive control of time-delayed teleoperation in a non-visible environment,” in Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc., Oct. 2017, pp. 7221–7226.
[15] D.-H. Lee, J.-H. Jung, and J. Lee, “Robust teleoperation in a non-visible environment with a new prediction scheme,” J. Mech. Sci. Technol., vol. 32, no. 2, pp. 835–843, Feb. 2018.
[16] A. P. Batista and F. G. Jota, “Performance improvement of an NCS closed over the internet with an adaptive Smith predictor,” Control Eng. Pract., vol. 71, pp. 34–43, Feb. 2018.
[17] T. Teng and P. R. Grant, “Adaptive Smith predictor for teleoperation of UAVs using parameter estimation techniques,” in Proc. AIAA Scitech Forum, 2019, p. 1077.
[18] R. Liu, D. Kwak, S. Devarakonda, K. Beksris, and L. Ilkodic, “Investigating remote driving over the LTE network,” in Proc. 9th Int. Conf. Automat. User Interfaces Interact. Veh. Appl., Sep. 2017, pp. 264–269.
[19] A. Tandon, M. J. Brudnak, J. L. Stein, and T. Ersal, “An observer based framework to improve fidelity in internet-distributed hardware-in-the-loop simulations,” in Proc. ASME Dyn. Syst. Control Conf., vol. 2. Palo Alto, CA, USA: ASME, Oct. 2013, Art. no. V002T21A004, doi: 10.1115/DSC2013-3878.
[20] X. Ge, Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, “Performance analysis of a model-free predictor for delay compensation in networked systems,” IFAC-PapersOnLine, vol. 48, no. 12, pp. 434–439, 2015.
[21] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, “A delay compensation framework for predicting heading in teleoperated ground vehicles,” IEEE/ASME Trans. Mechatronics, vol. 24, no. 5, pp. 2365–2376, Oct. 2019.
[22] M. Huba and K. Zakova, “Experimenting with constrained dead time compensators for FOTD systems,” in Proc. 4th Exp. Int. Conf., 2017, pp. 269–274.
[23] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, “Evaluation of a predictor-based framework in high-speed teleoperated military UGVs,” IEEE Trans. Hum.-Mach. Syst., vol. 50, no. 6, pp. 561–572, Dec. 2020.
[24] M. Fennel, A. Zea, and U. D. Hanebeck, “Haptic-guided path generation for remote car-like vehicles,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 4087–4094, Apr. 2021.
[25] T. Chen, “Methods for improving the control of teleoperated vehicles,” Ph.D. dissertation, Dept. Mech. Eng., Tech. Univ. Munich, Munich, Germany, Apr. 2015.
[26] J. Prakash, M. Vignati, S. Arrigoni, M. Bersani, and S. Mentasti, “Teleoperated vehicle-perspective predictive display accounting for network time delays,” in Proc. ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf., 21st Int. Conf. Adv. Vehicle Technol., 16th Int. Conf. Design Educ., vol. 3. Anaheim, CA, USA: ASME, Aug. 2019, Art. no. V003T01A022, doi: 10.1115/DETC2019-98159.
[27] D. Hearn and P. M. Baker, Computer Graphics: C Version. Upper Saddle River, NJ, USA: Prentice-Hall, 1997, ch. 9.
[28] O. J. Smith, “Closer control of loops with dead time,” Chem. Eng. Prog., vol. 53, no. 5, pp. 217–219, May 1957.
[29] M. Vignati, D. Tarisitano, M. Bersani, and F. Cheli, “Autonomous steer actuation for an urban quadricycle,” in Proc. Int. Conf. Electr. Electron. Technol. Automot., Jul. 2018, pp. 1–5.
[30] M. Vignati, D. Tarisitano, and F. Cheli, “On how to transform a commercial electric quadricycle into a full autonomously actuated vehicle,” in Proc. Int. Symp. Adv. Vehicle Control, Beijing, China, 2018, pp. 1–7.
[31] J. C. Lane, C. R. Carignan, B. R. Sullivan, D. L. Akin, T. Hunt, and R. Cohen, “Effects of time delay on telerobotic control of neutral buoyancy vehicles,” in Proc. IEEE Int. Conf. Robot. Automat., May 2002, pp. 2874–2879.
[32] F. Qiu, Q. Hu, J. Ma, and X. Han, “A simplified vehicle dynamics model for motion planner designed by nonlinear model predictive control,” Appl. Sci., vol. 11, no. 21, p. 9887, Oct. 2021.
Jai Prakash received the M.S. degree in mechanical engineering from Politecnico di Milano, Italy, in 2018, where he is currently pursuing the Ph.D. degree with the Mechanical Engineering Department. His research interests include computer-vision, robotics, vehicle dynamics, and vehicle teleoperation.

Michele Vignati (Member, IEEE) received the M.S. and Ph.D. degrees in mechanical engineering from Politecnico di Milano, in 2013 and 2017, respectively, with thesis on control strategies for distributed powertrain of hybrid and electric vehicles. Since 2019, he has been an Assistant Professor (RTDA) in the research field of applied mechanics. In particular, he focuses on mechanical systems, dynamics, and control applied in the automotive field. He worked on tire dynamics and modeling in cooperation with Pirelli. He works in autonomous driving field and vehicle modeling, testing, state estimation, and active control systems. He has more than 20 publications in international journals and conferences. In 2018, he has won the Best Paper Award for a paper presented at the AVEC–18 International Conference.

Daniele Vignarca received the M.S. degree in mechanical engineering from Politecnico di Milano, Italy, in 2020, where he is currently pursuing the Ph.D. degree with the Mechanical Engineering Department. His research activity is focused on connected and autonomous vehicles with a focus on V2I communication.

Edoardo Sabbioni received the Ph.D. degree in mechanical systems engineering in 2007. He is currently an Associate Professor with the Department of Mechanical Engineering, Politecnico di Milano. He is the author of more than 140 scientific publications, most of which published on peer-reviewed journals or presented at international conferences. His main research activities are in the field of stability, dynamics, and control of mechanical systems, with applications to road and rail vehicles. He is also involved in research activities within the fields of aerodynamics and mechatronics. In particular, he carried out numerical-experimental researches concerned with active safety of vehicles, design of control systems to improve vehicle handling and ride comfort, design of hardware-in-the-loop (HiL) test benches, electric and hybrid electric vehicles, autonomous and connected vehicles, ‘smart’ tires technology, and cross wind on road/rail vehicles. During his research activity, he cooperated and still cooperates with the main Italian automotive/railway industries and he was involved in several Italian and EC funded projects. He is a member of the Editorial Board of Shock and Vibration.

Federico Cheli (Member, IEEE) received the M.S. degree in mechanical engineering from Politecnico di Milano, Milano, Italy, in 1981. He is currently a Full Professor with the Department of Mechanical Engineering, Politecnico di Milano. He is the author of more than 380 publications on international journals and conferences. His scientific activity concerns research on vehicle performance, handling and comfort problems, active control, ADAS, and electric and autonomous vehicles. He is a member of the Editorial Board of the International Journal of Vehicle Performance and International Journal of Vehicle Systems Modeling and Testing.