Applied Soft Computing 120 (2022) 108646

Contents lists available at ScienceDirect

Applied Soft Computing

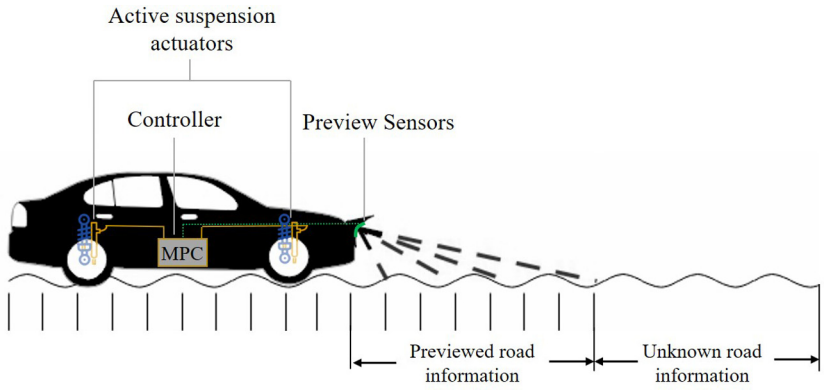
journal homepage: www.elsevier.com/locate/asoc

Active vehicle suspension control using road preview model predictive control and radial basis function networks

Myron Papadimitrakis, Alex Alexandridis *

Department of Electrical and Electronic Engineering, University of West Attica, Thivon 250, Aigaleo 12241, Greece

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 2 September 2021
Received in revised form 15 January 2022
Accepted 19 February 2022
Available online 1 March 2022

Keywords:
Active suspension
Chassis vibration control
Fuzzy means algorithm
Model predictive control
Radial basis function networks

ABSTRACT

Active suspension systems in road vehicles are applied in order to mitigate the road-induced chassis vertical accelerations more effectively than standard passive suspensions, thus increasing comfort and handling. Such systems are greatly assisted by road preview schemes, consisting of special sensors usually based on laser scanners (e.g. LiDAR sensors), which detect road irregularities ahead of the vehicle and feed this information to a control system, designed to manipulate the active suspension accordingly. In this paper, a model predictive controller (MPC) with road preview incorporating radial basis function (RBF) models, is presented as a control scheme for a full car active suspension system. The employed RBF models can efficiently approximate the nonlinear behavior of the suspension system, thus improving performance over linear MPC methods. Special care is taken to alleviate the increased computational complexity entailed in the RBF models, in order to ensure that online implementation of the controller is feasible. The proposed scheme is evaluated on a detailed simulated full car model under various road excitation types, while making use of a realistic approach for incorporating LiDAR road scanner noise. Comparisons to a passive suspension system, as well as a standard MPC controller with a fully linear plant model, demonstrate the performance potential of using RBF prediction models in a road preview MPC context.

1. Introduction

The suspension system of a conventional road vehicle serves to keep the wheels in a relative position with the chassis while traveling. The two main design objectives are ride comfort for passengers, which is directly linked to the vertical acceleration of the vehicle's chassis, and road holding capabilities, often expressed as a load variation on the vehicle's tire [1]. Active suspension systems are usually implemented in vehicles through a hydraulic system that powers a piston placed in parallel to a conventional spring and damper, allowing for the direct pursuit of

* Corresponding author.
E-mail address: alexx@uniwa.gr (A. Alexandridis).

https://doi.org/10.1016/j.asoc.2022.108646
1568-4946/

both objectives, i.e. ride comfort and vehicle handling [2]. The piston is controlled so as to exert desired forces on the wheel in the vertical direction. This system can achieve the aforementioned objectives in varying road conditions, as well as compensate for changes in the suspension dynamic behavior, which are inevitable through the life cycle of the vehicle [3]. The challenge of controlling the vertical dynamics of a vehicle is that its behavior is far from ideal. Notably, hydraulic and geometrical non-linearities, as well as cubic terms of displacement and velocity exhibited by the springs and dampers, are the main sources of this behavior; ignoring these characteristics can lead to sub-par results. In this respect, nonlinear control strategies need to be considered, so that the above requirements are addressed.

The development of such strategies has been a topic of research since 1970 and several methods have been proposed to date. In [4], an adaptive backstepping control method with a gray signal predictor used for the estimation of system states is applied for the integrated control of heave and pitch dynamics of a vehicle. A control method that also utilizes a state predictor is presented in [5], where a Kalman filter is used for the estimation of the road type, in order to toggle between energy-saving and high-performance control modes. A similar scheme is presented in [6] that takes the nonlinear dynamics of the actuators into account, albeit for a half-car suspension model. Various techniques that can circumvent the suspension nonlinearities have been built upon the classic methodology of skyhook damping. For example, in [7], a 6 DoF half-car active suspension is controlled by a combined skyhook damping and fuzzy logic controller, where the original non-linear actuator dynamics are approximated by a linear equation. In [8], an adaptive neuro-fuzzy inference controller is presented for the control of a full car active suspension. Initially, a fraction-order-PID controller is implemented, so that the necessary data for training the inference model can be generated. Then, the proposed controller is implemented on an FPGA module in order to accelerate computations. Other neural network-based approaches have been applied in feedback linearization control schemes [9,10] for the control of the full car active suspension vertical dynamics. Lastly, a fuzzy-PID control strategy is presented in [11] for the vibration control of a linear quarter-car active suspension, using an evolutionary computation algorithm for the optimization of the control parameters.

A control scheme that can directly accommodate for plant nonlinearities is model predictive control (MPC) [12]. Due to this advantage, along with its abilities to handle MIMO formulations, various physical constraints [2], and multiple objectives [13], MPC has emerged as a highly successful control scheme with applications in various fields during the last decades [12]. The MPC algorithm makes use of a dynamic model of the plant in order to calculate an approximation of the plant's response to the control inputs. Based on this model, a constrained optimization problem is formulated online to obtain the optimum sequence of control moves for a given time horizon. It is apparent that the MPC scheme effectiveness relies on the prediction accuracy of the plant model, as model inconsistencies can lead to poor choices regarding the control moves. Thus, in cases of highly nonlinear plant dynamics, it is a viable strategy to consider nonlinear plant models [14]; on the other hand, such an approach adds a significant computational load in solving the optimization problem. Due to this reason, nonlinear MPC approaches are usually coupled with techniques for alleviating the increased computational burden, like online linearization [15,16].

Though, as stated earlier, the vertical vehicle dynamics exhibit significant nonlinearities, up to now it was not the ability of MPC to incorporate nonlinear models, but rather its other merits, that have made it popular for active suspension design. An additional motivation is that MPC is suitable for integrating a road preview scheme, which can feed the active suspension control scheme with valuable information (Fig. 1). To be more specific, the vehicle can scan the road ahead using appropriate sensors, and supply this information to the controller, thus greatly enhancing performance [13]. The performance potential of MPC preview schemes has been confirmed for active suspensions in terms of energy expenditure [17]. In real-world applications, this preview technology is realized with LiDAR sensors, which, nevertheless exhibit a varying degree of measurement noise [18]. In [2] an MPC controller with preview (MPC-P) was applied to the active suspension full car problem to minimize roll and heave accelerations under several constraints, using a reduced model of the plant. Even though the wheel dynamics are not included in the MPC model on the basis that they exhibit higher frequency dynamics than the actuator, the proposed scheme achieves satisfactory performance. It should be noted that the hydraulic actuator dynamics are not included in the modeling stage but are substituted by the hydraulic actuators' displacement — this significantly reduces the plant nonlinearity, since hydraulic fluid flow through the actuator is a nonlinear phenomenon. In [13], an MPC-P controller with a linear model is created based on a simplified quarter car plant and compared to a skyhook-damping controller in a real world application.

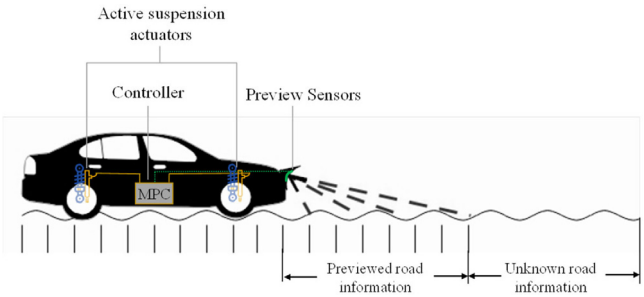


Fig. 1. Active suspension with a road preview MPC controller.

Even though the results reported in these works are very encouraging, the potential of using MPC for active suspension design is not yet fully reaped, as the employed linear models cannot account for the nonlinearities present. Methods based on computational intelligence, e.g. neural networks (NNs), seem to be an attractive alternative, as they are very effective in modeling nonlinear plants [19] and can be configured to accommodate for changes in plant parameters. Radial basis function networks (RBFN) in particular are widely considered for modeling nonlinear dynamics, mainly because of their simple structure and increased accuracy [20]. As these advantages are of paramount importance in the context of predictive control, RBFNs constitute a popular choice in conjunction with MPC [21].

In RBFNs, learning is performed by fitting a multidimensional surface to a set of training examples. This surface consists of simpler surfaces with radial basis symmetry around centers located in the input space. The number and location of these centers are critical parameters that affect the performance of the produced RBFN, and various algorithms have been proposed for their determination. A preliminary approach involved setting the training examples as centers, which unfortunately leads to excessively complicated models [22]. To overcome this, a clustering algorithm can be employed to reduce the number of required centers. One such approach is the *k*-means algorithm, which however is shown to be computationally intensive, while requiring a trial-and-error procedure for determining the number of centers [23]. The fuzzy means (FM) algorithm [24] on the other hand, allows for a fast selection of the number of centers and their locations, by utilizing a fuzzy clustering approach.

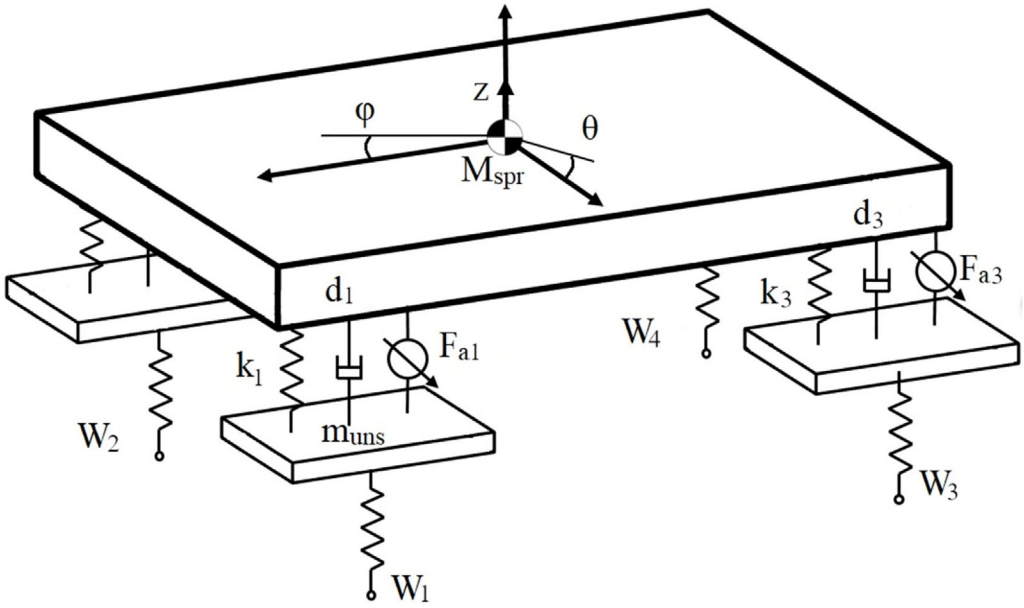


Fig. 2. 7 DoF full car model.

It should be noted that RBF networks have been used extensively in automotive system modeling and control. In [25], an RBFN is used for the modeling of the wheel-slip dynamics of an anti-lock braking system. In [3], adaptive RBFNs are applied for modeling the lateral and longitudinal dynamics of a 3 DoF vehicle model, for usage in a highway lane tracking PD-controller. In [21], an RBF network is used to explicitly model an MPC controller for the control of a semi-active suspension. Surprisingly, it seems that no RBFN approaches for modeling the vertical dynamics of a full car model have been reported in literature.

The main contributions of this work are the following:

- A method for developing a full car vertical dynamics model based on RBFNs is introduced. The new approach (a) is purely data-driven and does not make use of cumbersome first-principle equations, (b) can take into account plant nonlinearities, and (c) is computationally efficient as it makes use of the FM algorithm for training the networks.
- A nonlinear MPC framework is introduced for active suspension design, making use of road preview information. Nonlinearities that arise when modeling the vertical chassis dynamics of the full car are approximated with the aforementioned RBFN model. To the authors' best knowledge, no MPC-P full car active suspension control scheme that directly accommodates for model nonlinearities exists, much less one that applies RBFN models.
- A new method for initializing the MPC optimization problem using an inverse model of the plant [26], which is also based on RBFNs, is used in order to minimize the computational burden needed for calculating the control actions, without compromising the model's predictive abilities. This is an important practical consideration, so as to ensure real-time implementation of the proposed approach.
- A realistic representation of noise induced by the road preview functionality is devised, based on studies regarding LiDAR sensors [18,27]. The aim is to simulate the effect of road preview inaccuracies on the MPC controller performance, in order to confirm the robustness of the proposed scheme.

The rest of this paper is structured as follows: In the next section, the plant equations of the full car and electrohydraulic actuating system are presented, together with the LiDAR road preview scheme. Section 3 starts with an introduction to the RBFN architecture and the FM algorithm and continues with a detailed description of the modeling approach followed and the produced results. Section 4 introduces the proposed controller, including a description of the MPC formulation and a discussion about expediting the solution to the optimization problem through an appropriate solution initialization technique. Then, in Section 5 the case study is presented in detail, containing a description of the setup and the controller tuning procedure, followed by the produced results and relative discussion. Finally, the paper ends with concluding remarks and directions for future work.

2. Plant description

2.1. Electrohydraulic system

The electrohydraulic piston–valve system is the powerhouse of an active suspension, since it is where the control force originates from. Hydraulic fluid is pumped from the oil sump by the axial pump and raised to the supply pressure. When the power servo valve is open, high pressure fluid flows to either one of the actuator chambers, while low pressure fluid flows from the other one back to the sump. Between the two actuator chambers a pressure difference is created, which results in the actuator force. The dynamics of the actuator pressure are highly nonlinear [28], since they are nonlinearly related to the hydraulic fluid flow through the servo valve.

2.2. Full car active suspension

The vertical dynamics of the automobile can be approximated using a 7 DoF full car model. The model is based on an abstraction of the actual car, which is modeled as a rectangle representing the sprung inertia, with four masses at each corner, amounting to the unsprung masses, as shown in Fig. 2. This model can describe the heave z, roll θ and pitch φ modes of the sprung mass m_{spr}, as well as each unsprung mass heave displacement z_w. Each unsprung mass m_{uns} can be displaced along the vertical axis, and is connected to the sprung mass by a spring k, a damper d and a hydraulic actuator that exerts a force of F_a in parallel. The tire stiffness is modeled as a spring k_t between each unsprung mass and the road profile w_i. Each actuator is described by a

Table 1
Full car state equations and functions.

A. State differential equations	
$\dot{x}_{i,1} = x_{5,1}(b_{track}/2)(-1)^i + x_{5,2}(L_{base}/2)s(i) + x_{5,3}$	(1)
$\dot{x}_{i,2} = x_{i,4}$	(2)
$\dot{x}_{i,3} = \dot{x}_{5,1}(b_{track}/2)(-1)^i + \dot{x}_{5,2}(L_{base}/2)s(i) + \dot{x}_{5,3}$	(3)
$\dot{x}_{i,4} = (1/m_{unsp})\left(-F_{ks}\left(x_{i,1}, x_{i,2}\right) - F_{bs}\left(x_{i,3}, x_{i,4}\right) - F_{kt}\left(w_i, x_{i,2}\right) + A_p x_{i,5}\right)$	(4)
$\dot{x}_{i,5} = a\left[Q\left(x_{i,6}, x_{i,5}\right) - C_{tm}x_{i,5} - A_p\left(x_{i,3} - x_{i,4}\right)\right]$	(5)
$\dot{x}_{i,6} = (1/\tau)\left(-x_{i,6} + u_i\right)$	(6)
$\dot{x}_{5,1} = (b_{track}/2Ixx)\left(F_{strut}(2) + F_{strut}(4) - F_{strut}(1) - F_{strut}(3)\right)$	(7)
$\dot{x}_{5,2} = (L_{base}/2Iyy)\left(F_{strut}(1) + F_{strut}(2) - F_{strut}(3) - F_{strut}(4)\right)$	(8)
$\dot{x}_{5,3} = (1/m_{spr})\left(F_{strut}(1) + F_{strut}(2) + F_{strut}(3) + F_{strut}(4)\right)$	(9)
$\dot{x}_{5,4} = x_{5,1}$	(10)
$\dot{x}_{5,5} = x_{5,2}$	(11)
$\dot{x}_{5,6} = x_{5,3}$	(12)
B. Functions	
$s(i) = \{-1\ i \le 2, 1\ i > 2\}$	(13)
$Q\left(x_{i,6}, x_{i,5}\right) = C_{dP}S_{xv}x_{i,6}\sqrt{(P_s - sign\left(x_{i,6}\right)x_{i,5})/\rho}$	(14)
$F_{bs}\left(x_{i,3}, x_{i,4}\right) = b_s^{lin}\left(x_{i,4} - x_{i,3}\right) - b_s^{sym}\lvert x_{i,4} - x_{i,3}\rvert + b_s^{nonlin}\sqrt{\lvert x_{i,4} - x_{i,3}\rvert}\,\mathrm{sign}\left(x_{i,4} - x_{i,3}\right)$	(15)
$F_{ks}\left(x_{i,1}, x_{i,2}\right) = k_s^{lin}\left(x_{i,2} - x_{i,1}\right) + k_s^{nonlin}\left(x_{i,2} - x_{i,1}\right)^3$	(16)
$F_{kt}\left(x_{i,2}, w_i\right) = k_t\left(x_{i,2} - w_i\right)$	(17)
$F_{strut}(i) = F_{ks}\left(x_{i,1}, x_{i,2}\right) + F_{bs}\left(x_{i,3}, x_{i,4}\right) - A_P\ x_{i,5}$	(18)

valve displacement $x_{i,6}$ and a valve input u_i. The inputs to the full car plant are the four valve inputs and the four road profiles for each wheel. Table 1 depicts the derivative state equations for the model; for simplicity, numbered notation is adopted for each state, or input variable. The index $i \in \{1, 2, 3, 4, 5\}$ corresponds to {*Front Left, Front Right, Rear Left, Rear Right, Chassis*}. The index $j \in \{1, 2, 3, 4, 5, 6\}$ denotes a specific state, as shown in Table 1. For example, $x_{1,5}$ indicates the actuator pressure state of the front left wheel. The values employed for model parameters are shown in Table 2; these values were chosen to provide a realistic full car representation based on [28,29]. Note the nonlinear spring and damping terms.

In short, the full car active suspension plant can be described at each continuous time instant t as a system comprised of 30 states, stored in state matrix $\mathbf{X}$:

$$\mathbf{X}(t) = \begin{bmatrix} x_{1,1}(t) & \dots & x_{1,6}(t) \\ \vdots & \ddots & \vdots \\ x_{5,1}(t) & \cdots & x_{5,6}(t) \end{bmatrix} \tag{19}$$

that accepts as input a vector of each wheel's control signal $\mathbf{u}$:

$$\mathbf{u}(t) = [u_1(t),\ u_2(t),\ u_3(t),\ u_4(t)] \tag{20}$$

and a road profile vector $\mathbf{w}$, consisting of the load profile for each separate wheel:

$$\mathbf{w}(t) = [w_1(t),\ w_2(t),\ w_3(t),\ w_4(t)] \tag{21}$$

2.3. Road preview

In order to measure the real road height profile vector $\mathbf{w}(t)$, onboard LiDAR sensors that scan the road ahead can be used, as in [2,13]. A very detailed description about the practical aspects of road previewing in active suspensions takes place in [30], while a real-world implementation is given in [27]. LiDAR sensors are subject to a variety of measurement errors, mainly arising from photon interference and scattering [18]. In order to model such errors, an additive Gaussian noise to all measurements is assumed in [30]. However, it has been observed that LiDAR measurement errors can vary in magnitude, in relation to the distance of the measured profile, e.g. an almost linearly-increasing error in relation to distance for a kinematic scanning LiDAR is reported in [31]. A simple experiment is set up in [27], and an increasing relative error in relation to distance is also observed. In order to provide a more realistic representation of measurement error induced by the LiDAR sensor, we assume an additive Gaussian noise for the previewed road, with increasing standard deviation as the distance from the sensor increases:

$$w_m(d) = w_r(d) + L\left(0, \sigma^2(d)\right), \quad d \in [0, l_p] \tag{22}$$

Here, w_m denotes the road profile height as measured by the LiDAR sensor, w_r is the real road profile height at distance d from the sensor, l_p is the preview length, and L is a random value sampled from a Gaussian distribution, where the standard deviation σ is a linear function of d. An example of the measured road w_m is shown in Fig. 3.

Table 2
Full car plant and electrohydraulic system parameters.

Symbol	Description	Value	Units
P_s	Supply Pressure to Valve	$5\ 10^6$	Pa
A_p	Actuator Piston Area	$3.35\ 10^{-4}$	m^2
C_{tm}	Leakage constant	$22\ 10^{-11}$	–
α	Hydraulic Coefficient	$4.51\ 10^{12}$	N/m^5
ρ	Density of Hydraulic Fluid	850	kg/m^3
τ	Time constant of valve	$5\ 10^{-3}$	s
C_d	Discharge Coefficient	0.7	–
S_{xv}	Valve width	$4.1\ 10^{-3}$	m
b_s^{lin}	Linear damping coefficient	2000	N s/m
b_s^{sym}	Symmetric damping coefficient	400	N s/m
b_s^{nonlin}	Nonlinear damping coefficient	100	$N\ (m/s)^{-1/2}$
k_s^{lin}	Linear spring coefficient	25 000	N/m
k_s^{nonlin}	Nonlinear spring coefficient	$25\ 10^4$	N/m^3
k_t	Tire spring coefficient	$19\ 10^4$	N/m
b_{track}	Wheel track width	2.5	m
L_{base}	Wheelbase	4	m
Ixx	X moment of inertia	550	kg m^2
Iyy	Y moment of inertia	300	kg m^2
m_{spr}	Sprung mass	1800	kg
m_{uns}	Unsprung mass	40	kg

3. Modeling

3.1. Radial basis function networks & the fuzzy means algorithm

RBFNs constitute a popular NN architecture with numerous applications in nonlinear system identification and control, both

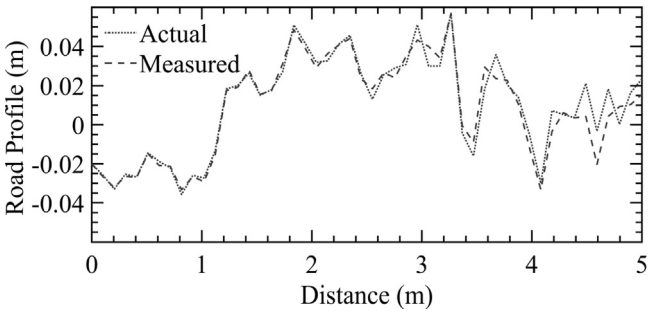


Fig. 3. 1.5 s of scanned-ahead road, in a road preview MPC context. The relative error increases with the distance from the LiDAR sensor.

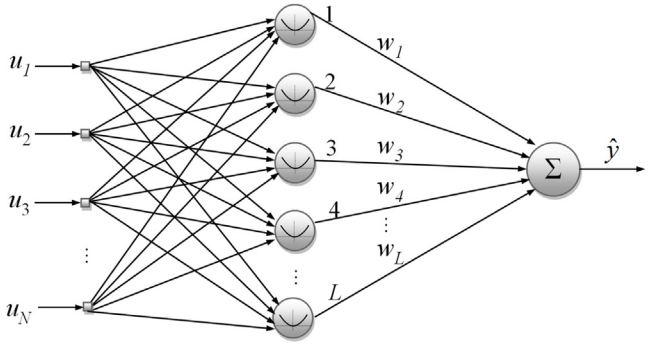


Fig. 4. Schematic of an RBF network: The input layer distributes N input variables to L nodes of the hidden layer. Each node of the hidden layer is comprised of a center with N dimensions. The hidden layer performs a nonlinear transformation that maps the input space on a new, higher dimensional space. The weighted sum of the nodes constitutes the output.

in industry and academia [32]. They are comprised of a single hidden layer which is linearly attached to the output layer of the network (Fig. 4). As a result, the employed training algorithms are faster and more efficient than their more complicated multilayer perceptron (MLP) counterparts [20]. In RBFN, the training procedure aims to find the multidimensional surface that best approximates the training data, which is constructed as a sum of simpler surfaces exhibiting radial basis symmetry around centers specifically placed in the input space.

Training an RBFN corresponds to finding the number of RBF centers, their coordinates in the input space, and the weights connecting the hidden layer to the output layer. A standard approach to determine the coordinates of the RBF centers involves the k-means algorithm [23], for a given number of centers; however, as this number is not a priori known, a tedious trial-and-error procedure is required to determine it. This procedure can be avoided by using an alternative approach known as the fuzzy means (FM) algorithm [24], which presents the ability to determine the number of nodes automatically through fuzzy clustering, while at the same time calculating center locations. The resulting number of nodes depends solely on the number of fuzzy sets s, which defines the density of the grid of potential centers. This enables a tractable tuning procedure of an RBFN through the FM algorithm, since the optimum number of fuzzy sets can be easily found by exhaustive search in a narrow discrete region.

The FM algorithm has found successful applications in diverse scientific fields [20], owing its success to its computational efficiency, as well as the produced model accuracy.

Table 3
Model comparison statistics.

States	Linear model		RBFN model			
	R^2	MAE	# of centers	# of fuzzy sets	R^2	MAE
$x_{1,1}$	0.9960	0.0021	–	–	–	–
$x_{1,2}$	0.9990	6.1e−4	–	–	–	–
$x_{1,3}$	0.9855	0.0371	–	–	–	–
$x_{1,4}$	0.9893	0.0287	5121	9	0.9910	0.0156
$x_{1,5}$	0.9652	0.5159	8131	12	0.9908	0.2425
$x_{1,6}$	0.9998	3.36e−5	–	–	–	–
$\dot{x}_{5,1}$	0.9865	0.3213	5121	9	0.9920	0.1656
$\dot{x}_{5,2}$	0.9909	0.1021	5121	9	0.9946	0.0628
$\dot{x}_{5,3}$	0.9875	0.1204	5121	9	0.9918	0.0852
$x_{5,4}$	0.9924	0.0014	–	–	–	–
$x_{5,5}$	0.9962	5.2e−4	–	–	–	–
$x_{5,6}$	0.9993	5.0e−4	–	–	–	–

The metrics were generated from a random road testing run. Only the results of the front left wheel are presented; the rest of the wheels are omitted because they exhibit almost identical metrics.

3.2. Model creation

The general mathematical notation for a discrete model f of the full car plant is:

$$\hat{\mathbf{X}}(k+1) = f(\mathbf{X}(k), \mathbf{u}(k), \mathbf{w}(k)) \tag{23}$$

$$\mathbf{w}(k) = [w_1(k),\ w_2(k),\ w_3(k),\ w_4(k)] \tag{24}$$

$$\mathbf{u}(k) = [u_1(k),\ u_2(k),\ u_3(k),\ u_4(k)] \tag{25}$$

The model f accepts as inputs the current state matrix $\mathbf{X}(k)$ (19), the current control vector $\mathbf{u}(k)$ and the current road profile vector $\mathbf{w}(k)$ at discrete time k, and generates the prediction for the state matrix $\hat{\mathbf{X}}(k+1)$ at the next discrete instance. In order to approximate the function matrix f and build a discrete data-driven model of the full car plant, a suitable input–output dataset must be created. The input data are comprised of a random road signal to the four wheels and a random valve displacement signal to the four actuators. The road signal is generated as per the ISO 8606 standard that specifies the power spectral density of random road profiles of different quality [33]; for the task at hand, a type 'E' road is chosen. The valve displacement signal is generated from a uniform distribution within the valves' operating range, namely ± 10 mm. The aforementioned input is applied on the full car plant, and its output, which consists of the state variables as described by (19), is used to create an input–output dataset of 40 000 datapoints. The dataset is subsequently split in a 50-25%-25% manner, in order to create the training, validation and testing subsets, respectively; the first subset is used for calculating the model parameters, the second for model selection and the third for independently evaluating model performance.

Once those datasets are acquired, least squares regression on the training subset can be used in order to obtain the parameters for a discrete linear state model. The goodness of fit is evaluated on the testing subset using the coefficient of determination (R^2) and the mean absolute error (MAE). The results are shown in Table 3; as expected, the states (or their derivatives for the chassis modes) that contain nonlinear terms, namely $x_{i,4}, x_{i,5}, \dot{x}_{5,1}, \dot{x}_{5,2}$ and $\dot{x}_{5,3}$, fail to be modeled sufficiently.

Having in mind that the produced model will be incorporated in a control scheme, it is important to note that the $x_{i,5}$ state corresponds to the actuator pressure of each wheel (and thus to the control force), hence it is directly related to the effectiveness of the applied control actions (5). If the MPC controller over/underestimates it after a control move, then the control performance will be directly compromised. Similar concerns are raised for the $\dot{x}_{5,1}$ state derivative, as it represents

the controlled variable, i.e. the heave acceleration. Lastly, state derivatives $\dot{x}_{5,2}, \dot{x}_{5,3}$, which concern roll and pitch accelerations, indirectly influence all plant states. Correctly estimating these particular quantities can have a significant effect to the overall model performance, but unfortunately the respective equations are nonlinear in nature; this is apparent from (1)–(12), but also confirmed in practice, as shown in Table 3. For this reason, RBFN dynamic models were used for the aforementioned states (or state derivatives), while for the rest of the states, linear models were applied, in order to reduce the computational cost of calculating the overall model response; it should be noted that when used for MPC, this model will be evaluated multiple times when solving the optimization problem, in order to calculate each control action.

A main consideration when training the RBFN models using the FM algorithm is the choice of the number of fuzzy sets s. The performance of an RBFN model will increase with increasing fuzzy sets, until it begins to overfit on the training data, thus compromising its generalization ability, i.e. its performance on new data. Overfitting occurs because the increasing network complexity allows for modeling the noise present in the training data [19], a fact which bears a detrimental effect on the network's ability to make accurate predictions for data not included in the training subset. In order to establish the optimum value for s, a validation data subset must be also employed; to be more specific, a different network is trained for each value of s in the range {4–20} using only the training subset, and then the value of s that maximizes the performance on the validation subset is chosen [24].

The results for the individual RBFN models used on the independent testing subset are shown in Table 3. These include the number of fuzzy sets and centers generated from the proposed training procedure, as well as prediction metrics on a random road testing subset. It is clear that the one-step ahead prediction performance of the individual RBFN models is superior to their linear counterparts, when modeling the highly nonlinear states $x_{i,4}, x_{i,5}, \dot{x}_{5,1}, \dot{x}_{5,2}, \dot{x}_{5,3}$. Thus, two full car models are created; one with linear models for all the states, and one applying RBFN models for the nonlinear ones.

The two resulting car models (hereby referred to as "linear" and "RBF-linear") are essentially one-step-ahead prediction models, but their performance when predicting multiple steps ahead must be also assessed, in order to evaluate their suitability for integration in an MPC context. This can be accomplished by recurrently applying the model, each time using as input states the state predictions produced by the model for the previous time step. Fig. 5 compares the 35-step ahead prediction performance of the two models for a random road profile and control input signal, selected from the aforementioned testing subset. The RBF-linear car model clearly outperforms the linear car model for multiple-step-ahead prediction of the controlled variable (chassis heave acceleration), as the former achieves a MAE of 0.1082 and an R^2 of 0.9812, while the latter a MAE of 0.1483 and an R^2 of 0.9528.

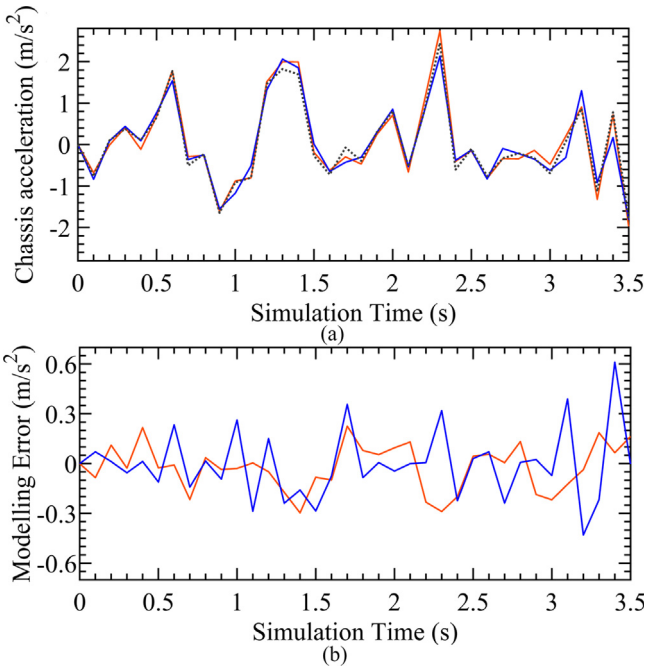


Fig. 5. Linear and RBF-linear car model comparison for multiple-step-ahead evaluation. (a) heave acceleration modeling comparison, (b) modeling error comparison.

4. Proposed MPC approach

4.1. Model predictive control scheme

The proposed active suspension control scheme involves using an MPC controller, in conjunction with road preview information. In MPC, the idea is to calculate the optimal control moves through the real time solution of a constrained optimization problem. The optimization problem is formulated at every discrete time instance k on a rolling time window comprised of two horizons; the first is the prediction horizon h_p, accounting for the number of time steps that the model will be used for future predictions, and the second is the control horizon h_c, representing the number of time steps that the control moves are allowed to change. Therefore, the objective function typically contains two terms, the first concerning the minimization of the deviations between model predictions for the controlled variables and their respective setpoints, over horizon h_p, and the second concerning the minimization of the total control actions over horizon h_c. The design variable vector contains the sequence of control actions $\mathbf{u}$ for every timestep up to horizon h_c, and the solution of the optimization problem yields the optimal sequence of control actions $\mathbf{u}^*$ that minimize the objective function for the current rolling horizon window.

There exist numerous formulations for the active suspension problem, as far as the design variables in the MPC objective function are concerned. In [13], wheel load and suspension displacement are explicitly minimized. This work aims to minimize the chassis heave acceleration, as the proposed scheme is concerned with vehicle passengers comfort [34].

At each discrete time point k, the following optimization problem is formulated:

$$\min_{\mathbf{u}(k),\mathbf{u}(k+1),\ldots,\mathbf{u}(k+hc-1)} \theta \sum_{j=1}^{h_p} \left[\left(\hat{x}_{5,3}(\mathrm{k}+j)\right) \Theta_j\right]^2 + \omega \sum_{i=1}^{4} \left[\sum_{j=1}^{h_c} \left[\left(u_i(\mathrm{k}+j)\right) \Omega_j\right]^2\right] \tag{26}$$

subject to:

$$\mathbf{u}_{\min} \leq \mathbf{u}\,(k+j) \leq \mathbf{u}_{\max},\ 1 \leq j \leq h_c \tag{27}$$

$$\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}\,(k+j) \leq \Delta\mathbf{u}_{\max},\ 1 \leq j \leq h_c \tag{28}$$

The first term of (26) is the setpoint deviation term, which, for this formulation, is the sum of the Θ-weighted squares of the chassis heave acceleration predictions $\hat{x}_{5,3}$ produced by the

RBF-linear model. The second term is the control move penalty term, which contains the sum of the Ω-weighted squares of the control moves $\mathbf{u}$. Θ and Ω are vectors, used to assign a different weight for each time step of the horizons h_p and h_c, respectively. The two terms are weighted by ω and θ factors, respectively. Inequalities (27), (28) represent physical constraints of the system, and describe input saturation and rate of change limits, respectively. In a real implementation, there are spool valve displacement and velocity limits, which constrain the input space; in this implementation these are chosen as ± 10 mm and ± 30 mm/s, respectively [28].

The optimization problem attempts to find the optimum sequence of control moves over the control horizon h_c, that will minimize the weighted sum of squares of the setpoint deviation and the control move penalty terms over the horizon h_p. Once the MPC optimization finds the optimal sequence of control moves $\mathbf{u}^*$, the controller applies the first control move $\mathbf{u}^*(1)$, for the current discrete timestep. It then reruns the optimization procedure for the next discrete timestep, formulated with the newly-sampled initial conditions.

4.2. Expediting the solution to the optimization problem

To ensure real-time implementation of the proposed control scheme, the optimization problem should be solved within one sampling time period, thus enabling the controller to calculate new actions for each discrete time step. In addition, in order to accurately track the fast dynamics imposed to the car by the high frequency road excitations, the controller sampling period should be kept as small as possible [35]. In the case of MPC controllers, the sampling time is essentially dictated by the time needed to solve the optimization problem, this being the bottleneck in the control action calculation procedure. Unfortunately, notwithstanding the improved approximation capabilities throughout the whole operating region offered by an RBF model, its use in MPC transforms the optimization problem to a nonlinear one, thus bearing a negative effect on the time required for obtaining a satisfactory solution.

A standard approach for speeding up the optimization procedure in MPC, is to initialize the optimization problem formulated during each time step with the optimal control moves $\mathbf{u}^*$ resulting from the previous time step, after removing the first move which has already been implemented. However, this technique may not be adequate for the particular application; the fast-changing road conditions may in turn cause the car's state to change rapidly, thus rendering the previous time step solution obsolete. In this case, the optimizer starts from a poor initial value, which can lead to longer computational times for solving the problem.

In this work, a more elaborate initialization technique is used, which has been shown to significantly expedite the solution procedure [26]. To be more specific, an inverse RBF model of the plant is trained offline. This model can be then applied at each time step to provide a sequence of actions $\mathbf{u}_{inv}$ for the entire control horizon; calculating $\mathbf{u}_{inv}$ is very fast, because no optimization procedure is involved. On the other hand, it should be noted that $\mathbf{u}_{inv}$ offers only a feasible trajectory to the setpoint, without taking into account the aspect of optimality; though directly applying it to the real plant could yield subpar results, it can be used as a start point for the MPC optimizer. Thus, for every new formulation of the MPC optimization problem occurring at each discrete time step, a suboptimal solution is first obtained using the inverse model, and then passed on to the optimization solver as an initial guess. More details about this type of initialization can be found in [26].

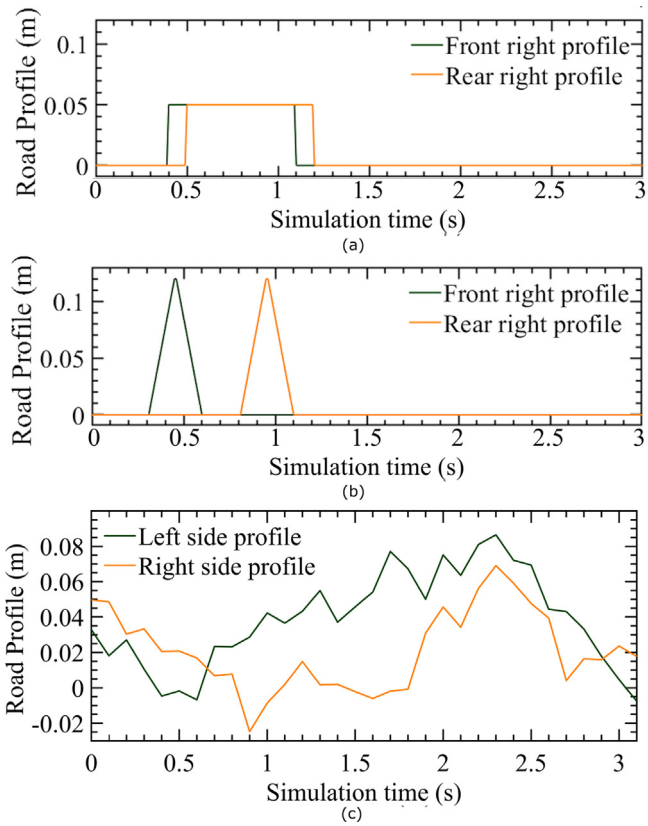


Fig. 6. Road scenarios for simulation: (a) Right-sided pulse bump, (b) Right-sided ramp bump, (c) Random road profile.

5. Case study

In this case study, the proposed RBFN-MPC approach is evaluated on a simulated full car plant, under various road excitation types. A linear MPC-P controller and a passive full car suspension are also presented for comparison purposes.

5.1. Setup

The full car plant described in Section 2 is simulated by numerically solving the state equations (1)–(12) using a Runge–Kutta 4–5 formula [36]. Two MPC controllers were created, hereby referred to as MPC-L and MPC-RBF-L, employing the linear and RBF-linear full car models, respectively. In both controllers, the MPC optimization problem was solved using an active-set method, as described in [26]. All simulations were realized in MATLAB environment and a desktop computer with an Intel i9-9960X CPU and 64 GB of RAM. The sampling time for both controllers was set to 100 ms of simulated time.

The control objective is to maximize passenger comfort. This could be taken as equivalent to minimizing the heave acceleration, as is done in [2]. However, different, more sophisticated performance metrics for comfort quantification also exist. ISO 2631 [37] sets standards for exposure of humans to vibration, and applies not only to vehicles, but to all vibrating environments as well. According to this standard, humans are more sensitive to heave acceleration in the range of 4–8 Hz. A more road-vehicle-oriented discussion about passenger comfort criteria takes place in [34]. The authors also note that vehicle comfort should be evaluated both on bump and random road tests, for completeness' sake.

In light of the above, it was decided to compare the competing schemes in three different tests (Fig. 6): a one-sided pulse bump

with a height of 5 cm, a one-sided symmetric ramp bump with a maximum height of 12 cm and a duration of 0.3 s, and a 30 m stretch of random road of class "E" [33], traversed at 10 m/s vehicle speed. For the bump tests, importance is placed on the max absolute acceleration (MAA), the sum of absolute errors (SAE) and the settling time; these are deemed important since any changes to them are directly perceived by the passengers. For the random road test, besides the sum of absolute errors, the fast Fourier transform (FFT) of the heave acceleration is computed, since it gives a clearer picture of controller performance, together with the squared heave displacement, as dictated by the Steffens comfort metric [34]:

$$X\left[\mathrm{cm}^2\right] = 7.62 \cdot 10^{-3}\left(1 + \frac{125}{f_c^2}\right) \tag{29}$$

X is the square of the chassis displacement in cm, and f_c is its frequency in Hz. The Steffens criterion sets a bound on the FFT of the chassis heave displacement, which signifies discomfort. Thus, the furthest away the frequency response of the chassis displacement is from that bound, the better the vehicle's suspension is performing.

Lastly, an important consideration must be made regarding the road holding capability of the proposed scheme. This capability can be quantified by the vertical acceleration of each wheel, which should not be too high, otherwise the tire contact patch may be perturbed or completely detached from the road, resulting in loss of road holding [1,38]. While increasing ride comfort remains as the sole control objective of the proposed scheme, it must be verified that the road holding capability does not deteriorate in comparison to the passive suspension case. Therefore, the average root mean square (RMS) values of the vertical acceleration of each wheel are calculated for the random road case, as it is the case most likely to cause such a phenomenon.

5.2. MPC tuning

Tuning of the two MPC controllers amounts to determining the optimum values for standard MPC parameters, namely the prediction and control horizons h_p and h_c, the $\boldsymbol{\Theta}$, $\boldsymbol{\Omega}$ vectors and the objective function weights ω and θ, but also for the road preview length l_p. The reason that l_p constitutes a tuning parameter is that the previewed road information quality deteriorates over distance due to noise, as discussed in Section 2.3; so, there exists a maximum effective length of previewed road that is useful for the MPC controller.

A low value for the h_c parameter is chosen a priori, since it will result to a lower number of control variables to be optimized online, and thus, to a faster solution of the MPC problem. The rest of the tuning parameters are optimized offline using the particle swarm optimization (PSO) method [39]. PSO is a favorable choice because of its effectiveness in solving nonlinear optimization problems, in conjunction with its simplicity. The tuning procedure was run on two cases: the first involves a one-sided pulse bump of 8 cm height, and the second a random road of class "E". The tuning optimization objective for both cases is the minimization of the sum of absolute heave acceleration over the timespan of the simulation. The results of the tuning procedure for the bump and random road cases for both controllers are shown in Tables 4 and 5, respectively, which contain the selected values for the tuning parameters described in the first paragraph of this subsection. Using two different tuning scenarios aims to cater to the differences of two distinct types of roads that can occur – a well-paved road with the occasional anomaly (bump case), or a badly paved road or dirt track (random road case). Note that this practice does not violate the practicality of the proposed scheme in a real-world implementation; the controller could toggle between on-road and off-road modes either manually or automatically [5], as is the case in most modern vehicles with adjustable suspension today.

Table 4
Bump road tuning parameters.

	MPC Linear	MPC Linear - RBF
h_p	15	15
h_c	2	2
l_p (m)	5	5
$\boldsymbol{\Theta}$	0.021, 0.034, 0.0005, 0.028, 0.020, 0.033, 0.036, 0.044, 0.049, 0.047, 0.016, 0.030, 0.013, 0.125, 0.191	0.021, 0.035, 0.0005, 0.028, 0.012, 0.013, 0.036, 0.044, 0.042, 0.047, 0.011, 0.035, 0.053, 0.125, 0.180
$\boldsymbol{\Omega}$	0.020, 0.033	0.020, 0.033
ω	$1.45\ 10^4$	$1.99\ 10^4$
θ	$9.42\ 10^4$	$8.17\ 10^4$

Table 5
Random road tuning parameters.

	MPC Linear	MPC Linear - RBF
h_p	7	7
h_c	2	2
l_p (m)	3.1	3.2
$\boldsymbol{\Theta}$	0.0213, 0.034, 0.045, 0.058, 0.071, 0.030, 0.031	0.207, 0.067, 0.365, 0.557, 0.980, 0.275, 0.428
$\boldsymbol{\Omega}$	0.199, 0.065	0.199, 0.065
ω	$1.15\ 10^4$	$5.93\ 10^4$
θ	$9.42\ 10^4$	$9.73\ 10^4$

Table 6
Pulse bump simulation results.

Performance metrics	Passive car	MPC linear	MPC linear - RBF
Max absolute acceleration (m/s^2)	2.79	2.3	2.33
Settling time (s)	2.85	2.85	1.81
Sum of absolute error (m/s^2)	113	90.6	81.4

Table 7
Ramp bump simulation results.

Performance metrics	Passive car	MPC linear	MPC linear - RBF
Max absolute acceleration (m/s^2)	2.054	1.489	1.31
Settling time (s)	1.66	2.02	1.65
Sum of absolute error (m/s^2)	101	97.1	81.0

Table 8
Random road simulation results.

Performance metrics	Passive car	MPC linear	MPC linear - RBF
Sum of absolute error (m/s^2)	126.61	112.50	101.05
Heave acceleration FFT at eigenfrequency (m/s^2)	81.6	57.92	45.14
Squared heave displacement FFT at eigenfrequency (cm^2)	199.1	198	108.1
Average RMS value for the vertical acceleration of each wheel (m/s^2)	5.494	5.864	5.754

5.3. Results and discussion

The proposed MPC-RBF-L controller is applied to the road scenarios shown in Fig. 6, and compared to MPC-L and a passive suspension. The responses of the two controllers, along with the passive suspension, are shown in Figs. 7–9, while numerical results are given in Tables 6–8. Each table corresponds to a specific road scenario, and contains the simulation metrics for the two controllers and the passive suspension, as discussed in Section 5.1.

As far as the pulse bump scenario (Fig. 7) is concerned, the MPC-RBF-L controller is able to reduce the SAE by 10% compared to MPC-L, and 28% compared to the passive suspension. The

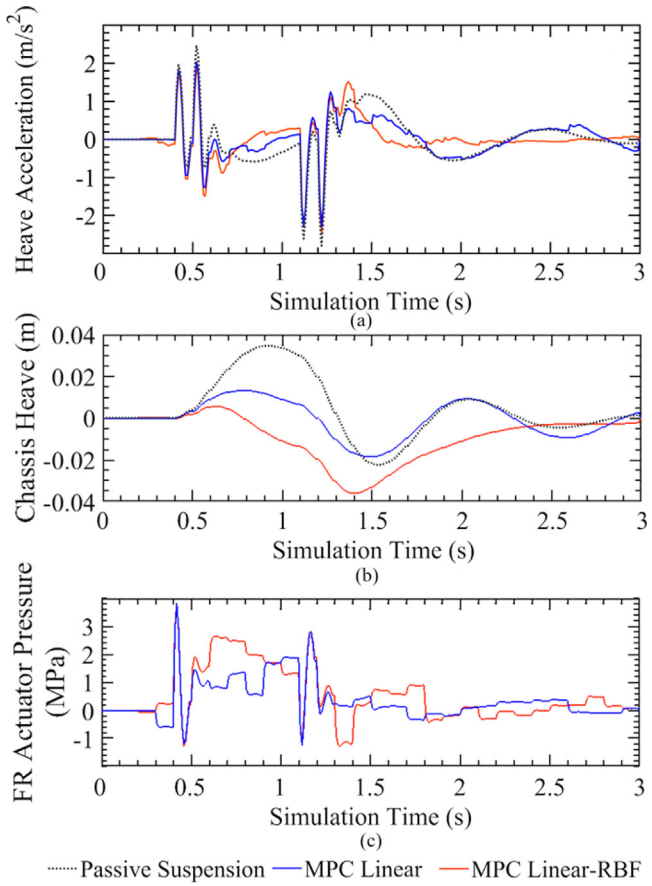


Fig. 7. Results of a pulse bump test: (a) chassis heave acceleration response, (b) chassis heave displacement, (c) front right pressure of the hydraulic actuator.

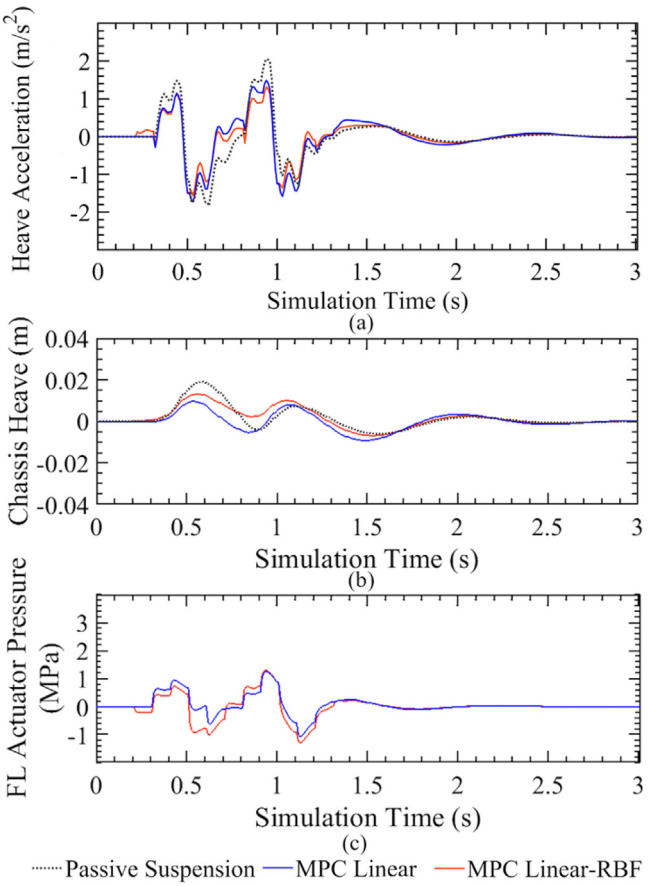


Fig. 8. Results of a ramp bump test: (a) chassis heave acceleration response, (b) chassis heave displacement, (c) front right pressure of the hydraulic actuator.

settling time also shows significant improvement. In the ramp bump test (Fig. 8), the MPC-RBF-L controller reduces the SAE by 11.5% compared to MPC-L, and 37% compared to the passive suspension. No reduction in settling time is achieved by the two MPC controllers when compared to the passive suspension in this case — this can be attributed to the fact that when the road disturbance ended, the absolute value of the heave acceleration was so small that the controllers decided not to act. The SAE is overall more effectively reduced on the ramp than the bump test for both MPC controllers. The reason is that the road disturbance is less abrupt and therefore both controllers can track it better.

On the random road (Fig. 9) the proposed controller reduces the SAE by 10% and 20%, compared to MPC-L and the passive suspension, respectively. Here, the frequency-domain response is of larger importance: the MPC-RBF-L generally performs well in the 4–8 Hz range, with a 44% reduction of the chassis acceleration response at the eigenfrequency compared to the passive car and a 22% reduction compared to MPC-L. This frequency range is of interest, since any vibrations herein incur the most discomfort for humans, according to ISO 2631 [37]. The chassis heave frequency response peaks at 4 Hz and then trails off towards zero for higher frequencies. It appears that both MPC controllers fail to control any heave acceleration responses over 10 Hz. This can be attributed to the high MPC sample time, which renders both controllers unable to monitor higher frequency responses. The Steffens control metric though, gives a clear indication of the superiority of the proposed controller over the rest of the competing schemes. This metric describes a threshold for vibratory environments that, once exceeded, human discomfort is induced; as shown in Fig. 9(c), the proposed scheme manages to remain largely below this threshold. In particular, for the eigenfrequency of the chassis, the MPC-RBF-L reduces the FFT of the heave displacement by 45.7% compared to the passive suspension, and by 45.4% compared to the MPC-L controller. Lastly, by assessing the average RMS value for the vertical acceleration of each wheel, one sees that both MPC controllers perform slightly worse than the passive suspension on this respect; the MPC-L vertical wheel acceleration value is 6.31% larger than the passive suspension one, while the respective MPC-RBF-L value is 4.5% larger. This result is to be expected, since the sampling frequency of 10 Hz of both MPC controllers is lower than the wheel-hop mode frequency, which usually resides in the 10–12 Hz range [38]. Still, this small difference in the aforementioned RMS values does not constitute a significant deterioration of road holding performance for the proposed scheme, thus its potential for practical applications in chassis vibration control is not diminished.

In general, when evaluating SAE as an overall performance metric on the three tests, both MPC controllers significantly increase riding comfort compared to the passive suspension. The MPC-L performance, while being better than the passive suspension, significantly deteriorates when applied to random road conditions; this is due to the limited accuracy of the linear prediction model. The MPC-RBF-L superior performance in all three tests is attributed to the better state approximation capabilities of the RBF-linear model, which enable the computation of more accurate control moves. It should be noted that the actual time needed by the MPC-RBF-L controller for solving the optimization problem was measured to be equal, or smaller to the

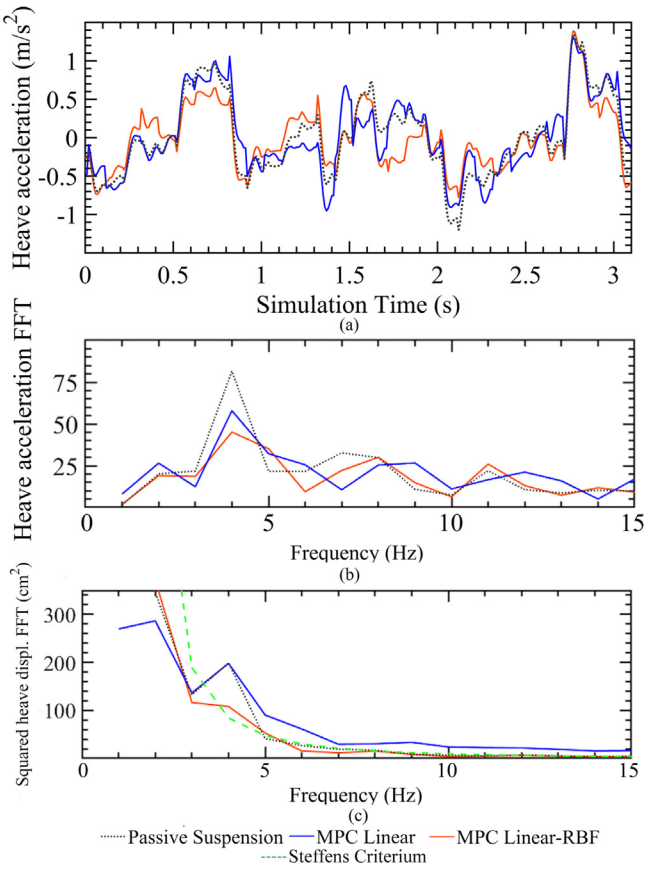


Fig. 9. Results of a random road test: (a) chassis heave acceleration, (b) chassis heave acceleration FFT, (c) chassis squared heave displacement FFT.

selected controller sampling time, thus allowing for the real-time implementation of the proposed scheme.

6. Conclusion

The MPC with preview information methodology has been proven suitable for the full car active suspension control; however, the nonlinear behavior of the plant should be taken into account when developing the MPC model, in order to avoid sub-par controller performance. In this work, we introduce the application of RBFN models trained with the FM algorithm, in order to approximate the nonlinear states of the full car model. This results to a significant improvement in modeling performance, as the modeling error for the hydraulic actuator pressure states is reduced up to 50% when compared to standard linear models. As the usage of nonlinear prediction models entails increased computational requirements, an approach reducing the overall computational burden of the MPC solution process is employed, through the usage of an inverse model of the plant for initializing the optimization problem. The proposed MPC scheme was implemented using a realistic LiDAR scanning noise model, tuned using PSO and simulated in three different road scenarios, namely pulse bump, ramp bump and random road. The proposed scheme was compared to an MPC controller with a linear model and a passive suspension, demonstrating superior performance on all scenarios. In particular, reductions in chassis heave acceleration of up to 28% and 37% are observed for the pulse and ramp bump tests, respectively. In addition, a 44% reduction of the chassis acceleration eigenfrequency response is achieved for the random road test, which confirms that the proposed scheme largely remains below the Steffens ride comfort criterium. It is emphasized that the aforementioned results are owed to the superior state prediction accuracy that RBF models allow.

Future research plans include the use of RBF networks for modeling the vehicle lateral and yaw dynamics [40], in order to accommodate for the significant nonlinearities arising from lateral tire force behavior. The resulting models could be incorporated in MPC schemes, thus producing vehicle stability [41], or active torque vectoring [42] controllers with nonlinear capabilities.

CRediT authorship contribution statement

Myron Papadimitrakis: Conceptualization, Formal analysis, Methodology, Investigation, Software, Visualization, Writing – original draft, Writing – review & editing. **Alex Alexandridis:** Conceptualization, Formal analysis, Methodology, Supervision, Resources, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EΔK-02191).

References

[1] A. Alleyne, J.K. Hedrick, Nonlinear adaptive control of active suspensions, IEEE Trans. Control Syst. Technol. 3 (1995) 94–101, http://dx.doi.org/10.1109/87.370714.

[2] C. Gohrle, A. Schindler, A. Wagner, O. Sawodny, Design and vehicle implementation of preview active suspension controllers, IEEE Trans. Control Syst. Technol. 22 (2014) 1135–1142, http://dx.doi.org/10.1109/TCST.2013.2272342.

[3] S. Kumarawadu, T.T. Lee, Neuroadaptive combined lateral and longitudinal control of highway vehicles using RBF networks, IEEE Trans. Intell. Transp. Syst. 7 (2006) 500–512, http://dx.doi.org/10.1109/TITS.2006.883113.

[4] S. Fergani, O. Sename, L. Dugard, An LPV Hinfin; Integrated vehicle dynamic controller, IEEE Trans. Veh. Technol. 65 (2016) 1880–1889, http://dx.doi.org/10.1109/TVT.2015.2425299.

[5] R. Ding, R. Wang, X. Meng, W. Liu, L. Chen, Intelligent switching control of hybrid electromagnetic active suspension based on road identification, Mech. Syst. Signal Process. 152 (2021) 107355, http://dx.doi.org/10.1016/j.ymssp.2020.107355.

[6] W. Sun, H. Pan, H. Gao, Filter-based adaptive vibration control for active vehicle suspensions with electrohydraulic actuators, IEEE Trans. Veh. Technol. 65 (2016) 4619–4626, http://dx.doi.org/10.1109/TVT.2015.2437455.

[7] T. Yoshimura, Y. Isari, Q. Li, J. Hino, Active suspension of motor coaches using skyhook damper and fuzzy logic control, Control Eng. Pract. 5 (1997) 175–184, http://dx.doi.org/10.1016/S0967-0661(97)00224-4.

[8] A. Aldair, W.J. Wang, Adaptive neuro fuzzy inference controller for full vehicle nonlinear active suspension systems, in: EPC-IQ01 2010-2010 1st Int Conf Energy, Power Control, Vol. 1, 2010, pp. 97–106.

[9] J.O. Pedro, M. Dangor, O.A. Dahunsi, M.M. Ali, Dynamic neural network-based feedback linearization control of full-car suspensions using PSO, Appl. Soft Comput. J. 70 (2018) 723–736, http://dx.doi.org/10.1016/j.asoc.2018.06.002.

[10] J.O. Pedro, M. Dangor, O.A. Dahunsi, M.M. Ali, Intelligent feedback linearization control of nonlinear electrohydraulic suspension systems using particle swarm optimization, Appl. Soft Comput. J. 24 (2014) 50–62, http://dx.doi.org/10.1016/j.asoc.2014.05.013.

[11] W. Wang, Y. Song, Y. Xue, H. Jin, J. Hou, M. Zhao, An optimal vibration control strategy for a vehicle's active suspension based on improved cultural algorithm, Appl. Soft Comput. J. 28 (2015) 167–174, http://dx.doi.org/10.1016/j.asoc.2014.11.047.
[12] J.H. Lee, Model predictive control: Review of the three decades of development, Int. J. Control Autom. Syst. 9 (2011) 415–424, http://dx.doi.org/10.1007/s12555-011-0300-6.
[13] J.K. Hedrick, M.D. Donahue, Implementation of an active suspension, in: Nonlinear Hybrid Syst Automot Control XVIII, 2003, p. 446.
[14] M. Ławryńczuk, P. Tatjewski, Offset-free state-space nonlinear predictive control for Wiener systems, Inf. Sci. (Ny) 511 (2020) 127–151, http://dx.doi.org/10.1016/j.ins.2019.09.042.
[15] M. Ławryńczuk, Constrained computationally efficient nonlinear predictive control of Solid Oxide Fuel Cell: Tuning, feasibility and performance, ISA Trans. 99 (2020) 270–289, http://dx.doi.org/10.1016/j.isatra.2019.10.009.
[16] M. Ławryńczuk, Nonlinear predictive control of a boiler-turbine unit: A state-space approach with successive on-line model linearisation and quadratic optimisation, ISA Trans. 67 (2017) 476–495, http://dx.doi.org/10.1016/j.isatra.2017.01.016.
[17] C. Pilbeam, R.S. Sharp, Performance potential and power consumption of slow-active suspension systems with preview, Veh. Syst. Dyn. 25 (1996) 169–183, http://dx.doi.org/10.1080/00423119608968963.
[18] Y. Hu, Theory and Technology of Laser Imaging Based Target Detection, Springer, 2017, http://dx.doi.org/10.1007/978-981-10-3497-8.
[19] S. Haykin, Nueral Network, second ed., Pearson Prentice Hall, 2005.
[20] A. Alexandridis, E. Chondrodima, N. Giannopoulos, H. Sarimveis, A fast and efficient method for training categorical radial basis function networks, IEEE Trans. Neural Netw. Learn. Syst. 28 (2017) 2831–2836, http://dx.doi.org/10.1109/TNNLS.2016.2598722.
[21] L.H. Cseko, M. Kvasnica, B. Lantos, Explicit MPC-based RBF neural network controller design with discrete-time actual Kalman filter for semiactive suspension, IEEE Trans. Control Syst. Technol. 23 (2015) 1736–1753, http://dx.doi.org/10.1109/TCST.2014.2382571.
[22] T. Poggio, F. Girosi, Networks for approximation arid learning, Proc. IEEE 78 (1990) 1481–1497, http://dx.doi.org/10.1109/5.58326.
[23] C. Darken, J. Moody, Fast, adaptive K-means clustering: some empirical results, in: Proc. IEEE Int. Jt. Conf. Neural Networks (IJCNN 1990), 1990.
[24] A. Alexandridis, E. Chondrodima, H. Sarimveis, Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization, IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 219–230, http://dx.doi.org/10.1109/TNNLS.2012.2227794.
[25] H. Mirzaeinejad, Robust predictive control of wheel slip in antilock braking systems based on radial basis function neural network, Appl. Soft Comput. J. 70 (2018) 318–329, http://dx.doi.org/10.1016/j.asoc.2018.05.043.
[26] M. Stogiannos, A. Alexandridis, H. Sarimveis, Model predictive control for systems with fast dynamics using inverse neural models, ISA Trans. 72 (2018) 161–177, http://dx.doi.org/10.1016/j.isatra.2017.09.016.
[27] A. Schindler, Neue Konzeption und Erstmalige Realisierung eines Aktiven Fahrwerks mit Preview-Strategie, Karlsruher Institut für Technologie, 2009, http://dx.doi.org/10.1007/s13398-014-0173-7.2.
[28] H.E. Merritt, Hydraulic Control System, John Wiley & Sons, 1973.
[29] W.R. Garrott, Measured vehicle inertial parameters-NHTSA's data through 1992, SAE Tech. Pap. 1 (1993) http://dx.doi.org/10.4271/930897.
[30] C. Göhrle, A. Schindler, A. Wagner, O. Sawodny, Road profile estimation and preview control for low-bandwidth active suspension systems, IEEE/ASME Trans. Mechatronics 20 (2015) 2299–2310, http://dx.doi.org/10.1109/TMECH.2014.2375336.
[31] C. Glennie, Rigorous 3D error analysis of kinematic scanning LIDAR systems, J. Appl. Geod. 1 (2008) 147–157, http://dx.doi.org/10.1515/jag.2007.017.
[32] A. Alexandridis, H. Sarimveis, Nonlinear adaptive model predictive control based on self-correcting neural network models, AIChE J. 51 (2005) 2495–2506, http://dx.doi.org/10.1002/aic.10505.
[33] F. Tyan, Y.-F. Hong, S.-H. Tu, W.S. Jeng, Generation of random road profiles, J. Adv. Eng. 4 (2009) 151–156, doi: 10.1.1.474.1875.
[34] A.L. Do, O. Sename, L. Dugard, LPV modeling and control of semi-active dampers in automotive systems, in: J. Mohammadpour, C. Scherer (Eds.), Control Linear Param. Varying Syst. with Appl., Vol. 9781461418, Springer, Boston, MA, 2012, pp. 381–411, http://dx.doi.org/10.1007/978-1-4614-1833-7_15.
[35] K. Worthmann, M. Reble, L. Grüne, F. Allgöwer, The role of sampling for stability and performance in unconstrained nonlinear model predictive control, SIAM J. Control Optim. 52 (2014) 581–605, http://dx.doi.org/10.1137/12086652X.
[36] J.R. Dorm, P.J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26, http://dx.doi.org/10.1016/0771-050X(80)90013-3.
[37] International Organization for Standardization. ISO 2631-5:2018. 2nd ed. 2018.
[38] W.F. Milliken, D.L. Milliken, Race Car Vehicle Dynamics, Volume 1, SAE International, 1995.
[39] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. ICNN'95 - Int. Conf. Neural Networks, Vol. 4, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.
[40] N. Ahmadian, A. Khosravi, P. Sarhadi, Integrated model reference adaptive control to coordinate active front steering and direct yaw moment control, ISA Trans. 106 (2020) 85–96, http://dx.doi.org/10.1016/j.isatra.2020.06.020.
[41] M. Choi, S.B. Choi, Model predictive control for vehicle yaw stability with practical concerns, IEEE Trans. Veh. Technol. 63 (2014) 3539–3548, http://dx.doi.org/10.1109/TVT.2014.2306733.
[42] A. Tahouni, M. Mirzaei, B. Najjari, Novel constrained nonlinear control of vehicle dynamics using integrated active torque vectoring and electronic stability control, IEEE Trans. Veh. Technol. 68 (2019) 9564–9572, http://dx.doi.org/10.1109/TVT.2019.2933229.