

Robust Iterative Learning for Collaborative Road Profile Estimation and Active Suspension Control in Connected Vehicles^{*}

Harsh Modi^{*} Mohammad R Hajidavalloo^{**} Zhaojian Li^{***}
Minghui Zheng^{****}

^{*} *Texas A&M University, College Station 77840, TX, USA (e-mail: harsh.modi@tamu.edu).*

^{**} *Michigan State University, East Lansing, MI 48824, USA (e-mail: hajidava@msu.edu)*

^{***} *Michigan State University, East Lansing, MI 48824, USA (e-mail: lizhaoj1@egr.msu.edu)*

^{****} *Texas A&M University, College Station 77840, TX (e-mail: mhzheng@tamu.edu).*

Abstract: This paper presents the development of a novel collaborative road profile estimation and active suspension control framework in connected vehicles, where participating vehicles iteratively refine the road profile estimation and enhance suspension control performance through an iterative learning scheme. Specifically, we develop a robust iterative learning approach to tackle the heterogeneity and model uncertainties in participating vehicles, which are important for practical implementations. In addition, the framework can be adopted as an add-on system to augment existing suspension control schemes. Comprehensive simulations are performed to confirm the effectiveness of the proposed framework.

Keywords: Iterative Learning Control (ILC), Disturbance Observer (DOB), Active Suspension, Connected Vehicles, Disturbance Estimation, Road Profile Estimation

1. INTRODUCTION

There is a growing interest in utilizing road profile information to enhance suspension control with improved comfort and safety [Yu et al. (2023); Hajidavalloo et al. (2022)]. In addition, the estimated road profile information can also be used for optimized budget allocation for pavement maintenance [Peraka and Biligiri (2020)]. Traditionally, the road profile estimation process involves specialized sensors such as contact-based [Healey et al. (1977), Doumiati et al. (2011)] and laser-based sensors [McCann and Nguyen (2007), Ni et al. (2020)], which are costly to acquire and maintain and can only provide limited coverage [Li et al. (2017)].

On the other hand, modern vehicles are equipped with a multitude of sensors [Massaro et al. (2016)], which can be readily integrated with advanced machine learning and communication telematics for efficient road data crowdsourcing [Ma et al. (2020); Whaiduzzaman et al. (2014)]. The crowdsourced data has been proven effective in accurately estimating road and traffic information such as real-time traffic data (e.g. Google Maps, Waze). Similarly, with advancements in onboard vehicular sensing and communication, the process of estimating the road profile can also take advantage of the crowdsourced data. Specifically, with multiple participating vehicles estimating and sharing the information on the same road segment, vehicle-specific

biases and errors can be tackled and mitigated. Also, the widespread road profile estimations can be done in a very cost-effective manner.

The onboard sensor-based road profile estimation has been done in some studies. Frej et al. (2023) utilized H-infinity based observer and onboard sensors to estimate the road profile in the passive suspension vehicles. Tudón-Martínez et al. (2015) utilizes the Q-parametrization approach to estimate the road profile for semi-active suspension systems. Göhrle et al. (2014) utilizes the onboard sensor capable of measuring the road height profile in front of the vehicle to control the active suspension force. Song and Wang (2020) utilized model predictive control to predict the road profile using the lead vehicle preview. Some other studies used methodologies such as observer and dynamic response of the vehicle [Hassen et al. (2019), Li et al. (2019), Rath et al. (2014)]. However, all of these research methods were focused on using only one vehicle, which is prone to inaccuracies due to vehicle-specific characteristics.

Some studies have worked on using multiple vehicles and iterative learning for estimating the road profile [Chen et al. (2022), Gao et al. (2020)]. However, they are focused on passive suspension systems, which do not have closed-loop feedback. The learning-based approach with a closed-loop active suspension system can estimate the road profiles more accurately as the closed-loop feedback inherently ensures the correction of estimation errors. This kind of approach is used in other domains such as estimating

^{*} This work is supported by the U.S. National Science Foundation (Grants: No. 2030375, 2030411, and 2422579)

Fig. 1. Overall learning schematic along with a quarter car model used for one of the agents

the disturbance in UAVs and improving their trajectory tracking [Chen et al. (2020), Zheng et al. (2020)]. In this study, we utilize the disturbance observer along with iterative learning control for estimating the road profile. To the best of our knowledge, this is the first attempt at utilizing closed-loop active suspension with learning among multiple vehicles to estimate the road profiles.

In this research, the learning is among the vehicles with different dynamics. The difference in dynamics can arise either from different type of vehicles used or from the differences in payload, tire pressure, suspension coefficients, etc. within the same vehicle type. The learning framework is designed to account for these differences. Also, some vehicle parameters may change with time, hence the learning framework is designed to be robust enough to account for these variations. The basic outline of the learning framework is as follows:

- (1) Each vehicle equipped with active suspension estimates the road profile on its own using the disturbance observer (DOB).
- (2) The vehicle also receives anonymous data from the previously passed vehicle. This data includes information about its nominal dynamics model and its errors.
- (3) Based on this data, the learning filters generate a learning signal, which is added to the road profile estimate from the DOB, improving the estimation accuracy.

The remaining paper is organized as follows: In section 2, we will establish the theoretical learning framework. In section 3, we will present the numerical studies and we will conclude the article in section 4.

2. LEARNING FRAMEWORK

The learning framework is implemented in a cascaded format, i.e. an agent $\#(j)$ learns from the information of agent $\#(j-1)$ and agent $\#(j)$ passes on the data to be used by agent $\#(j+1)$. Fig. 1 shows the overall learning information flow among different agents (vehicles). They share information such as their own learning signal, the sprung mass displacement, and their dynamics to a shared database. We use a quarter-car model for developing the learning framework. Fig. 1 also shows a detailed version of this quarter car model for one of the agents.

In Fig. 1, m_s is a sprung mass, m_{us} is an unsprung mass, k_s and k_{us} are suspension stiffness and tire stiffness coefficients respectively, and C_s and C_{us} are spring and tire damping coefficients respectively. The Active suspension force is described as F_a . z_s , z_{us} , and z_r are sprung mass displacement, unsprung mass displacement, and road profile displacement respectively. It is important to note that we treat the road profile z_r as a function of time instead of a function of space. With the speed of the vehicle readily available in the onboard sensing, the road profile estimation can be converted into a spatial domain. In this study, we will consider the road profile as a dis-

turbance. Hence, we subtract the equivalent disturbance estimate from the control signal generated by the baseline controller. This research aims to accurately estimate this disturbance. To establish a disturbance estimation process, let us first describe the governing equations of this quarter-car suspension system:

$$\begin{aligned} m_s \ddot{z}_s &= F_a + C_s(\dot{z}_{us} - \dot{z}_s) + k_s(z_{us} - z_s) \\ m_{us} \ddot{z}_{us} &= -F_a - C_s(\dot{z}_{us} - \dot{z}_s) - k_s(z_{us} - z_s) \\ &\quad + C_{us}(\dot{z}_r - \dot{z}_{us}) + k_{us}(z_r - z_{us}) \end{aligned} \quad (1)$$

The systems of Eq. (1) have two independent inputs: 1. active suspension force (F_a), and 2. the road profile (z_r). As we want to minimize the sprung mass displacement (z_s), we consider z_s as the output of the system. In this study, we assume can measure z_s using the onboard sensors [Sisi et al. (2024)]. Using Eq. (1) and assuming the suspension operates in a linear regime, we can derive the transfer functions relating the output (z_s) to each of the inputs (F_a and z_r) respectively as:

$$\begin{aligned} P(1) &= z_s / F_a \\ &= \frac{m_{us} \cdot s^2 + c_{us} \cdot s + k_{us}}{m_s m_{us} s^4 + (c_s m_s + c_s m_{us} + c_{us} m_s) s^3 \\ &\quad + (k_s m_s + k_s m_{us} + k_{us} m_s + c_s c_{us}) s^2 \\ &\quad + (c_s k_{us} + c_{us} k_s) s + k_s k_{us}} \end{aligned} \quad (2)$$

$$\begin{aligned} P(2) &= z_s / \dot{z}_r \\ &= \frac{(k_s + c_s s)(k_{us} + c_{us} s)}{m_s m_{us} s^5 + (c_s m_s + c_s m_{us} + c_{us} m_s) s^4 \\ &\quad + (k_s m_s + k_s m_{us} + k_{us} m_s + c_s c_{us}) s^3 \\ &\quad + (c_s k_{us} + c_{us} k_s) s^2 + k_s k_{us} s} \end{aligned} \quad (3)$$

Fig. 2. (a) Original block diagram of the system (b) Equivalent block diagram after block diagram manipulation

The transfer function system derived in Eq. (2) and Eq. (3) is arranged in block diagram form in Fig. 2 (a). In this block diagram, C represents a baseline controller, e represents an error in trajectory tracking (i.e. $-z_s$), and s is a derivative transfer function. The learning framework is developed such that the baseline controller C remains untouched by the framework. Hence, the learning framework can be applied to any active suspension system regardless of what controller is used in the system.

As we consider using a DOB for a primary estimation of the road profile, we need to treat the road profile as a disturbance. Hence, we move the signal corresponding to z_r upstream of $P(1)$ using the block diagram manipulation as shown in Fig. 2 (b). Considering this, the equivalent disturbance based on the road profile can be given as:

$$z_{req} = d = \delta\{z_r\} \quad (4)$$

where

$$\delta = sP^{-1}(1)P(2) \quad (5)$$

As z_{req} is being added between the baseline controller and the plant, it can be considered a disturbance to the system. For simplicity, we will represent z_{req} as d in further equations and figures. The notation $\{\}$ means that the signal inside the notation is sent to a system which can be represented by the outside transfer function.

Fig. 3. Block diagram of the system along with learning framework for Agent#(j)

Fig. 3 shows a block diagram of the system with the disturbance observer and learning framework added to the system. As the learning framework establishes the learning relationship between different agents (vehicles), each agent's dynamics can be different from any other agent. In order to denote this difference, we use the index j as the subscript of each term. In Fig. 3, $M_j \approx Q_j P_j^{-1}$ is an inverse approximation of the plant $P_j(1)$, Q_j is a low pass filter, $F_{a,j}$ is an active suspension actuator force. $\hat{d}_j$ is a disturbance estimate from the DOB. We complement this estimate with the learning signal $d_{f,j}$ in order to generate the final disturbance estimate $\hat{d}_j$. This learning signal is generated using the trajectory tracking error of the previous agent (e_{j-1}), the learning signal of the previous agent ($d_{f,j-1}$), and respective to-be-designed learning filters $L_{1,j}$ and $L_{2,j}$. Similarly, the data from the current agent is passed to the next agent.

With this agent description established, we will derive the learning filters $L_{1,j}$ and $L_{2,j}$ in the following subsections such that disturbance estimation error is reduced iteratively. As we want to reduce the disturbance estimation error of agent#(j) (i.e. $e_{d,j}$) compared to the disturbance estimation error of agent#(j-1) (i.e. $e_{d,j-1}$), we will establish a relationship between $e_{d,j}$ and $e_{d,j-1}$.

2.1 Establishing relationship between $e_{d,j}$ and $e_{d,j-1}$:

Let us first define a few system parameters:

System Parameters: Based on the block diagram in Fig. 3, $G_{d,j}$ (dynamics from disturbance d_j to output $z_{s,j}$), $G_{f,j}$ (dynamics from learning signal $d_{f,j}$ to output $z_{s,j}$), and Ω_j (dynamics from disturbance d_j to DOB disturbance estimate $\hat{d}_j$) can be described by:

$$G_{d,j} = [1 - Q_j + P(1)_j(M_j + C_j)]^{-1} P(1)_j(1 - Q_j) \quad (6)$$

$$G_{f,j} = [1 - Q_j + P(1)_j(M_j + C_j)]^{-1} (-P(1)_j) \quad (7)$$

$$\Omega_j = [1 - Q_j + P(1)_j(M_j + C_j)]^{-1} \cdot (M_j + Q_j C_j) P(1)_j \quad (8)$$

Using Fig. 3 and Eq. (8), the error in disturbance estimation of agent#(j) can be given by:

$$\begin{aligned} e_{d,j} &= d_j - \hat{d}_j = d_j - (\hat{d}_j + d_{f,j}) \\ &= (1 - \Omega_j)\{d_j\} - d_{f,j} \end{aligned} \quad (9)$$

Similarly, $e_{d,j-1}$ can be described by:

$$e_{d,j-1} = (1 - \Omega_{j-1})\{d_{j-1}\} - d_{f,j-1} \quad (10)$$

Subtracting Eq. (9) with Eq. (10), we get:

$$\begin{aligned} e_{d,j} - e_{d,j-1} &= (1 - \Omega_j)\{d_j\} - (1 - \Omega_{j-1})\{d_{j-1}\} \\ &\quad - d_{f,j} + d_{f,j-1} \end{aligned} \quad (11)$$

Now, using Eq. (4), we replace d_j with a term containing d_{j-1} and rewrite Eq. (11) as:

$$\begin{aligned} e_{d,j} - e_{d,j-1} &= \left[(1 - \Omega_j) \frac{\delta_j}{\delta_{j-1}} - (1 - \Omega_{j-1}) \right] \{d_{j-1}\} \\ &\quad - d_{f,j} + d_{f,j-1} \end{aligned} \quad (12)$$

Now, Let us define a learning signal $d_{f,j}$ using the to-be-designed learning filters $L_{1,j}$ and $L_{2,j}$ as:

$$d_{f,j} = L_{1,j}\{e_{j-1}\} + L_{2,j}\{d_{f,j-1}\} \quad (13)$$

Using Eq. (13) and with some simplifications, Eq. (12) can be written as:

$$\begin{aligned} e_{d,j} - e_{d,j-1} &= \left[(1 - \Omega_j) \frac{\delta_j}{\delta_{j-1}} - (1 - \Omega_{j-1}) \right] \{d_{j-1}\} \\ &\quad - L_{1,j}\{e_{j-1}\} + (1 - L_{2,j})\{d_{f,j-1}\} \end{aligned} \quad (14)$$

Now, using Eq. (6) and Eq. (7), the displacement of the sprung mass for agent#(j-1) can be given by:

$$z_{s,j-1} = G_{d,j-1}\{d_{j-1}\} + G_{f,j-1}\{d_{f,j-1}\} \quad (15)$$

Hence, with the aim of $z_{s,j} = 0$, the error in trajectory tracking can be described as:

$$e_{j-1} = -z_{s,j-1} = -G_{d,j-1}\{d_{j-1}\} - G_{f,j-1}\{d_{f,j-1}\} \quad (16)$$

Using Eq. (16), we can simplify Eq. (14) as:

$$\begin{aligned} e_{d,j} - e_{d,j-1} &= \left[(1 - \Omega_j) \frac{\delta_j}{\delta_{j-1}} - (1 - \Omega_{j-1}) + L_{1,j} G_{d,j-1} \right] \{d_{j-1}\} \\ &\quad + (1 - L_{2,j} + L_{1,j} G_{f,j-1}) \{d_{f,j-1}\} \end{aligned} \quad (17)$$

Now, using Eq. (10), we substitute d_{j-1} in Eq. (17) and simplify the Eq. (17) as:

$$\begin{aligned} e_{d,j} &= \left[\frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{L_{1,j} G_{d,j-1}}{(1 - \Omega_{j-1})} \right] \{e_{d,j-1}\} \\ &\quad + \left[\frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{L_{1,j} G_{d,j-1}}{(1 - \Omega_{j-1})} - L_{2,j} + L_{1,j} G_{f,j-1} \right] \{d_{f,j-1}\} \end{aligned} \quad (18)$$

For simplicity, let us define new terms $T_{e1,j}$ and $T_{e2,j}$ as:

$$\begin{aligned} T_{e1,j} &= \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{L_{1,j} G_{d,j-1}}{(1 - \Omega_{j-1})} \\ T_{e2,j} &= \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{L_{1,j} G_{d,j-1}}{(1 - \Omega_{j-1})} - L_{2,j} + L_{1,j} G_{f,j-1} \end{aligned} \quad (19)$$

Using Eq. (19), the Eq. (18) can be written as:

$$e_{d,j} = T_{e1,j}\{e_{d,j-1}\} + T_{e2,j}\{d_{f,j-1}\} \quad (20)$$

Eq. (20) is the desired relationship between $e_{d,j}$ and $e_{d,j-1}$ for designing the learning filters. In subsection 2.2, we will introduce the proposed learning filters in a theorem and prove the error reduction using this relationship

2.2 Designing the learning filters

We will define a few notations to be used in the theorem. Let us define an error reduction factor $0 < \alpha < 1$, where

$$\|e_{d,j}\| = \alpha \cdot \|e_{d,j-1}\| \quad (21)$$

and let us define η_j as zero order approximation of a transfer function δ_j/δ_{j-1} . Also, for determining the learning filters for any agent, we will use the estimated plant models. These plant models may not be perfectly accurate due to modeling errors or due to variations in the model parameters. In order to quantify this uncertainty, let us define $\Delta_{1,j}$ and $\Delta_{2,j}$ where:

$$P_j(1) = (1 + \Delta_{1,j})\hat{P}_j(1) \quad (22)$$

$$P_j(2) = (1 + \Delta_{2,j})\hat{P}_j(2) \quad (23)$$

with $\hat{P}_j(1)$ and $\hat{P}_j(2)$ being the estimated plant models. $\Delta_{1,j}$ and $\Delta_{2,j}$ are small gain transfer functions to account for inaccuracies in these estimated plant models. From hereon, the hat symbol ($\hat{\cdot}$) over any transfer function indicates it is the estimated transfer function of the corresponding actual transfer function. With these notations, let us introduce the learning filters and prove the error reduction in the following theorem:

Theorem: With the learning filters

$$L_{1,j} = \hat{G}_{d,j-1}^{-1} \left[\alpha(1 - \hat{\Omega}_{j-1}) - \eta_j(1 - \hat{\Omega}_j) \right] \quad (24)$$

and

$$L_{2,j} = \alpha + L_{1,j}\hat{G}_{f,j-1} \quad (25)$$

we can achieve iterative estimation error reduction of factor $\approx \alpha$ in each iteration, where $0 < \alpha < 1$

Proof:

Using Eq. (24) and Eq. (25), the Eq. (19) can be written as:

$$\begin{aligned} T_{e1,j} &= \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{\hat{G}_{d,j-1}^{-1} [\alpha(1 - \hat{\Omega}_{j-1}) - \eta_j(1 - \hat{\Omega}_j)] G d_{j-1}}{(1 - \Omega_{j-1})} \\ T_{e2,j} &= \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \frac{\delta_j}{\delta_{j-1}} + \frac{\hat{G}_{d,j-1}^{-1} [\alpha(1 - \hat{\Omega}_{j-1}) - \eta_j(1 - \hat{\Omega}_j)] G d_{j-1}}{(1 - \Omega_{j-1})} \\ &\quad - (\alpha + L_{1,j}\hat{G}_{f,j-1}) + L_{1,j}Gf_{j-1} \end{aligned} \quad (26)$$

Now, Let us expand Eq. (26) using Eq. (6), Eq. (7), Eq. (8), Eq. (22), and Eq. (23). Also, let us assume that the low pass filter for all the systems is the same (i.e. $Q_j \approx Q$ for $\forall j$). With this, Eq. (26) can be expressed as:

$$\begin{aligned} T_{e1,j} &= \left(1 + \frac{\Delta_{1,j-1}Q}{1 + C_{j-1}P_{j-1}(1)} \right) \cdot \\ &\quad \left[\frac{(\alpha - \eta_j) \left(1 + \frac{\Delta_{1,j-1}}{1 + C_{j-1}P_{j-1}(1)} \right)}{\left(1 + \frac{\Delta_{1,j-1}Q}{1 + C_{j-1}P_{j-1}(1)} \right)} + \frac{\delta_j}{\delta_{j-1}} \frac{(1 + \Delta_{1,j-1})(1 + \Delta_{2,j})}{(1 + \Delta_{1,j})(1 + \Delta_{2,j-1})} \right] \\ T_{e2,j} &= T_{e1,j} - \eta_j - (\alpha - \eta_j) \left(1 + \frac{\Delta_{1,j-1}}{1 + C_{j-1}P_{j-1}(1)} \right) \end{aligned} \quad (27)$$

In Eq. (27), the term

$$\left(\frac{1}{1 + C_j P_j(1)} \right)$$

reflects the transfer function from reference to the trajectory tracking error for agent#(j). For a well designed baseline active suspension controller, this term will have a gain very close to zero. Additionally,

$$\left(\frac{\Delta_{1,j}}{1 + C_j P_j(1)} \right) \text{ and } \left(\frac{\Delta_{1,j}Q}{1 + C_j P_j(1)} \right)$$

will have even smaller gain, hence:

$$\left(1 + \frac{\Delta_{1,j}}{1 + C_j P_j(1)} \right) \approx 1 \text{ for } \forall j \quad (28)$$

and

$$\left(1 + \frac{\Delta_{1,j}Q}{1 + C_j P_j(1)} \right) \approx 1 \text{ for } \forall j \quad (29)$$

Also, we can assume

$$\frac{(1 + \Delta_{1,j-1})(1 + \Delta_{2,j})}{(1 + \Delta_{1,j})(1 + \Delta_{2,j-1})} \approx 1 \quad (30)$$

as $\Delta_{1,j}$ and $\Delta_{2,j}$ have a small gain for reasonably well modelled plants for $\forall j$

Using Eq. (28), Eq. (29), and Eq. (30), the Eq. (26) reduces to:

$$\begin{aligned} T_{e1,j} &\approx \alpha + \left(\frac{\delta_j}{\delta_{j-1}} - \eta_j \right) \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \\ T_{e2,j} &\approx \left(\frac{\delta_j}{\delta_{j-1}} - \eta_j \right) \frac{(1 - \Omega_j)}{(1 - \Omega_{j-1})} \end{aligned} \quad (31)$$

For a suspension system, δ_j/δ_{j-1} can be approximated as a static gain in the desired road profile frequency range. As η_j is the zero-order approximation of the transfer function δ_j/δ_{j-1} , $\delta_j/\delta_{j-1} - \eta_j \approx 0$. With this, Eq. (31) can be simplified as:

$$\begin{aligned} T_{e1,j} &\approx \alpha \\ T_{e2,j} &\approx 0 \end{aligned} \quad (32)$$

Using Eq. (32), Eq. (20) can be written as:

$$e_{d,j} \approx \alpha \{e_{d,j-1}\} \quad (33)$$

Hence, the estimation error will be reduced by a factor $\approx \alpha$ in each iteration. This is the end of the theorem proof.

3. NUMERICAL EVALUATION

Parameter	Actual Model	Nominal Model
m_s (kg)	$(2.45 + \beta \times j)$	$(2.45 + \beta \times j)f_{m_s}$
m_{us} (kg)	$(1 + \beta \times j)$	$(1 + \beta \times j)f_{m_{us}}$
k_s (N/m)	$(950 + 100\beta \times j)$	$(950 + 100\beta \times j)f_{k_s}$
k_{us} (N/m)	$(1250 + 100\beta \times j)$	$(1250 + 100\beta \times j)f_{k_{us}}$
c_s (N s/m)	$(7.5 + \beta \times j)$	$(7.5 + \beta \times j)f_{c_s}$
c_{us} (N s/m)	$(5 + \beta \times j)$	$(5 + \beta \times 7j)f_{c_{us}}$

Table 1. Simulation agent characteristics. In current simulations, $\beta = 1/15$

figures/TF1s-eps-converted-to.pdf

Fig. 4. Bode plots for the actual model transfer function from active suspension force to sprung mass displacement ($P_j(1)$) for all the agents

figures/TF2s-eps-converted-to.pdf

Fig. 5. Bode plots for the actual model transfer function from road profile to sprung mass displacement ($P_j(2)$) for all the agents

We evaluate the performance of the learning framework numerically in the MATLAB simulations. We perform the simulation over 90 dynamically different agents running on the same road profile in succession. We use a quarter-car model of the vehicle suspension system as we had illustrated in Fig. 1.

figures/road_profiles-eps-converted-to.pdf

Fig. 6. Road profiles introduced

The specific characteristics of the agents used in the simulation are described in Table 1. The definitions used to generate both the actual models and the nominal models are shown in the table. The simulator uses the actual model to simulate the response of the system, but the learning framework has access to the inaccurate nominal model only. This ensures the numerical evaluation reflects modeling uncertainties described by $\Delta_{1,j}$ and $\Delta_{2,j}$ in the theorem.

Each agent's dynamics is varied using the index multiplier j varying from 1 to 90. In our current simulations, we choose $\beta = 1/15$. Hence, for example, the actual sprung mass varies in the range of $2.45 + 1/15 \times 0 = 2.45 \text{ kg}$ to $2.45 + 1/15 \times 90 = 8.45 \text{ kg}$. We also randomize the order of j in the simulation to ensure that none of the characteristics are steadily increasing or decreasing with each successive agent. The f_x (e.g. f_{m_s} , $f_{c_{us}}$, etc.) indicates the relevant model uncertainty added to defer the nominal model from the actual model. This factor represents the modeling error either due to incorrect system identification or variations in the parameters with time (e.g. change in the payload, change in tire pressure, etc). The f_x is chosen as a random number with a mean of 1.0 and an upper bound chosen before starting the simulation. Importantly, f_x is not known to the learning framework. In our current setup, we have chosen this upper bound on uncertainty as 10% (i.e. f_x varies from 0.9 to 1.10 randomly for each agent and each parameter). Fig. 4 and Fig. 5 show the bode plots of actual model $P_j(1)$ and $P_j(2)$ respectively for all the agents.

For each of our agents, we are using a PID controller as the baseline controller. Iterating again, the learning framework does not require any specific controller type as long as the controller is stable. Table 2 shows the PID controller constants used in our simulations. Here, also, j varies from 1 to 90 in a random order. Unlike the agent plant models,

figures/sin_4_fig_5rads-eps-converted-to.pdf

Fig. 7. (a) Comparison of road profile estimations for a few agents for a sinusoidal road profile (b) Comparison of “Root Mean Square Errors” of the road profile estimations for all agents for a sinusoidal road profile

PID Controller Constant	Value
P	$(1500 + 2j)$
I	$(200 + 0.067j)$
D	$(500 + j)$

Table 2. Simulation agent controllers

we do not need to add any uncertainties to the controller as the controller is known to us.

We perform simulation over 2 different road profile scenarios as shown in Fig. 6. In scenario 1, a sinusoidal road profile of magnitude 15 *mm* and a frequency 5 *rad/s* is used. In scenario 2, we introduce a type-C road profile

[Hassen et al. (2019)]. We also perform the simulation without learning but with the disturbance observer for all the agents. This is done to compare the results of the “no learning” cases with the “learning” cases. In the following subsections, we will describe these results for each of the scenarios in detail.

3.1 Sinusoidal Road Profile

Fig. 7 shows the results from the simulation for scenario 1 with a sinusoidal road profile. Fig. 7(a) compares the road profile estimations without learning (only DOB running) and with learning. We show only a few agents’ estimates out of 90 agents to de-clutter the plots. The actual road

Fig. 8. Learning signals for a few agents for the sinusoidal road profile

profile is also shown in both figures for ease of comparison. In the "without learning" case, the estimates from all the agents are almost similar. The variations in these estimates are due to differences in the vehicle models. The important observation from this plot is that the estimates have a significant delay than the actual road profile. This is due to the disturbance observer using the outputs of the system to estimate the disturbances, which is often delayed. These estimates are also scaled down compared to the actual road profile. However, with learning, the road profile estimates gradually approach the actual road profile with each vehicle. The agent#(90) has a road profile estimate very close to the actual road profile.

Fig. 7(b) shows the "Root Mean Square Error (RMSE)" of the road profile estimation for all the agents. This plot represents the overall "closeness" of the road profile estimates to the actual road profiles. In the "without learning" cases, all the agents have almost the same RMSE of around 10.25 mm . However, in the "with learning" cases, the error decreases exponentially in each iteration. The RMSE errors converge to around 1.22 mm . The exponential curve that fits the data with 95% confidence bounds is:

$$y(x) = 8.75e^{-0.039x} + 0.273e^{0.014x} \quad (34)$$

Fig. 8 shows the learning signals generated by the learning framework for a few agents. This plot represents how the inaccuracies of the disturbance observer are corrected by the iterative learning process. The learning signal for the agent#(1) is a constant 0 as there is no previous agent to learn from. From agent#(2) onwards, the learning signal tries to compensate for the errors of the disturbance observer. The learning signals slowly converge toward the learning signal of the last agent (i.e. agent#(90)).

3.2 Type-C Road Profile

Fig. 9 shows the results from the simulation for scenario 2 with a type-C road profile. Similar to scenario 1, the

agents with learning are able to iteratively improve the road profile estimation. Fig. 9 (a) shows the comparison of road profile estimates for a few agents in "without learning" and "with learning" cases. Without learning, the road profile estimates are delayed. They are also missing the higher frequency components of the road profile and are only estimating the slowly varying signal. In the learning cases, the agents gradually improve their estimations in each iteration. Also, the estimation of high-frequency components is gradually improving with each agent. Agent#(90) is able to capture the actual road profile with much more detail and with almost negligible delay. For ease of visualization, Fig. 10 shows the road profile estimation errors for agent#(1) (i.e. the first agent) and agent#(90) (i.e. the last agent) in the learning case. It can be clearly confirmed that the estimation errors in the last agent are negligible compared to the first agent.

Fig. 9 (b) shows the RMSE of the road profile estimation errors. Here also, in the "without learning" case, the errors are almost constant at about 4 mm . In the learning cases, the error exponentially decreases and converges to around 0.8 mm . The exponential curve that fits the data with 95% confidence bounds is:

$$y(x) = 3.38e^{-0.039x} + 0.28e^{0.012x} \quad (35)$$

Fig. 11 shows the learning signals for a few agents in this scenario. Again, the learning signal for the agent#(1) is a constant 0 as there is no previous agent to learn from. The high-frequency components in the learning signals indicate the learning process is able to capture the frequencies that otherwise would have been missed out without learning. This plot explains why the road profile estimation in Fig. 9 (a) is able to estimate the higher frequency changes in the road profile. In this scenario also, The learning signals change very rapidly for initial agents but then converge to almost the same signal for the final few agents.

4. CONCLUSION

This paper established the learning framework for estimating the road profile using disturbance observer and iterative learning control using the onboard sensors. The framework utilizes multiple dynamically different active suspension vehicles to estimate the road profile accurately simultaneously increasing the passenger comfort in each vehicle. The learning framework can incorporate modeling uncertainties to iteratively reduce road profile estimation errors. The framework has been evaluated in robust simulations and they confirm the effectiveness of the learning process.

REFERENCES

- Chen, Z., Hajidavalloo, M.R., Li, Z., and Zheng, M. (2022). A cascaded learning framework for road profile estimation using multiple heterogeneous vehicles. *Journal of Dynamic Systems, Measurement, and Control*, 144(10), 104501.
- Chen, Z., Liang, X., and Zheng, M. (2020). Knowledge transfer between different uavs for trajectory tracking. *IEEE Robotics and Automation Letters*, 5(3), 4939–4946.
- Doumiati, M., Victorino, A., Charara, A., and Lechner, D. (2011). Estimation of road profile for vehicle dynamics

figures/cc_4_fig_trial-eps-converted-to.pdf

Fig. 9. (a) Comparison of road profile estimations for a few agents for a type-C road profile (b) Comparison of “Root Mean Square Errors” of the road profile estimations for all agents for a type-C road profile

- motion: experimental validation. In *Proceedings of the 2011 American control conference*, 5237–5242. IEEE.
- Frej, G.B.H., Moreau, X., Guridis, R., Benine-Neto, A., and Hernette, V. (2023). Road profile estimation from onboard sensor measurements through a combination of h-infinity and unknown inputs observers. In *2023 31st Mediterranean Conference on Control and Automation (MED)*, 113–118. doi: 10.1109/MED59994.2023.10185895.
- Gao, H., Li, Z., and Wang, Y. (2020). Privacy-preserved collaborative estimation for networked vehicles with application to road anomaly detection. *arXiv preprint arXiv:2008.02928*.
- Göhrle, C., Schindler, A., Wagner, A., and Sawodny, O. (2014). Road profile estimation and preview control for low-bandwidth active suspension systems. *IEEE/ASME Transactions on Mechatronics*, 20(5), 2299–2310.
- Hajidavalloo, M.R., Li, Z., Xia, X., Louati, A., Zheng, M., and Zhuang, W. (2022). Cloud-assisted collaborative road information discovery with gaussian process: Application to road profile estimation. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 23951–23962. doi:10.1109/TITS.2022.3194093.
- Hassen, D.B., Miladi, M., Abbes, M.S., Baslamisli, S.C., Chaari, F., and Haddar, M. (2019). Road profile estimation using the dynamic responses of the full vehicle model. *Applied Acoustics*, 147, 87–99.



Fig. 10. Comparison of road profile estimation errors between the first and the last agent with learning for type-C road profile



Fig. 11. Learning signals for a few agents for the type-C road profile

- Healey, A.J., Nathman, E., and Smith, C.C. (1977). An Analytical and Experimental Study of Automobile Dynamics With Random Roadway Inputs. *Journal of Dynamic Systems, Measurement, and Control*, 99(4), 284–292. doi:10.1115/1.3427121. URL <https://doi.org/10.1115/1.3427121>.
- Li, Z., Kolmanovsky, I.V., Atkins, E.M., Lu, J., Filev, D.P., and Bai, Y. (2017). Road disturbance estimation and cloud-aided comfort-based route planning. *IEEE Transactions on Cybernetics*, 47(11), 3879–3891. doi: 10.1109/TCYB.2016.2587673.
- Li, Z., Zheng, M., and Zhang, H. (2019). Optimization-based unknown input observer for road profile estimation with experimental validation on a suspension station. In *2019 American Control Conference (ACC)*, 3829–3834. IEEE.
- Ma, Y., Wang, Z., Yang, H., and Yang, L. (2020). Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2), 315–329. doi: 10.1109/JAS.2020.1003021.
- Massaro, E., Ahn, C., Ratti, C., Santi, P., Stahlmann, R., Lamprecht, A., Roehder, M., and Huber, M. (2016). The car as an ambient sensing platform [point of view]. *Proceedings of the IEEE*, 105(1), 3–7.
- McCann, R. and Nguyen, S. (2007). System identification for a model-based observer of a road roughness profiler. In *2007 IEEE Region 5 Technical Conference*, 336–343. IEEE.
- Ni, T., Li, W., Zhao, D., and Kong, Z. (2020). Road profile estimation using a 3d sensor and intelligent vehicle. *Sensors*, 20(13), 3676.
- Peraka, N.S.P. and Biligiri, K.P. (2020). Pavement asset management systems and technologies: A review. *Automation in Construction*, 119, 103336.
- Rath, J.J., Veluvolu, K.C., and Defoort, M. (2014). Estimation of road profile for suspension systems using adaptive super-twisting observer. In *2014 European Control Conference (ECC)*, 1675–1680. IEEE.
- Sisi, Z.A., Mirzaei, M., and Rafatnia, S. (2024). Estimation of vehicle suspension dynamics with data fusion for correcting measurement errors. *Measurement*, 114438.
- Song, S. and Wang, J. (2020). Incremental model predictive control of active suspensions with estimated road preview information from a lead vehicle. *Journal of Dynamic Systems, Measurement, and Control*, 142(12), 121004.
- Tudón-Martínez, J.C., Fergani, S., Sename, O., Martínez, J.J., Morales-Menendez, R., and Dugard, L. (2015). Adaptive road profile estimation in semiactive car suspensions. *IEEE Transactions on Control Systems Technology*, 23(6), 2293–2305. doi: 10.1109/TCST.2015.2413937.
- Whaiduzzaman, M., Sookhak, M., Gani, A., and Buyya, R. (2014). A survey on vehicular cloud computing. *Journal of Network and Computer applications*, 40, 325–344.
- Yu, M., Evangelou, S., and Dini, D. (2023). Advances in active suspension systems for road vehicles. *Engineering*.
- Zheng, M., Lyu, X., Liang, X., and Zhang, F. (2020). A generalized design method for learning-based disturbance observer. *IEEE/ASME Transactions on Mechatronics*, 26(1), 45–54.