GPU based Stochastic Parameterized NMPC scheme for Control of Semi-Active Suspension System for Half Car Vehicle

Karthik Murali Madhavan Rathai, Mazen Alamir, Olivier Sename

GPU based Stochastic Parameterized NMPC scheme for Control of Semi-Active Suspension System for Half Car Vehicle ⋆

Karthik Murali Madhavan Rathai * **Mazen Alamir** *
Olivier Sename *

* *Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France*
(e-mail: karthik.murali-madhavan-rathai@gipsa-lab.grenoble-inp.fr)
(e-mail: mazen.alamir@gipsa-lab.grenoble-inp.fr)
(e-mail: olivier.sename@gipsa-lab.fr)

Abstract: Control of complex systems with inherent randomness in process dynamics poses a serious concern for control engineers, especially in situations where performance and constraint satisfaction are highly demanded. In this paper, we propose a real time (RT) scenario based stochastic parameterized NMPC (SS-pNMPC) scheme for control of semi-active (SA) system for a half car vehicle. The method utilizes graphic processing unit (GPU) to generate several RT scenarios of the random road profile for each parameterized input and through Monte-Carlo (MC) simulations, the expected objective function along with a probabilistic constraint violation certificate are numerically obtained. The optimal input is elicited by finding the input either with minimum expected objective or with the lowest probabilistic constraint violation certificate. The method was implemented on NVIDIA Jetson embedded boards and also, tested in MATLAB/Simulink environment for different ISO road profiles and the simulation results exhibits better performance of the proposed method in comparison to passive systems.

Keywords: Automotive control, GPGPU computing, Real-time Stochastic Non-linear Model Predictive Control, Embedded control, Vertical Dynamics

1. INTRODUCTION

With increasing demands from automotive industry on performance and safety of vehicles, over the last few years Model Predictive Control (MPC) has gained immense attraction. However, the major hindrance for the method from being pervasive is the need for high computational time for solving the online optimization problem. Over the last few decades with the development of advanced and sophisticated embedded processors, the gap between theory and practice for MPC controller has abridged. At the same time the MPC methods, techniques and its variants have evolved substantially in large proportions. Thus, it can be scarcely denied that with increased complexity of methods, a need for increased computational resource is inevitable to meet the real-time (RT) requirements for fast sampled systems. Under this premise, the multicore Graphic Processing Unit (GPU) displays a strong potential to implement several state of the art MPC methods.

There have been several research contributions on utilization of GPUs for solving MPC problems. In Abughalieh and Alawneh (2019), a detailed survey on various types of parallel implementation of MPC methods are described. In Sampathirao et al. (2017), a scenario based Stochastic MPC (SMPC) method is proposed where the structure of the system is exploited and the problem is solved using proximal gradient method which is parallelized on GPU. In Ohyama and Date (2017), a sampling based parallelized nonlinear MPC (NMPC) scheme is proposed and experimentally validated for control of inverted pendulum system. In Williams et al. (2016), a path integral based MPC method is proposed and experimentally validated for a remote controlled (RC) car. In Rogers (2013), a guidance law for guided projectiles is proposed, where the GPUs are utilized to generate RT scenarios to predict the impact point and probability of violating impact area constraints. Concerning the control of semi-active suspensions system, Savaresi et al. (2010) provides a comprehensive collection of all classical and modern control methods such as Sky-hook, SH-ADD, Hybrid MPC with preview, H_∞ and LPV methods etc. However, in the line of research of application of SMPC for control of suspension systems, not many research have been conducted in the past. To the best of knowledge of the authors, in Guanetti and Borrelli (2017) a cloud aided SMPC method is proposed for control of active suspension system for a quarter car vehicle.

In this paper, we propose a GPU based scenario-stochastic parameterized NMPC (SS-pNMPC) method for control of semi-active system for half car vehicle to provide comfort and safety for onboard passengers. The work presented in this paper is an extension to the previous work Rathai

⋆ This work was supported by the ITEA3 European project, 15016 EMPHYSIS (Embedded systems with physical models in the production code software).

et al. (2019b), where the novelty lies in the inclusion of stochastic road model into control design. The key contributions of the paper are :
a) **Modeling flexibility** - The proposed method requires no a-priori assumptions such as linear time invariant (LTI) dynamics, stationary property, Gaussian distribution of noise etc. This feature is highly sought these days as proposed in Guanetti and Borrelli (2017), the road model stochastic information can be dynamically obtained from cloud servers.
b) **Validation/Verification on Embedded platforms** - The proposed method was tested on multiple GPU based embedded boards to calculate the computation time and also to assess the RT feasibilty of the method. The method was tested on the NVIDIA Jetson embedded boards - Nano, TX1, TX2 and Xavier.

This paper is organized as follows. Section 2 discusses the vertical dynamics model for the vehicle, non-linear quasi static model for the ER dampers and modeling of ISO road profiles in detail. Section 3 discusses the stochastic NMPC problem formulation. Section 4 describes the SS-pNMPC method, pseudo-code and scenario generation techniques in detail. Finally, the paper is concluded with Section 5 and Section 6 with results, simulation, conclusions and future works respectively.

2. HALF CAR MODEL WITH SEMI-ACTIVE SUSPENSION SYSTEM

2.1 Half car mathematical model

The mathematical model of the vehicle is comprised of a) Dynamical model of vertical motion of half car vehicle extended with road model and b) Kinematics model for longitudinal motion of the vehicle Savaresi et al. (2010). Let the left and right corners of the vehicle be indexed with $i \in \{l, r\}$ respectively. The equations of the model is expressed as follows

$$\begin{cases} m_s \ddot{z}_s = -(F_{s,l} + F_{s,r}) \\ I_x \ddot{\theta} = (l_l F_{s,l} - l_r F_{s,r}) \\ m_{us,l} \ddot{z}_{us,l} = (-F_{s,l} + F_{t,l}) \\ m_{us,r} \ddot{z}_{us,r} = (-F_{s,r} + F_{t,r}) \\ \dot{z}_{r,l} = -\alpha v_x z_{r,l} + \xi_l \\ \dot{z}_{r,r} = -\alpha v_x z_{r,r} + \xi_r \\ \dot{v}_x = a_x \end{cases} \tag{1}$$

where, z_s, θ represents the heave/chassis position and roll angle of the vehicle w.r.t. the centre of gravity (COG) respectively. $z_{us,i}$, $z_{r,i}$, ξ_i $\forall i \in \{l, r\}$ represents the wheel/unsprung mass position, vertical road profile of the vehicle and random disturbances respectively. v_x and a_x denotes the longitudinal velocity and acceleration of the vehicle. m_s, $m_{us,l}$, $m_{us,r}$ represents the chassis mass, unsprung masses for the left and right corners. I_x represents the moment of inertia along the roll axis. l_l and l_r represents the length of the chassis from the left and right corners with respect to COG. α represents the ISO road profile parameter. $F_{s,i}$ represents the chassis forces and $F_{t,i}$ represents the wheel forces $\forall i \in \{l, r\}$ which are expressed with

$$\begin{aligned} F_{s,i} &= -k_{s,i}(z_{s,i} - z_{us,i}) + u_i \\ F_{t,i} &= -k_{t,i}(z_{us,i} - z_{r,i}) \end{aligned} \tag{2}$$

where, $k_{s,i}$ and $k_{t,i}$ represents the stiffness coefficent of the SA suspension system and wheel respectively. $z_{r,i}$ and $z_{us,i}$ represents the vertical road displacement and unsprung mass position $\forall i \in \{l, r\}$. u_i represents the actuation force obtained from the nonlinear SA damper model (see Section 2.2). $z_{s,i}$ represents the sprung mass displacement at each corner which are obtained from the following equations

$$\begin{aligned} z_{s,l} &= z_s + l_l \sin\theta \\ z_{s,r} &= z_s - l_r \sin\theta \end{aligned} \tag{3}$$

2.2 Nonlinear quasi-static SA damper model

The SA damper force u_i in (2) is expressed by utilizing the Guo's damper force model (Savaresi et al. (2010)) $\forall i \in \{l, r\}$ with

$$u_i = k_0 z_{d,i} + c_0 \dot{z}_{d,i} + f_c \phi_i \tanh(a_1 \dot{z}_{d,i} + a_2 z_{d,i}) \tag{4}$$

where k_0, c_0, f_c, a_1 and a_2 represents the damper stiffness coefficient, viscous damping coefficient, dynamic yield force of the fluid, hysteresis coefficient due to velocity and position respectively. ϕ_i, $\forall i \in \{l, r\}$ represents the duty cycle (PWM-DC) input signal which manipulates the damper characteristics online by changing the input voltage. $z_{d,i} = z_{s,i} - z_{us,i}$ and $\dot{z}_{d,i} = \dot{z}_{s,i} - \dot{z}_{us,i}$ represents the deflection position and velocity $\forall i \in \{l, r\}$ between the chassis and wheel respectively.

2.3 ISO road profile

The ISO road profile is primarily dependent upon two factors which are a) Longitudinal velocity of the vehicle and b) Road roughness coefficient, i.e. the road surface (Tyan et al. (2009)). In accordance with the ISO-8608 standard (de Normalización (Ginebra)), the road profile can be expressed as a time-varying first order auto regressive process, which is expressed with

$$\dot{z}_r(t) = -\alpha v_x(t) z_r(t) + \xi(t) \tag{5}$$

where, $z_r(t)$ is the road profile as defined in (1) and $\xi(t)$ is the stochastic road disturbance drawn from a normal distribution defined with $\xi(t) \sim \mathcal{N}(0, \Psi_{z_r}(t))$, where $\Psi_{z_r}(t) = 2\alpha v_x(t)\sigma^2$ represents the spectral density of the Gaussian white noise and σ^2 denotes the road roughness variance. The parameters for different road surfaces are listed in Table 1.

Table 1. ISO road roughness parameters

Road surface	Road roughness (σ^2)	α (rad/s)
ISO A (Very Good)	4×10^{-6} m	0.127
ISO B (Good)	16×10^{-6} m	0.127
ISO C (Average)	64×10^{-6} m	0.127
ISO D (Poor)	256×10^{-6} m	0.127
ISO E (Very Poor)	1024×10^{-6} m	0.127

Let $\mathbf{X} = [z_s, \theta, z_{us,l}, z_{us,r}, \dot{z}_s, \dot{\theta}, \dot{z}_{us,l}, \dot{z}_{us,r}, z_{r,l}, z_{r,r}, v_x]$ denote the state vector, $\mathbf{U} = [\phi_l, \phi_r]$ denote the input vector and $\mathbf{\Xi} = [\xi_l, \xi_r]$ denote the disturbance vector, then the half car model (1) can be compactly expressed with

$$\dot{\mathbf{X}}(t) = f(\mathbf{X}(t), \mathbf{U}(t), \mathbf{\Xi}(t), a_x(t)) \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{11}$, $\mathbf{U} \in \mathbb{R}^2$ and $\mathbf{\Xi} \in \mathbb{R}^2$. The parameters are obtained from the INOVE test platform at GIPSA lab, Grenoble. The INOVE test platform discussed is a 1:5-scaled baja style racing car which consists of 4 controllable

Electro-Rheological (ER) SA dampers (ER-SA) and 4 DC motors to generate different road profiles for each wheel corner (Sename (2019)).

3. MODEL PREDICTIVE CONTROL DESIGN

3.1 Objective requirements

The objective design in time domain can be briefly classified as a) comfort and b) ride handling objective Savaresi et al. (2010).

(1) **Comfort objective**: The goal of the comfort based objective is to minimize the vertical acceleration of the chassis ($\ddot{z}_s$), governed by (1). The comfort objective for a given look ahead period T_l is expressed with

$$J^{acc}_{T_l}(\mathbf{X}(.), \mathbf{U}(.), \mathbf{\Xi}(.), a_x(.)) = \int_0^{T_l} (\ddot{z}_s(t))^2 dt \quad (7)$$

(2) **Ride handling objective**: The goal of the ride handling objective is to minimize the roll angle (θ) of the vehicle. The ride handling objective for a given look ahead period T_l is expressed with

$$J^{roll}_{T_l}(\mathbf{X}(.), \mathbf{U}(.), \mathbf{\Xi}(.), a_x(.)) = \int_0^{T_l} (\theta(t))^2 dt \quad (8)$$

3.2 Constraint requirements

The constraints for the SA suspension system primarily arise from the physical limitations and secondarily from performance requirements (Baumal et al. (1998)). For the MPC design, the included constraints are

(1) **ER-SA damper input constraints**:
   (a) Damper force constraint (Physical): The ER-SA damper force is bounded i.e. $u_i \in [\underline{u_i},\ \overline{u_i}]$, $\forall i \in \{l, r\}$.
   (b) PWM-DC input constraints: The operating DC for the PWM signal is constrained to $\phi_i \in [\underline{\phi_i},\ \overline{\phi_i}]$, $\forall i \in \{l, r\}$.

(2) **State constraints**:
   (a) Stroke deflection constraint (Physical): This forms a linear state constraint i.e. $z_{d,i} \in [\underline{z_{d,i}},\ \overline{z_{d,i}}]$, $\forall i \in \{l, r\}$.
   (b) Wheel rebound constraint (Performance): This bounds the deflection position between the wheel and road. This is to ensure the tyre deflection forces are bounded i.e. $z_{us,i} - z_{r,i} \in [\underline{z_{reb,i}},\ \overline{z_{reb,i}}]$, $\forall i \in \{l, r\}$.

(3) **Road variable assumption**: The vertical road displacement at the current time instant is assumed to be measured by means of adaptive road profile observers (Doumiati et al. (2017)) or from cloud servers (Zhang et al. (2017)).

(4) **Longitudinal acceleration assumption**: The longitudinal acceleration a_x is assumed to be constant over the prediction horizon (T_l). In a real vehicle setting a_x is typically obtained from Inertial Measurement Unit (IMU) of the vehicle.

The mixed input-state constraint set is compactly expressed with $(\mathbf{X}, \mathbf{U}) \in \Omega_{(\mathbf{X},\mathbf{U})} \subset \mathbb{R}^{11} \times \mathbb{R}^2$ and the input constraint set is compactly expressed with $\mathbf{U} \in \Omega_{\mathbf{U}} \subset \mathbb{R}^2$. The disturbance has a probabilistic support $\mathbb{P}$ which is normally distributed with $\mathbf{\Xi} \sim \mathcal{N}(0, \Sigma(\mathbf{X}(t)))$, where $\Sigma(\mathbf{X}(t)) = \text{diag}(\Psi_{z_{r,l}}(\mathbf{X}(t)), \Psi_{z_{r,r}}(\mathbf{X}(t)))$. The variance is dependent on longitudinal velocity (v_x) of the vehicle, which is a state variable of the system (see Section 2.3).

3.3 SNMPC problem formulation

Due to the stochastic disturbances acting on the system, it is of paramount importance to model the MPC formulation to deal with the uncertainties. Thus, stochastic measures are adopted for the MPC problem formulation. The stochastic measure adopted for the objective is the expectation operator ($\mathbb{E}$) and the mixed state-input constraints ($\Omega_{(\mathbf{X},\mathbf{U})}$) are encapsulated in a probabilistic framework with a finite level of violation $\eta \ll 1$ (as the problem is not always feasible). In summary, with the proposed objective and constraints function, the SNMPC OCP is casted as

$$\begin{aligned} J^*_{obj}(\mathbf{X}_0, \Gamma_0, a_{x,0}, \mathbf{U}^*(.)) = \min_{\mathbf{U}(.)} \ & \mathbb{E}[\max(\Gamma^1_0 J^{acc}_{T_l}, \Gamma^2_0 J^{roll}_{T_l})] \\ \text{s.t.} \ \ & \dot{\mathbf{X}}(t) = f(\mathbf{X}(t), \mathbf{U}(t), \mathbf{\Xi}(t), a_x(t)) \\ & \mathbf{X}(0) = \mathbf{X}_0, a_x(.) = a_{x,0}, \mathbf{U}(.) \in \Omega_{\mathbf{U}} \\ & \mathbb{P}[(\mathbf{X}(.), \mathbf{U}(.)) \notin \Omega_{(\mathbf{X},\mathbf{U})}] \leq \eta \end{aligned} \quad (9)$$

where $\mathbf{X}_0$ represents the initial state vector and $a_{x,0}$ represents the constant longitudinal acceleration over the prediction horizon. $\Gamma_0 = [\Gamma^1_0,\ \Gamma^2_0]$ with Γ^1_0 and Γ^2_0 being the convex weights for the two objectives comfort and ride handling respectively. The rationale behind utilizing the min-max objective is to minimize the objective with maximum cost at the current instant. It is also important to note that the two objectives are conflicting in nature (See Savaresi et al. (2010)). Once the optimal input trajectory is computed, the first control action is injected into the system and this procedure is repeated in receding horizon fashion. In this work, a constant input profile is assumed over the control horizon. As the SA suspension system is inherently stable the foregoing assumption is apposite for performance requirements for fast sampled systems.

4. SS-PNMPC METHOD

4.1 Method description

The method is an extension of the work proposed in Rathai et al. (2018) and Rathai et al. (2019b), where the stochastic road model is accounted in the model dynamics. The fundamental idea of the method is to parameterize the input set ($\Omega_{\mathbf{U}}$) into finite number of control inputs (similar to finite control set MPC) and for each parameterized control input, the SMPC is solved by performing MC simulations with several scenarios of the road profile. From the simulations, the expected objective function is numerically obtained by empirical mean and a probabilistic constraint violation certificate (PCVC) is numerically obtained by computing the ratio between number of constraint violation and number of scenarios generated. The optimal input is selected by finding the minimum expected objective along with the consideration of PCVC less than or equal to the specified level (η). If none of the input satisfies the above criteria, then the input with the least PCVC is selected. The pseudo-code for the method is shown in Algorithm 1.

Algorithm 1 SS-pNMPC pseudocode

Data Initialization: Model/Constraint parameters
Input: $\mathbf{X}_0, \Gamma_0, a_{x,0}, n_g, \gamma := N_s \times N_\tau$
Output: $\mathcal{U}_{i^*}$

1: **function** __CPU__ **SSpNMPC**($\mathbf{X}_0, \Gamma_0, a_{x,0}$)
2: $\mathcal{U}_{1:n_g} \leftarrow \texttt{grid}(\Omega_{\mathbf{U}}, n_g)$
3: **parfor** $i \leftarrow 1 : n_g$ **do**
4: **parfor** $l \leftarrow 1 : N_s$ **do**
5: $(\texttt{Obj}_i[l], \texttt{CV}_i[l]) \leftarrow \textbf{SIM}(\mathbf{X}_0, \Gamma_0, a_{x,0}, \mathcal{U}_i)$
6: **end parfor**
7: $\texttt{EObj}_i \leftarrow \frac{\sum_l \texttt{Obj}_i[l]}{N_s}$; $\texttt{PCVC}_i \leftarrow \frac{\sum_l \texttt{CV}_i[l]}{N_s}$
8: **end parfor**
9: **if** $(\texttt{EObj}_{i^*} \leq \texttt{EObj}_{\forall i \setminus \{i^*\}}$ & $\texttt{PCVC}_{i^*} \leq \eta)$ **then**
10: $i^* \leftarrow \texttt{indexmin}(\texttt{EObj})$
11: $J^*_{obj} \leftarrow \texttt{EObj}_{i^*}$
12: **else**
13: $i^* \leftarrow \texttt{indexmin}(\texttt{PCVC})$
14: **end if**
15: **return** $\mathcal{U}_{i^*}$
16: **end function**
17: **function** __GPU__ **SIM**($\mathbf{X}_0, \Gamma_0, a_{x,0}, \mathcal{U}_i$)
18: $\texttt{CV} \leftarrow 0$; $\texttt{Obj} \leftarrow 0$
19: **for** $j \leftarrow 0 : N_\tau$ **do**
20: $\texttt{BO} \leftarrow 1$; $\texttt{TObj} \leftarrow 0$; $\mathbf{X}_{em} \leftarrow \mathbf{X}_0$
21: **for** $t_{loop} \leftarrow 0 : h : T_l$ **do**
22: $\boldsymbol{\Xi} \sim \mathcal{N}(0, \Sigma(\mathbf{X}_{em})h)$
23: $\mathbf{X}_{em} \leftarrow \mathbf{X}_{em} + hf(\mathbf{X}_{em}, \mathcal{U}_i, \boldsymbol{\Xi}, a_{x,0})$
24: **if** $(((\mathbf{X}_{em}, \mathcal{U}_i) \notin \Omega_{(\mathbf{X},\mathbf{U})})$ & $\texttt{BO})$ **then**
25: $\texttt{CV} \leftarrow \texttt{CV} + 1$; $\texttt{BO} \leftarrow 0$
26: **end if**
27: $\texttt{TObj} \leftarrow \texttt{TObj} + h\texttt{max}\{\Gamma_0^1 J_{l_l}^{acc}, \Gamma_0^2 J_{l_l}^{roll}\}$
28: **end for**
29: $\texttt{Obj} \leftarrow \texttt{Obj} + \texttt{TObj}$
30: **end for**
31: **return** $\{\frac{\texttt{Obj}}{N_\tau}, \frac{\texttt{CV}}{N_\tau}\}$
32: **end function**

Explanation of Implementation:

(1) **Initialization, I/Os and Syntax declaration**:
   (a) The data initialization step sets the parameter values for the half-car model and the constraints.
   (b) The input variables are $\mathbf{X}_0, \Gamma_0, a_{x,0}, n_g, \gamma$, where n_g and γ are the number of input parameterization and number of scenarios respectively. The output variable is $\mathcal{U}_{i^*}$, i.e. the optimal input vector injected into the system.
   (c) The qualifiers __CPU__ and __GPU__ denotes the function operation in CPU and GPU respectively. The entry point is **SSpNMPC** function.

(2) **SSpNMPC** function:
   (a) In line 2, the input set $\Omega_{\mathbf{U}}$ is finitely discretized into n_g points and collected in the grid $\mathcal{U}_{1:n_g}$.
   (b) From line 3-8, the **parfor** i.e. parallel **for** function is utilized to dispatch each and every discretized input from $\mathcal{U}_{1:n_g}$ to GPU and for each input the **SIM** function is utilized to conduct N_s number of MC simulations. The respective objective function $\texttt{Obj}_i[l]$ and constraint violation $\texttt{CV}_i[l]$ for the l^{th} MC simulation and i^{th} input are obtained as return arguments. Finally, the expected objective function $\texttt{EObj}_i$ and probabilistic constraint violation certificate $\texttt{PCVC}_i$ are numerically obtained for the i^{th} input in $\mathcal{U}_{1:n_g}$.
   (c) The lines 9-15, looks for the optimal input i^* index with minimum expected objective i.e. $\texttt{EObj}_{i^*}$ and also satisfies the violation condition $\texttt{PCVC}_{i^*} \leq \eta$. If the aforementioned condition is true, i^* is obtained from the function **indexmin** over the vector EObj, otherwise, i^* with the least violating constraint is obtained from the function **indexmin** over the vector PCVC. Once the index for the optimal input (i^*) is obtained, $\mathcal{U}_{i^*}$ is injected to the system.

(3) **SIM** function:
   (a) The lines from 19 to 30 executes the MC simulation for the i^{th} input $\mathcal{U}_i$. The simulation consists of two **for** loops. The outer loop runs the MC simulation N_τ ($N_\tau = \gamma/N_s$) and the inner loop is dedicated for problem (9), where the stochastic ODE is simulated and if the constraints are violated, the counter variable CV registers the violation and the objective is numerically approximated by means of Riemann sum. It is also important to note that the total number of MC simulations for each input is $\gamma = N_s \times N_\tau$ and the total number of simulations for all the inputs is $N = \gamma \times n_g$.
   (b) The line 31 returns the average objective and constraint violation certificate with respect to N_τ simulations.

4.2 Scenario generation

As the SNMPC optimization problem in (9) is numerically solved by means of MC simulations, it is of paramount importance to sample enough number of scenarios to approximate the solution for the problem. Consider the task of approximating a stochastic function defined by

$$h(\mathbf{y}) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_\mathbf{x}}[\psi(\mathbf{x}, \mathbf{y})] \tag{10}$$

where, $\mathbf{x} \in \mathcal{X}$ is distributed w.r.t. the distribution $\mathbb{P}_\mathbf{x}$ over $\mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ (One can liken the $\mathbf{x}$ to the road disturbance scenarios and $\mathbf{y}$ with the damper inputs in the context of this paper). Let the empirical mean approximation of the function in (10) be defined with $\hat{h}(\mathbf{y})$. Let the parameters for accuracy, probability level and confidence level be defined with $\epsilon, \beta, \delta \in [0, 1]$. Given these parameters, the study of randomized algorithms is to derive a lower bound for the number of scenarios required to achieve the following requirement.

$$\mathbb{P}[\mathbb{P}[|\hat{h}(\mathbf{y}) - h(\mathbf{y})| > \epsilon] \leq \beta] \geq 1 - \delta, \ \forall \mathbf{y} \in \mathcal{Y} \tag{11}$$

Theorem 1. Choose the integers n_g and γ defined with

$$n_g \geq \frac{\ln(\frac{2}{\delta})}{\ln(\frac{1}{1-\beta})} \text{ and } \gamma \geq \frac{1}{2\epsilon^2} \ln \frac{4n_g}{\delta} \tag{12}$$

Generate i.i.d. samples $\mathbf{y}_1, \ldots, \mathbf{y}_{n_g} \in \mathcal{Y}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_\gamma \in \mathcal{X}$ according to $\mathbb{P}_\mathbf{x}$. Define

$$\hat{h}(\mathbf{y}_i) = \frac{1}{\gamma} \sum_{j=1}^{\gamma} \psi(\mathbf{x}_j, \mathbf{y}_i), i = 1 \ldots n_g \text{ and } \hat{h}^* = \min_{1 \leq i \leq n_g} \hat{h}(\mathbf{y}_i)$$

Then with confidence $1 - \delta$ it can be said that $\hat{h}^*$ is a probably approximate near the minimum $h(.)$ to accuracy ϵ and level β. The result is universal and applicable for all family of functions (Vidyasagar (2001)).

Using the results from Theorem 1, the empirical means in the SNMPC problem (9) both in the objective as well as the chance constraints can be numerically approximated. An important point to note is that the chance constraint can be recasted with expectation formulation with

$$\mathbb{P}[(\mathbf{X}(.),\mathbf{U}(.)) \notin \Omega_{(\mathbf{X},\mathbf{U})}] = \mathbb{E}[\mathbb{1}\{(\mathbf{X}(.),\mathbf{U}(.)) \notin \Omega_{(\mathbf{X},\mathbf{U})}\}] \tag{13}$$

where, $\mathbb{1}\{A\}$ represents the indicator function over the set $\{A\}$. Thus, by leveraging the results propounded in Theorem 1, the number of scenarios γ for Algorithm. 1 can be derived. Setting $\beta = \delta = 0.05$ yields the total number of input parameterization to $n_g = 64$. Utilizing the previous result and setting $\epsilon = 0.125$, the number of scenarios for each parameterized input is $\gamma \approx 270$. In total, the GPU simulates approximately 270×64 simulations over the prediction horizon at every sampling period.

5. RESULTS AND SIMULATIONS

5.1 RT embedded tests on NVIDIA boards

The test was conducted to study the RT viability of the proposed method and it involved execution of the proposed SS-pNMPC method (Algorithm. 1) with the aforementioned scenario parameters on multiple NVIDIA embedded boards as listed in Table 2. The key metrics which define the H/W performance of the board are a) Number of CUDA cores and b) Compute capability (CC) of the board i.e. the underlying architecture of the board (See Sanders and Kandrot (2010)). The performance of the method is gauged with the mean computation time (CT) and maximum computation time (CT) in terms of milliseconds (ms). The method was programmed in C++ using CUDA libraries. The histogram for the computational time for different NVIDIA embedded platforms is shown in Fig. 1. The natural sampling period (T_s) of the platform is 5

Table 2. NVIDIA boards H/W configuration and results

Board	#Cores	CC	Mean CT (ms)	Max CT (ms)
Nano	128	5.3	20.81	29.83
TX1	256	5.3	22.37	30.64
TX2	256	6.2	7.62	7.74
Xavier	512	7.2	6.04	6.59

ms and the best of embedded NVIDIA boards - Jetson Xavier hovers around 6 ms. Yet, the method is viable for the INOVE test platform as the optimal sampling period was ascertained to be 26 ms (See Rathai et al. (2019a)).

5.2 Pareto optimality of objectives

As mentioned in Section 3.3, the two objectives (comfort and ride handling) are conflicting in nature. Thus, the convex weights Γ_0^1 and Γ_0^2 ought to be tuned in order to strike a proper balance between the two objectives. To study the effects on variations of the convex weights, a Pareto optimality analysis was performed for several simulations. The simulation involved the vehicle moving at a constant velocity $v_x = 20m/s$ on a ISO-C road profile for a duration of 10s. In Fig. 2, the Pareto optimal front is plotted and this aids in detecting the right weights for the best results.

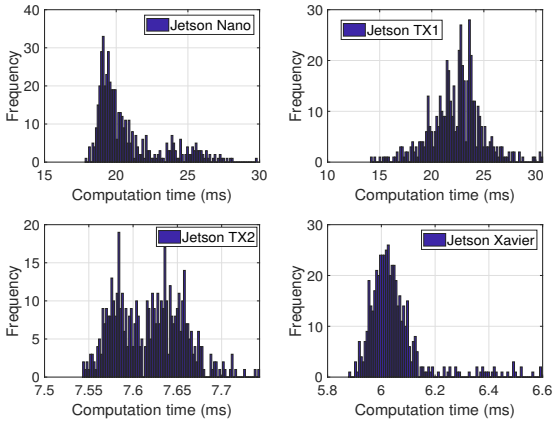

Fig. 1. Histogram of computation time on different embedded GPU platforms

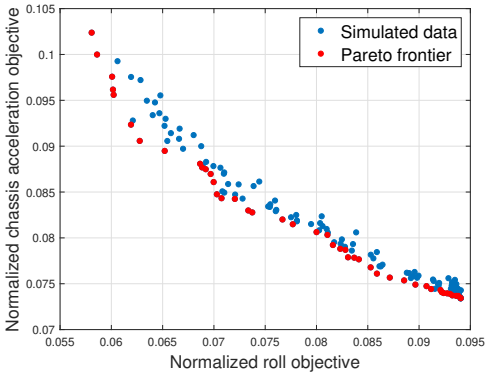

Fig. 2. Pareto optimal front between comfort and ride handling objective

5.3 Road profile simulation test

The simulation test involved the vehicle moving with longitudinal acceleration and velocity profile as shown in Fig. 3 and Fig. 4 on all the ISO road profiles. These profile correspond to the scenario where the vehicle is subjected to sudden braking. The convex weights used in the study are $\Gamma_0^1 = 0.75$ and $\Gamma_0^2 = 0.25$. Three passive controllers - $\underline{u_i}$, $\overline{u_i}$ and nominal $u_{n,i} = \frac{u_i + \overline{u_i}}{2}$, $\forall i \in \{l, r\}$ were chosen to compare the performance with the proposed controller.

The RMS values of the chassis acceleration for all the systems are listed in Table 3. It is clearly evident that the RMS value of the proposed method is less than the nominal passive system and in par with the minimum passive system (Comfort design). Also, at the same time, the proposed method minimizes the ride handling objective judiciously in comparison with other passive systems as shown in Fig. 5 for ISO-E road profile.

Table 3. RMS value of chassis acceleration

ISO	SS-pNMPC(ms^{-2})	$u_n(ms^{-2})$	$\underline{u}(ms^{-2})$	$\overline{u}(ms^{-2})$
A	0.012	0.014	0.011	0.017
B	0.023	0.029	0.021	0.034
C	0.049	0.058	0.042	0.076
D	0.093	0.118	0.087	0.138
E	0.194	0.234	0.178	0.277

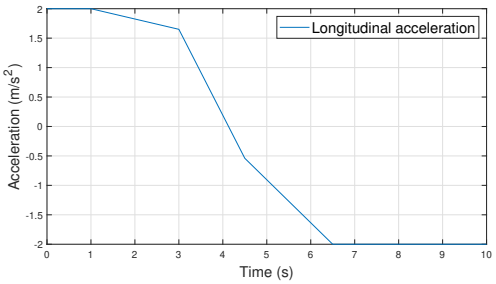

Fig. 3. Longitudinal acceleration profile

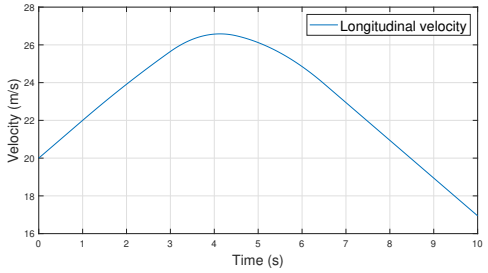

Fig. 4. Longitudinal velocity profile

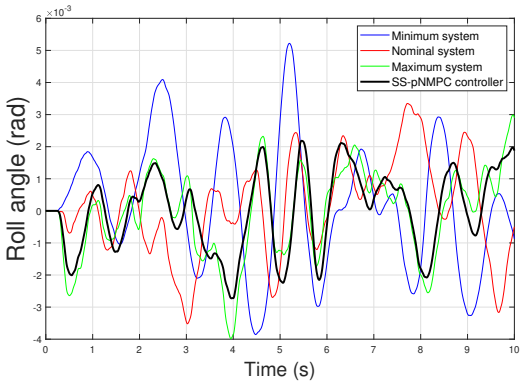

Fig. 5. Roll angle plot for different controllers for ISO-E road profile

6. CONCLUSIONS AND FUTURE WORKS

In this work, a scenario-stochastic parameterized NMPC (SS-pNMPC) method is proposed for a Half car semi-active suspension system. The method was simulated in MATLAB/Simulink environment and also, the RT feasibility of the method was validated and verified using embedded GPU boards. The method in general, provides a framework to implement highly nonlinear stochastic MPC by virtue of GPUs using the input parameterization technique. For the future line of work, the method will be implemented for full car suspension system and tested on the INOVE test platform at GIPSA lab.

REFERENCES

Abughalieh, K.M. and Alawneh, S.G. (2019). A survey of parallel implementations for model predictive control. *IEEE Access*.

Baumal, A., McPhee, J., and Calamai, P. (1998). Application of genetic algorithms to the design optimization of an active vehicle suspension system. *Computer methods in applied mechanics and engineering*, 163(1-4), 87–94.

de Normalización (Ginebra), O.I. (1995). *Mechanical vibration-road surface profiles-reporting of measured data*. ISO.

Doumiati, M., Martinez, J., Sename, O., Dugard, L., and Lechner, D. (2017). Road profile estimation using an adaptive youla–kučera parametric observer: Comparison to real profilers. *Control Engineering Practice*, 61, 270–278.

Guanetti, J. and Borrelli, F. (2017). Stochastic MPC for cloud-aided suspension control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 238–243. IEEE.

Ohyama, S. and Date, H. (2017). Parallelized nonlinear model predictive control on GPU. In *2017 11th Asian Control Conference (ASCC)*, 1620–1625. IEEE.

Rathai, K.M.M., Alamir, M., and Sename, O. (2019a). Experimental Implementation of Model Predictive Control Scheme for Control of Semi-active Suspension System. In *9th IFAC International Symposium on Advances in Automotive Control*. Orleans, France. URL `https://hal.archives-ouvertes.fr/hal-02160145`.

Rathai, K.M.M., Alamir, M., Sename, O., and Tang, R. (2018). A parameterized NMPC scheme for embedded control of semi-active suspension system. *IFAC-PapersOnLine*, 51(20), 301–306.

Rathai, K.M.M., Sename, O., and Alamir, M. (2019b). GPU-based parameterized NMPC scheme for control of half car vehicle with semi-active suspension system. *IEEE Control Systems Letters*, 3(3), 631–636.

Rogers, J. (2013). GPU-enabled projectile guidance for impact area constraints. In *Modeling and Simulation for Defense Systems and Applications VIII*, volume 8752, 87520I. International Society for Optics and Photonics.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., and Patrinos, P.P. (2017). GPU-accelerated stochastic predictive control of drinking water networks. *IEEE Transactions on Control Systems Technology*, 26(2), 551–562.

Sanders, J. and Kandrot, E. (2010). *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional.

Savaresi, S.M., Poussot-Vassal, C., Spelta, C., Sename, O., and Dugard, L. (2010). *Semi-active suspension control design for vehicles*. Elsevier.

Sename, O. (2019). Integrated approach for observation and control of vehicle dynamics (INOVE). `http://www.gipsa-lab.fr/projet/inove/index.html`.

Tyan, F., Hong, Y.F., Tu, S.H., and Jeng, W.S. (2009). Generation of random road profiles. *Journal of Advanced Engineering*, 4(2), 1373–1378.

Vidyasagar, M. (2001). Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37(10), 1515–1528.

Williams, G., Drews, P., Goldfain, B., Rehg, J.M., and Theodorou, E.A. (2016). Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1433–1440. IEEE.

Zhang, L., Yin, X., Shen, J., and Yu, H. (2017). Cloud-aided state estimation of a full-car semi-active suspension system. *arXiv preprint arXiv:1701.03343*.