Intelligent feedback linearization control of nonlinear electrohydraulic suspension systems using particle swarm optimization

Jimoh O. Pedro a,*, Muhammed Dangor a, Olurotimi A. Dahunsi a, M. Montaz Ali b

a School of Mechanical, Industrial and Aeronautical Engineering, University of the Witwatersrand, Johannesburg, South Africa
b School of Computational and Applied Mathematics, Faculty of Science, and TCSE, Faculty of Engineering and Built Environment, University of the Witwatersrand, Johannesburg, South Africa

**A R T I C L E  I N F O**

Article history:
Received 23 February 2013
Received in revised form 14 May 2014
Accepted 16 May 2014
Available online 27 June 2014

Keywords:
Feedback linearization
Dynamic neural networks
Particle swarm optimization
Active vehicle suspension systems
Ride comfort
Road holding

**A B S T R A C T**

The core factors governing the performance of active vehicle suspension systems (AVSS) are the inherent trade-offs involving suspension travel, ride comfort, road holding and power consumption. In addition to this, robustness to parameter variations is an essential issue that affects the effectiveness of highly nonlinear electrohydraulic AVSS. Therefore, this paper proposes a nonlinear control approach using dynamic neural network (DNN)-based input–output feedback linearization (FBL) for a quarter-car AVSS. The gains of the proposed controllers and the weights of the DNNs are selected using particle swarm optimization (PSO) algorithm while addressing simultaneously the AVSS trade-offs. Robustness and effectiveness of the proposed controller were demonstrated through simulations.

© 2014 Elsevier B.V. All rights reserved.

**Introduction**

Vehicle suspensions are subsystems that aim to improve the performance of an automobile by isolating the vehicle from road-induced disturbances, improving passenger ride comfort, and enhancing the road holding performance of the vehicle. However, better ride comfort demands a softer suspension and superior road holding requires a stiffer suspension. The desire to manage these trade-offs has led to the development of active vehicle suspension systems (AVSS) which incorporates an actuator to deal with these compromises in real time [1].

AVSS are highly nonlinear systems with complex actuator dynamics and need to be designed carefully to manage its sensitivity to parameter variations. Nonlinear AVSS have been successfully designed with linear controllers [1]. However, these controllers lack robustness when dealing with variations in vehicle speed, sprung mass and tyre load. On the contrary, nonlinear control schemes such as Fuzzy Logic Control (FLC), backstepping, FBL and neural network (NN)-based control have been able to deal with these issues more effectively [2–10]. In contrast to linear controllers, nonlinear control methods generate an input that aims to remove or significantly reduce the effects of nonlinearities in the system.

The nonlinear AVSS model presented by Shi et al. [3] incorporated a servo-hydraulic system controlled by a combination of sliding mode and feedback linearization control methods. This effort and those of Yagiz and Sakman [2] and Chamasedine et al. [9] using sliding mode control were plagued with chattering challenges.

Backstepping control has been performed by [4,5,10] for half-car and full-car models respectively. Their solutions showed a significant improvement over the passive vehicle suspension system (PVSS) and additionally provided an adequate bandwidth where they were able to reject a large range of road disturbances, which emphasizes the robustness of this control technique.

However, these preceding control laws are fundamentally based on the mathematical models which have been chosen to be either linear or nonlinear. In the case of backstepping, highly nonlinear and realistic models would require rigorous interlaced backstepping which may in some cases prove impossible to solve. Furthermore, the associating zero-dynamics present in real models may be unstable under certain conditions. Furthermore, the systems may not be completely understood in reality and this would...
require the use of some model predictive controller to learn the dynamics of the plant.

Intelligent nonlinear control seeks to emulate human logic as well as the brain. They do not require the mathematics of the system to be completely understood whilst developing a control law. FLC and NN-based control form part of this set and have been successfully implemented for AVSS designs to deal with robustness issues and to better manage AVSS trade-offs [13–25].

Documented works in the literature have demonstrated that good AVSS performances can be achieved using FLC [24–26]. Combination of FLC with neural networks in AVSS applications has also shown remarkable improvements in the robustness of the controller designed. Rajeswari and Lakshmi [14], Lian [26], and Aldair and Wang [18] have proposed hybrid neuro-fuzzy controllers. They realized that the control structure performs better than a NN-based PID control architecture, which makes it more suitable for AVSS applications. Furthermore, there is no need to mathematically model the system since the NN can approximate it through the process of system identification.

The accomplishments of FLC in AVSS have made it a suitable candidate for optimization algorithms. Chiou et al. [13], Rajeswari and Lakshmi [14], and Pekgökgöz et al. [25] used evolutionary algorithms to derive the membership functions of a FLC. This method was successful in improving either the body-heave acceleration or the suspension travel with larger success than a PID-controlled AVSS.

With regards to the application of NN-based control to AVSS, intelligent controllers using multilayer NNs in system identification and control have improved the AVSS response as compared to the PVSS. Tang et al. [15] investigated the performance of a half-car AVSS that was controlled using a multilayer feedforward NN and genetic algorithm (GA). There was an improvement in the passenger's seat vertical response as compared to that of a PVSS. In terms of training a multilayer NN through PSO for AVSS, Alfi and Fateh [17] showed that this method performs better than the conventional NN training algorithms and GA-based training with quicker convergence speeds, improved accuracy, and had no premature convergence problem.

Guclu and Gulez [27], and Aldair and Wang [18] utilized network inversion to control a full-car nonlinear AVSS with actuator dynamics. The NN-based controllers for each case displayed superior performance as compared to the PVSS. Eski and Yildirim [16] also used an adaptive multilayer NN to create a robust PID controller for a full-car model. The system displayed high identification and tracking capabilities as compared to offline supervised learning algorithms.

Pedro et al. [21] designed a direct adaptive NN-based FBL controller for nonlinear quarter-car AVSS using radial basis function NN (RBFNN). However, the model did not contain any actuator dynamics and ignored zero dynamics that may exist as a result of FBL. The ride comfort and road holding improved as compared to the PVSS and PID-controlled AVSS. Pedro and Dahuni [20] later utilized a multilayer feedforward NN to perform indirect adaptive control of a servo-hydraulic nonlinear AVSS using FBL. They considered subsequent zero dynamics and their resulting system displayed superior performance as compared to the case where linear controllers were used.

DNN uses differential equation to model the neuron and contains feedback elements. DNN offers several benefits above static neural networks (radial basis function neural network (RBFNN) and multilayer perceptron neural network (MLPNN)) especially as regard computational efficiency. DNN has capacity to learn complex nonlinear systems especially when static neural networks fails to represent the model appropriately [30–32]. Application of DNN with PSO training is very rare, especially with respect to AVSS controller design. However, DNN has been used for various control systems: linear AVSS [23], semi-active suspension [22,33], evaporators [30,36], continuously stirred tank reactors (CSTR) [29,35] and flexible manipulators [37].

Yildirim [23] successfully identified a linear AVSS using a recurrent neural network (RNN) and thereafter carried out network inversion to control the system. He achieved an improvement over the PVSS. Zapaterio et al. [22] performed recurrent neural network (RNN)-based backstepping control on a semi-active suspension that utilized a magnetorheological (MR) damper. Metered et al. [33] carried out network inversion on a MR-based semi-active suspension that was identified with a RNN.

Becerikli et al. [34] presented a DNN to identify and control a CSTR. The system displayed adequate performance in the presence of a wide range of disturbances. The start-up and regulation problems of CSTR were resolved better with this configuration as compared to the currently employed control schemes for CSTR tanks.

Al Seyab and Cao [35] created a DNN to identify and control a double CSTR plant. It was concluded that the DNN decreased the training time and improved the accuracy in the identification process as compared to conventional model predictive control configurations. Nanayakkara et al. [36] successfully trained a DNN to identify an evaporator with the use of an evolutionary algorithm. The DNN outperformed the static NNs and required fewer neurons to learn the dynamics of the plant.

Tian and Collins [37] designed a neuro-fuzzy controller for a flexible manipulator. The dynamics of the nonlinear model was learnt using a DNN. The identification results were satisfactory and this control method was superior to conventional industrial robots controllers. Deng et al. [28], Deng et al. [29] and Garces et al. [30] utilized DNN-based FBL for a variety of control systems. In each case, the network was trained using genetic algorithm and the response of the system displayed superior results as compared to conventional control architectures. Additionally they noticed that this control law can be implemented with linear control such as PID to improve system response.

Both linear and nonlinear AVSS controllers have been optimized using heuristic search methods since they are effective methods in finding global minima. These methods are based on a random search methodology and do not require any function based methods to find the minimum. Such methods include PSO and GA. Wai et al. [12], Crews et al. [11], Chiou et al. [13], and Rajeswari and Lakshmi [14] optimized the gains of a single loop linear PID-based AVSS using PSO and GA techniques respectively. Pekgökgöz et al. [25] and Chiou et al. [13] optimized the membership functions and FLC control parameters using GA and PSO. The fitness function of these approaches incorporated sprung mass acceleration and body displacement. These optimal policies outperformed the manually tuned AVSS in terms of acceleration and body displacement at the expense of the actuation force.

The major contribution of the paper is the demonstration of the impact of the combination of a PSO-optimized cascaded PID control with dynamic neural network (DNN)-based feedback linearization control. The weights of the DNN are chosen using PSO algorithm. Performance results for these controllers are presented for each controller when applied to a 2 degree-of-freedom (DOF) nonlinear electro-hydraulic vehicle suspension system. These results are also compared with that of a controller that combines both schemes. The controller with combined controlled schemes is characterized with better performance for all the vehicle suspension performance parameters considered. The controller is also tested for robustness using response to parameter variation within the 20% range for speed, mass and tire stiffness. The difference in the results obtained is marginal. Frequency domain analysis of the controllers are also presented within the whole-body-vibration range of 0–80 Hz. The trend in performance is the same, corroborating
the previous results and showing that all controllers attenuated vibration disturbance within the frequency range considered.

The paper is organized as follows. In Active suspension system: overview and modelling section we present the active suspension system overview and mathematical modelling. Controller design section introduces the AVSS performance specifications followed by a brief description of the PID controller. PSO algorithm is also described in this section. Detailed description of the DNN-based system identification and FBL control are presented. Simulation results and discussion section presents both for deterministic and random disturbance inputs. Results for robustness analysis are also presented and discussed in this section. Finally, concluding remarks and recommendations for future work are given in Conclusion and future work section.

Active suspension system: overview and modelling

A schematic of the quarter-car model used in this investigation is presented in Fig. 1. The mass of the wheel assembly is $m_w$ and that of the chassis is $m_s$. These two components are coupled through the suspension elements (spring $k_s$ and damper $b_s$). In the case of AVSS, an actuator is placed in parallel with the suspension elements and supplies an actuator force $F_a$ which supports the PVSS components. The flexural nature of the wheel is captured by means of a spring with stiffness $k_c$.

Reference frames are created at the wheel, chassis and road surface with $x_w$, $x_c$ and $w$ denoting the vertical movement of the wheel, vertical movement of the chassis and the road profile respectively. The associated velocity and accelerations of these bodies are represented as $\dot{x}$ and $\ddot{x}$ respectively.

The governing equations of the system are derived by applying Newton’s laws to both the wheel and chassis:

$$m_w \ddot{x}_w = F_{gs} + F_{bs} - F_a,$$

$$m_s \ddot{x}_c = -F_{ks} - F_{bs} + F_a + F_r,$$

where $F_{gs}$ and $F_{bs}$ are the respective spring and damping forces exerted by the suspension, $F_a$ is the force produced by the road disturbance and $F_r$ is the actuator force. It is assumed that the wheel and suspension elements compress such that $w > x_w$ and $x_w > x_c$.

Both the spring and damping forces are functions of the suspension travel $(x_w - x_c)$ and suspension travel velocity $(\dot{x}_w - \dot{x}_c)$, respectively. The suspension components have linear, symmetric and nonlinear elements which are fundamentally a function of the suspension travel and its velocity and are described as follows [20]:

$$F_s = k_s(x_w - x_c) + k_{nl}^s(x_w - x_c)^3,$$

$$F_b = b_s(x_w - x_c) + b_{nl}^s \sqrt{|x_w - x_c|} \text{sgn}(x_w - x_c) - b_{sym}^s |x_w - x_c|,$$

where $k_s^s$ and $b_s^s$ are the linear spring stiffness and linear damping constant of the suspension system, $k_{nl}^s$ and $b_{nl}^s$ are the corresponding nonlinear spring stiffness and damping constant of the suspension system, and $b_{sym}^s$ is the associating symmetric damping constant. The elastic behaviour of the tyre is assumed linear and the force produced due to its interaction with the road is:

$$F_w = k_c(w - x_w),$$

where $(w - x_w)$ is the deflection of the tyre.

The actuator force $F_a$ is manipulated through an electrohydraulic servo-valve which aims to return the system to rest after the vehicle is disturbed by the road disturbance. A schematic of the actuator explaining the flow of hydraulic fluid and pressure changes in the system is shown in Fig. 2.

The electrohydraulic servo-valve consists of two components: a voltage-regulated electromechanical device and a three-land four-way spool-valve hydraulic system. The dynamics of the actuator are described through Newtonian fluid mechanics. The governing equations for the electrohydraulic actuator may be structured to a simpler form that is suitable for FBL using the following relations [38,39,41]:

$$\dot{p}_t = \gamma \Phi x_v - \beta \dot{p}_t + \alpha A(x_w - x_c),$$

where $\Phi = \phi_1 \times \phi_2$ with $\phi_1 = \text{sgn}(P_t - \text{sgn}(x_v)P_1)$ and $\phi_2 = \sqrt{|P_t - \text{sgn}(x_v)P_1|}$, $\alpha = \frac{4\rho_0}{\gamma V}$, $\beta = \alpha CIP$, $\gamma = C_q \sqrt{1/\beta}$. $A$ is the cross-sectional area of the piston, $P_t$ is the change in pressure experienced across the piston, $x_v$ is the servo-valve displacement, $P_1$ is the supply pressure into the hydraulic cylinder, $P_t$ is the return pressure from the hydraulic cylinder, $P_a$ and $P_1$ represent the oil pressure in the upper and lower portion of the cylinder respectively, $V_t$ is the total actuator volume, $\beta$ is the effective bulk modulus of the system, $\Phi$ is the hydraulic load flow, $C_{IP}$ is the total leakage
Table 1
System parameters for the quarter-car model [20,40,41].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Numerical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis or sprung mass, (m_c)</td>
<td>290 kg</td>
</tr>
<tr>
<td>Wheel or unsprung mass, (m_u)</td>
<td>40 kg</td>
</tr>
<tr>
<td>Suspension spring linear stiffness, (k_s)</td>
<td>(2.35 \times 10^4) N/m</td>
</tr>
<tr>
<td>Suspension spring nonlinear stiffness, (k_{nl})</td>
<td>(2.35 \times 10^4) N/m³</td>
</tr>
<tr>
<td>Tyre stiffness, (k_t)</td>
<td>(190 \times 10^2) N/m</td>
</tr>
<tr>
<td>Suspension linear damping coefficient, (b_s)</td>
<td>700 Ns/m</td>
</tr>
<tr>
<td>Suspension nonlinear damping coefficient, (b_{nl})</td>
<td>(400 \text{Ns}^2/\text{m}^3)</td>
</tr>
<tr>
<td>Actuator parameters, ((\alpha, \beta, \gamma))</td>
<td>(4.515 \times 10^{11}, 1, 1.545 \times 10^6)</td>
</tr>
<tr>
<td>Piston area, (A)</td>
<td>(3.35 \times 10^{-4}) m²</td>
</tr>
<tr>
<td>Supply pressure, (P_s)</td>
<td>10,342,500 Pa</td>
</tr>
<tr>
<td>Time constant, (\tau)</td>
<td>(1/30) s</td>
</tr>
<tr>
<td>Servo valve gains, (k_s)</td>
<td>(0.001 \text{m/V})</td>
</tr>
</tbody>
</table>

The coefficient of the piston, \(C_d\) is the discharge coefficient, \(S\) is the spool-valve area gradient and \(\rho\) is the hydraulic fluid density.

In order to reduce complexity, it is assumed that the electromagnetic device that controls the motion of the spool valve is a first-order element with a time constant \(\tau\) and is described as:

\[
\dot{x}_v = \frac{1}{\tau}(K_v u - x_v),
\]

where \(K_v\) is the valve gain and \(u\) is the control input voltage. Values of the parameters used in the quarter-car model are given in Table 1.

The performance of the AVSS is investigated as the vehicle travels over a deterministic road bump at a speed of 40 km/h. The road bump has a sinusoidal profile with a length of 2.5 m and amplitude of 11 cm. The equation describing the profile of the bump is presented in Eq. (8) and is displayed in Fig. 3.

\[
w(t) = \begin{cases} 
\frac{a(1 - \cos 2\pi(V/\lambda)t)}{2} & 0.45 \leq t \leq 0.9, \\
0 & \text{otherwise}, 
\end{cases} 
\]

where \(a\) is the bump height, \(V\) is the vehicle speed, \(\lambda\) is the half wavelength of the sinusoidal road undulation [20].

The system may be further rearranged into a state-space form by defining the following state variables: \(x_1 = x_3, x_1 = x_3, x_v = x_4, x_v = x_4, x_5 = P_1, \) and \(x_6 = x_t\). [20]:

\[
\dot{x} = f(x) + g(x)u + w(x),
\]

\[
y = h(x) = x_1 - x_2,
\]

where the state vector is given by \(x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T\).

The system matrices \(f\) and \(g\) are denoted by:

\[
f(x) = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]^T,
\]

\[
g(x) = \begin{bmatrix} 0 & 0 & 0 & 0 & k_{v1} \\
0 & 0 & 0 & 0 & k_{v2} \\
0 & 0 & 0 & 0 & k_{v3} \\
0 & 0 & 0 & 0 & k_{v4} \\
\end{bmatrix}.
\]

The disturbance matrix \(w\) is represented by:

\[
w(x) = \begin{bmatrix} 0 & 0 & w(t) & 0 & k_{v1} & 0 \end{bmatrix}^T.
\]

The elements of these matrices are as follows [20]:

\[
f_1(x) = x_3,
\]

\[
f_2(x) = x_4,
\]

\[
f_3(x) = \frac{1}{m_s} \left[ k_s(x_2 - x_1) + k_{nl}(x_2 - x_1)^3 + b_s(x_4 - x_3) - b_{nl}(x_4 - x_3) \right],
\]

\[
f_4(x) = \frac{1}{m_u} \left[ -k_s(x_2 - x_1) - k_{nl}(x_2 - x_1)^3 - b_s(x_4 - x_3) + b_{nl}(x_4 - x_3) \right],
\]

\[
f_5(x) = -\alpha x_6 - \beta x_5 + A(x_4 - x_3),
\]

\[
f_6(x) = \frac{1}{\tau}(-x_6).
\]

Controller design

AVSS performance specifications

The fundamental objective of the controller is to return the system to steady state after being disturbed by a deterministic road bump (Eq. (8)). The controller must also satisfy the following performance specifications for the AVSS [20]:

1. The controller should demonstrate good low frequency disturbance rejection.
2. Satisfactory transient response with minimal oscillations after the disturbance has disappeared; that is:
   • the rise time should not be greater than 0.1 s.
   • the maximum overshoot should be less than 5%.
   • and zero steady-state error.
3. The body-heave acceleration, \(\ddot{x}_1\), should be less than 4.5 m/s² in order to keep the system in the Less Discomfort region as [43].
4. The suspension travel, \(y = (x_1 - x_3)\), must remain within ±0.1 m.
5. The control voltage is limited to ±10 volts.
6. The maximum actuation force, \(F_{\text{max}}\), must be less than the static weight of the vehicle.
7. For good road holding, the dynamic load that is transmitted from the road should be less than the static weight of the vehicle, i.e., \(F_{\text{tyre}} < m_gg\).
8. The controller should minimize a performance index which includes suspension travel (disturbance rejection), body-heave acceleration (ride comfort), wheel deflection (road holding), actuation force and control input voltage (powerconsumption);
controller to achieve improved performance [30]. PSO algorithm is thereafter used to optimize the PID gains, FBL control parameters and the DNN weightings in order to obtain an optimal control law that best manages the trade-offs associated with the AVSS. $R_d$ is the desired suspension travel as shown in Fig. 5. The rest of this subsection is devoted to providing a detailed presentation of the system identification and controller design.

**DNN-based nonlinear system identification**

Fig. 6 shows the structure of a typical DNN. The dynamics of the neurons are described by a first-order differential equation with a time constant $\beta$. Additionally, each neuron receives feedback from neurons in its respective hidden layer $\Xi_{-1}$ of the neural network, as well as from the input layer of the neural network $u$. Both the network and neuron-to-neuron inputs are essentially added to the right-hand side of the differential equation that describes the dynamics of the neuron.

The output of each neuron is passed through an activation function $\sigma(x)$ before it is fed back to each neuron in the corresponding hidden layer. Additionally, each neuron has two external inputs: control input $u$ and a delayed system state $\Xi_{t-1}$, each of which possesses its own associating weighting value. Hence, a neuron in the first hidden layer of the neural network is described by:

$$\dot{x} = -\beta x + \omega(x) + \gamma u + \zeta \Xi_{t-1},$$

where $x$ is a vector denoting the outputs of each neuron, $\beta$ is a diagonal matrix containing the time constants for each neuron in the hidden layer, $\sigma(x)$ is the vector containing the neuron outputs after it had passed through the activation function, $\omega$ is the interconnecting neuron weights, $u$ is a vector that holds the various control input signals that are being passed into the real system, $\gamma$ is a matrix which holds the weighting contributions that each control input has on each neuron, $\Xi_{t-1}$ is a vector that holds the actual system output or delayed output at the previous time step, and $\zeta$ is the contribution of these aforementioned outputs on each neuron.

The output layer of the DNN comprises of a single neuron and is fundamentally an algebraic equation, which is essentially the weighted sum of the neuron outputs from the preceding hidden layer $x$. Thus, the neuron in this layer is described as follows:

$$\hat{y} = h_n(x) = \sum_{i=1}^{n_n} w_i x_i,$$

where $x_i$ is a vector comprising of the output of the $i$th neuron from the hidden layer, $w_i$ is the associating weighting contribution of the $i$th neuron in the hidden layer, $n_n$ is the number of neurons in the hidden layer.

In order to further simplify the model and the subsequent computation, the output of the network $\hat{y}$ will depend solely on the output from the first neuron in the preceding hidden layer. Consequently, the output layer will be simplified to:

$$\hat{y} = h_1 = w_1 x_1.$$  (27)

Furthermore, the selection of the network parameters such as: the number of hidden layers, size of the hidden layer $n_n$ and the activation function $\sigma(\cdot)$ is based on two items: the network stability and the method of pruning [32]. In the course of pruning, the primary network parameters such as the hidden layer size are increased until the predicted system output $\hat{y}$ stops changing topology with further increase in the hidden layer size. Garces et al. [30] suggest that $\sigma(\cdot)$ should be bounded to within $\pm 1$ so that the free response of the DNN converges to zero and thus stabilizes once the networks inputs are removed. Hence, the hyperbolic tangent function is chosen as activation function $\sigma(\cdot)$ to meet this

**PID controller design**

The PID control for the AVSS comprises of two control loops: the outer loop, which regulates the controlled variable (suspension travel, $y$) and the inner loop which ensures stable operations of the electrohydraulic actuator [1]. A schematic of the multi-loop PID control system is shown in Fig. 4.

The setpoint $y_d$ is set to zero to address a regulation problem, $e_1$ and $e_2$ are error signals that will be minimized in the outer and inner control loops respectively, $F_o$ is the actuator force that will be regulated in the inner control loop with $F_D$ being its respective setpoint, and $u$ is the control input voltage that is passed into the hydraulic actuator of the AVSS. PID controllers operate according to the following equations:

$$e_1(t) = y_d(t) - y(t) = y_d(t) - x_w(t) + x_c(t),$$

$$F_d(t) = K_pe_1(t) + K_d \frac{de_1(t)}{dt} + K_i \int_0^t e_1(t)dt,$$

$$e_2(t) = F_d(t) - F_o(t),$$

$$u(t) = K_p e_2(t) + K_d \frac{de_2(t)}{dt} + K_i \int_0^t e_2(t)dt,$$

where $K_p$ and $K_p$ are the proportional gains of the outer and inner loops respectively; $K_i$ and $K_i$ are the corresponding integral gains of the controllers; and $K_o$ and $K_o$ are the derivative gains of the respective control loops.

**Proposed control approach**

The proposed controller architecture is presented in Fig. 5. The DNN predicts the response of the plant for a specified input. The controller performs FBL on the trained DNN with the intention of producing a control signal that cancels out the system’s nonlinearities. The FBL controller is augmented with a multi-loop PID

![Fig. 4. Architecture of the multi-loop PID-controlled quarter-car AVSS.](image-url)
condition. The next condition may only be fulfilled after the selection of appropriate input–output data.

An important step in system identification is to select a range of input–output data that covers the range of signals that will be encountered in reality. In indirect adaptive control the DNN must predict the output of the suspension system for a given set of control input voltage. White-Band-Limited (WBL) noise is used to create a set of input data because it can successfully create a random set of input signals which span the space of the expected input signals.

Selection of an appropriate data set is a rather rigorous process that requires several conditions to be met. Firstly, the dynamics of the subsystem with the smallest time constant must be captured and this demands that the seed strength of the WBL be significantly high. Secondly, the input data must span the space of all possible input voltages, which is known to be in the range of ±10 V. Similarly, the set of suspension travel output must span the region in which it is expected to operate, which corresponds to ±0.1 m. “Whole Body Vibration” (WBV) frequency range is classified to be between 0.5 and 80 Hz. Human occupants are susceptible to vibrations within this range. The resonant frequency range for the suspension system is smaller and fits into both the WBV as well as the “low frequency” range. It is thus paramount that the system identification input covers these frequencies [42]. In order to satisfy the preceding conditions, WBL was set as follows:

i. Control input \( u \) operates within ±10 V.

ii. WBL has the following properties:

- Seed strength of 22,641.
- 0.001 s sampling time.

iii. Hyperbolic Tangent is used for the activation function \( \sigma(x) \) as this ensures the DNN stability [28–30].

With regards to pruning and the choice of hidden layer size \( n_h \) the response of the DNN is analyzed for a range of \( n_h \) starting from one. The network size was increased until satisfactory results were attained for a credible range of randomly selected network parameters. Fig. 7 shows the general trend of the suspension travel output for the various hidden layer sizes. It is evident from this figure that a hidden layer size \( n_h \) of 8 is capable of capturing the sensitive dynamics of the system as it can pick up the sudden rate of change of suspension travel more adequately than the 4-neuron and 6-neuron configurations. Hence, it is suitable for system identification. The structure of the DNN is summarized in Table 2.

This DNN is trained offline using PSO algorithm. PSO was first introduced by Kennedy and Eberhart [44] and they mimic the food searching process of swarms. As in the swarm of animals, each

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Configuration of applied dynamic neural networks for the quarter-car system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Numerical value</td>
</tr>
<tr>
<td>Number of hidden layers</td>
<td>2</td>
</tr>
<tr>
<td>Number of neurons in first hidden layer</td>
<td>8</td>
</tr>
<tr>
<td>Number of delayed system inputs</td>
<td>1</td>
</tr>
<tr>
<td>Number of delayed system outputs</td>
<td>1</td>
</tr>
</tbody>
</table>
animal or particle searches a search space for a specific item. Each particle thereafter relays their success in relation to finding the desired item. Each particle will subsequently travel towards the position of the particle with the highest success at various speeds. At the same time, it will examine the area for a personal best position to achieve. The position of the particle refers to the location of the particle in n dimensional space where n is the number of parameters that are being altered. Searching towards the vicinity of the best particle and in the neighbourhood its personal best are referred to as the global and local searches respectively. The process repeats itself until the stopping criterion is met. The algorithm is summarized in the following steps [17]:

Step 1: Produce a random population of particles using the uniform distribution.
Step 2: Provide an initial small velocity to each particle using pseudo-random normal distributions.
Step 3: Choose the fittest particle as the best particle.
Step 4: If the stopping criteria has been met, then stop the algorithm, otherwise proceed to Step 5.
Step 5: Adjust the local and global search parameters according to:

\[ w_{\text{local}} = w_1 \left( \frac{n}{n_{\text{max}}} \right), \]
\[ w_{\text{global}} = w_2 \left( 1 - \frac{n}{n_{\text{max}}} \right), \]

where \( w_{\text{local}} \) and \( w_{\text{global}} \) are the local and global search parameters respectively; whilst \( w_1 \) and \( w_2 \) are the maximum local and global search weighting respectively. \( n \) and \( n_{\text{max}} \) are the current iteration and maximum number of iterations respectively.

Step 6: Compute the new positions of each particle in the search space using:

\[ X(t + 1) = X(t) + V(t + 1), \]

where \( X \) is a matrix comprising of the neural network parameters, \( V \) is a matrix consisting of velocity as each parameter varies

\[ V(t + 1) = w_1 V(t) + w_{\text{local}} \times \text{rand1}(P_{\text{best}} - X(t)) + w_{\text{global}} \times \text{rand2}(G_{\text{best}} - X(t)), \]

where \( G_{\text{best}} \) denotes the vector containing the parameters of the global best particle, \( P_{\text{best}} \) represents the matrix containing the personal best parameters of each particle. \( \text{rand1} \) and \( \text{rand2} \) are pseudo-random numbers.

Step 7: If the fitness of its new position is better than the fitness of its personal best, replace its personal best position with its current position and proceed to Step 8, otherwise continue straight to Step 8 without any adjustments.

Step 8: Find the fittest particle in the population and choose it as the best particle and return to Step 4.

In this learning process, the DNN parameters \( \beta, W, \gamma, \text{and } \xi \) are the problem variables that are determined by the PSO algorithm. \( \beta \) is a \( 1 \times 8 \) vector with \( \beta_n \) denoting the time constant of the \( n \)th neuron from the 8 present in the first hidden layer. The same applies to \( \gamma \) and \( \xi \) as well. \( W \) is the weighting matrix that connects each of the neurons of the hidden layer to each other and it is a square \( 8 \times 8 \) matrix with \( W_{ji} \) denoting the feedback weighting of the \( j \)th neuron into the \( i \)th neuron.

The objective function of the PSO algorithm during the system identification process is the mean square error (MSE) of the deviation between the actual and predicted outputs [20]:

\[ J = \text{MSE} = \frac{1}{2N} \sum_{j=1}^{N} (y_j - \hat{y}_j)^2, \]

where \( N \) is the total number of samples used in the input–output data. The DNN parameters chosen are listed in Table 3 and the variation of the global fitness with each iteration is presented in Fig. 8. The resulting verification and validation data are shown in Figs. 9 and 10, respectively.

Before controller design can begin, the stability of the DNN model must be ensured. Deng et al. [28], Deng et al. [29] and Garces et al. [30] suggest that the DNN stability will be guaranteed if the following conditions hold:

\[ \text{Table 3} \]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>200</td>
</tr>
<tr>
<td>Maximum number of iterations</td>
<td>200</td>
</tr>
<tr>
<td>Velocity inertia weight ((w_1))</td>
<td>0.5</td>
</tr>
<tr>
<td>Social inertia weight ((w_2))</td>
<td>2</td>
</tr>
<tr>
<td>Maximum global inertia weight ((w_3))</td>
<td>2</td>
</tr>
<tr>
<td>Fitness function</td>
<td>MSE</td>
</tr>
<tr>
<td>Goal for MSE</td>
<td>(9 \times 10^{-6})</td>
</tr>
</tbody>
</table>
Step i. The activation function $\sigma(x)$ is continuously differentiable.

Step ii. $\sigma(x)$ is bounded to $0 \leq \sigma(x) \leq 1$.

Step iii. Given $u_t \in \mathbb{R}^n$, there is a symmetric and positive solution $P$ to the following equation:

$$\beta^T P - P \beta = -\mu I.$$  \hspace{1cm} (33)

$I$ is an identity matrix and $\mu$ is a scaling factor which [30] suggested should have a value of 1.

Step iv. The inequality of Eq. (33) must be satisfied:

$$||W||^2 \leq \frac{\mu - 2||P||}{||P||},$$  \hspace{1cm} (34)

where $||\cdot||$ is the Euclidean norm of the specified matrix.

As the activation function $\sigma(x)$ is the hyperbolic tangent function, conditions i. and ii. are fulfilled. With the network computed through PSO algorithm, both Eq. (33) and the inequality of Eq. (34) are satisfied. Hence, it may be concluded that the DNN model attained through training is indeed stable.

**Control law formulation**

The DNN model is rearranged into the compatible control affine form that is required to derive the feedback linearizing law as follows:

$$\dot{x} = f(x) + g(x)u_t + \zeta x_{t-1},$$  \hspace{1cm} (35)

where $f(x) = W(x) - \beta x$, and $g(x) = \gamma$. The following steps involve computing the consecutive derivatives of the DNN model outputs until a corresponding derivative is explicitly a function of the control input $u_t$. The first derivative of the network output is computed as follows:

$$\dot{y} = \frac{\partial y}{\partial t} = \frac{\partial y}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial h_1(x)}{\partial x},$$

$$= \frac{\partial h_1}{\partial x}[f(x) + g(x)u_t] = \omega y h_1(x) + \omega g_1 h_1(x)u_t,$$

$$= \sum_{i=1}^d \frac{\partial h_1}{\partial x_i} \frac{\partial h_1}{\partial x} \ldots \ldots \frac{\partial h_1}{\partial x} [f(x) + g(x)u_t]^T,$$

$$= w_1 [-\beta x + \sum_{i=1}^d W_i \sigma(x_i)] = \omega y h_1(x).$$  \hspace{1cm} (36)

where $\omega y h_1(x) = \frac{\partial y}{\partial x} f(x)$, is known as the Lie derivative of $h_1$ along $f$, $g(x)$ is the consequence of DNN training, and its resulting matrix has its first element $g_1(x) = 0$. Furthermore, $\frac{\partial h_1}{\partial x_1}, \ldots, \frac{\partial h_1}{\partial x_d}$ are zero as $h_1$ is a function of $x_1$ only as per Eq. (27). Such values give rise to Eq. (36), where clearly the first derivative of the DNN model output $\dot{y}$ is not a function of the control input $u_t$. Subsequent computation of the second derivative of the network output produces:

$$\ddot{y} = \frac{\partial^2 y}{\partial t^2} = \frac{\partial y}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial h_2 h_1(x)}{\partial x} [f(x) + g(x)u_t],$$

$$= w_1 [-\beta x + \sum_{i=1}^d W_i (1 - \sigma(x_i)^2) + W_1 (1 - \sigma(x_2)^2)$$

$$+ \ldots + W_1 (1 - \sigma(x_d)^2)] [f(x) + g(x)u_t],$$

$$= d(x) + e(x) u_t = \omega y^2 h_1(x) + \omega g_2 h_1(x)u_t,$$  \hspace{1cm} (37)

where $d(x)$ or $\omega y^2 h_1(x)$ is the free response of the system and $e(x) u_t$ or $\omega g_2 h_1(x) u_t$ is the forced response of the system. In the above derivative of the output, the DNN of the PSO training yielded a matrix where $g_1(x)$, $g_2(x)$, $\ldots$, $g_d(x)$ were considerably large constants. Hence the computation of the second derivative of the network output $\dot{y}$ produced a solution which was explicitly dependent on the control input $u_t$. Hence, the relative degree of the system is two, which infers that the DNN is input–output linearizable as its relative degree is less than the number of states of the DNN (which corresponds to 8 as there are eight neurons in the hidden layer).

The next step in the controller formulation demands that the DNN dynamics now be transformed into a coordinate system which separates the observable and zero dynamics. The DNN may be described in terms of its observed and unobserved zero dynamics using the diffeomorphism as follows:

$$\eta = f_0(\xi, \eta),$$  \hspace{1cm} (38)

$$\dot{\xi} = A_\xi \dot{\xi} + B_\xi u_t + \bar{P}(w),$$  \hspace{1cm} (39)

$$\dot{\bar{\xi}} = A_\bar{\xi} \dot{\bar{\xi}} + B_\bar{\xi} \frac{u_t - \alpha(x)}{b(x)} + \bar{P}(w),$$  \hspace{1cm} (40)

$$\dot{\bar{x}} = C_\bar{\xi} \bar{\xi}. $$  \hspace{1cm} (41)
As the relative degree of the DNN is 2, the transformation yields a slightly different set of system matrices which are:

$$A_c = \begin{bmatrix} 0 & 1 \\ \lambda_0 & \lambda_1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \quad \Phi(w) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T.$$

(42)

The control law aims to create a linear mapping between the virtual input $\tilde{u}$ and the system network output $\tilde{y}$ as explained in Fig. 6 such that:

$$\lambda_2 \tilde{y} = \tilde{u}.$$  

(43)

$\Psi$ must be invertible such that:

$$\mathcal{L}_f \Psi_{i} = \frac{d\Psi_{i}}{dx}g(x), \quad r + 1 \leq i \leq n$$

(44)

where $n$ is the number of state variables. With regards to guaranteed stability, both the observable $\xi$ and unobservable $\eta$ system dynamics must be stable. The unobservable system dynamics $\eta$ are defined as non-trivial internal dynamics that remain once the output and observable system dynamics are forced to zero such that $\xi = 0$ and hence $\eta_{0}(\eta, 0)$. Such dynamics is termed the zero dynamics, and they tend to have a significant impact on the stability of the system. Asymptotic stability of the system is confirmed if the origin of the transformed system $\eta_{0}(\eta, 0)$ is an equilibrium point. Such stability reduces the dynamics of the rth derivative of the output described by Eq. (37) to:

$$\lambda_2 \tilde{y} = \tilde{u}.$$  

(45)

Hence the FBL control law required to linearize the DNN and to acquire the linear mapping preferred in Eq. (45) is of the form:

$$u = \frac{1}{\lambda_2 \mathcal{L}_g \mathcal{L}_f \Psi_{i}(x)} \left[ \tilde{u} - \lambda_2 \mathcal{L}_g \mathcal{L}_f \Psi_{i}(x) \right].$$

(46)

The virtual input $\tilde{u}$ may be designed using pole placement approach such that:

$$\tilde{u} = -\lambda_1 \tilde{y} - \lambda_0 \tilde{y} + \mathcal{T}.$$  

(47)

Consequently, the actual control law will take the form:

$$u = \frac{1}{\lambda_1 \mathcal{L}_g \mathcal{L}_f \Psi_{i}(x)} \left[ \mathcal{T} - \lambda_1 \mathcal{L}_g \mathcal{L}_f \Psi_{i}(x) - \sum_{i=0}^{1} \lambda_{i-1} \mathcal{L}_g \mathcal{L}_f \Psi_{i}(x) \right].$$

(48)

The new virtual control input $\mathcal{T}$ is determined through a multi-loop PID control system described in Fig. 5.

With regards to controller gains, there are now 9 controller gains to be optimized; namely the 6 PID gains of the multi-loop PID controller, and the 3 FBL controller gains $(\lambda_0, \lambda_1, \lambda_2)$. The performance index used to select these gains is the same presented in Eq. (20). The process of the manual tuning of the intelligent controller was rather cumbersome and rigorous as very small variations in $\lambda_0$, $\lambda_1$, and $\lambda_2$ cause considerable variations in the system response.

The best gains that could be obtained through manual tuning are listed in Table 4. The gains of the controller are selected using the PSO algorithm and the optimization parameters are listed in Table 3.

Table 4

<table>
<thead>
<tr>
<th>Controllers</th>
<th>Outer loop</th>
<th>Inner loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Kp</td>
<td>Kd</td>
</tr>
<tr>
<td>PID + PSO</td>
<td>1700</td>
<td>1200</td>
</tr>
<tr>
<td>DNNFBL</td>
<td>19,518</td>
<td>2107</td>
</tr>
<tr>
<td>DNNFBL + PSO</td>
<td>15,000</td>
<td>0</td>
</tr>
<tr>
<td>DNNFBL + PSO</td>
<td>23,500</td>
<td>35</td>
</tr>
</tbody>
</table>

Table 4: Selected gains for each algorithm.

Fig. 11 clearly indicates an improvement in the performance index from that of the PSO-optimized PID-controlled AVSS tuning. This infers that PSO-optimized controller tuning is successful in improving system performance. This figure also shows a better performance index for the DNNFBL controller, which infers that DNNFBL performs overall better than the PID case. However, this figure does not show if the inherent conflicting performance criteria of AVSS have been resolved. Thus investigation of each aspect of AVSS performance needs to be conducted to account for this. The corresponding optimal controller gains are listed in Table 4.

![Fig. 11. Convergence history of performance indices through the use of PSO algorithm for DNN-based FBL-controlled and PID-controlled AVSS.](image-url)
the objective function which gave suspension travel an appropriate weighting. However, improved settling time and transient response were achieved at the cost of increased peak and root mean squared (RMS) suspension travel. This may be attributed to the considerable weightings that were given to the remaining trade-offs in the cost function.

The DNNFBL + PSO controlled AVSS possesses better RMS and peak values of the suspension travel than its non-optimized counterpart. This proves that the PSO algorithm is an effective tool in improving the response of the system. Controller tuning of the DNNFBL case proved to be rather sensitive to changes in gains, which hence produced a weaker response than the optimized PID case as well. Therefore, it may be stated that the tuning of FBL gains is more rigorous and less intuitive than that of the PID tuning.

The ride comfort (body-heave acceleration) and road holding (wheel deflection) characteristics plots shown in Figs. 13 and 14 clearly depict the DNNFBL + PSO case in a favourable light as it was able to bring the ride comfort to the Less Discomfort range and contain both improved peak and RMS values of body-heave acceleration and wheel deflection. Furthermore, the DNNFBL + PSO

Table 5
Performance evaluations for the passive, PID-controlled, DNN-based feedback linearization controlled and PSO-augmented, DNN-based feedback linearization controlled suspension systems to a deterministic road bump.

<table>
<thead>
<tr>
<th>Cases</th>
<th>Passive</th>
<th>PID</th>
<th>PID + PSO</th>
<th>DNNFBL</th>
<th>DNNFBL + PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suspension travel (m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>0.025</td>
<td>0.023</td>
<td>0.0024</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td>PEAK</td>
<td>0.087</td>
<td>0.065</td>
<td>0.0075</td>
<td>0.079</td>
<td>0.075</td>
</tr>
<tr>
<td>Tyre deflection (m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>0.0064</td>
<td>0.0021</td>
<td>0.0019</td>
<td>0.0027</td>
<td>0.0016</td>
</tr>
<tr>
<td>PEAK</td>
<td>0.0206</td>
<td>0.0098</td>
<td>0.0095</td>
<td>0.0101</td>
<td>0.0084</td>
</tr>
<tr>
<td>Control voltage (Volt)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>N/A</td>
<td>0.679</td>
<td>0.68</td>
<td>0.846</td>
<td>0.683</td>
</tr>
<tr>
<td>PEAK</td>
<td>N/A</td>
<td>2.91</td>
<td>2.8</td>
<td>3.67</td>
<td>3.39</td>
</tr>
<tr>
<td>Body heave acceleration (m/s²)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>4.10</td>
<td>1.1383</td>
<td>0.895</td>
<td>1.74899</td>
<td>0.791</td>
</tr>
<tr>
<td>PEAK</td>
<td>13.35</td>
<td>5.3</td>
<td>4.7</td>
<td>7.2</td>
<td>4.3</td>
</tr>
<tr>
<td>Actuation force (N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>N/A</td>
<td>476</td>
<td>480</td>
<td>386</td>
<td>635</td>
</tr>
<tr>
<td>PEAK</td>
<td>N/A</td>
<td>1996</td>
<td>2000</td>
<td>1200</td>
<td>2600</td>
</tr>
</tbody>
</table>
| Settling time (s)      | 2.8     | 2.5  | 3.5       | 2.2    | 2.1          

Fig. 12. Comparison of the suspension travel responses for the PVSS, optimized PID-controlled, DNN-based feedback linearization controlled and PSO-augmented, DNN-based feedback linearization controlled suspension systems to a deterministic road bump.

Fig. 13. Comparison of the body-heave acceleration responses for the PVSS, optimized PID-controlled, DNN-based feedback linearization controlled and PSO-augmented, DNN-based feedback linearization controlled suspension systems to a deterministic road bump.

Fig. 14. Comparison of the wheel deflection responses for the PVSS, optimized PID-controlled, DNN-based FBL-controlled and PSO-augmented, DNN-based FBL-controlled AVSS to a deterministic road bump.
case displayed a lower degree of chattering in these aspects. These results show that the objective function for the PSO algorithm that contains each of the AVSS trade-offs is an effective tool for resolving design specifications particularly on ride comfort, road holding and system transient response. On the other hand, the PID + PSO controlled case performed better than the DNNFBL case and fell marginally short than the DNNFBL + PSO case in terms of peak and RMS values of the wheel deflection and the body-heave acceleration. It did also maintain a greater degree of chattering than the intelligent controller.

**Figs. 15 and 16** indicate that the success of the DNNFBL + PSO controller was achieved at the cost of actuation force and voltage. This is expected as a control input and hence actuation force is required to bring down the acceleration considerably into the Less Discomfort range of the ISO standards. However, hydraulic actuators in general are sensitive to changes in control input, where a slightly smaller case in control voltage as in the case of the optimized PID and DNNFBL controllers produces a larger variation in actuator forces. Hence, in order to satisfy both hard design specifications (such as ride comfort) and a reduced performance index, the actuator force will naturally be larger. In conclusion this data argues that actuator dynamics play a significant role in AVSS design and that it should not be neglected.

To test the sensitivity to parameter variations of the DNNFBL + PSO controller, several realistic suspension parameters were altered. These included tyre stiffness, vehicle mass and speed. Each parameter was varied by ±20% and the corresponding suspension travel responses are plotted in **Figs. 17 and 18**, respectively. **Figs. 17 and 18** indicate that the DNNFBL + PSO case has a satisfactory robustness to parameter variations. The system remained Bounded-Input Bounded-Output (BIBO) stable, and the steady-state error increased upon the variation of the various parameters. Alteration in vehicle mass contained the most severe steady-state error, followed by the adjustments in vehicle speed and tyre loading respectively. Its transient response did however remain similar to the original case where no topology was lost whatsoever. The optimized PID controller on the other hand displayed no steady-state error, but its transient response was worst than the DNNFBL + PSO controller with greater settling times and additional peaks. This implies that the intelligent controller has a better robustness to parameter variations.

**Fig. 15.** Comparison of the actuation force responses for the optimized PID-controlled, DNN-based FBL-controlled and PSO-augmented, DNN-based FBL-controlled AVSS to a deterministic road bump.

**Fig. 16.** Comparison of the control input voltage responses for the optimized PID-controlled, DNN-based FBL-controlled and PSO-augmented, DNN-based FBL-controlled AVSS to a deterministic road bump.

**Fig. 17.** Comparison of the suspension travel responses for the PSO-augmented, DNN-based FBL-controlled AVSS to −20% parameter variations in vehicle speed, mass and tyre loading.

**Fig. 18.** Comparison of the suspension travel responses for the PSO-augmented, DNN-based FBL-controlled AVSS to +20% parameter variations in vehicle speed, mass and tyre loading.
The robustness of the proposed controller was also tested for the case of a random road disturbance presented in Fig. 19. The suspension travel responses for the optimized PID and PSO-augmented DNNFBL subjected to this disturbance is presented in Fig. 20.

The optimized PID case reported larger peaks with a marginally greater RMS value. This implies that the DNNFBL+PSO performed better in this aspect as well, which further highlights its improved robustness.

System sensitivity is also investigated through frequency domain plots. The most sensitive frequencies of vibrations for human exposure range between 2 and 80 Hz [43]. These standards also explain that body-heave acceleration is used to quantify human exposure and hence body-heave acceleration under these frequencies is investigated. The frequency response of the proposed AVSS schemes covering this range is presented in Fig. 21 and it was generated using the Power-Spectral-Density (PSD) estimates from Matlab signal processing toolbox. The settings for this estimate are listed in Table 6.

In this frequency range, the DNNFBL+PSO controller showed substantially lower power ratios than its counterparts across most frequencies. It maintained the best power ratio in the most sensitive low frequency range. Beyond 10 Hz it performed worst than the PVSS, but the power ratios in this range for all the control methods are effectively negligible. These results once again highlight the robustness of the DNNFBL+PSO controller.

**Table 6**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>Algorithm Welch</td>
</tr>
<tr>
<td>Windowing function</td>
<td>Hanning</td>
</tr>
<tr>
<td>Number of points included</td>
<td>1024</td>
</tr>
<tr>
<td>in fourier transform (NNFT)</td>
<td></td>
</tr>
<tr>
<td>Length of window (NWind)</td>
<td>256</td>
</tr>
<tr>
<td>Sampling frequency</td>
<td>80Hz</td>
</tr>
</tbody>
</table>

**Conclusion and future work**

In light of the preceding discussion, the following conclusions are made. Firstly, the DNN was successful in learning the dynamics of the AVSS. However, the DNNFBL-controlled case produced weaker data than the benchmark PID controller as it proved rather tedious to tune. PSO overcame this tuning issue and produced significantly better results which completely outperformed the optimized PID-controlled case. The DNNFBL+PSO controlled case is characterized with improved ride comfort, road holding, suspension travel, settling time and contained a significantly lower degree of chattering as compared to the optimized PID-controlled case.

However, this success came at the price of increased actuation force. This is required to further minimize the effects of the transmitted road disturbance and hence improve the suspension travel response, ride comfort, settling time and road holding properties. On the contrary, the control input voltage did not alter significantly for the various cases. This may be attributed to the nature of the hydraulic actuator, where small increments in voltage develop large changes in force. Furthermore, the optimal intelligent controller displayed an acceptable sensitivity to parameter variations.

The DNNFBL+PSO control displayed the best robustness with the most desirable response for variations in parameter values, even though it had the shortfall of steady-state error. It showed better response than its PID counterpart in the frequency domain as well as when subjected to a random road disturbance.

In relation to future work, it is worth stating that success of the proposed controller may be extended to a full-car model to resolve its associating trade-offs. This is necessary as full-car models are much more complex and realistic. Furthermore, experimental validation should be carried out as real-world model contains additional complexities and introduces various issues that have been ignored in numerical simulations.
References


