Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Learning nonlinear robust control as a data-driven zero-sum two-player game for an active suspension system

Mircea-Bogdan Radac* and Timotei Lala**

* Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania (e-mail: mircea.radac@upt.ro)
** Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania (e-mail: timotei.lala@student.upt.ro)

Abstract: An optimal robust control data-driven learning solution is proposed for an active suspension system. The problem is formulated as a zero-sum two-player differential game (ZS-TP-DG), where the optimal control law and the worst-case disturbance control law must be searched for. The distinctive features of the proposed solution are: a Q-learning-like data-driven model-free (with unknown process dynamics) algorithm relying on collected input-state data from the process; neural networks being used as generic function approximators; validation on an active suspension system that is easily amenable to artificial road profile disturbance generation. The superiority of the ZS-TP-DG controller over another optimal controller learned in a disturbance-free context is validated and proven.

Keywords: active suspension system, hydraulic actuator, neural networks, optimal control, approximate dynamic programming, reinforcement learning, two-player zero-sum games.

1. INTRODUCTION

Improving passenger ride comfort in cars has been in continuous attention since the early developments of the first hydro-pneumatic suspension almost 70 years ago. While pure active suspensions remained a costlier option designated to higher-end cars, the cheaper semi-active systems are more widely found. The car suspension system is already a well-established benchmark within automotive control systems design (Acosta Lua et al., 2015; D'Andrea Novel et al., 2016; Radac and Precup, 2018a; He et al., 2019; Sardarmehni and Heydari, 2019), being subjected to a diversity of control techniques (Rettig and von Stryk, 2004; Wang et al., 2018; Hua et al., 2018; Rathai et al., 2019). Its main attractive feature from a control perspective is the highly underdamped character resulting from the common two-masses-springs-dampers modelling. A large body of scientific literature deals with the optimal active suspension control and in particular in that of reinforcement learning applied for the suspension control, to name a few (Howell et al., 1997), (Tognetti et al., 2009), (Bucak et al., 2012), (Akraminia et al., 2015), (Wang, 2018).

Approximate Dynamic Programming (ADP) (Wang et al., 2009) is the name by which Reinforcement learning (RL) (Busoniu et al., 2018) is better known to control engineering and it suggests an attractive optimal control design concept, owing its ever-increasing popularity to the ability of obtaining high-performance control when the process dynamics are (partially) unknown and nonlinear, under complete or incomplete process state measurement. With more recent applications (de Bruin et al., 2018; Tang et al., 2019; Treesatayapun 2019), by better exploiting the recent computational advances of generic function approximators such as neural networks (NNs), ADP has proved its capacity to better scale and deal with complex systems with many states and control inputs such as the ones stemming from video games (Mnih et al., 2016). This way, ADP can better handle the "curse of dimensionality" issue and mainstream as one of the representative data-driven model-free control techniques (Chi et al., 2018; Salvador et al., 2019; Radac and Precup, 2019).

Within the active suspension control problem, the road condition is an external disturbance treated as a process input that affects the ride comfort. Therefore, the control problem straightforward lends itself to the methods employed by H-infinity optimal robust control design. Fortunately, the H-infinity framework was translated to an L_2-gain optimal control problem for general nonlinear systems (Basar and Bernhard 1995; Van der Schaft, TAC 1992), where the objective is to find the (non-computable analytically) solution to the continuous (discrete) time Hamilton-Jacobi-Isaacs (HJI) equations. The L2-gain control problem has been extensively treated as a zero-sum two-player differential game (ZS-TP-DG) where the optimal control law and the worst-case disturbance law must be calculated as the minimax saddle-point solution to the HJI equation, assuming that one exists.

Several methods for finding state feedback controllers that solve the HJI equation using ADP have been developed in various works, including model-based implementations like (Abu-Khalaf et al., 2006) and (Vamvoudakis et al., 2010) for continuous-time nonlinear systems, (Liu et al., 2013) for discrete-time nonlinear systems and also model-free versions, like (Al-Tamimi et al., 2007, Kim et al., 2010) for discrete-time linear systems via Q-learning.

In the spirit of the model-free Q-learning methods dedicated to solving the ZS-TP-DG problem (Al Tamimi et al., 2007), learning the two optimal controllers must rely on collected data from the process, in the form of transition samples. While an active suspension could be setup to run on realistic road conditions for transition samples collection, it must be noticed that measuring the unknown road profile disturbance is not an acceptable solution in practice. However, artificial disturbances emulating road conditions are easily produced in fixed stands with the car left on-site. These artificial disturbances lead to enhanced state-action space exploration. Additionally, that the worst-case disturbance controller is a virtual one, used only in the controller learning phase and it does not have to be employed in feedback, after the optimal controller is found. After terminating the learning process, the car can be used in real-world road conditions without road profile measurement.

Upon the above aspects, the paper shows that it is possible to learn optimal robust controller in a model-free setting, using generic function approximators such as NNs. From the author's knowledge, this is a first successful attempt on a nonlinear active suspension, in a model-free context.

The following Section defines the optimal control problem formulation and proposes a model-free Q-learning-based solution. The case study in Section 3 validates the proposed solution on a realistic quarter-car active suspension model and provides discussions and implementation details. Final conclusions are the subject of the fourth Section.

2. THE CONTROL PROBLEM AND THE PROPOSED SOLUTION

In order to solve the ZS-TP-DG problem for a general nonlinear process with unknown dynamics

$$x_{k+1} = P(x_k, u_k, d_k), \tag{1}$$

where, at sample instant *k*, the state is $x_k \in \Omega_x \subset R^n$, the control input is $u_k \in \Omega_u \subset R^m$ and the disturbance input is $d_k \in \Omega_d \subset R^p$, with $P: \Omega_x \times \Omega_u \times \Omega_d \to \Omega_x$ and with domains $\Omega_x, \Omega_u, \Omega_d$ assumed compact convex subsets of the real numbers of corresponding dimensions. The goal is to solve the optimization problem minimizing a cost function (c.f.) as

$$\begin{aligned} &C^*, D^* = \arg\min_C \max_D J(x_k), \\ &J(x_k) = \sum_{j=k}^{\infty} F(x_j) + C(x_j)^T W_C C(x_j) - \gamma^2 D(x_j)^T W_D D(x_j), \end{aligned} \tag{2}$$

where $u_k = C(x_k), C: \Omega_x \to \Omega_u$ is a state feedback controller and $d_k = D(x_k), D: \Omega_x \to \Omega_d$ is a disturbance controller, $F(x_k) > 0 \in R$ is a state penalty function, W_C, W_D are positive definite square weighting matrices and $\gamma \geq \underline{\gamma} > 0 \in R$ is a given constant greater than its smallest value $\underline{\gamma}$ for which the state feedback control system resulting from (1) combined with $C(x_k), D(x_k)$ is stabilized. In particular, the class of *admissible controllers* are those who render $J(x_k)$ finite when starting from any x_k.

In general, some restrictions on γ apply, in order for (2) to be solvable (Al Tamimi et al., 2007). Assuming a solvable ZS-TP-DG problem (2), the optimal controller $C^*(x_k)$ ensures that the L_2 gain of the closed-loop makes

$$\sum_{k=0}^{\infty} F(x_k) + C^*(x_k)^T W_C C^*(x_k) \leq \gamma^2 \sum_{k=0}^{\infty} d_k{}^T W_D d_k, \tag{3}$$

for any disturbance $d_k \in l_2$ within the l_2 space of square-integrable functions. Moreover, $C^*(x_k)$ and $D^*(x_k)$ must be in minimax saddle-point equilibrium.

To solve (2) in a model-free style, a batch-fitted variant of Q-learning is employed. First, $J(x_k)$ is extended with the Q-function defined the cost of taking any actions u_k, d_k in current state and then following the fixed control strategies $C(x_k), D(x_k)$ as

$$\begin{aligned} &Q(x_k, u_k, d_k) = F(x_k) + u_k^T W_C u_k - \gamma^2 d_k^T W_D d_k + \\ &+ J(x_{k+1}) = \$(x_k, u_k, d_k) + J(x_{k+1}). \end{aligned} \tag{4}$$

The previous Q-function fulfils the Bellman equation and its optimal version $Q^*(x_k, u_k, d_k) = \min_C \max_D Q(x_k, u_k, d_k)$ also leads to the optimal c.f. value $J^*(x_k) = Q^*(x_k, C^*(x_k), D^*(x_k))$ of (2).

The search for the optimal Q-function is proposed in the following. Function approximators are considered for the Q-function and for the controllers $C(x_k), D(x_k)$ and let them be parameterized as $\hat{Q}(x_k, u_k, d_k, \pi_Q), \hat{C}(x_k, \pi_C), \hat{D}(x_k, \pi_D)$, respectively, with $\pi_i, i \in \{Q, C, D\}$ vectors of dimension corresponding to the number of tunable weights of the approximator (e.g. NN weights).

Starting with an available dataset of transition samples containing tuples of the form $S = \{(x_k, u_k, d_k, x_{k+1})\}$ and with initialization $\pi_i^0, i \in \{Q, C, D\}$ not necessarily corresponding to admissible controllers, the following steps are alternated at each iteration *j* of a Value Iteration-like Q-learning algorithm:

Step 1. Update the c.f. $\hat{Q}^{j+1}$ (i.e. find π_Q^{j+1}) based on current iteration parameters $\pi_Q^j, \pi_C^j, \pi_D^j$. Such an update can be formulated, e.g. as the optimization

$$\pi_Q^{j+1} = \arg\min_{\pi} \frac{1}{|S|} \sum_{k=1}^{|S|} \left(\begin{aligned} &\hat{Q}(x_k, u_k, d_k, \pi) - \$(x_k, u_k, d_k) - \\ &\hat{Q}(x_{k+1}, \hat{C}(x_{k+1}, \pi_C^j), \hat{D}(x_{k+1}, \pi_D^j), \pi_Q^j) \end{aligned} \right)^2 \tag{5}$$

over the entire batch of $|S|$ transition samples.

Step 2. Improve the controller $\hat{C}(x_k)$ (i.e. find π_C^{j+1}) as

$$\pi_C^{j+1} = \arg\min_{\pi} \hat{Q}(x_k, \hat{C}(x_k, \pi), \hat{D}(x_k, \pi_D^j), \pi_Q^{j+1}). \quad (6)$$

Step 3. Improve the controller $\hat{D}(x_k)$ (i.e. find π_D^{j+1}) as

$$\pi_D^{j+1} = \arg\max_{\pi} \hat{Q}(x_k, \hat{C}(x_k, \pi_C^j), \hat{D}(x_k, \pi), \pi_Q^{j+1}). \quad (7)$$

Step 4. If termination condition is met (*j* reaches a predefined value or no more changes in π_Q^j), stop the iterations, else go to *Step 1*.

Noticeable, the c.f. (5) is a mean squared errors (MSE) commonly used in the NNs training phase. Then, a NN with input $[x_k^T, u_k^T, d_k^T]^T$ and output $\$(x_k, u_k, d_k) + \hat{Q}(x_{k+1}, \hat{C}(x_{k+1}, \pi_C^j), \hat{D}(x_{k+1}, \pi_D^j), \pi_Q^j)$ built from transition samples from *S*, when subjected to training procedure in term of its weights π_Q^j, actually solves (5). On the other hand, (6) and (7) are differently solved. One solution is to set the targets of the cascaded NN $\hat{Q}(x_k, \hat{C}(x_k, \pi_C^j), \hat{D}(x_k, \pi_D^j), \pi_Q^{j+1})$ equal to zero for all inputs x_k and then minimize (maximize) w.r.t. π_C^j (π_D^j) by normal NN training procedure, with π_Q^{j+1} kept fixed (Radac et al., 2018; Radac and Precup, 2018b; Radac and Lala, 2019). Otherwise, the gradient descent (ascent) steps

$$\pi_C^{[j+1]} = \pi_C^{[j]} - \frac{\alpha_1}{B_1} \sum_{k=1}^{B_1} \left. \frac{\partial \hat{Q}(x_k, u_k, \hat{D}(x_k, \pi_D^j), \pi_Q^{j+1})}{\partial u_k} \right|_{\pi_C^{[j]}} \left. \frac{\partial \hat{C}(x_k, \pi)}{\partial \pi} \right|_{\pi_C^{[j]}}, \quad (8)$$

$$\pi_D^{[j+1]} = \pi_D^{[j]} + \frac{\alpha_2}{B_2} \sum_{k=1}^{B_2} \left. \frac{\partial \hat{Q}(x_k, \hat{C}(x_k, \pi_C^j), d_k, \pi_Q^{j+1})}{\partial d_k} \right|_{\pi_D^{[j]}} \left. \frac{\partial \hat{D}(x_k, \pi)}{\partial \pi} \right|_{\pi_D^{[j]}}, \quad (9)$$

with positive step size learning rates α_1, α_2 can be called for a number of times T_1 (T_2), with gradients cumulated on mini-batches of randomly selected states x_k (counted as $B_1 \le |S|, B_2 \le |S|$) or on all states x_k from *S*, starting from π_C^j and π_D^j, respectively. Upon convergence (maximum number of iterations T_1, T_2), π_C^{j+1} and π_D^{j+1} are obtained.

Algorithm 1 summarizes the model-free Q-learning based solution for the ZS-TP-DG with NNs as follows.

Algorithm 1. NN-based ZS-TP-DG Q-learning

1. Available input: dataset *S*.

2. Select $\bar{j}, \gamma, B_1, B_2, T_1, T_2, \alpha_1, \alpha_2, \Delta_\pi$. Select NNs architecture and training settings. Initialize $j = 0, \pi_Q^0, \pi_C^0, \pi_D^0$.

3. Train the Q-function NN by solving (5) to find π_Q^{j+1}.

4. Select a random mini-batch of B_1 states x_k from *S*. Perform T_1 gradient descent steps (8) to find π_C^{j+1}.

5. Select a random mini-batch of B_2 states x_k from *S*. Perform gradient ascent steps (9) for T_2 times to find π_D^{j+1}.

6. If $j < \bar{j}$ and $\left\| \pi_Q^{j+1} - \pi_Q^j \right\| > \Delta_\pi$, make *j*=*j*+1 and jump to 3, else stop.

Upon exiting Algorithm 1, the ZS-TP-DG controller holds. Algorithm 1 is exemplified in the following section, on a case study related to a hydraulic active suspension process.

3. CASE STUDY

3.1 The active suspension process

The continuous-time state-space model of the active suspension system for a quarter-car is (Huang et al., 2018)

$$\begin{cases} \dot{\bar{x}}_1 = \bar{x}_2 \\ \dot{\bar{x}}_2 = \frac{1}{m_s}(-b_s(\bar{x}_2 - \bar{x}_4) - k_s(\bar{x}_1 - \bar{x}_3) - k_{sn}(\bar{x}_1 - \bar{x}_3)^3 + \frac{A}{\varphi}\bar{x}_5) \\ \dot{\bar{x}}_3 = \bar{x}_4 \\ \dot{\bar{x}}_4 = \frac{1}{m_u}(b_s(\bar{x}_2 - \bar{x}_4) + k_s(\bar{x}_1 - \bar{x}_3) + k_{sn}(\bar{x}_1 - \bar{x}_3)^3 + \\ \qquad - k_t(\bar{x}_3 - \bar{x}_6) - b_t(\bar{x}_4 - \Xi_1 d) - \frac{A}{\varphi}\bar{x}_5) \\ \dot{\bar{x}}_5 = -\beta\bar{x}_5 - \varphi A\alpha(\bar{x}_2 - \bar{x}_4) + \varphi\phi\kappa\Xi_2 u, \\ \dot{\bar{x}}_6 = \Xi_1 d \end{cases} \quad (10)$$

$$\text{with} \quad \kappa = \text{sgn}[P_s - \text{sgn}(\Xi_2 u)\frac{\bar{x}_5}{\varphi}]\sqrt{\left|P_s - \text{sgn}(\Xi_2 u)\frac{\bar{x}_5}{\varphi}\right|},$$

where the numerical values of the model parameters used in simulation are (Huang et al., 2018) $m_s = 600kg$, $m_u = 60kg$, $k_t = 200000\,N/m$, $b_t = 1000\,Ns/m$, $k_{sn} = 1000\,N/m$, $k_s = 18000\,N/m$, $b_s = 2500\,Ns/m$, $\varphi = 1\times10^{-7}$, $\beta = 1s^{-1}$, $A = 3.35\times10^{-4}m^2$, $P_s = 10342500Pa$, $\alpha = 4.151\times10^{13}\,N/m^{5/2}$, $\phi = 1.545\times10^9\,N/m^{5/2}$. Here, $\bar{x}_1$ and $\bar{x}_3$ are the displacements of the sprung and unsprung masses, w.r.t. their rest position. The state $\bar{x}_5$ in the fifth equation is the force generated by the four-way valve-piston hydraulic actuator which is voltage-driven by the control input *u*. The input disturbance *d* is the second process input and it is the derivative of the road profile. The normalization scaling constants $\Xi_1 = 0.03, \Xi_2 = 0.001$ are introduced in (10) to ensure that $d \in [-1;1]$ results in an equivalent disturbance of amplitude 3 *cm/s* that models the road conditions and that $u \in [-1;1]$ operates in the active area of the active force generator. The signum function is sgn(.). The states are normalized after transition samples collection phase and illustrated in the following sub-section.

An equivalent discrete-time model of dynamics (10) is obtained with sample period $T_s = 0.01$ seconds (s) and used for input-state transition samples collection in the following. The actual model (10) is not used in the learning process. The discrete-time counterpart of (10) is of the form

$x_{k+1} = P(x_k, u_k, d_k)$ where $x_k = [x_{1,k}, ..., x_{6,k}]^T$ groups the discrete-time states.

3.2 Transition samples collection

The transition samples tuples (x_k, u_k, d_k, x_{k+1}) are collected using the following settings for u_k, d_k. The control input $u_k \in [-1;1]$ is modelled as successive piece-wise constant steps with uniformly random amplitude lasting 0.5 s, upon which additive uniform random noise is added every T_s seconds, with amplitude $\upsilon_1 \in [-1e-3; 1e-3]$. The disturbance input $d_k \in [-1;1]$ is modelled as successive piece-wise constant steps lasting 0.6 s having uniformly random amplitude and additively perturbed by a similar uniform random noise. After a 200 s experiment time, $|S| = 20000$ tuples are collected and stored in the database S. The states are normalized to $x_{i,k} \in [-1;1], i = \overline{1,6}, k = \overline{1,|S|}$ by dividing each state with its corresponding maximal absolute value $\max_k |x_{i,k}|$ from the recorded history.

3.3 NNs approximators and learning settings

To implement Algorithm 1, the next settings are used. The NN $\hat{Q}(x_k, u_k, d_k, \pi_Q^j)$ is a feed-forward fully connected one with 8 inputs (6 states and 2 control inputs), one hidden layer with hyperbolic tangent activation having 30 neurons and an output layer with one linearly activated neuron (shortly, 8–30–1). A fast scaled conjugate batch training is used for maximum 500 episodes and the training data is randomly divided in 80% actual training data and 20% validation data, the latter being used to force early stopping after ten successive increases of the MSE (criterion is also used in training) measured on the validation data.

The two controllers $\hat{C}(x_k, \pi_C^j)$ and $\hat{D}(x_k, \pi_D^j)$ have similar architectures (i.e. 6–6–1), but their training is performed on mini-batches using gradient descent/ascent according to (8), (9) for a given number of steps. Each NN has 6 inputs (the states) and one output (two in total, corresponding to the two process control input signals u_k, d_k). All three NNs weights $\pi_i, i \in \{Q, C, D\}$ are initialized as zero-mean small uniformly random numbers in [–0.005; 0.005].

Other learning settings are presented. In (2), the penalty in the c.f. is selected as $20x_{1,k}^2 + u_k^2 - \gamma^2 d_k^2$, for $\gamma = 2$ (i.e. $F(x_k) = 20x_{1,k}^2$, $W_C = W_D = 1$ in (2)). The objective is to limit the displacement of the sprung mass (i.e. the car) when changes in the road profile occur. In Algorithm 1, $\bar{j} = 500, T_1 = T_2 = 50, \alpha_1 = \alpha_2 = 1e-3, \Delta_\pi = 1e-5$. The same mini-batch of B_1=B_2=128 states x_k are randomly extracted from S at each iteration j of Algorithm 1, then used in the gradient descent/ascent Steps 4 and 5 of the Algorithm 1. After $\bar{j} = 500$ iterations, the final optimal controllers $\hat{C}^*(x_k, \pi_C^{500})$ and $\hat{D}^*(x_k, \pi_D^{500})$ of the converged algorithm result. The ZS-TP-DG saddle-point solution corresponding to $\hat{Q}^*(0, u_k, d_k, \pi_Q^{500})$ near $x_k = 0$ is verified in Fig. 1.

3.4 Obtained results and discussions

For comparison, a simple optimal controller (SOC) is learned, not using a ZS-TP-DG-type c.f. Instead, it solves

$$C^* = \arg\min_C J(x_k), \quad \text{where } J(x_k) = \sum_{i=k}^{\infty} 20x_{1,i}^2 + u_i^2, \tag{11}$$

and the first part of the penalty is the same with that of the ZS-TP-DG c.f. The problem (2) is solved in a quite similar manner as the ZS-TP-DG problem, using a variant of Algorithm 1 in which only the step 5 is not employed, since the SOC relies only on the Q-function estimate $\hat{Q}(x_k, u_k, \pi_Q^j)$ and on the controller estimate $\hat{C}(x_k, \pi_C^{\bar{j}})$. The transition samples were collected in the same settings for u_k, but without a disturbance (d_k=0). Meaning that the model (10) is reduced to a fifth order one. The NNs architectures for $\hat{Q}$ and $\hat{C}$ are 6–30–1 and 5–5–1, respectively. The gradient descent steps in (8) are similarly used for $T_1 = 50$ times. The rest of the parameters in Algorithm 1 are the same and after $\bar{j} = 500$ iterations, the optimal controller and Q-function result.

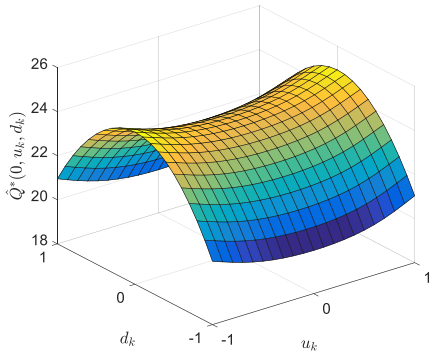

Fig. 1. Saddle point of $\hat{Q}^*(0, u_k, d_k, \pi_Q^{500})$ after Algorithm 1.

To assess the performance of the ZS-TP-DG and SOC controllers, the attenuation c.f. (Mehraeen et al., 2013; Liu et al., 2013)

$$J_{test} = \left(\sum_{k=0}^{10000} 20x_{1,k}^2 + u_k^2\right) \Big/ \left(\gamma^2 \sum_{k=0}^{10000} d_k^2\right) \tag{12}$$

is defined and measured on a 100 s test scenario where the used disturbance $d_k \in [-1;1]$ is a succession of piece-wise constant steps having uniform random amplitudes and lasting for 0.5 s. This test disturbance was not seen during the transition samples collection for the ZS-TP-DG controller learning. Note that the ZS-TP-DG disturbance controller $\hat{D}^*(x_k, \pi_D^{500})$ is not actually used in the loop and it is necessary only in the learning phase of Algorithm 1.

For measuring attenuation, both the ZS-TP-DG and the SOC controllers use their respective learned $\hat{C}^*(x_k, \pi_C^{500})$ in closed loop. It is obtained that $J_{test}^{ZS-TP-DG} = 9.98e-4$, $J_{test}^{SOC} = 57.34e-4$, clearly indicating the effective attenuation with the ZS-TP-DG controller. $J_{test}^{ZS-TP-DG} < 1$ shows that the finite-horizon version of (3) is verified.

The CS response with respect to the test disturbance input d_k obtained with the two controllers (ZS-TP-DG and SOC) in closed-loop is supplemented with the open-loop response (no controller $C(x_k)$ used, meaning $u_k = 0$). The results are shown in Fig. 2 only for the first 2000 samples (20 s) and reveal that the ZS-TP-DG controller manages to keep $x_{1,k}$ near zero in the presence of the road profile derivative test disturbance input d_k, better than the SOC controller.

A frequency response function estimator (Tognetti et al., 2009) is used to measure the transmissibility from the disturbance input d_k to the output $x_{1,k}$, for the approximately linear closed-loop CS. The result is shown in Fig. 3. The ZS-TP-DG controller reduces the CS resonant mode and also offers more attenuation at lower frequencies, with respect to the SOC one.

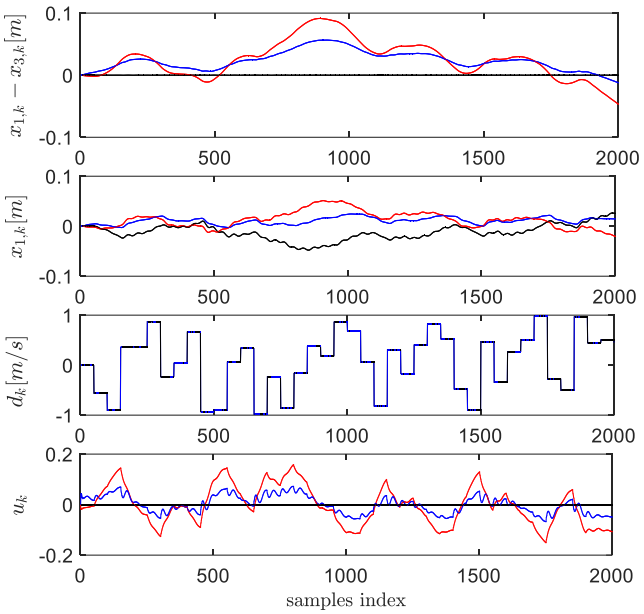

Fig. 2. Obtained responses in open-loop (u_k=0) (black), and in closed-loop with the ZS-TP-DG controller (blue) and with the SOC controller (red).

4. CONCLUSIONS

The value of the measured attenuation c.f. J_{test}, together with the results from Fig. 2 and Fig. 3, leads to the conclusion that the ZS-TP-DG controller is robust to road disturbances, fulfilling the design objective. The ZS-TP-DG control better handles disturbances than the SOC, even though the latter inherently possesses some disturbance rejection capability.

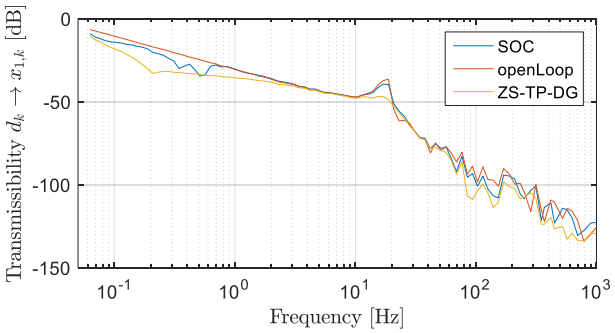

Fig. 3. Transmissibility $_{d_k \rightarrow x_{1,k}}$ in open-loop (u_k=0), and with the controllers ZS-TP-DG and SOC in closed-loop.

Especially for the active suspension system, artificial disturbances emulating road conditions are easily produced in fixed stands, for collecting transition samples, after which learning takes place. After the ZS-TP-DG control learning process has ended, the disturbance controller $\hat{D}^*(x_k)$ is not needed and the disturbance input itself must not be measured for feedback purposes, since the closed-loop uses only the controller $\hat{C}^*(x_k)$. The proposed design presents significant practical interest, being attractive for physical prototypes.

REFERENCES

Acosta Lua, C., Toledo, B.C., Di Gennaro, S., and Martinez-Gardea, M. (2015). Dynamic control applied to a laboratory antilock braking system. *Math Probl. Eng.*, Article ID 896859, 10 pp.

Abu-Khalaf, M., Lewis, F.L., and Huang, J. (2006). Policy iterations and the Hamilton-Jacobi-Isaacs equation for the H∞ state feedback control with input saturation. *IEEE Trans. Autom. Control,* 51, 1989-1995.

Akraminia, M., Tatari, M., Fard., M., and Jazar, R.N. (2015). Designing active vehicle suspension system using critic-based control strategy. *Nonl. Eng.*, 4(3), 141-154.

Al-Tamimi, A., Lewis, F.L., and Abu-Khalaf, M. (2007). Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3), 473-481.

Basar, T., and Bernhard, P. (1995). *H∞* optimal control and related minimax design problems. Cambridge, MA: Birkhauser.

Bucak, I.O., and Oz, H.R. (2012). Vibration control of a nonlinear quarter-car active suspension system by reinforcement learning. *Intl. J. Syst. Sci.*, 43(6), 1177-1190.

Busoniu, L., de Bruin T., Tolic, D., Kober, J., and Palunko, I. (2018). Reinforcement learning for control: Performance, stability, and deep approximators. *Annu. Rev. Control*, 46, 8-28.

Chi, R., Hou, Z., Jin, S., and Huang, B. (2018). Computationally efficient data-driven higher order

optimal iterative learning control. *IEEE Trans. Neural Netw. Learn. Syst*., 29(12), 5971-5980.

D'Andrea Novel, B., Menhour, L., Fliess., M., and Mounier., M. (2016). Some remarks on wheeled autonomous vehicles and the evolution of their control design. *Proc. of 9th IFAC Symp. Intell. Autonom. Veh.* (*IAV 2016*), Leipzig, Germany, 49(15), 199-204.

de Bruin, T., Kober, J., Tuyls, K., and Babuska, R. (2018). Integrating state representation learning into deep reinforcement learning. *IEEE Robot. Autom. Lett*., 3(3), 1394-1401.

He, Y., Lu, C., Shen, J., and Yuan, C. (2019). Design and analysis of output feedback constraint control for antilock braking system with time-varying slip ratio. *Math. Probl. Eng.*, Article ID 8193134, 11 pp.

Howell, M.N., Frost, G.P., Gordon, T.J., and Wu, Q.H. (1997). Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics*, 7(3), 263-276.

Hua, C., Chen, J., Li, Y., and Li, L. (2018). Adaptive prescribed performance control of half-car active suspension system with unknown dead-zone input. *Mech. Syst. Signal Process.*, 111, 135-148.

Huang, Y., Na, J., and Gao, G. (2018). Approximation-free control for vehicle active suspension with hydraulic actuator. *IEEE Trans. Ind. Electr*., 65(9), 7258-7267.

Kim, K. H., and Lewis, F. L., (2010). Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI. *Automatica*, 46, 1320-1326.

Liu, D., Li., H, and Wang, D. (2013). Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm. *Neurocomputing*, 110, 92-100.

Mehraeen, S., Dierks, T., Jagannathan, S., and Crow, M.L. (2013). Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks. *IEEE Trans. Cybern*., 43, 1641-1655.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu A.A., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529-533.

Radac, M.-B., and Precup, R.-E. (2018a). Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning. *Neurocomput.*, 275, 317-329.

Radac, M.-B., and Precup, R.-E. (2019). Data-Driven model-free tracking reinforcement learning control with VRFT-based adaptive actor-critic. *Appl. Sci.*, 9(9), 1807.

Radac, M.-B., and Precup, R.-E. (2018b). Data-driven MIMO model-free reference tracking control with nonlinear state-feedback and fractional order controllers. *Appl. Soft Comp.*, 73, 992-1003.

Radac, M.-B., Precup, R.-E., and Roman R.-C. (2018). Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-Learning. *ISA Trans.*, 73, 227-238.

Radac, M.B., and Lala, T. (2019). Learning output reference model tracking for higher-order nonlinear systems with unknown dynamics. *Algorithms*, 12(6), 121.

Rathai, K.M.M., Alamir, M., and Sename, O. (2019). Experimental implementation of model predictive control scheme for control of semi-active suspension system. *Proc. 9th IFAC Intl. Symp. Adv. Automotive Control*, Orleans, France, 261-266.

Rettig, U., and von Stryk, O. (2004). Optimal and robust damping control for semi-active vehicle suspension. In: Progress in Industrial Mathematics at ECMI 2002, (Buikis A., Čiegis R., Fitt A. D., eds.), vol. 5, 353-361, Springer, Berlin Heidelberg.

Salvador, J.R., Ramirez, D.R., Alamo, T., and de la Pena, D.M. (2019). Offset free data driven control: application to a process control trainer. *IET Control Theory Appl*., 13(18), 3096 - 3106.

Sardarmehni, T., and Heydari, A. (2019). Sub-optimal switching in anti-lock brake systems using approximate dynamic programming. *IET Control Theory Appl.*, 13(9), 1413-1424.

Tang, D., Chen, L., Tian, Z.F., and Hu, E. (2019). Modified value-function-approximation for synchronous policy iteration with single-critic configuration for nonlinear optimal control. *Int. J. Control*, DOI: 10.1080/00207179.2019.1648874.

Treesatayapun, C. (2019). Knowledge-based reinforcement learning controller with fuzzy-rule network: experimental validation. *Neural Comput. Appl.*, DOI: 10.1007/s00521-019-04509-x, pp. 1-15.

Tognetti, S., Savaresi, S.M., Spelta, C., and Restelli, M. (2009). Batch reinforcement learning for semi-active suspension control. *Proc. of 18th IEEE Intl. Conf. Control Appl.*, Saint Petersburg, Russia, 582-587.

Vamvoudakis, K. G., and Lewis, F. L., (2010). Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46, 878-888.

Van Der Schaft, A. J. (1992). L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control. I*EEE Trans. Autom. Control*, 37(6), 770-784.

Wang, F.-Y., Zhang, H., Liu D. (2009). Adaptive dynamic programming: An introduction. *IEEE Comput. Intell. Mag.*, 39-47.

Wang, X. (2018). Semi-active adaptive optimal control of vehicle suspension with a magnetorheological damper based on policy iteration. *J. Intell. Material Syst. Struct*., 29(2), 255-264.

Wang, H.P., Mustafa, G.I.Y., and Tian, Y. (2018). Model-free fractional-order sliding mode control for an active vehicle suspension system. *Adv. Eng. Software*, 115, 452-461.