Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform

Tulga Ersal a, Mark Brudnak b, Ashwin Salvia a, Jeffrey L. Stein c, Zoran Filipia, Hosam K. Fathy a,⇑

a Department of Mechanical Engineering, University of Michigan, 1231 Beal Ave., Ann Arbor, MI, USA
b The US Army Tank-Automotive Research, Development and Engineering Center, 6501 E. 11 Mile Road, Bldg 215, (RDTA-RS) MS 157, Warren, MI, USA
c Department of Mechanical Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI, USA

⇑ Corresponding author. Tel.: +1 734 936 5295.
E-mail address: hfathy@umich.edu (H.K. Fathy).

A R T I C L E   I N F O

Article history:
Received 11 January 2010
Accepted 3 August 2010

Keywords:
Real-time hardware-in-the-loop simulation
Internet-distributed simulation
Transparency

A B S T R A C T

This paper summarizes efforts to integrate, for the first time, two geographically dispersed hardware-in-the-loop simulation setups over the Internet in an observer-free way for an automotive application. The two setups are the engine-in-the-loop simulation setup at the University of Michigan (UM) in Ann Arbor, MI, USA, and the driver-in-the-loop ride motion simulator at the US Army Tank-Automotive Research, Development and Engineering Center (TARDEC) in Warren, MI, USA. The goal of such integration is to increase the fidelity of experiments and to enable concurrent geographically dispersed systems engineering. First, experiments with the actual hardware are presented. The concept of transparency is discussed, and the infeasibility of performing a baseline experiment with ideally integrated hardware is presented as a challenge to characterize the transparency of the experimental setup. This motivates the second half of the paper, in which a model-based approach is taken to analyze the transparency of the system. The conclusion is that an observer-free solution is feasible for integrating the two pieces of hardware over the Internet in a transparent manner, even if the nominal delay is increased by four times. It is also found that different signals in the system can exhibit different levels of transparency.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Hardware-in-the-loop simulation (HILS) provides a bridge between physical prototyping and virtual experiments by uniquely combining their advantages and allowing for cost-effective, high-fidelity experiments [1,2]. It strongly promotes concurrent system engineering and has therefore become indispensable in many application areas, such as automotive [3–11], aerospace [12–15], manufacturing [16], robotics [17,18], defense [19,20], and structural analysis [21,22].

A natural evolution of the basic HILS idea is to integrate multiple HILS setups to fully exploit their benefits [23]. However, if the setups are geographically dispersed, bringing them together and establishing a physical connection may be infeasible. In that case, a virtual coupling can be created through a communication medium.

Among the different communication media that are available for a distributed HILS setup, the Internet is an attractive choice due to its prevalence. The Internet-distributed HILS (ID-HILS) idea has emerged within the last decade and has found applications in the earthquake simulation [21], telerobotics [24], and automotive powertrain areas [25]. Within the earthquake simulation literature, the idea of geographically distributing experimental substructures within a network of laboratories linked through numerical simulations using the Internet was proposed by Campbell and Stojadinovic in 1998 [21]. Today, the George E. Brown Jr. Network for Earthquake Engineering Simulation (NEES) [26] provides an outstanding example of the capabilities and impact of the ID-HILS idea, and the earthquake literature presents many applications of the ID-HILS idea to earthquake simulation [27–32]. Within the telerobotics literature, the possibility of achieving stable force-reflecting teleoperation over the Internet was first explored by Niemeyer and Slotine, also in 1998 [24], and attracted the attention of many other researchers [33–42]. The notion of ID-HILS did not appear in the automotive powertrain systems engineering area until 2006, when US Tank-Automotive Research, Development, and Engineering Center (TARDEC) researchers successfully integrated a ride motion simulator in Warren, MI, USA, with a hybrid-power-system simulator in Santa Clara, CA, USA [25,43,44].

These works highlighted that using the Internet as the communication medium also creates some challenges due to the Internet’s inherent delay, jitter, and loss. Much of the delay is related to routing and processing in the network. Jitter refers to the variability of the delay, and loss occurs because not every packet sent through Internet necessarily arrives at its destination. Although loss can be remedied by transport-level protocols like TCP, such protocols increase the delay and are therefore not considered in this work.
These characteristics of the Internet jeopardize system stability, robustness, and transparency – a measure of how close the dynamics of the Internet-distributed system are to the dynamics of the directly coupled system [45,46]. Thus, maintaining stability, robustness, and transparency in the presence of delay, jitter, and loss is a major challenge in using the Internet to couple HILS setups.

This paper specifically focuses on the automotive powertrain application area. As mentioned above, the only application of ID-HILS to automotive powertrains is the work by TARDEC researchers [25,43,44]. To overcome the abovementioned challenges, they proposed an observer-based method. Specifically, each physical location had a local mathematical model, i.e., observer, of the remote component with which it needed to couple and interacted with the remote component not directly, but through its observer [25]. The integration ensured, through feedback control, that the states of the observers remained close to the actual states of the corresponding components. Different feedback control strategies were employed and analyzed, including PI [25], sliding mode, and $H_{\infty}$ control [43]. Duty cycle experiments have been performed successfully with this setup [44]. In addition to this observer-based approach, there are other techniques, such as the event-based framework [32,33,38] or the passivity-based framework [36,47–49], to address the challenges mentioned above. However, these alternative methods have not yet been investigated within this particular application area.

The goal of this paper is to adopt, for the first time, an event-based solution into the ID-HILS in the automotive powertrain application domain to investigate its viability in this particular domain and to enable an observer-free solution. The question of viability is based on the fact that the dynamics and goals of a powertrain ID-HILS system are different than the dynamics of an earthquake ID-HILS simulator or a telerobotic setup, the only application areas in which the event-based ideas were tested so far. Furthermore, earthquake ID-HILS setups are typically pseudo-dynamic or quasi-static [21,22,27,50–53], in the sense that the inertial effects are simulated numerically rather than experimentally. In the powertrain application area, however, real-time simulation and accurate capturing of dynamics of the hardware components become more critical. Whether the event-based idea can meet those requirements is still an open research question. The motivation for seeking an observer-free solution is that appropriate models, i.e., models that are accurate enough and can run in real time, may be difficult or even infeasible to obtain. Even if the models are available, they may be proprietary, making their utilization difficult. To make ID-HILS still feasible under such conditions, an observer-free solution is sought in this work.

The two hardware components of interest for this work are the engine in-the-loop simulation setup at the University of Michigan (UM) in Ann Arbor, MI, USA, and the abovementioned driver-in-the-loop ride motion simulator at the TARDEC Simulation Laboratory in Warren, MI, USA. The UM setup uses a 6L V8 diesel engine in combination with a vehicle dynamics and driveline model [54]. The motivation for having the engine in the loop stems from the fact that an accurate and fast prediction of the transient dynamics of an engine with all of its aspects, including torque generation, fuel economy, and emissions, still remains to be a challenge. Simulating soot formation processes in a diesel engine, for example, is prohibitively slow in the context of the systems work, since it requires coupling of sophisticated computational fluid dynamics models and chemical kinetics [54]. Thus, the only currently available feasible solution for a driver-in-the-loop, real-time, system-level analysis that requires an accurate representation of transient engine dynamics may be to actually put the physical engine into the loop. This need for engine- or powertrain-in-the-loop simulation has also been recognized in the literature by other researchers [55–58].

The TARDEC setup, on the other hand, has a human operator driving a virtual vehicle through a ride motion simulator. This enables the inclusion of a human driver in the loop and allows for a more realistic analysis with all the aspects of human driving. It is desired to couple UM’s engine with TARDEC’s simulator to enable the operator to drive a virtual car with a real engine, thereby increasing the fidelity of experiments and enabling concurrent engineering.

Towards these goals, Sections 2 and 3 present the experimental part of this work. The two pieces of HILS setups that are integrated are described in Section 2, and the adopted event-based communication is outlined. In Section 3, the experiments are explained, and the results from the ID-HIL experiments are given and interpreted. Sections 4 and 5 present a model-based study of the transparency of the adopted observer-free integration approach. The models are described in detail in Section 4, and simulation results are given along with interpretations for transparency in Section 5. The conclusions are given in Section 6.

2. Description of the ID-HILS setup

Fig. 1 illustrates a high-level overview of the ID-HILS setup considered in this study. The major components are denoted as driver, ride motion simulator (RMS), vehicle, internet, drivetrain, flywheel, idle controller, and engine, with regular and italic typefaces designating hardware and model components, respectively. In addition, the signals exchanged between these components are also shown along with the geographic locations of the components. These subsystems will be described next in detail.

2.1. Description of the hardware

The two hardware components of interest for this work are the engine at UM and the ride motion simulator at TARDEC (Fig. 2). The engine is a 6L V8 diesel engine with 240 kW rated power at 3300 rpm and a rated torque of 760 Nm. It is intended for a variety of medium-duty truck applications covering the range between classes IIB and VII. A high-fidelity, AC electric dynamometer couples the physical engine with the models in real time. Because this engine is approximately twice as powerful as the standard engine used in a High-Mobility Multipurpose Wheeled Vehicle (HMMWV), the vehicle of interest in this study, the measured engine torque is scaled by 50% before it is fed to the Flywheel model. Physically, this corresponds to running the engine with only half of the cylinders active.

The ride motion simulator, on the other hand, is a 6-DOF, hydraulic Stewart platform equipped with a seat and instrumentation to emulate a vehicle’s cockpit (Fig. 2b). The platform is coupled with a high-fidelity, multibody model of the HMMWV, which provides the motion signals to the platform. The platform and the computer displays mounted on it provide the human driver with motion and visual cues, which then provides the throttle, brake, and steering commands back to the vehicle model, closing the loop.

2.2. Description of the models

The flywheel model is an inertia element that takes the load torque and the engine torque as inputs and determines the pump/engine speed as the output. This speed signal is fed to the driveline model as the pump speed, as well as to the engine as the engine speed.

The idle controller is a PI controller with saturation and anti-windup, and is activated when the throttle demand from the driver falls below 11% to maintain an engine idle speed of 750 rpm.
The drivetrain model includes the torque converter, transmission, and shift logic. The torque converter model is a static model that takes pump and turbine speeds as inputs and generates pump and turbine torques according to the equations:

\[
\tau_{\text{pump}} = \left(\frac{\omega_{\text{pump}}}{\kappa} \right)^2 \text{sign}(1 - \omega_t)
\]

\[
\tau_{\text{turbine}} = 2(\omega_t)\tau_{\text{pump}}
\]

where \( \omega_t = \omega_{\text{turbine}}/\omega_{\text{pump}} \) is the speed ratio between turbine and pump speeds, \( \kappa(\omega_t) \) is a piecewise function approximating a desired capacity factor curve, and \( 2(\omega_t) \) is a piecewise linear function approximating a desired torque ratio curve. The pump side of the torque converter is connected to the engine flywheel; thus, pump speed and engine speed are the same. The turbine side of the torque converter, on the other hand, is connected to the transmission model.

The transmission model takes into account the transmission shaft inertia, stiffness, and damping, as well as the gear inefficiencies and torque losses due to fluid churning. Specifically, the speed reduction in each gear is assumed to be ideal, while the torque multiplication is assumed to be scaled by an efficiency factor. Furthermore, the torque lost due to fluid churning is modeled as variable nonlinear resistance of the form:

\[
\tau_{\text{churning loss}} = r_1(\text{gear})\omega_{\text{shaft}} + r_2(\text{gear})\omega_{\text{shaft}}^2
\]

where \( r_1 \) and \( r_2 \) are coefficients that change depending on the gear.

The inputs to the shift logic, the final element in the drivetrain model, are the transmission output shaft speed and the throttle demanded by the driver. The simple chart shown in Fig. 3 is used to determine if a shift is to be initiated. The solid and dashed lines in Fig. 3 indicate upshift and downshift thresholds, respectively. Note that this chart is only a crude approximation of a real shift map, but it is employed here for simplicity.

Finally, the vehicle model is a 3D, high-fidelity, multibody model of an HMMWV developed by TARDEC.
2.3. Description of the communication architecture

This work partially adopts the event-based framework proposed by Xi and Tarn [33] and further developed by Elhajj et al. [38]. The premise of this framework is to find a variable s, called “the event reference,” that is a monotonically increasing function of time and is the only independent variable for all the signals in the system. Then, under certain conditions, the event-based framework may have guaranteed stability, event-transparency, and event-synchronization independent of the time delay.

This work does not fully implement the event-based framework described in [33,38], because there does not exist a non-time-based event variable s with respect to which all signals can be referenced. Rather, this work adopts an event-based communication architecture only, primarily because of its ease of practical implementation. Due to the lack of a non-time-based event variable s that serves as a reference for all the other signals, there are no stability or transparency guarantees associated with the adopted architecture.

In the adopted framework, the TARDEC setup acts as the client and the UM setup as the server. The client sends an updated transmission speed and throttle signal at a frequency of 20 Hz using the User Datagram Protocol (UDP), regardless of whether it receives a response. The use of UDP implies that packets may arrive out of order or go missing, but it also reduces the communication delay, which is more critical for the purposes of ID-HILS. The server, on the other hand, only responds to the packets it receives; i.e., it sends an updated transmission torque only when it receives a packet from the client. All packets are time-stamped, and only the most recent ones are used, ignoring older packets if they arrive later than a more recent packet. This method ensures that both sides respond to the most up-to-date signals available.

An increase in the network delay is simulated by bouncing the packets between TARDEC and UM multiple times before processing them. This way, the delay can be increased artificially in multiples of the nominal delay. For example, if the nominal round-trip delay between TARDEC and UM is approximately 25 ms, an effective network delay of 125 ms can be simulated by introducing four additional round trips. Using the Internet itself to increase the delay, the jitter and loss statistics would also scale; however, this might not correspond exactly to the characteristics of a longer distance connection.

2.4. Description of the network characteristics

To characterize the Internet quality of service between UM and TARDEC, a series of experiments were conducted in which packets ranging from 64 bytes to 1024 bytes were exchanged between a computer at UM and a computer at TARDEC on different days and at different times of day using the UDP/IP protocol. A typical result for round trip time delay vs. time of day obtained from one those experiments is shown in Fig. 4. The figure clearly shows a multi-modal character in the sense that some packets experience a delay around 25 ms, and some around 350 ms (spikes), whereas others are dropped (shown as zero delay in the figure). A packet is considered dropped in this case if it does not arrive within 1 s. Table 1 provides some statistics of the results shown in Fig. 4. The percentages of spikes and drops being low and the variation in the dominant mode around 25 ms being small make this network particularly suitable for the purposes of this study.

3. ID-HIL experiments

3.1. Description of experiments

The stability and safety of the setup were tested experimentally with flat and hilly terrains and with accelerations and decelerations ranging from mild (ramp inputs with small slopes) to aggressive (step inputs with various amplitudes). After successful completion of these tests, full-scale experiments were implemented.

The full-scale experiments were performed with two different terrain models simulating two different closed courses at Aberdeen. The first one is the Munson Standard Fuel Course, which is a part gravel, part paved 1.67 miles long closed loop with a grade ranging from −15% to 30%. The second one is the Churchville B course, which is an all gravel 3.7 miles long close loop with moguls and a grade ranging from −23% to 29%. This course also contains four stop signs.

Four drivers were used in the experiments. Each driver drove on both courses twice; once with the nominal 25 ms average delay, and once with an increased average delay of 125 ms. The drivers were instructed to maintain an average speed of 20 mph and 15 mph on the Munson and Churchville courses, respectively.

3.2. Experimental results

Figs. 5 and 6 show example results from one of the drivers for all experiments (Munson and Churchville with nominal and increased delay). The results are similar for the other three drivers in the sense that the experiments were stable even with the increased delay and the drivers were able to maintain the desired average speeds on both courses.

Once stability and robustness are established, the next major question of interest is transparency, i.e., how close the dynamics of the system are to the ideal case in which the two HILS setups are integrated directly, instead of over the Internet. According to their evaluations, the drivers did not perceive any difference between the nominal and increased delay cases; however, these evaluations are subjective. An objective evaluation of the transparency of the setup is desired, but since a direct coupling of the two hardware components is not feasible, a baseline experiment to which other experiments can be compared cannot be performed. This prevents a rigorous transparency analysis of the experiments with the two hardware components. Therefore, in the remainder of the paper, a model-based study is presented to objectively evaluate the transparency of this setup.
where \( b_a \) is the aerodynamic drag coefficient, \( J_{\text{wheel}} \) is wheel slip, \( F_z \) is the normal tire force, \( \mu \) is the friction coefficient, \( \mu_{\text{max}} \) is the wheel slip for force saturation; 

\[
 f_{\text{slip}} = \frac{F_z \mu}{\mu_{\text{max}}} \begin{cases} \lambda_x < \lambda_{\text{slip}} & \lambda_x \geq \lambda_{\text{slip}} \end{cases}
\]

(4)

where \( \lambda_x = (\dot{\theta}_{\text{wheel}} - \dot{x})/x \) is wheel slip, \( F_z \) is the normal tire force, \( \mu \) is the friction coefficient, \( \mu_{\text{max}} \) is wheel slip for force saturation; 

\[
 f_{\text{aero}} = \frac{1}{2} A \rho \text{air} C_d |x| x
\]

(5)

where \( A \) is the vehicle frontal area, \( \rho_{\text{air}} \) is the air density, \( C_d \) is the aerodynamic drag coefficient, and \( x \) is vehicle velocity; 

\[
 f_{\text{rolling}} = \text{sign}(r_{\text{wheel}} \dot{x}) \left( a_0 + a_1 F_z + a_2 F_z^2 + a_3 \right)
\]

(6)

where \( a_i \) are empirical coefficients, and \( P \) is the tire pressure; and 

\[
 \tau_{\text{brake}} = (b_2 \dot{x} + \text{sign}(\dot{x}) F_{\text{Coulomb}}) c_{\text{brake}}
\]

(7)

where \( b_2 \) is viscous damping coefficient, \( F_{\text{Coulomb}} \) is the static Coulomb force, and \( c_{\text{brake}} \) is the brake command from the driver.

### 4.2. Modeling the hardware components

Two hardware components are necessary to model for the purposes of this part of the study: the driver and the engine. The ride motion simulator is merely a display device that provides the driver with motion cues and thus is not considered in this section. The actual Internet is used in these simulations and is, therefore, not modeled.

The driver model is a PI controller with saturation and antiwindup (Fig. 7). It takes the difference between the desired and actual vehicle velocities as input and generates an output within the interval [−1, 1], positive values corresponding to throttle and negative values to a brake command.

A map-based engine model is considered as shown in Fig. 8. The map is a static map obtained experimentally from the physical 6L V8 diesel engine employed in the UM setup. The input to the map is the fuel rate and the engine speed, and the output is the engine torque. A gain is used to scale the engine torque and simulate engines with more or less power. Fig. 9 shows the engine map.

The fuel rate is given by the fuel controller, which fulfills two purposes. First, it implements the turbo lag as a first order system of the form \( k [[(\dot{z} + 1) \text{ input to the turbo lag is the difference between the maximum fuel rate for the given engine RPM and the naturally aspirated fuel rate, both determined experimentally. The output of the turbo lag is added to the naturally aspirated fuel rate. This sum is then compared to the maximum possible fuel rate for the given engine RPM and throttle, and the minimum of the two is taken as the unadjusted fuel rate. Second, the fuel controller monitors the unadjusted fuel rate and adjusts it if the engine speed falls below 650 rpm or exceeds the maximum rated speed of 3300 rpm to bring the speed back to the desired operating region.

The engine model is validated using experimental data obtained from engine-in-the-loop experiments with the simplified vehicle and driver models presented above. In these experiments, the driver model is given an FTP75 drive cycle to follow. Fig. 10 compares the torque obtained with the model to the experimental data. Fig. 11 shows the error in torque. A root-mean-square value of 64.7 Nm is obtained for the torque error, which is approximately 8.5% of the rated torque of the engine.
5. Model-based evaluation of transparency

The HILS model described in the previous section is utilized in a series of simulation studies in this section to evaluate the transparency of the ID-HILS setup. First, the driver model is given an FTP75 drive cycle to follow, and the model is simulated for the ideally coupled case, i.e., in a single computational environment without the event-based framework or Internet. This simulation serves as the baseline that is necessary for the transparency analysis.

Next, the models corresponding to TARDEC and UM sides are put into two different computational environments in the remote and local locations, respectively, and coupled through the event-based framework over the Internet. The same driving scenario (FTP75) is repeated for the nominal delay, as well as the increased delay, which is again obtained by rebounding packets four times between the local and remote computers before processing them.

The simulation results are summarized in Figs. 12–14 for the variables that are communicated over the Internet; namely, throttle, shaft speed, and shaft torque. Two important conclusions can be drawn from these figures.

First, qualitatively, all the curves are almost the same as their corresponding baseline cases, suggesting that the integration over the Internet does not affect the dynamics significantly. Quantitatively, the relative $l^2$ norm of error in the throttle, shaft speed, and shaft torque signals are 2.40%, 0.19%, and 5.44%, respectively, for the nominal delay condition. For the increased delay condition, the corresponding values are 5.77%, 0.46%, and 11.46%, respectively. In light of this evidence, it can be concluded that the ID-HILS setup is transparent for the signals and delay conditions considered.

Second, it is interesting to note the differences in the relative $l^2$ norms of error for the three signals considered. These differences show that the different aspects of the system are affected to different extents by the integration over the Internet. This is a result of the different bandwidths of the subsystems comprising the system, and indicates that transparency is not necessarily a property of the
Finally, it is also worth noting that the bandwidth of a human driver is expected to be lower than the bandwidth of the PI-controller-based driver model employed in this part of the study, since the driver model aims to correct for the smallest errors in the vehicle speed. Thus, the experiments may actually appear more transparent to a human driver than what is predicted by the driver model.

6. Summary and conclusions

Integrating HIL setups over the Internet presents itself as an important enabler of concurrent and geographically dispersed systems engineering. It also presents unique challenges in terms of maintaining stability, robustness, and transparency. Even though the literature developed numerous methods to address those challenges, the application of those methods within the automotive powertrain application area has been limited to the observer-based approach. This paper, for the first time, investigated the possibility of using an event-based framework within the powertrain application domain to seek an observer-free solution. The original contributions of the paper can be summarized as follows:

1. The adoption of a variation of the event-based framework developed in [33,38] into the ID-HILS framework in the automotive application domain to enable, for the first time, an observer-free ID-HILS framework for a powertrain application.

2. Establishment of the transparency of the developed ID-HILS application through a model-based simulation study, thereby showing that the adopted event-based framework can still be successfully applied despite that the assumptions critical for the stability and transparency proofs in [33,38] may not hold.

3. Establishment of the output-signal-dependent nature of transparency through the model-based simulation study.

Future work can focus on identifying the fundamental limitations of an observer-free solution to ID-HILS and investigating how to push those limits further. It is also important to adopt and develop other communication methods, such as the passivity-based framework, within the context of automotive powertrains.

Acknowledgments

This work was supported by a grant from the ILIR program at TARDEC to Dr. Hosam K. Fathy. The authors gratefully acknowledge this support, and would also like to thank Fernando Tavares and Rajit Johri for their help with the experiments.

References


