Synchronization and Feedback Loop Integration of a Non-real Time Microscopic Traffic Simulation with a Real-time Driving Simulator using Model-Based Prediction

Satya Prasad Maddipatla and Sean Brennan

Abstract— **Driving simulators are a common tool to study human responses to driver assistance algorithms interacting with vehicle behavior. The accuracy of human responses, particularly to other vehicles, depends on the fidelity of representing the traffic surrounding the ego-vehicle. The traffic surrounding the ego-vehicle must also react to the ego-vehicle and be updated at a high enough rate such that the driver can perceive continuity. Microscopic traffic simulation tools are well developed to simulate traffic behavior replicating the real world over large networks. But the speed of microscopic simulations depends on the size of the network, traffic volume, simulation's computational hardware, and the traffic simulation software itself. The frequency of updating the traffic in many traffic simulators is not frequent enough for direct use in a driving simulator, thus causing discontinuous jumps in vehicle motion perceived by drivers. However, these traffic simulation results can be made continuous in perception by performing trajectory-level smoothing and time up-sampling as a real-time process occurring parallel to the driving simulator's rendering software. This ensures a realistic human perception of traffic behavior around the ego-vehicle. However, this integration creates the problem of synchronizing the traffic simulation dynamics with the ego-vehicle's dynamics within a driving simulator. This is challenging because traffic simulations are not typically programmed to be real-time, and thus the time offsets of the traffic simulator in relation to real-time can be time-varying. This paper describes a model-predictive feedback control method to synchronize a traffic simulator with a driving simulator in real-time by projecting the ego-vehicle into the future while adjusting time offsets via a feedback loop taking the traffic simulator's speed and communication delays into account. The update rate of the traffic is enhanced using a uniform acceleration observer that utilizes a localized simulation of the traffic network, and thus three dynamic systems are integrated at once: a large-scale network traffic simulation, a local real-time reduced-order model of the traffic, and the behavior of an ego-vehicle. This method of integrating traffic simulation with a driving simulator is demonstrated for a high-speed vehicle motion in a highway driving simulator, and the error analysis shows the success of this approach.**

I. INTRODUCTION

Driving is a complicated and dangerous activity requiring training and licensing. Automotive companies have been developing vehicular systems like Advanced Driver Assistance Systems (ADAS) and Intelligent Transportation Systems (ITS) to assist the driver, reduce road accidents, and improve road safety. These systems can influence driver behavior and reaction to traffic situations underlining the need to consider human factors before deploying. Driving simulators (DS) have the potential to study the impact of vehicular systems like ADAS and ITS on driver behavior from the early stages of the design process, as in [1]–[5]. To study driver behavior interacting with vehicular systems and traffic, driving simulators require a capability to generate realistic traffic behavior around the driver's ego-vehicle.

Simulating traffic only in the vicinity of the ego-vehicle can cause DSs to fail to reflect the aggregate traffic behavior, such as a change in traffic flows due to congestion - particularly if the congestion is due to the driver's behavior in the simulator. This challenge has led the past research to integrate a traffic simulator (TS) with a DS [6], [10]. At the same time, microscopic traffic simulations are capable of simulating realistic traffic behavior over large networks. These microscopic traffic simulations' speed depends on the size of the network, traffic flow, hardware, and operating system. In general, the time taken by many TSs for executing a simulation step does not correspond to the real-time; many TSs update at a frequency of 10 Hz maximum or slower such that, if interacting live with a human, would cause human drivers to identify discontinuous trajectory jumps in the traffic. Thus, TS software is usually not suitable for studying driver behavior to traffic within DSs, at least not without modification.

Traffic motion predicted by a TS surrounding the ego-vehicle within a DS should be updated at a rate that the driver perceives as continuous, a rate that is generally 20 Hz or faster. This creates the need to up-sample the traffic information from TS to use in a DS [6]. This up-sampling is not difficult as traffic positions can be interpolated or extrapolated assuming either continuous velocity or acceleration between two successive simulation steps.

The literature on integration of a TS with a DS focuses on both the spatial and temporal integration requirements that link these together. For example, [1] achieves road matching between the TS and DS using a geographic database as a reference for traffic network models in both. [6] proposes a framework to synchronize a faster-than-real-time TS with a DS. In [7], a nano traffic model in DS controls traffic near the ego-vehicle, whereas TS manages the traffic in a larger network. Frameworks to integrate networked DSs and different simulators (vehicles, bikes, pedestrians) with the TS are proposed in [8], [9]. Open-source TS (SUMO) is

Satya Prasad Maddipatla is a graduate student in Mechanical Engineering at The Pennsylvania State University, University Park, PA 16802, USA. email: szm888@psu.edu

Sean Brennan is with the Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA 16802, USA. IEEE Member. email: snb10@psu.edu

integrated with a DS in [11], [12]. This past work shows a strong and growing research interest in TS integrated with DS software. Yet a remaining challenge is determining how to integrate a functionally slower-than-real-time TS with a DS.

The challenge of the time-synchronization of two processes is a well-studied area. Perhaps the most famous example is that of a Phase-Locked Loop (PLL) which is a feedback system that forces one system (often a voltage-controlled oscillator) to replicate and track both the frequency and phase of the input when in lock [13]. The PLL literature defines several concepts that are useful in studying TS/DS interactions. For example, there is a master/slave behavior where the slave clock tries to track both the frequency and phase of the master; as well, PLLs use a negative feedback configuration to account for phase error between master and slave. This literature also introduces the notion of jitter: a statistical measure of the deviation of the PLL's slave clock edges compared to master clock edges. Feed-forward control is another well-studied area in PLLs where disturbances are measured and accounted for before they affect the system [14].

While the concepts of a PLL can assist in constructing a feedback-based software architecture for TSs interacting with DSs, there are also key differences. For example, the master/slave behavior used in PLL systems is not clearly defined with driving simulators. The DS/TS synchronization requires real-time tracking of traffic position by the DS as the slave to the TS as the master. However, the TS as the slave must accept ego-vehicle motion updates from the DS as the master. Thus, the TS/DS behaviors are more coupled than would the typical PLL model. Further, a time-varying TS speed can act as a disturbance to the traffic.

In this paper, a co-simulation framework is proposed to integrate slower-than-real-time TS with a real-time DS. The contribution of this work compared to the previous work is to present a method that corrects for the occasional time slip in a TS relative to a real-time DS. If this time slip is ignored, the traffic motion predicted by the TS could exhibit discontinuous motions within the DS. The remainder of this paper is as follows: Section II describes the framework. Section III shows the performance of the algorithm, and section IV analyzes the effect of tuning parameters, particularly the proportional gain, on the framework's performance.

II. CO-SIMULATION FRAMEWORK

For purposes of clarity, we first define the concept of time "slip" as follows: simulations exhibit "slip" if the time to execute a simulation step is more than the simulation's step size; the resulting time error is the *slip*. As shown in Fig. 1, a real-time simulation should run each simulation step in time equal to the simulation's step size. A slower-than-real-time simulation takes more time than the simulation's step size causing the simulation to slip. A faster-than-real-time simulation can execute the computations more quickly than the simulation's step size, thus can wait some remaining time to perform the next step to be a real-time simulation.

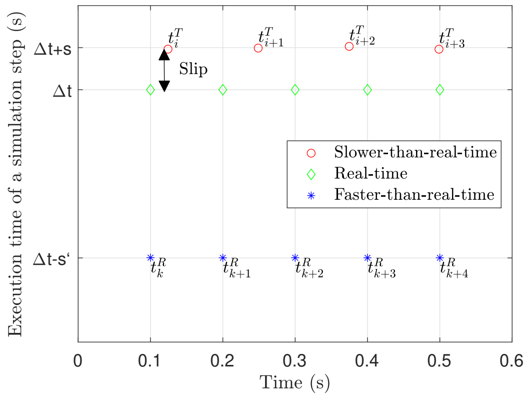


Fig. 1. A slower-than-real-time simulation takes more time per step than the step size. A real-time simulation takes time equal to the step size. A faster-than-real-time takes less time than the step size. Here Δt is the step size, 0.1s in this case, and s is the time slip.

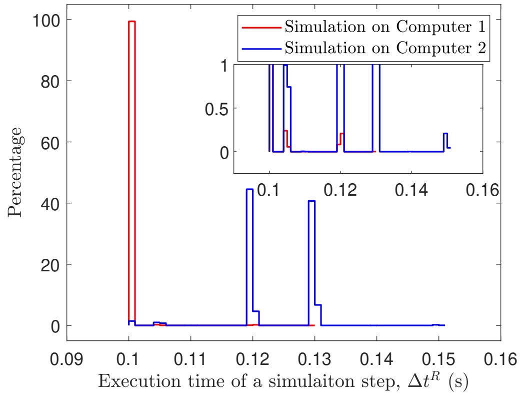


Fig. 2. Time to execute a simulation step in the TS (AIMSUN) on different computers. Simulation step size in both the trials is $0.1s$. TS's aggregate slip ratio in the simulation on Computer 1 is 1.002, and in the simulation on Computer-2 is 1.24.

The strategy and framework used in this work take advantage of the observation that microscopic traffic simulations, even though their update rates are low, are often close to real-time or even real-time for long intervals with infrequent slip events. For these situations, the microscopic traffic simulation can emulate real-time behavior by artificially simulating traffic moving at very slightly higher speeds, with the result that they are slightly slowed, on average, due to slip down to the average speed. This method of artificially increasing TS speeds assumes that the TS's slip is small and does not violate "rules of the road" behavior, driver reaction behavior, or any other speed-dependent traffic dynamics within the TS. If a TS is nearly real-time, this slight speed-up of traffic can be used to gradually recover from the periodic slip in the TS, allowing one to estimate traffic moving at realistic speeds in real-time. For reference, in this work it was found that the TS speeds had to be increased by a factor of 1.002, or 0.2 percent, to account for slip events.

To prevent the infrequent TS slip events from affecting the DS, the method used in this work was to synchronize the TS with a faster-than-real-time reduced-order model for local traffic around the ego-vehicle, e.g. the traffic that the human driver can "see". The control of time synchronization

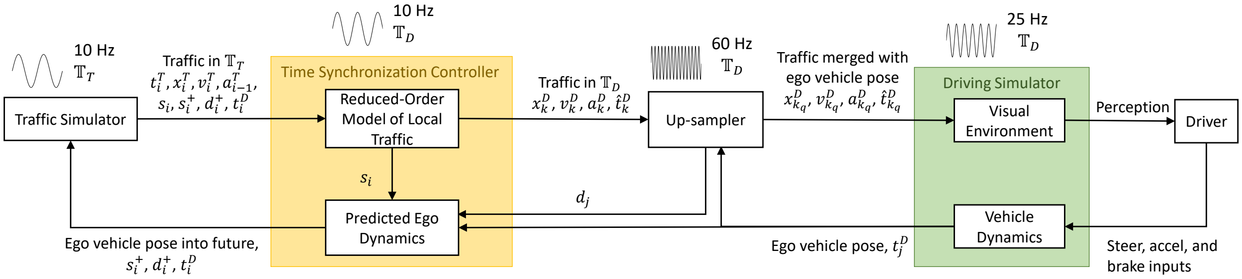

10 Hz
$\mathbb{T}_T$
Traffic Simulator
Traffic in $\mathbb{T}_T$
$t_i^T, x_i^T, v_i^T, a_{i-1}^T,$
s_i, s_i^+, d_i^+, t_i^D
10 Hz
$\mathbb{T}_D$
Time Synchronization Controller
Reduced-Order Model of Local Traffic
s_i
Predicted Ego Dynamics
Ego vehicle pose into future,
s_i^+, d_i^+, t_i^D
Traffic in $\mathbb{T}_D$
$x_k^D, v_k^D, a_k^D, \hat{t}_k^D$
60 Hz
$\mathbb{T}_D$
Up-sampler
d_j
Traffic merged with ego vehicle pose
$x_{k_q}^D, v_{k_q}^D, a_{k_q}^D, \hat{t}_{k_q}^D$
25 Hz
$\mathbb{T}_D$
Driving Simulator
Visual Environment
Vehicle Dynamics
Ego vehicle pose, t_j^D
Perception
Driver
Steer, accel, and brake inputs

Fig. 3. Co-simulation framework to synchronize a slower-than-real-time traffic simulator with a real-time driving simulator

of three models – the near-real-time TS, the faster-than-real-time reduced-order model, and the real-time DS – is the general strategy employed in this work.

For examples of real-world slip values, Fig. 2 shows the distribution of time taken by the TS to execute simulation steps in two different trials. Few instances in the simulation on Computer-1 and many instances in the simulation on Computer-2, where the execution time is more than the simulation step size, indicates that the simulation slips in real-time. This non-real time behavior of the TS necessitates a co-simulation framework, as shown in Fig. 3, to update traffic around ego-vehicle in the DS using the TS's traffic information.

The co-simulation framework proposed in Fig. 3 synchronizes the traffic from the TS simulating slower-than-real-time with ego-vehicle in the DS. It uses three clocks: 1) the traffic clock, 2) the clock for the reduced-order model, and 3) the clock for the driving simulator and ego-vehicle. Each is explained in the section that follows. The framework also consists of six modules: vehicle dynamics, visual environment (Blender 2.79b), anticipated ego dynamics, traffic simulator (AIMSUN NEXT 8.3), reduced-order model of local traffic, and an up-sampler module. All the modules except the visual environment and traffic simulator are implemented within the Robot Operating System (ROS Kinetic on Ubuntu 16.04).

A. Time Coordinates

The challenge of the proposed algorithm is related to the goal to synchronize three different time coordinates simultaneously. The first time system, the *Traffic simulator time coordinate*, $\mathbb{T}_T$, is a non-real time coordinate that increases by the TS's step size after every TS simulation step. The second time system is the *Reduced-order model time coordinate*, $\mathbb{T}_R$, which is used to simulate a local, reduced-order model of the traffic simulation. And the third time system is the *Driving simulator time coordinate*, $\mathbb{T}_D$, which is the time system in which the driver is interacting with the driving simulator. The second and third time systems, $\mathbb{T}_R$ and $\mathbb{T}_D$, are both real-time whereas the first, $\mathbb{T}_T$, is not – thus creating difficulty. The origin, or zero-time point, of each coordinate (denoted $\mathbb{T}_i(0)$) is as follows: $\mathbb{T}_T(0)$ is when the first message is sent from the TS to the reduced-order model; $\mathbb{T}_R(0)$ is when the first message is received by the reduced-order model from the TS; and $\mathbb{T}_D(0)$ is when the first message is sent from the vehicle dynamics module to the predicted ego dynamics module.

The next goal is to relate individual time measurements, t_i, within each time coordinate to each other. We begin first by relating the traffic simulator time, $\mathbb{T}_T$, to the reduced-order model time, $\mathbb{T}_R$. Specifically, the i^{th} time in the reduced-order model, denoted by t_i^R, is related to the i^{th} time in the traffic simulator, denoted t_i^T by equation (1).

$$t_i^R = t_i^T + \Sigma_{p=1}^{i-1} s_p + s_i^+ \tag{1}$$

We expect a one-to-one relation, (2), to exist between the reduced-order model time, $\mathbb{T}_R$ and the driving simulator time, $\mathbb{T}_D$, which is quantified as t_i^D. This mapping is unknown, but given by:

$$\boxed{t_i^R} \longleftrightarrow \boxed{t_i^D + s_i^+ + d_i^+} \tag{2}$$

We define s_p as the TS's instantaneous slip, e.g. the difference between the time to execute a simulation step and step size. The amount of time ego-vehicle will be projected into the future to compensate for the TS's slip is s_i^+, and to account for the delay between the ego-vehicle and traffic time coordinates is d_i^+.

We now define the aggregate slip-ratio, $\bar{r}_{(i-N)\to i}$, as the ratio of the cumulative time in $\mathbb{T}_R$ necessary to receive the data from the last N simulation steps in the TS divided by N times the TS's step size, Δt, as in equation (3).

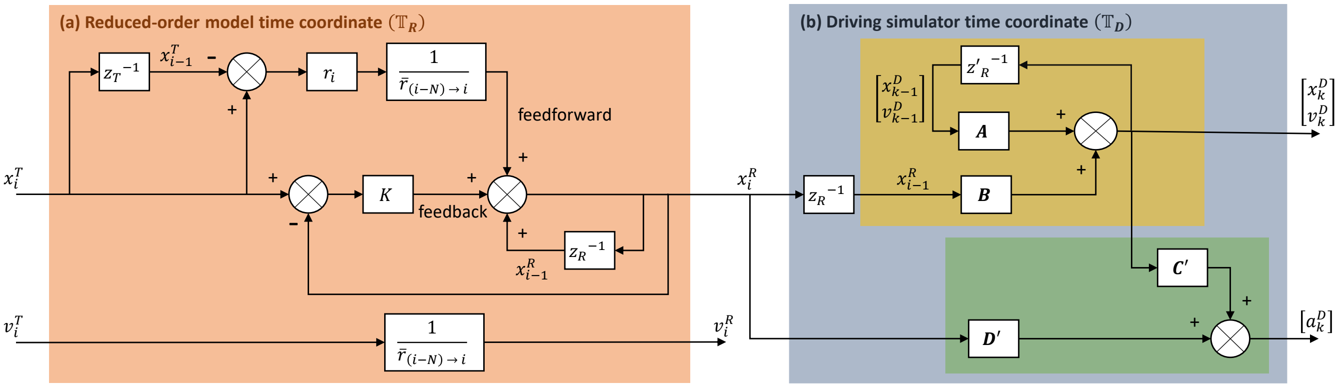
$$\bar{r}_{(i-N)\to i} = \frac{t_i^R - t_{i-N}^R}{N\,\Delta t} \tag{3}$$

N is chosen to be a large enough value to capture infrequent intervals of slip and thus act similar to a low-pass filter; for this work $N = 100$. In comparison, the TS's instantaneous slip-ratio, r_i, is calculated from the time in $\mathbb{T}_R$ for executing a simulation step in the TS divided by the TS's step size, as in equation (4).

$$r_i = \frac{t_i^R - t_{i-1}^R}{\Delta t} \tag{4}$$

B. Modules

The *vehicle dynamics* module receives the human driver's inputs to convert them into the ego-vehicle pose using vehicle dynamics. In this study, a bicycle model formulation with nonlinear tire behavior is used to simulate vehicle chassis behavior as this model is quite accurate for typical driving situations [15], [16]. The *predicted ego dynamics* module projects the ego-vehicle pose from the vehicle dynamics module into the future to compensate for TS's slip and transmission delay, assuming constant velocity and yaw rate.



(a) Reduced-order model time coordinate ($\mathbb{T}_R$)
z_T^{-1}
x_{i-1}^T
-
+
r_i
$\frac{1}{\bar{r}_{(i-N)\to i}}$
feedforward
x_i^T
+
-
K
+
+
feedback
+
z_R^{-1}
x_{i-1}^R
v_i^T
$\frac{1}{\bar{r}_{(i-N)\to i}}$
v_i^R
x_i^R
(b) Driving simulator time coordinate ($\mathbb{T}_D$)
$\begin{bmatrix} x_{k-1}^D \\ v_{k-1}^D \end{bmatrix}$
$z'_R{}^{-1}$
A
+
+
$\begin{bmatrix} x_k^D \\ v_k^D \end{bmatrix}$
z_R^{-1}
x_{i-1}^R
B
C'
+
+
D'
$[a_k^D]$

Fig. 4. Reduced-Order Model of local traffic. (a) Feed-forward and feedback control to transform TS-vehicles' states from $\mathbb{T}_T$ to $\mathbb{T}_R$. (b) Model-based prediction to transform TS-vehicles' states from $\mathbb{T}_R$ to $\mathbb{T}_D$.

The extent of projection, $s_i^+ + d_i^+$, depends on the TS's maximum instantaneous slip and the delay between ego-vehicle and traffic, d_j. The *traffic simulator* module updates the microscopic traffic simulation with the ego-vehicle pose from the module - *predicted ego dynamics*. The TS sends the traffic's vehicles that are close to the ego-vehicle – a 1km radius was used in this study – to the *reduced-order model* model at 10Hz in $\mathbb{T}_T$. It simulates traffic without vehicle dynamics using the car-following and lane-change model. The traffic data need to be processed to include dynamics. For example, the velocities can be uniformly reduced in a low friction scenario, thereby having an appropriate deceleration behavior. While not part of this work, more complicated reduced-order models of each vehicle in the local traffic could be used. The module *reduced-order model of local traffic* transforms traffic states from $\mathbb{T}_T$ to $\mathbb{T}_D$. The *up-sampler* module increases the frequency of the traffic states in $\mathbb{T}_D$ to 60Hz by assuming uniform acceleration, thus matching the frequency of traffic with that of ego-vehicle. The traffic and the ego-vehicle states from the up-sampler are updated in the *visual environment* at 25Hz in real-time. Figure 3 shows the data flow between different modules in the co-simulation framework.

C. Reduced-Order Model

We define TS-vehicle and DS-vehicle to describe traffic information flow from the TS to DS for clarity purposes. A *TS-vehicle* is a unique vehicle within a local region of interest in the TS. In this study, this local region moves with the ego-vehicle and is defined as a 1km radius around the ego-vehicle. A *DS-vehicle* is a vehicle model in the DS. A DS-vehicle can represent several TS-vehicles during a co-simulation run. A TS-vehicle is assigned to a DS-vehicle until it leaves the local region. Typically, there are roughly 60 to 80 vehicles within this local region (including both directions of travel) for the simulations performed in this study on free-flow highway driving with two lanes (slow lane and passing) on a divided highway, assuming relatively congested conditions.

The reduced-order model applies only to the TS-vehicles. It transforms TS-vehicles' states from $\mathbb{T}_T$ to $\mathbb{T}_D$, as shown in Fig. 4. The change in TS's speed adds disturbance to TS-vehicles' position in $\mathbb{T}_R$. The disturbance is rejected using feed-forward control based on TS's slip ratios as in the second term on the right-hand side of equation (5).

$$x_i^R = x_{i-1}^R + \frac{1}{\bar{r}_{(i-N)\to i}}(x_i^T - x_{i-1}^T)r_i + K(x_i^T - x_i^R) \tag{5}$$

The disturbance rejection using a feed-forward control induces spatial jitter in TS-vehicles' position. Spatial jitter is defined here as the difference between the position in $\mathbb{T}_T$ and the position in $\mathbb{T}_R$; it is compensated using proportional control, K, acting on spatial jitter, as shown in equation (5). Equation (5) is rearranged to estimate position of TS-vehicles in $\mathbb{T}_R$, as in equation (6).

$$x_i^R = \frac{1}{K+1}(x_{i-1}^R + \frac{1}{\bar{r}_{(i-N)\to i}}(x_i^T - x_{i-1}^T)r_i + Kx_i^T) \tag{6}$$

where x_i^T is the position of a TS-vehicle in station coordinates in $\mathbb{T}_T$, x_i^R is the position of the same vehicle in station coordinates in $\mathbb{T}_R$, and K is the proportional gain. The initial position of a TS-vehicle in $\mathbb{T}_R$ is set to be equal to its position in $\mathbb{T}_T$, as in equation (7).

$$x_{init}^R = x_{init}^T \tag{7}$$

The velocity of a TS-vehicle in $\mathbb{T}_R$ is related to its velocity in $\mathbb{T}_T$ by equation (8).

$$v_i^R = \frac{1}{\bar{r}_{(i-N)\to i}} v_i^T \tag{8}$$

where v_i^T is the velocity of a TS-vehicle in $\mathbb{T}_T$ and v_i^R is the velocity of the same vehicle in $\mathbb{T}_R$.

Figure 4(b) shows the process of model-based estimation of position, velocity, and acceleration of TS-vehicle in $\mathbb{T}_D$. In order for the time in the $\mathbb{T}_R$ to be mapped to the estimator's outputs in the $\mathbb{T}_D$ frame, the following constraint must apply to the ordering of respective times:

$$t_{k-1}^R < t_k^R \leq t_{i-1}^R < t_{k+1}^R < t_i^R \tag{9}$$

where the subscript i corresponds to terms in the $\mathbb{T}_R$ frame, and k corresponds to terms in the $\mathbb{T}_D$ frame. The estimator's output interval is related to the TS step size, as in equation

(10). The time in $\mathbb{T}_D$ mapped to the estimator output, $\hat{t}_k^D$, is given by equation (11).

$$\Delta t = t_k^R - t_{k-1}^R \tag{10}$$

$$\hat{t}_k^D = t_i^D + s_i^+ + d_i^+ - t_i^R + t_k^R \tag{11}$$

The position and velocity of TS-vehicles in $\mathbb{T}_D$ are estimated in equation (12) using a uniform acceleration model in the interval given by the equation (10).

$$\begin{aligned}
\begin{bmatrix} x_k^D \\ v_k^D \end{bmatrix} &= \mathbf{A} \begin{bmatrix} x_{k-1}^D \\ v_{k-1}^D \end{bmatrix} + \mathbf{B} \begin{bmatrix} x_{i-1}^R \end{bmatrix} \\
\mathbf{A} &= \begin{bmatrix} 1 - \frac{\Delta t^2}{(t_{i-1}^R - t_{k-1}^R)^2} & \Delta t - \frac{\Delta t^2}{(t_{i-1}^R - t_{k-1}^R)} \\ -\frac{2\Delta t}{(t_{i-1}^R - t_{k-1}^R)^2} & 1 - \frac{2\Delta t}{(t_{i-1}^R - t_{k-1}^R)} \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} \frac{\Delta t^2}{(t_{i-1}^R - t_{k-1}^R)^2} \\ \frac{2\Delta t}{(t_{i-1}^R - t_{k-1}^R)^2} \end{bmatrix}
\end{aligned} \tag{12}$$

where x_k^D, v_k^D, and a_k^D are the position, velocity, and acceleration estimate of a TS-vehicle in $\mathbb{T}_D$. $\mathbf{A}$ and $\mathbf{B}$ are time dependent. The acceleration of a TS-vehicle in $\mathbb{T}_D$ is estimated from its position in $\mathbb{T}_R$ as in equation (13) assuming uniform acceleration between t_k^R and t_i^R.

$$\begin{aligned}
\begin{bmatrix} a_k^D \end{bmatrix} &= \mathbf{C}' \begin{bmatrix} x_k^D \\ v_k^D \end{bmatrix} + \mathbf{D}' \begin{bmatrix} x_i^R \end{bmatrix} \\
\mathbf{C}' &= \begin{bmatrix} -\frac{2}{(t_i^R - t_k^R)^2} & -\frac{2}{(t_i^R - t_k^R)} \end{bmatrix} \\
\mathbf{D}' &= \begin{bmatrix} \frac{2}{(t_i^R - t_k^R)^2} \end{bmatrix}
\end{aligned} \tag{13}$$

where $\mathbf{C}'$, and $\mathbf{D}'$ are time dependent. The position, velocity, and acceleration of a TS-vehicle in $\mathbb{T}_D$ are initialized as in equation (14).

$$\begin{aligned}
x_{init}^D &= x_{init}^R - v_{init}^R (t_i^R - t_k^R) \\
v_{init}^D &= v_{init}^R \\
a_{init}^D &= 0
\end{aligned} \tag{14}$$

The accuracy of the reduced-order model depends on the radius of the local region in the TS. The minimum radius is the radius at which ego-vehicle's driver behavior does not change with an increase in the radius, e.g. where the driver can no longer perceive effects of error due to the TS. This radius is assumed to be 1km in this study.

D. Up-sampler

The traffic data can be up-sampled n times as in equation (15), where n is the desired frequency ratio to the input frequency and δ is the up-sampling interval. In this study, $n = 6$ to match the frequency of traffic (10 Hz) with that of ego-vehicle simulation (60 Hz).

$$\Delta t = n\delta \tag{15}$$

The position, velocity, and acceleration of TS-vehicles between the interval in equation (10) are given by equation (16) assuming uniform acceleration within the interval, Δt.

$$\begin{aligned}
x_{k_q}^D &= x_k^D + v_k^D (q\delta) + \frac{1}{2} a_k^D (q\delta)^2 \\
v_{k_q}^D &= v_k^D + a_k^D (q\delta) \\
a_{k_q}^D &= a_k^D
\end{aligned} \tag{16}$$

where $x_{k_q}^D$, $v_{k_q}^D$, and $a_{k_q}^D$ are the up-sampled position, velocity, and acceleration of a TS-vehicle in $\mathbb{T}_D$. q is an integer in the interval $0 \le q < n-1$. The up-sampler also estimates the instantaneous delay between the ego-vehicle and traffic as in equation (17).

$$d_j = t_j^D - (\hat{t}_k^D + q\delta) \tag{17}$$

III. Results

A summary analysis of the simulation results reveals that the overall synchronization of the DS and TS strongly depends on the proportional feedback gain, K. The spatial jitter and acceleration of DS-vehicles in $\mathbb{T}_D$ quantify the performance of the framework. The acceleration in $\mathbb{T}_D$ is useful to illustrate deviations in normal traffic behavior. Ideally, we expect zero spatial jitter and no deviation from natural traffic behavior. The analysis reveals that achieving zero spatial error is impossible without largely deviating from natural traffic behavior and vice-versa. So there is a need for a trade-off to minimize both spatial jitter and deviation from natural traffic behavior. The analysis also reveals that the required artificial speed-up of traffic in the TS is negligible because of the TS's near-real-time nature; this is reflected in TS's aggregate slip ratios, which are close to one.

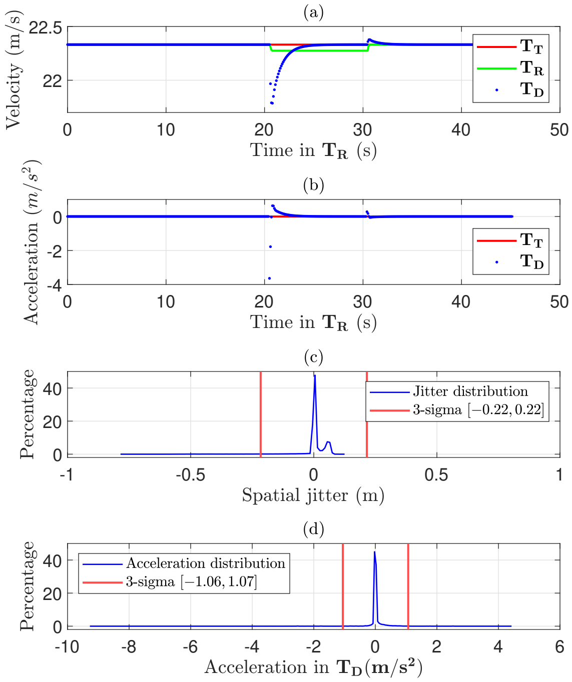


Fig. 5. Proportional gain, $K = 0.1$. (a) Velocity of a TS-vehicle in $\mathbb{T}_T$, $\mathbb{T}_R$, and $\mathbb{T}_D$. (b) Acceleration of a TS-vehicle in $\mathbb{T}_T$, and $\mathbb{T}_D$. (c) Spatial jitter of a DS-vehicle. (d) Acceleration of a DS-vehicle in $\mathbb{T}_D$.

As specific examples, Figures $5(b)$, $6(b)$, and $7(b)$ show the acceleration of one TS-vehicle for different proportional gains. The spatial jitter distribution of one DS-vehicle for

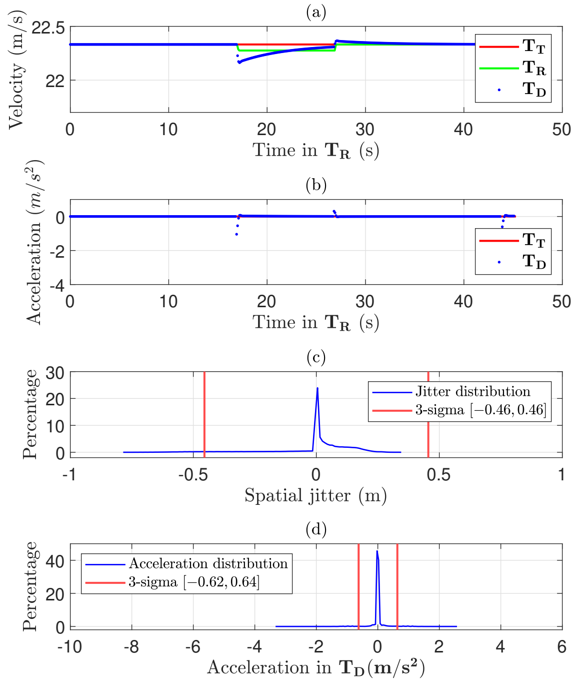


Fig. 6. Proportional gain, $K = 0.02$. (a) Velocity of a TS-vehicle in $\mathbb{T}_T$, $\mathbb{T}_R$, and $\mathbb{T}_D$. (b) Acceleration of a TS-vehicle in $\mathbb{T}_T$, and $\mathbb{T}_D$. (c) Spatial jitter of a DS-vehicle. (d) Acceleration of a DS-vehicle in $\mathbb{T}_D$.

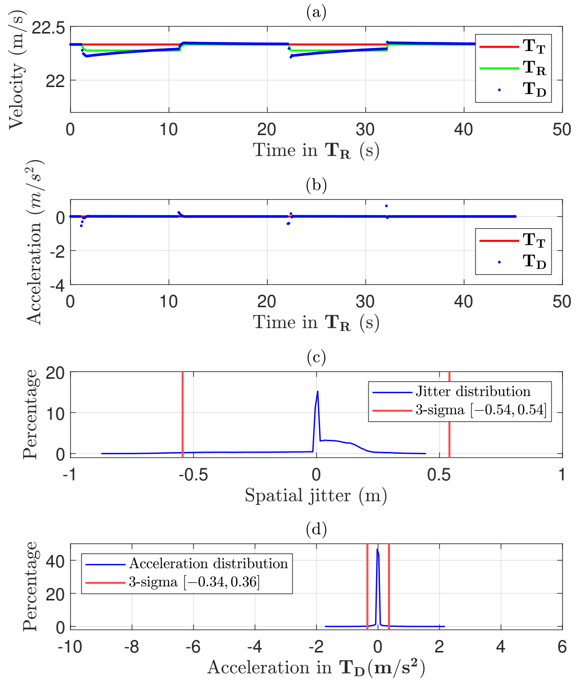


Fig. 7. Proportional gain, $K = 0.01$. (a) Velocity of a TS-vehicle in $\mathbb{T}_T$, $\mathbb{T}_R$, and $\mathbb{T}_D$. (b) Acceleration of a TS-vehicle in $\mathbb{T}_T$, and $\mathbb{T}_D$. (c) Spatial jitter of a DS-vehicle. (d) Acceleration of a DS-vehicle in $\mathbb{T}_D$.

different values of proportional gain is shown in Fig. 5(c), 6(c), and 7(c). Spatial jitter can be seen to decrease with an increase in the proportional gain. Like jitter, the acceleration of DS-vehicles in $\mathbb{T}_D$ shows a similar trend among different DS-vehicles. For brevity, Fig. 5(d), 6(d), and 7(d) shows acceleration distribution of one of the DS-vehicles in $\mathbb{T}_D$ for different values of proportional gain. The acceleration of a DS-vehicle in $\mathbb{T}_D$ depicts a more natural behavior with decreased proportional gain. We found that a gain, $K = 0.02$, results in acceptable trade-offs in spatial jitter while the corresponding acceleration in $\mathbb{T}_D$ depicts a more natural traffic behavior.

To illustrate how small the artificial velocity changes are within the TS to emulate real-time behavior, the TS's instantaneous slip ratio for the highway scenario mentioned previously is shown in Fig. 8(a). One can observe the infrequent slips in the TS as there are relatively few instances where the instantaneous slip ratio is greater than one. The TS's aggregate slip ratio shown in Fig. 8(b) is close to one, reflecting the near-real-time nature of the TS. The spatial jitter of DS-vehicles shows a similar trend among different DS-vehicles. For brevity, Fig. 8(c) shows jitter of just one of the DS-vehicles. Figure 8(b),(c) shows that the fall and rise in the spatial jitter correspond to the rise and fall in the TS's aggregate slip ratio. This dependence is due to the feed-forward term in equation (5).

For comparison, Fig. 5(a), 6(a), and 7(a) show the velocity of one TS-vehicle for different proportional gains. The velocity in $\mathbb{T}_D$ and $\mathbb{T}_R$ is very close to the velocity in $\mathbb{T}_T$, which is possible due to the near-real-time TS. The required speed-up of traffic in the TS based on the aggregate slip ratio is only 0.2%, which is negligible in impact on traffic behavior. The ego-vehicle and traffic states are synchronized to be perceived by humans as smooth, as shown in Fig. 9. The delay between the ego-vehicle and traffic is less than 0.05s for most of the time. Figure 10 shows the maximum error in position of the ego-vehicle, showing that in the TS the ego vehicle is typically projected roughly 6.5 meters ahead of where it would be in the DS. This error, which is the difference in ego-vehicle's position in $\mathbb{T}_D$ and $\mathbb{T}_T$, can affect immediate traffic behavior around the ego vehicle, which can cause erroneous results if very close car-following is occurring in the TS. The source of this error is mainly the transmission delay from the TS to DS. Fortunately, typical TS behaviors rarely if ever have TS vehicles so near to the

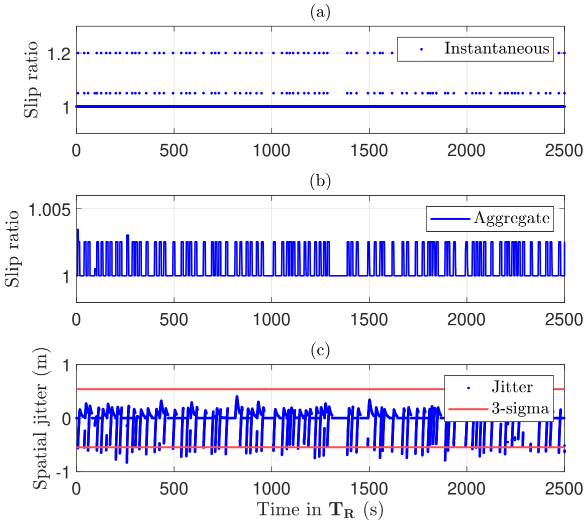


Fig. 8. (a) Instantaneous slip ratio of the TS. (b) Aggregate slip ratio of the TS. In this study, $N = 100$ to calculate the TS's aggregate slip ratio. (c) Spatial jitter of a DS-vehicle.

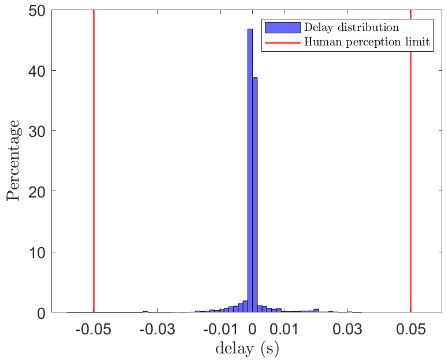


Fig. 9. Delay is the difference between time associated with ego-vehicle and traffic after compensating for transmission delays.

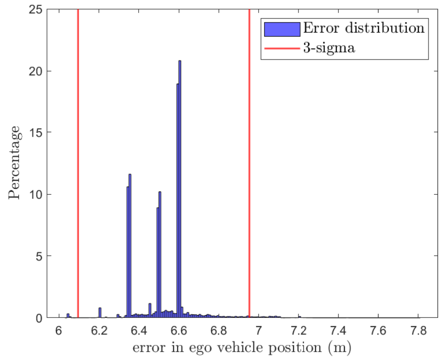


Fig. 10. Error is the difference between position of ego-vehicle in the DS and position of ego-vehicle in the TS. This error is visible only to the traffic in the TS.

ego vehicle (e.g. within two car lengths), and thus this error is assumed to have small impact on TS behavior. Overall, the use of a non-real-time TS to increase fidelity of real-time DS traffic representations will require such trade-offs; this work is therefore presented as a guide to show algorithms and analyses that facilitate this integration.

IV. CONCLUSIONS

In this paper, a co-simulation framework to synchronize a slower-than-real-time TS with real-time DS is presented. The co-simulation framework involves a reduced-order model for local traffic to estimate position, velocity, and acceleration of traffic by reducing spatial jitter and maintaining acceleration behavior similar to normal road vehicles. A feedback loop, feed-forward, and model-based prediction are used in this work to implement the reduced-order local traffic model. The feedback loop's proportional gain was then tuned to balance the goals of decreasing jitter versus reducing acceleration behaviors in traffic. For $K = 0.1$, the spatial jitter is less, but the accelerations in $\mathbb{T}_D$ are largely deviating from normal traffic behavior. Spatial jitter for $K = 0.02$ is less than a meter, and accelerations in $\mathbb{T}_D$ reflect the normal traffic behavior; this value was found to be a suitable trade-off for this driving simulation application. Further, there is little impact due to the artificial speed-up of traffic in the TS as the maximum TS's aggregate slip ratio is less than 1.004 and typically 1.002 or less.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under grant no. CNS-1932509 "CPS: Medium: Collaborative Research: Automated Discovery of Data Validity for Safety-Critical Feedback Control in a Population of Connected Vehicles."

REFERENCES

[1] Ming Jin and Soi-Hoi Lam, "A virtual-reality based integrated driving-traffic simulation system to study the impacts of intelligent transportation system (ITS)," Proceedings. 2003 International Conference on Cyberworlds, Singapore, 2003, pp. 158-165, doi: 10.1109/CYBER.2003.1253449.

[2] C. Chang and Y. Chou, "Development of Fuzzy-Based Bus Rear-End Collision Warning Thresholds Using a Driving Simulator," in IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 2, pp. 360-365, June 2009, doi: 10.1109/TITS.2009.2020204.

[3] C. Olaverri-Monreal, P. Gomes, M. K. Silvéria and M. Ferreira, "In-Vehicle Virtual Traffic Lights: A graphical user interface," 7th Iberian Conference on Information Systems and Technologies (CISTI 2012), Madrid, 2012, pp. 1-6.

[4] P. R. J. A. Alves, J. Gonçalves, R. J. F. Rossetti, E. C. Oliveira and C. Olaverri-Monreal, "Forward collision warning systems using heads-up displays: Testing usability of two new metaphors," 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, 2013, pp. 1-6, doi: 10.1109/IVS.2013.6629438.

[5] Ali Akbari, Farshidreza Haghighi, "Traffic calming measures: An evaluation of four low-cost TCMs' effect on driving speed and lateral distance," IATSS Research, Volume 44, Issue 1, 2020, pp. 67-74.
Ali Akbari, Farshidreza Haghighi, "Traffic calming measures: An evaluation of four low-cost TCMs' effect on driving speed and lateral distance," IATSS Research, Volume 44, Issue 1, 2020, Pages 67-74, ISSN 0386-1112, https://doi.org/10.1016/j.iatssr.2019.07.002.

[6] V. Punzo and B. Ciuffo, "Integration of Driving and Traffic Simulation: Issues and First Solutions," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 354-363, June 2011, doi: 10.1109/TITS.2010.2095846.

[7] Thomas Nguen That, Jordi Casas, "An integrated framework combining a traffic simulator and a driving simulator," Procedia - Social and Behavioral Sciences, Volume 20, 2011, Pages 648-655, ISSN 1877-0428, https://doi.org/10.1016/j.sbspro.2011.08.072.

[8] Paz A, Veeramisti N, Khaddar R, de la Fuente-Mella H, Modorcea L, "Traffic and Driving Simulator Based on Architecture of Interactive Motion," ScientificWorldJournal, 2015.

[9] K. Abdelgawad, S. Henning, P. Biemelt, S. Gausemeier, A. Trächtler, "Advanced Traffic Simulation Framework for Networked Driving Simulators," IFAC-PapersOnLine, Volume 49, Issue 11, 2016, pp. 101-108.

[10] Mansoureh Jeihani, Shiva NarooieNezhad, Kaveh Bakhsh Kelarestaghi, "Integration of a driving simulator and a traffic simulator case study: Exploring drivers' behavior in response to variable message signs," IATSS Research, Volume 41, Issue 4, 2017, Pages 164-171, ISSN 0386-1112, https://doi.org/10.1016/j.iatssr.2017.03.001.

[11] Cristina Olaverri-Monreal, Javier Errea-Moreno, Alberto Diaz Alvarez, Carlos Biurrun-Quel, Luis Serrano-Arriezu, Markus Kuba, "Connection of the SUMO Microscopic Traffic Simulator and the Unity 3D Game Engine to Evaluate V2X Communication-Based Systems," Sensors (Basel, Switzerland), volume 18, 2018.

[12] Jakob Kaths, Benedikt Schott, Frederic Chucholowski, "Co-simulation of the virtual vehicle in virtual traffic considering tactical driver decisions," SUMO User Conference 2019, volume 62, pp. 21-28.

[13] D. Wolaver, "Phase-Locked Loop Circuit Design", Prentice Hall, 1991.

[14] B. J. Thijssen, E. A. M. Klumperink, P. Quinlan and B. Nauta, "Feedforward Phase Noise Cancellation Exploiting a Sub-Sampling Phase Detector," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 11, pp. 1574-1578, Nov. 2018, doi: 10.1109/TCSII.2017.2764096.

[15] W.F. Milliken, D.L. Milliken, "Race car vehicle dynamics," Premiere series, SAE International, 1995.

[16] R. Rajamani, "Vehicle dynamics and control," Mechanical engineering series 9781461414322, Springer, US, 2011.