

Real-Time Preview Control of Active Suspensions Under Actuation Limits via Learned MPC Surrogates and Online Identification

Zuqiang Yuan

Ben-Gurion University of the Negev

Muxin Weng

Guangdong Technion-Israel Institute of Technology

Mingyi Liu

`mingyi.liu@gtiit.edu.cn`

Guangdong Technion-Israel Institute of Technology

Research Article

Keywords: Vibration suppression, MPC, Imitation learning, Real-time deployment

Posted Date: September 18th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10923316/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Real-Time Preview Control of Active Suspensions Under Actuation Limits via Learned MPC Surrogates and Online Identification

Abstract — Vehicle active suspensions can significantly improve ride comfort, but their practical deployment is constrained by limited actuator authority and real-time computational requirements. This paper presents a real-time, preview-enabled, constraint-aware control framework for active suspension systems operating under actuation saturation. The proposed architecture integrates road-preview model predictive control (MPC), a learned surrogate controller, and online system identification to achieve high-performance vibration suppression under changing operating conditions. First, a road-preview MPC formulation with explicit actuator constraints is developed to investigate the interaction between preview information and actuator saturation. Results show that preview provides measurable benefits only when sufficient actuator authority is available, achieving up to 12% lower vibration acceleration than a feedforward PD baseline across ISO road profiles A–H. This analysis reveals fundamental limitations on the effectiveness of preview control in constraint-limited systems. To enable real-time deployment, a neural-network surrogate is trained to approximate the constrained MPC policy while preserving its saturation characteristics. The learned controller reduces online computation by approximately $80\times$ while maintaining comparable performance and constraint satisfaction. To address model mismatch caused by variations in sprung mass, an online identification module continuously updates the control model. The adaptive framework achieves an additional 32% reduction in vibration acceleration compared with a fixed-model controller. Validation on quarter- and half-vehicle suspension models demonstrates consistent performance improvements and real-time feasibility.

Index Terms — Vibration suppression, MPC, Imitation learning, Real-time deployment

I. INTRODUCTION

Vibration suppression reduces unwanted oscillations and improves the performance, safety, reliability, and durability of vehicle, machines, civil structures, *etc.* Early approaches rely on inherent material properties and mechanical designs, such as rubber pads and mechanical springs, to attenuate vibrations without the need for external power [1]. Over time, the field progressed with the advent of semi-active and active vibration control in the 20th century, each tailored to specific applications [2] [3]. Active methods utilize external energy, along with sensors and actuators, to dynamically counteract vibrations in real-time [4]. Semi-active systems serve as an intermediary between passive and active approaches, offering adaptability by modulating damping or stiffness [5], while requiring only intermittent power input rather than continuous energy supply.

Vehicle active suspension systems, which can dynamically adjust to road conditions in real-time, play a vital role in improving safety, ride comfort, road handling, and protecting vehicle components. However, until now, active suspension systems are used only on a few of the luxury models. One of the biggest challenges for massive adoption is the huge cost of building an actuator with a large force output and small-time delay. In addition, large energy consumption comes with this design. As a result, the majority of the vehicles nowadays are not equipped with active suspension control. In this project, we propose to use a bio-inspired control strategy, which leverages the merits of MPC-based optimal control, deep learning, and real-time system identification to improve vibration suppression performance in a wide range of excitation frequency and amplitudes, which can potentially lead to the massive adoption of vehicle active suspension. The key contribution is a constraint-faithful surrogate that reproduces MPC's active-set/saturation structure while enabling preview control at real-time rates under online-identified model updates. The contributions are summarized as follows:

1. With preview and MPC control, the algorithm can achieve global optimal control and reduce the RMS vibration acceleration by 12% compared with feedforward PD control under the actuator saturation limitation.
2. The imitation learning can improve computational speed by a factor of 80, while achieving nondifferentiable MPC saturation-preserving and constraint-aware performance, and potentially enables real-time deployment.
3. With real-time system identification, the deep neural network can consider the changing of the system model and take actions accordingly, and the RMS vibration acceleration can be reduced by 32% compared with the baseline model.
4. This paper provides technical insights for dealing with actuation saturation and preview window length, which can be guidelines when designing the preview-based MPC controller.

The rest of the paper is organized as follows. In *Section 2*, the literature review is carried out. *Section 3* is the method part. First, the problem formulation is introduced. Control system design is built based on MPC with preview. Deep learning is used to fit the control function. In the meantime, real-time system identification is used to deal with model with a changing dynamic model. *Section 4* is simulation results and discussion. Conclusions are drawn and presented in *Section 5*.

II. LITERATURE REVIEW

A. Active vibration suspension control

Active vibration suspension control research mainly started back in 1970th and underwent four stages of development. In the first stage, skyhook and ground-hook controls are proposed to realize ride comfort [6] and road handling performances [7]. Later, with increased on-board computing, optimal state space control (such as LQR, Linear Quadratic Regulator) became practical, and can minimize a quadratic cost, balancing body acceleration, suspension deflection, and actuator effort [8]. Around 2000th, robust and nonlinear control, such as H_∞ , sliding-mode, and adaptive controllers, are proposed to reject disturbance with strong robustness to uncertainties [9]. Around 2010th, Model Predictive Control (MPC) emerges to leverage a rolling horizon optimization to handle constraints (actuator limits, tire contact) [10], however, a much more powerful computer is required. Fuzzy logic and neural network controllers learn nonlinear suspension behavior or driver preferences [11]. Active Disturbance Rejection Control (ADRC) estimates and cancels disturbances in real time without an exact model [12] and becomes a very powerful upgrade since the invention of PID.

B. MPC with preview for active suspension

Since the early adoption of MPC with preview technology [13], the development of the preview technology enables a rich family of controllers for better ride comfort and road handling [14] [15]. Papadimitrakakis *et al.* proposed to use RBF (Radial Basis Function Network) to efficiently approximate the nonlinear behavior of the suspension system and improved the chassis heave acceleration performance by 28% compared with the baseline [16]. Achnib *et al.* proposed to use discrete time preview robust controller design method for rejecting road disturbance for an automotive semi-active suspension. The design is based on the use of a preview feedforward filter in order to limit the control force damper demand and improve its reliability while helping the rejection of the disturbance road effect. The implementation of this approach on a real scaled-vehicle test bench proves the accuracy and robustness of the proposed approach [17]. Kim *et al.* present a model predictive control of a semi-active suspension with a shift delay compensation using preview road information for both ride comfort and road handling performance. However, the authors report the MPC carries a sizeable computational load and the “shift time delays” prevent ideal real-time applications. They have carried out a successful implementation of the proposed algorithms into an actual vehicle, where the real-world performance of the proposed algorithms has been assessed [18]. Dong *et al.* propose damping control of speed-dependent active suspensions. The control laws are determined offline using multi-parameter linear programming, simplifying online implementation to searching for optimal solutions in a lookup table.

Simulation results demonstrate a significant improvement in active suspension control under changing vehicle speeds compared to passive control [19].

Literature search finds the effectiveness of MPC control with preview information. However, the computation burden will hence impair system performance in real-time implementation [15]. To address this problem, [20] proposed to use a Deep Neural Network to fit the MPC control policy and reduce the execution time.

C. Algorithms deal with saturation in actuators

In a real system, saturation in the actuator is a common problem. In order to improve the system performance with a saturated actuator, Lin *et al.* propose a performance-based activation mechanism for anti-windup compensation. The signals representing the transient performance of the system are passed through a static performance compensator and then injected into the traditional anti-windup activation module. Simulation results show that, compared with the existing methods, their proposed approach has the ability to acquire a significantly larger estimate of the domain of attraction, a significantly tighter estimate of the nonlinear gain, and significantly higher transient quality in reference tracking [21].

Regionless e-MPC and classical e-MPC can save the time for MPC optimization and potentially enables real-time implementation [Regionless Explicit Model Predictive Control of Active Suspension Systems with Preview], however they become increasingly difficult to construct as the state and preview dimensions grow.

D. Bio-inspired real-time control

In nature, various birds can achieve robust vibration suppression control with seemingly little effort. From the literature, birds are observed to be able to suppress vibration base position input up to 10 cm amplitude and 8 Hz frequency [22]. Birds are able to sense the motion input and figure out a control strategy quickly. Research reveals that the birds, like other animals, do not calculate the optimal control strategy in real-time. But rather they have an optimal control strategy stored in their brain and use it “sub-consciously” [23]. Inspired by this, we propose to use a neural network to fit a control function and use it for real time control of the suspended mass.

III. METHODS

A. Problem Formulation

In this research, we look into the classical problem of active vehicle suppression. For simplicity consideration, a quarter vehicle model (as shown in *Figure 1 A*) is considered. The control of the vehicle vibration is enabled by a motor (denoted as M) between the unsprung mass and the sprung mass. Due to the fact that the tire stiffness k_{us} is typically an order higher than the stiffness of the suspension k_s [24] (shown in *Figure 1*), the system is further simplified as *Figure 1B*), with the unsprung mass M_{us} , the unsprung spring k_{us} , and the unsprung damping c_{us} all neglected. The motion of M_{us} is assumed to be the same as that of the ground.

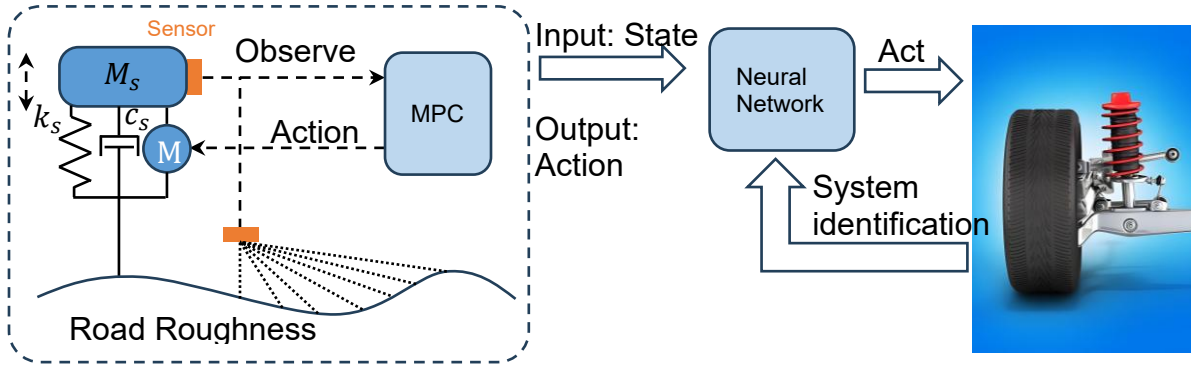


Figure 1. Framework for the proposed controller.

Table 1. Typical passenger vehicle quarter model [24], where the tire stiffness is an order of magnitude larger than the suspension stiffness.

Parameter	Sprung mass	Unsprung mass	Suspension stiffness	Suspension damping	Tire stiffness	Tire damping
Typical Value	300 kg	40 kg	$20,000 \frac{N}{m}$	$2,000 \frac{Ns}{m}$	$200,000 \frac{N}{m}$	neglectable

To accurately model the simulation environment, the active suspension system is described using a second-order differential equation that captures the relationship among external excitations, suspension dynamics, and control inputs. The system consists of a suspended mass connected to a spring and damper, subjected to external displacement excitation. The governing equation is formulated as:

$$m\ddot{y} = k_s(x - y) + 2m\omega_n\zeta_a(\dot{x} - \dot{y}) + u \quad (1)$$

where x represents the external excitation displacement, characterizing road surface irregularities (with the tire deflection neglected), and $\dot{x}$ is its corresponding first-order derivation of the excitation displacement. The terms y , $\dot{y}$, and $\ddot{y}$ denote the displacement, velocity, and acceleration of the suspended mass, respectively.

Additionally, k_s is the spring stiffness coefficient, and $\omega_n = \sqrt{\frac{k_s}{m}}$ denotes the resonance frequency of the system. m is the quarter vehicle suspended mass. The excitation force u comes from an electric motor to the suspended mass, and the control action is determined by the control algorithm. Due to actuation saturation, the $u_{optimal}$ is clipped by the following equation.

$$u = clip(u_{optimal}, -u_{max}, u_{max}) \quad (2)$$

where u_{max} is the actuation limitation.

B. Method overview

In this paper, we propose a framework for MPC-based optimal control with preview and actuation saturation. As shown in *Figure 1*, MPC will obtain optimal control with preview technology (*i.e.*, the displacement input of the future road) and actuation saturation. The optimal policy will be learnt by the neural network for a fast real-time deployment. Since the vehicle model will change due to the load and unload of passengers, real-time system identification will be implemented and feed the suspended mass estimation result to the neural network. Thus enables real-time adaptation to system model changes.

C. Preview-enhanced state augmentation and MPC-based optimal control

For an ideal linear system with unlimited actuation, Proportional Derivative with feedforward (PDFF) can suppress the vehicle body motion with ideal performance. However, due to the actuation saturation, the system becomes nonlinear, and PDFF's performance is limited. As a result, the controller has to consider the disturbance ahead and take actions beforehand. The disturbance ahead can be irregular roads, such as potholes and speed bumps. As a result, the state vector $\mathbf{S} = [x, \dot{x}, y, \dot{y}]$, which relies solely on current observations, is not enough.

MPC with an excitation preview mechanism is used in this project. The preview augments the state space with forecasted future external excitation displacements, enabling proactive actuation. Specifically, we extend the state dimension from 4 to $4 + H$ by appending a H -step preview horizon of future ground actuation x_{k+1} to x_{k+10} . The augmented state is:

$$\mathbf{S}_{k,aug} = [x_k, \dot{x}_k, y_k, \dot{y}_k, x_{k+1}, x_{k+2}, \dots, x_{k+H}] \quad (3)$$

The system equation of motion in state space format is

$$\begin{bmatrix} \dot{y} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} 0 \\ u \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{c}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (4)$$

where we define $\mathbf{y} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$, $\mathbf{A}_c = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}$, $\mathbf{B}_c = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix}$, $\mathbf{u} = \begin{bmatrix} 0 \\ u \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} 0 & 0 \\ \frac{k}{m} & \frac{c}{m} \end{bmatrix}$, and $\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$, the equation of motion becomes

$$\dot{\mathbf{y}} = \mathbf{A}_c \mathbf{y} + \mathbf{B}_c \mathbf{u} + \mathbf{G}_c \mathbf{x} \quad (5)$$

For numerical integration, we define

$$\mathbf{A}_d = \mathbf{I} + \Delta t \mathbf{A}_c, \mathbf{B}_d = \Delta t \mathbf{B}_c, \mathbf{G}_d = \Delta t \mathbf{G}_c \quad (6)$$

where Δt is a discrete time step. Then the numerical integration is

$$\mathbf{y}_{k+1} = \mathbf{A}_d \mathbf{y}_k + \mathbf{B}_d \mathbf{u}_k + \mathbf{G}_d \mathbf{x}_k \quad (7)$$

With the system simulation into the next H steps, the expression in stacked form can be obtained as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{k+1} \\ \mathbf{y}_{k+2} \\ \mathbf{y}_{k+i} \\ \dots \\ \mathbf{y}_{k+H} \end{bmatrix} = \mathbf{\Phi} \mathbf{y} + \mathbf{\Gamma} \mathbf{u} + \mathbf{S} \mathbf{x} \quad (8)$$

where $\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{S}$ are vertically stacked by $\mathbf{\Phi}_i, \mathbf{\Gamma}_i, \mathbf{S}_i$, with $\mathbf{\Phi}_i = \mathbf{A}_d^i$, $\mathbf{\Gamma}_i = \sum_{j=0}^{i-1} \mathbf{A}_d^{i-1-j} \mathbf{B}_d$, $\mathbf{S}_i = \sum_{j=0}^{i-1} \mathbf{A}_d^{i-1-j} \mathbf{G}_d$.

Since human comfort is largely indicated by acceleration [25], $\ddot{\mathbf{y}}$ is used as an optimization goal. The acceleration is expressed as

$$\ddot{\mathbf{y}} = \begin{bmatrix} -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \mathbf{y} + \frac{\mathbf{u}}{m} + \begin{bmatrix} \frac{k}{m} & \frac{c}{m} \end{bmatrix} \mathbf{x} = \mathbf{C}_a \mathbf{y} + \mathbf{D}_a \mathbf{u} + \mathbf{E}_a \mathbf{x} \quad (9)$$

where $\mathbf{C}_a = \begin{bmatrix} -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}$, $\mathbf{D}_a = \frac{1}{m}$, $\mathbf{E}_a = \begin{bmatrix} \frac{k}{m} & \frac{c}{m} \end{bmatrix}$.

Define the cost function as

$$\begin{aligned} J &= \sum_{k=1}^H (\ddot{\mathbf{y}}_{k+1}^T \mathbf{Q} \ddot{\mathbf{y}}_{k+1} + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k) \\ &= \sum_{k=1}^H ((\mathbf{C}_a (\mathbf{\Phi} \mathbf{y} + \mathbf{\Gamma} \mathbf{u} + \mathbf{S} \mathbf{x}) + \mathbf{D}_a \mathbf{u} + \mathbf{E}_a \mathbf{x})^T \mathbf{Q} (\mathbf{C}_a (\mathbf{\Phi} \mathbf{y} + \mathbf{\Gamma} \mathbf{u} + \mathbf{S} \mathbf{x}) + \mathbf{D}_a \mathbf{u} \\ &\quad + \mathbf{E}_a \mathbf{x}) + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k) \end{aligned} \quad (10)$$

where $\mathbf{Q} = \text{diag}(Q_{pos}, Q_{vel})$, and $\mathbf{R} = [R_u]$. The gradient of J over $\mathbf{u}_k$ is

$$\begin{aligned} \nabla J &= 2(\mathbf{C}_a \mathbf{\Phi} \mathbf{y}_k + (\mathbf{C}_a \mathbf{S} + \mathbf{E}_a) \mathbf{x}_k)^T \mathbf{Q} (\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma}) + 2((\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma}))^T \mathbf{Q} ((\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma})) \mathbf{u}_k \\ &\quad + \mathbf{R} \mathbf{u}_k = \mathbf{H}' \mathbf{u}_k + \mathbf{g}' \end{aligned} \quad (11)$$

where $\mathbf{H}' = ((\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma}))^T \mathbf{Q} ((\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma})) + \mathbf{R}$, and $\mathbf{g}' = 2(\mathbf{C}_a \mathbf{\Phi} \mathbf{y}_k + (\mathbf{C}_a \mathbf{S} + \mathbf{E}_a) \mathbf{x}_k)^T \mathbf{Q} (\mathbf{D}_a + \mathbf{C}_a \mathbf{\Gamma})$

Then use gradient descent to obtain the optimal $\mathbf{u}$ as

$$\mathbf{u}_{k+1} = \begin{cases} \mathbf{u}_k + \alpha(\mathbf{H}'\mathbf{u}_k + \mathbf{g}'), & \text{if } -\mathbf{u}_{max} \leq \mathbf{u} \leq \mathbf{u}_{max} \\ \mathbf{u}_{max}, & \text{if } \mathbf{u} > \mathbf{u}_{max} \\ -\mathbf{u}_{max}, & \text{if } \mathbf{u} < -\mathbf{u}_{max} \end{cases} \quad (12)$$

where α is the step and it is defined by $\frac{0.9}{\max(\text{diag}(\mathbf{H}'\mathbf{u}_k + \mathbf{g}'))}$ to make sure the step length is neither too large to cause numerical instability nor too small to cause unnecessary calculation time.

In order to get the minimum vibration acceleration, $\mathbf{Q} = \text{diag}(1, 1)$, and since the control effort is not considered in this research, $\mathbf{R} = [0.001]$.

With this setup, the MPC obtained the optimal action with respect to the augment state, expressed

$$u_k = f(\mathbf{s}_{k,aug}) \quad (13)$$

Since this is a quadratic cost function and the nonlinearity comes only from the saturation, the convex nature ensures the optimal solution provided by MPC is globally optimal.

For the baseline system, we use Feedforward + PD (Proportional + Derivative) control, the equation of motion is

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) + u_{FF} + k_d\dot{x} + k_p x = 0 \quad (14)$$

where the feedforward term is $u_{FF} = c\dot{y} + ky$.

D. Supervised imitation and policy distillation

Since MPC uses gradient descent method to get to the optimal solution, it takes dozens or even hundreds of steps before the optimal solution can be obtained. Considering in a real application scenario, the control loop happens every 0.01 s or even short time, it is challenging to do it in real-time on an on-board computer chip. In order to enable real-time deployment, we propose to use a neural network to fit the state-action pairs.

Considering the versatility of neural networks to fit almost any continuous function, it is ideal to fit the control policy using the MPC. The system runs for N steps and collect N pairs of optimal state action pairs from Eq. 13. In order to consider the change of the system due to passenger load and unload from the vehicle, simulation with suspension mass m (including the vehicle mass and the passenger mass) is collected and form pairs as follows:

$$u_k = f(\mathbf{s}_{k,aug}, m) \quad (15)$$

Hence the goal of the neural network is to fit this function. The loss is designed as the Mean Squared Error (MSE) between the neural network output and the target action. The neural network has a 2 hidden layer with 128 nodes in each layer. The learning rate is set to be 0.001.

IV. Results

A. Optimal solution by preview and MPC

Simulation results with MPC preview will be presented in this section. For this research, the base excitation amplitude ranges from 20 mm to 40 mm, and frequency ranges from 1 Hz all the way to 3 Hz. This is characterized as a specific road profile at a constant vehicle motion speed. The phase of the excitation is random and can be anywhere, ranging from 0 to 2π . The saturation reaches $u_{max} = 500 \text{ N}$. Experiments are done with different preview windows with time length T ranging from 0.1 s all the way to 1 s. The results are shown in Figure 2.

Show not only average improvement, but **worst-case** or robustness envelopes. Use real road profile starting from now?

- Report confidence intervals or at least multiple seeds/roads.

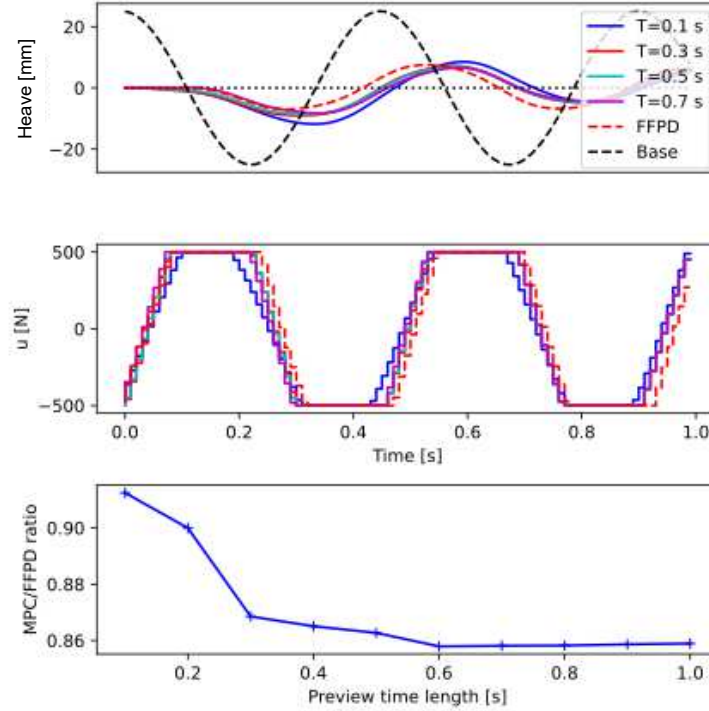


Figure 2. System performance vs. preview window length T . The last subfigure is the ratio between the RMS value of the acceleration (*i.e.*, $\ddot{y}$) with the MPC and FFPD method.

The results are compared with the baseline method, *i.e.*, FFPD. It can be observed from *Figure 2* that all MPC and FFPD methods achieve reasonable vibration suppression performance with actuation saturation, as the heave motion is much smaller than that of the excitation. It can also be observed that all MPC methods are better than the FFPD method, which is indicated by the RMS acceleration of the MPC/FFPD ratio $\frac{RMS(\ddot{y}_{MPC})}{RMS(\ddot{y}_{FFPD})}$, and its value is less than 1. Another important trend is that as the preview horizon increases, the system performance becomes better and better, which is indicated by the decrease of the MPC/FFPD ratio.

Another interesting phenomenon is that the system performance improvement mainly happens when the preview window is within the first 0.5 seconds. This roughly aligns with around 0.5 seconds, *i.e.*, 1 cycle of the excitation period. After that, the increase of the preview window will not help with the performance, while it will increase the computation burden significantly. The explanation of such a phenomenon is that the MPC and preview basically expect the peak to come and will act in advance. With a periodic signal, the peak comes within one cycle of the excitation period (for the case shown in *Figure 2*, it is 0.5 seconds). This provides guidelines for practical tuning of the preview window length. Further statistical experiment with 1000 experiments with 20~40 mm, 1~3 Hz base excitation with random phase shows that the RMS of the mass acceleration is reduced by $12 \pm 8.3\%$ compared with feedforward PD control.

In order to further showcase the contribution of MPC with respect to the saturation u_{max} , we have done simulations with different u_{max} values, and the results are shown in *Figure 3*.

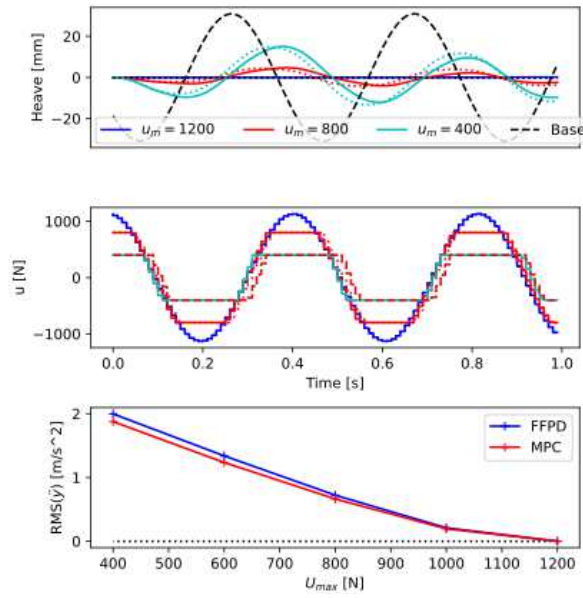


Figure 3. System performance vs. saturation amplitude. The first two subfigures share the same legend. The solid lines are for FFPD, while the dotted lines are for MPC (except for the base motion). u_m represents u_{max} due to space constraints.

It can be observed from *Figure 3* that as the decrease of u_{max} , the system performance becomes worse and worse, which is indicated by the amplitude of $RMS(\ddot{y})$. As the increases of u_{max} , $RMS(\ddot{y})$ is approaching zero, and reaches zero when the u_{max} reaches 1200 N, *i.e.*, the saturation is not effective anymore. At this stage, both FFPD and MPC can achieve perfect vibration suppression performance. However, when the maximum actuation reaches u_{max} , the MPC has around 10% margin better than the MPC, which indicates the effectiveness of the algorithm. This observation indicates that preview is most beneficial not when actuators are strong, but when they are weak and saturated, because preview and MPC reallocate limited actuation earlier in time.

In order to show the systematic performance improvement with different excitation frequencies, we have carried out experiments, and the results are shown in *Figure 4*.

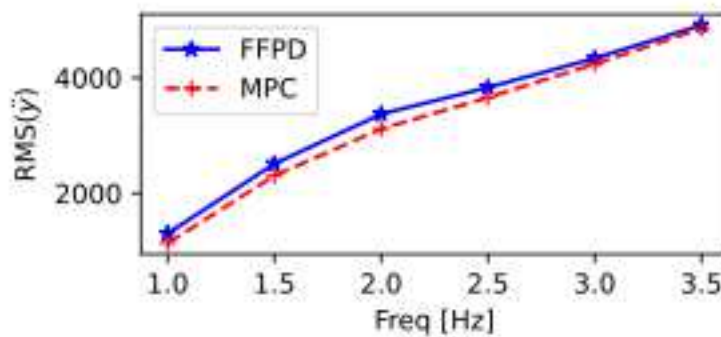


Figure 4. System performance vs. different excitation frequencies.

It can be observed from *Figure 4* that the RMS of acceleration decreased by 10% on average compared to the baseline in frequencies ranging from 1 Hz to 3.5 Hz.

From the experiment, it is observed that MPC takes a very long time due to the complexity of the calculation. In order to provide a warm start, the exact solution without constraint is used as an initial guess $\mathbf{u}$ by solving $\mathbf{H}'\mathbf{u}_k + \mathbf{g}' = 0$. On a Mac (M2 chip) with Python, a full simulation cycle takes 0.484 s, with time breakdown shown in *Figure 5*.

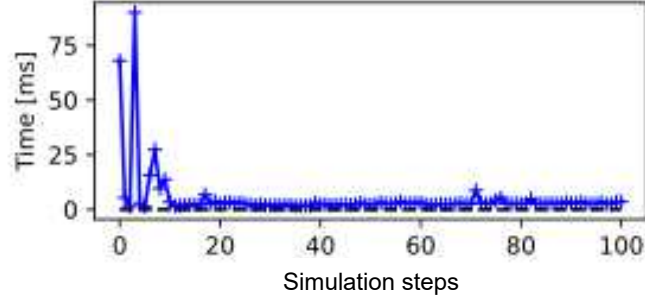


Figure 5. Time breakdown for the MPC algorithm. The first time point is the system matrix build-up.

From *Figure 5*, it is shown that the system matrix build-up takes around 75 ms and the first few iterations take more time. As time goes on, the rest takes around 3 ms per time step, which indicates it is impossible to achieve real-time control with a time step less than 3 ms. Considering this is carried out by an advanced 64-bit CPU with a 3.5 GHz main clock frequency, is even more challenging to carry out on an on-board CPU.

B. Control strategy distillation by neural network

The fitting process of the neural network is shown in *Figure 6*.

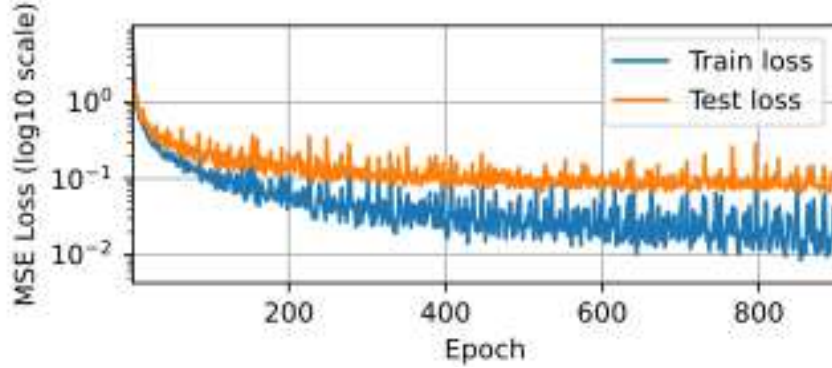


Figure 6. Neural Network fitting MSE loss vs. training epoch.

As observed in *Figure 6*, it takes around 400 epochs before the MSE loss stops from decreasing in both the training loss and test loss. After that, the loss change is slow, and hence we choose to stop the training. Considering the test loss is only $10^{-1} = 0.1 N$, and the maximum force is 500 N, it is more than enough for the task. In order to verify the performance of the neural network, we have carried out a simulation with MPC as well as Neural Network (NN) at the same time and plotted the results next to each other, and show in *Figure 7*.

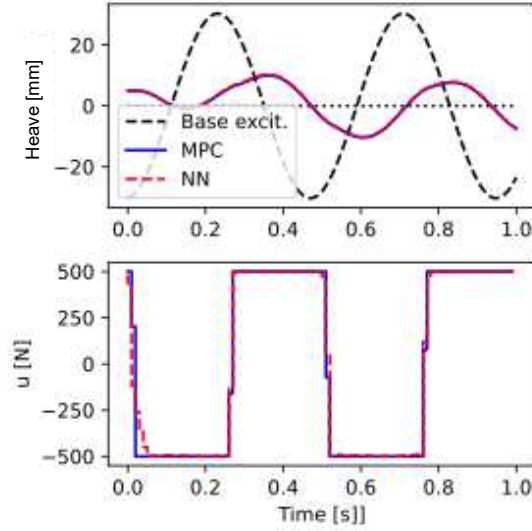


Figure 7. Different strategy: MPC vs. the imitation control strategy.

As observed in *Figure 7*, the control force u is almost identical for both cases. As the displacement is integration of velocity, and velocity is integration of acceleration (which is proportional to force), there is no observable difference in the mass displacement motion y , or heave motion of the vehicle.

To evaluate the improved execution speed when using the neural network, a comparison to the MPC method is done, and the results are shown in *Figure 8*.

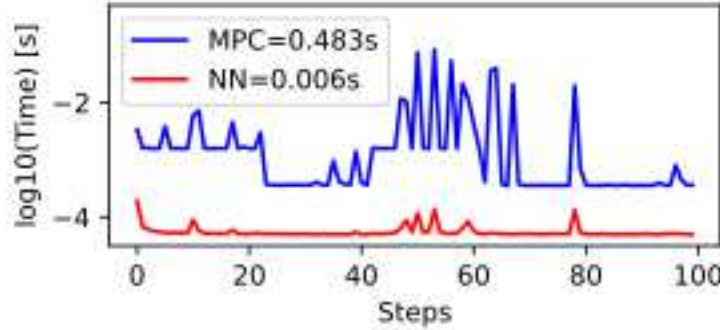


Figure 8. Time spent on each control step and total simulation time for the whole cycle comparison.

From *Figure 8*, it is observed that both methods finished the 100 steps of simulation. For the same task, the neural network only takes 6 ms, while the MPC takes 483 ms. This indicates that the neural network almost only takes two orders of magnitude shorter time compared to the MPC method.

“The learned controller achieves 0.3 ms inference time on an Intel i7 CPU, enabling 1 kHz control, whereas MPC requires 24 ms...”

C. Test on speed bump

The most commonly seen unevenness is the speed bump and was used to test the active suspension. In this section, we compare the performance of the FFPD and MPC in detail. The speed bump is formulated as follows, with a typical height of 8 cm and a width of 30 cm [26]. The shape can be reasonably assumed to be of a 60° arc, hence the math function to describe it is

$$z = \begin{cases} \sqrt{0.3^2 - (0.45 - x)^2} - 0.15\sqrt{2}, & 0.3 < x < 0.6 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

The test result with a real speed bump profile is shown in *Figure 9*.

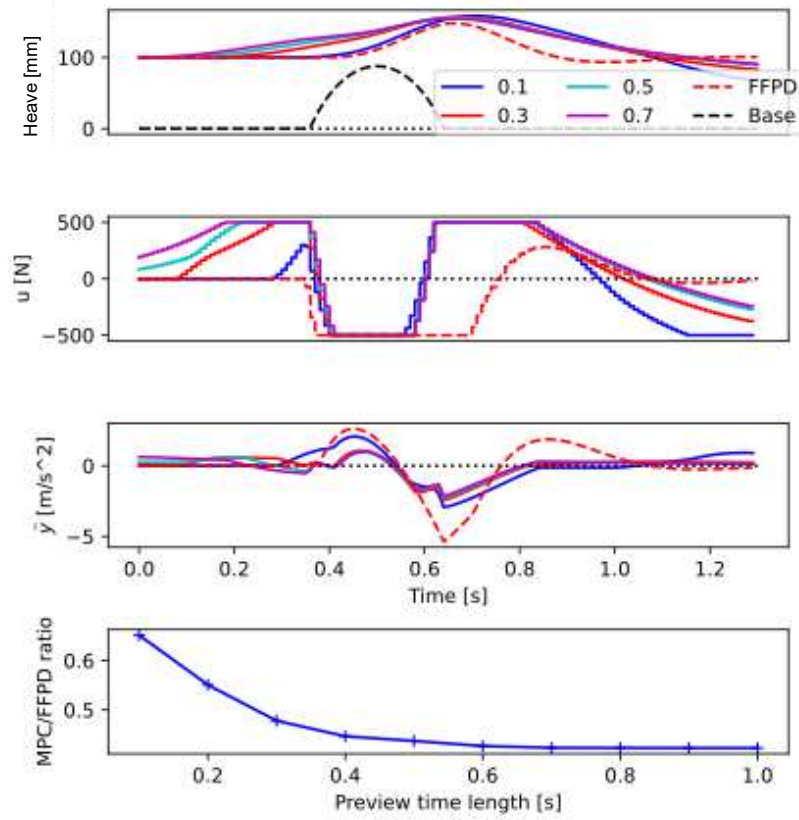


Figure 9. System performance test with a real speed bump.

From *Figure 9*, it can be observed that the RMS of the MPC/FFPD acceleration ratio $\frac{RMS(\ddot{y}_{MPC})}{RMS(\ddot{y}_{FFPD})}$ is significantly smaller when applying MPC-based preview control. With a 0.7 s preview window, the ratio can be as low as 0.43, which indicates the RMS acceleration is reduced by 57%. It can be observed in the heave figure that in all the MPC cases, control u starts to raise before actually hitting the bump, which indicates that the preview plays a key role here and the MPC algorithm starts the control effort in advance to avoid saturation later. With the action u figure, it can be observed that the action is active long before hitting the speed bump, and it increases as the preview window increases. However, for FFPD, the actuation starts only when the tire hits the bump, indicating the response is not proactive.

D. Test on real road profile

To validate the algorithm's performance on actual road profiles, we used road condition measurement data from the Danish Roads Authority (DRD) [27]. The data was acquired using a P79 profilometer with 25-point lasers measuring longitudinal and lateral road profiles. The selected profile segment is shown in *Figure 10* and serves as a representative road condition.

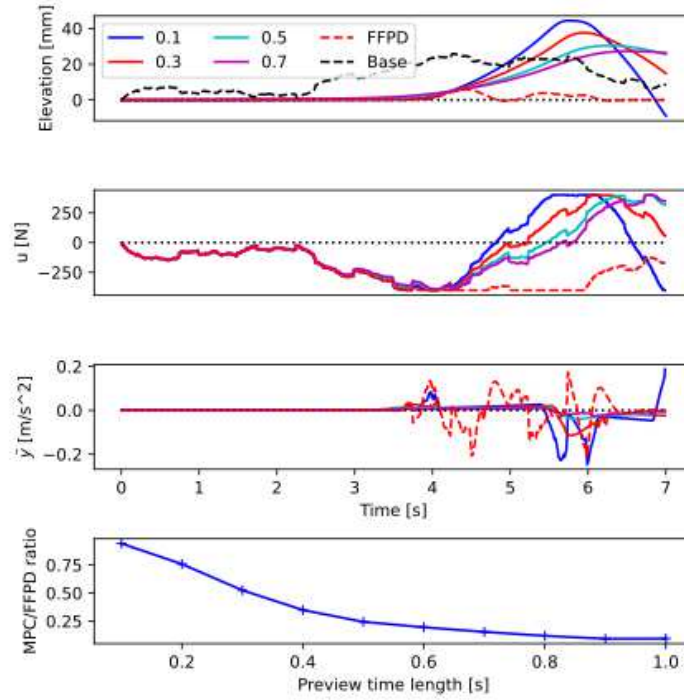


Figure 10. Performance test with real road profile.

As observed in *Figure 10*, the same performance trend is observed as in the speed bump case, with all the performance better than the FFPD. With a longer preview window, the MPC/FFPD ratio is smaller, indicating an improved performance. It is also interesting to observe that the MPC method suppressed the high frequency vibration while leaving the low frequency vibration, which is a good strategy because most of the acceleration comes from high frequency.

E. System identification and improved performance

The system identification can be carried out by the system modeling and least square method [28]. From the simulation, it turns out that the parameters can be identified with 0.4% of accuracy with 5% standard noise. With system identification and feeding that to the control algorithm, the performance improvement is shown as in *Figure 5*.

- Demonstrate model mismatch scenarios that are realistic (load changes, damper changes)
- No, with passenger load and unload. The system model is different.

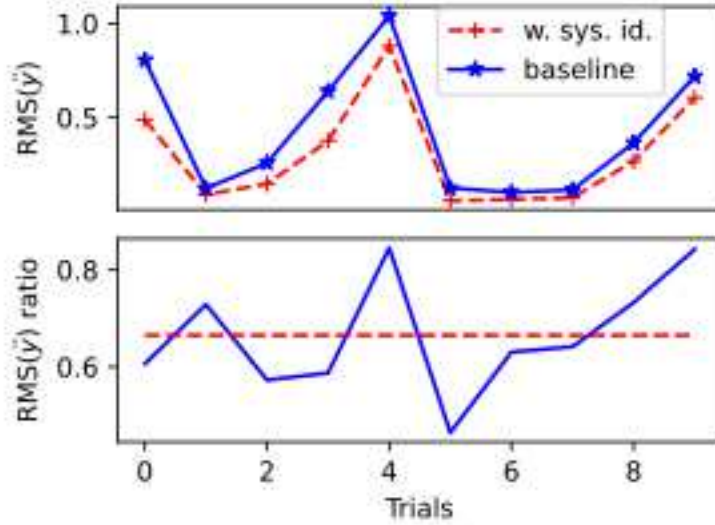


Figure 11. System performance with vs. without system identification.

From *Figure 5*, it is shown that with system identification, the RMS of $\ddot{y}$ decreases by an average of 32% compared with the baseline without system identification.

Hardware in the loop

- **Real embedded controller + real-time plant simulator**

This is the better version.

A practical setup for you could be:

- **Plant side:** desktop PC real-time simulating quarter-car / half-car dynamics
- **Controller side:** Arduino Due/Teensy
- **Communication:** serial
- **Road preview:** supplied by the plant simulator as a preview channel
- **Actuator saturation / lag / delay:** inside the plant model

VI. DISCUSSION

A. Possibility for real-world implementation

To verify the scalability of the proposed method, we tested it on a half-vehicle model (*Figure 12*) that captures both heave and pitch dynamics. This two-degree-of-freedom model better represents actual vehicle behavior than the quarter-vehicle model, as it includes front and rear suspensions with independent actuation and accounts for coupled vertical-rotational motions. For the MPC control, the objective function becomes the acceleration of the heave and pitch motion of the main body.

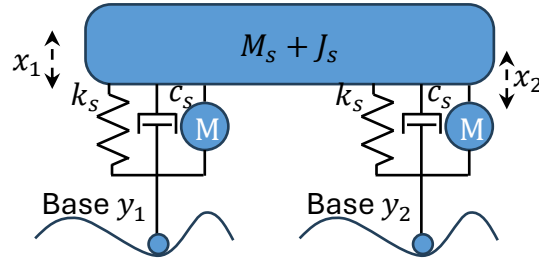


Figure 12. Algorithm application on half vehicle model.

The simulation experiment results and their comparison with FFPD are shown in Figure 13.

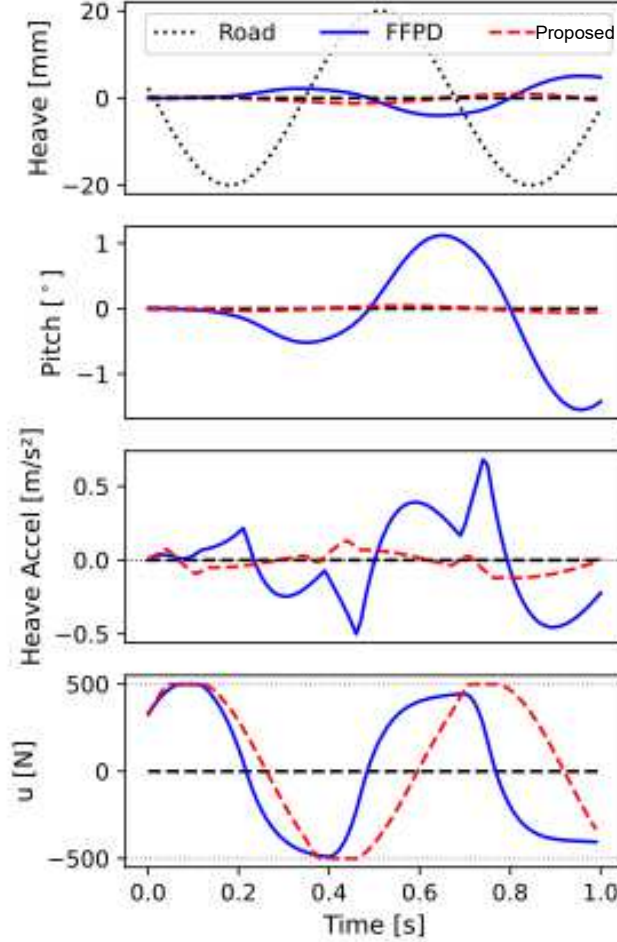


Figure 13. Performance comparison of the two algorithms with half vehicle model.

As observed in Figure 13, the proposed method (neural network fitting model fitted from MPC) can have better performance compared to the FFPD method, not only in heave acceleration, but also in the pitch of the half vehicle. For heave motion, the MPC-based approach achieves significantly lower RMS acceleration than PDFF. More notably, the pitch angle remains nearly zero with MPC control, while PDFF exhibits larger pitch variations. This difference partly stems from the control strategy: PDFF controls the front and rear suspensions independently, whereas MPC targets the vehicle center point, naturally coordinating the two actuators to minimize pitch. The results demonstrate that the proposed method maintains its performance advantages when applied to configurations closer to real vehicles.

B. Sensitivity to system parameter change

In engineering applications, practitioners often face the choice of which parameter to watch, or whether all the damper, spring, and mass parameters need to be estimated in real time. Although engineering

considerations, such as the cost of estimation, need to be considered, the key question has not yet answered: of all three system parameters, which one is more sensitive to system performance? In order to answer this question, we have done a simulation with MPC control with different parameter variations, and the results are shown in *Figure 14*.

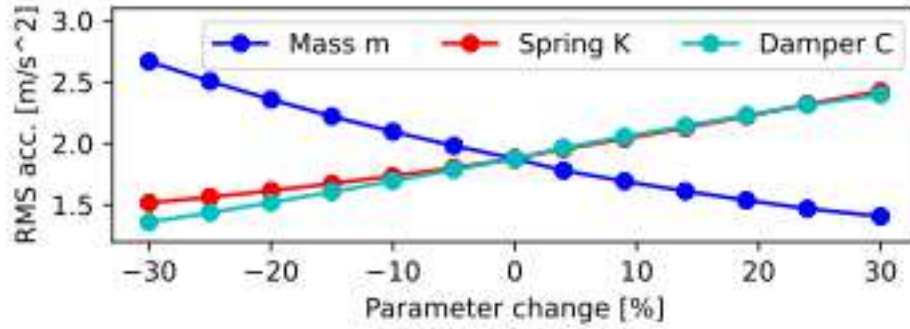


Figure 14. System performance sensitivity to spring, mass, and damper parameter changes.

As shown in *Figure 14*, it is observed that the system performance will improve/deteriorate due to the change of parameters, with a ratio of -1.12, 0.81, and 0.92, respectively. This provides a guideline for the practitioners when designing the system identification module, *i.e.*, tells which parameter to watch.

C. Comparison to other frameworks

In the early phase of this research project, we strived to use deep reinforcement learning (DRL) with a preview window to obtain the optimal control policy. However, it took us too much effort and forced us to use MPC, which is guaranteed to be globally optimal and uses much less computational power compared to reinforcement learning. Since the neural network just fits the MPC control policy and is tested to perform almost the same with guarantee, we can claim this control algorithm is the same as MPC, and it is physically interpretable. On the other hand, the policy learnt by deep reinforcement learning takes too much effort and is not physically interpretable. A comparison of the proposed framework with other options is detailed in *Table 2*.

Table 2. Comparison of this framework to existing methods.

	FFPD	MPC	DRL	Our work
Guaranteed global optimal	×	✓	×	✓
Low computation cost	✓	×	✓	✓
Adapt to system change	×	×	✓	✓
Physically interpretable	✓	✓	×	✓
Small training effort	✓	✓	×	✓

VI. CONCLUSIONS AND FUTURE WORK

In summary, we have conceived and tested a real-time adaptive optimal control architecture that preserves preview-based MPC performance under actuation saturation by combining deep learning policy approximation with online system identification. With the MPC and the preview window, global optimal control is achieved. The neural network is used to fit the control function with high accuracy and low computation cost, hence enabling real-time and physically interpretable control. The neural network also considers the system parameter change, and with real-time system identification, performance is further improved by considering the change. This research has the potential to accelerate the adoption rate of active suspension in vehicles. The work here provides design guidelines, as well as engineering insights for

practitioners. The work here is based on vehicle suspension but can be applied to a broader range of engineering applications, especially when the hardware (actuation, computation power) is limited.

This work focused on the simulation study and did not go through rigorous tests in real vehicles. Since safety is a special concern in the vehicle industry, further work should be done to ensure it can be implemented without accidents. The algorithm we proposed has to go through tests before it can be applied in real vehicles. In the meantime, this research did not consider the road handling performance simultaneously, which can be a good topic for future work.

Acknowledgement

Removed for review purposes.

References

1. Ibrahim, R. A. (2008). Recent advances in nonlinear passive vibration isolators. *Journal of Sound and Vibration*, 314(3-5), 371-452.
2. Burton, A. W. (1993). Active vibration control in automotive chassis systems. *Computing and Control Engineering*, 4(5), 225-232.
3. Jalili, N. (2002). A comparative study and analysis of semi-active vibration-control systems. *J. Vib. Acoust.*, 124(4), 593-605.
4. Lv, X., Ji, Y., Zhao, H., Zhang, J., Zhang, G., & Zhang, L. (2020). Research review of a vehicle energy-regenerative suspension system. *Energies*, 13(2), 441.
5. Hyniova, K., Stribrsky, A., Honcu, J., & Kruczek, A. (2009). Active suspension system–energy control. *IFAC Proceedings Volumes*, 42(19), 146-152.
6. Karnopp D, Crosby MJ, Harwood RA. Vibration control using semi-active force generators.
7. Valášek M, Novak M, Šika Z, Vaculin O. Extended ground-hook-new concept of semi-active control of truck's suspension. *Vehicle System Dynamics*. 1997 Jun 1;27(5-6):289-303.
8. Kalman, R.E. (1960) Contributions to the Theory of Optimal Control. *Boletín de la Sociedad Matemática Mexicana*, 5, 102-119
9. Yu M, Evangelou SA, Dini D. Advances in active suspension systems for road vehicles. *Engineering*. 2024 Feb 1; 33:160-77.
10. Morari M, Lee JH. Model predictive control: past, present and future. *Computers & chemical engineering*. 1999 May 1;23(4-5):667-82.
11. Hamza A, Ben Yahia N. Heavy trucks with intelligent control of active suspension based on artificial neural networks. Proceedings of the Institution of Mechanical Engineers, Part I: *Journal of Systems and Control Engineering*. 2021 Jul;235(6):952-69.
12. Han J. From PID to active disturbance rejection control. *IEEE transactions on Industrial Electronics*. 2009 Feb 27;56(3):900-6
13. Cho, B. K. (1999). Active suspension controller design using MPC with preview information. *KSME International Journal*, 13(2), 168-174.
14. Theunissen, J., Tota, A., Gruber, P., Dhaens, M., & Sorniotti, A. (2021). Preview-based techniques for vehicle suspension control: A state-of-the-art review. *Annual Reviews in Control*, 51, 206–235.
15. Di Cairano, S., & Kolmanovsky, I. V. (2018). Automotive applications of model predictive control. In *Handbook of model predictive control* (pp. 493-527). Cham: Springer International Publishing.
16. Papadimitrakakis, M., & Alexandridis, A. (2022). Active vehicle suspension control using road preview model predictive control and radial basis function networks. *Applied Soft Computing*, 120, 108646.
17. Achnib, A., & Senane, O. (2023). Discrete-time multi-model preview control: Application to a real semi-active automotive suspension system. *Control Engineering Practice*, 137, 105553.

-
18. Kim, J., Lee, T., Kim, C.-J., & Yi, K. (2023). Model predictive control of a semi-active suspension with a shift delay compensation using preview road information. *Control Engineering Practice*, 137, 105584.
 19. Li, Q., Chen, Z., Song, H., & Dong, Y. (2024). Model Predictive Control for Speed-Dependent Active Suspension System with Road Preview Information. *Sensors*, 24(7), 2255.
 20. Cao, Y., & Gopaluni, R. B. (2020). Deep neural network approximation of nonlinear model predictive control. *IFAC-PapersOnLine*, 53(2), 11319-11324.
 21. Lai, W., Li, Y. L., & Lin, Z. (2024). Performance-based activation of anti-windup compensation for control of linear systems subject to actuator saturation. *Automatica*, 159, 111298.
 22. Katzir, G., Schechtman, E., Carmi, N., & Weihs, D. (2001). *Head stabilization in herons*. *Journal of Comparative Physiology A*, 187(6), 423–432.
 23. Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269(5232), 1880–1882.
 24. Rajamani, R. (2006). *Vehicle dynamics and control*. Boston, MA: Springer US.
 25. ISO 2631-1:1997 “Mechanical vibration and shock — Evaluation of human exposure to whole-body vibration — Part 1: General requirements.” Geneva, Switzerland.
 26. <https://www.dimensions.com/element/speed-bump-standard> (accessed 2025.12.10)
 27. . https://data.dtu.dk/articles/dataset/Data_subset_for_road_condition_modelling_-_reference_data/23097002 (accessed 2026.1.15)
 28. Wang, Z., Zhou, A., Cook, A., Shao, Y., & Wang, J. “Active Suspension Parameters Identification: An Algebraic Approach and its Application to Suspension Travel Control.” *IFAC-PapersOnLine*, 56(3), 73–78, 2023. DOI: 10.1016/j.ifacol.2023.12.003.