24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-0963
STATE DEPENDENCE RICCATI EQUATION FOR THE TRAINING OF THE ECHO STATE NETWORK FOR CONTROL HALF-CAR SYSTEM

Angelo Marcelo Tusset
Eduardo Moletta
Jose Carlos de Pontes
Hugo Valadares Siqueira
Federal University of Technology – Paraná, 84016-210, Ponta Grossa – PR, Brazil
edumoletta@gmail.com; jcapontes@utfpr.edu.br; hugosiqueira@utfpr.edu.br; tusset@utfpr.edu.br

Abstract. *In this work is proposed an echo state network (ESN) from the controlled State Dependence Riccati Equation (SDRE) for control of the nonlinear half-car model subject to sinusoidal perturbation. To reduce the computational cost of the SDRE control is considered one recurrent neural network ESN, that after the training of the ESN is necessary only product of matrices to obtain control signal.*

Keywords: *SDRE control, ESN, half-car model, nonlinear model.*

1. INTRODUCTION

The half-car system is the nonlinear dynamic model for the vehicular suspension, in which there is a nonlinearity through dissipative forces of the shock absorbers, restorers of the springs and of the tires. The control of half-car system is necessary to maintain the stability of vehicle and comfort passengers. In the half-car mathematical model, the system subjected to a disturbance by a sinusoidal signal was studied, in which combined to the nonlinear terms can lead to a chaotic behavior. (Tusset *et al*., 2014).

Figure 1 represents a half-car model widely used in the designing of active control (Tusset *et al*., 2014).

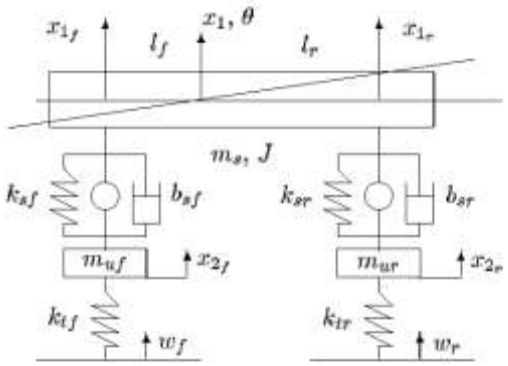

Figure 1. Half-car model

The model is composed by three masses, where m_s is the chassis mass, m_{uf} is the front wheel mass and m_{ur} is the rear wheel mass. The mathematical model of the system can be written with four-degrees-of-freedom. The mass of the chassis is considered as a rigid body with vertical displacement and pitch rotation. It was defined x_1 as the displacement of the center of gravity of the chassis and θ is the pitch angle of m_s. The vertical displacement of the front mass m_s is denoted by x_{1f} and the displacement of the rear mass m_s is denoted by x_{1r}. The vertical displacement of mass m_{uf} is denoted by x_{2f} and the displacement of mass m_{ur} of the rear wheel by x_{2r}. The support motions are $w_f = \alpha \sin(2\pi\omega t)$ in the front and $w_r = \alpha \sin(2\pi\omega t + \pi)$ at the rear of the car, that are caused by road irregularities. The control forces are F_f in front and F_r in rear. Suspension for both front and rear is comprised of a leaf springs k_{sf} in front and k_{sr} in rear, and dampers in front b_{sf} and in rear b_{sr} . The tires are represented by springs, k_{tf} for the front tire and k_{tr}, for the rear tire. The forces exerted by dampers are represented by Eq. (1):

$$F_{bsf} = b_f^l\left(\dot{\Delta}_f\right) - b_{sf}^y\left|\dot{\Delta}_f\right| + b_{sf}^{nl}\sqrt{\left|\dot{\Delta}_f\right|}\,\mathrm{sgn}\left(\dot{\Delta}_f\right);\ F_{bsr} = b_r^l\left(\dot{\Delta}_r\right) - b_{sr}^y\left|\dot{\Delta}_r\right| + b_{sr}^{nl}\sqrt{\left|\dot{\Delta}_r\right|}\,\mathrm{sgn}\left(\dot{\Delta}_r\right) \quad (1)$$

where: $\dot{\Delta}_f = \dot{x} - \dot{x}_{2f} - l_f\dot{\theta}\cos\theta$, $\dot{\Delta}_r = \dot{x} - \dot{x}_{2r} + l_r\dot{\theta}\cos\theta$.

The damping coefficient $b_{s\_}$ is composed by ($b_{s\_}^l$), the linear damping coefficient, and $b_{s\_}^{nl}$, the nonlinear characteristics of the damper. $b_{s\_}^y$ represents the asymmetric damping behavior.

The restoring forces provided by either front or rear suspensions can be represented by Eq. (2):

$$F_{ksf} = k_{sf}^l(\Delta_f) + k_{sf}^{nl}(\Delta_f)^3;\ F_{ksr} = k_{sr}^l(\Delta_r) + k_{sr}^{nl}(\Delta_r)^3 \quad (2)$$

where: $\Delta_f = x - x_{2f} - l_f\sin\theta$, $\Delta_r = x - \dot{x}_{2r} + l_r\sin\theta$,

The spring coefficient $k_{s\_}$ is composed by ($k_{s\_}^l$), the linear spring coefficient, and $k_{s\_}^{nl}$, the non-linear characteristics of the spring.

The tire can be represented by a linear spring which provide force denoted by Eq. (3).

$$F_{ktf} = k_{tf}(x_{2f} - w_f);\ F_{ktr} = k_{tr}(x_{2r} - w_r) \quad (3)$$

The vertical force, pitch moment and force acting on the displacement of the chassis can be represented by:

$$F_{ms} = m_s\ddot{x}_1;\ F_\theta = J\ddot{\theta};\ F_{muf} = m_{uf}\ddot{x}_{2f};\ \ F_{mur} = m_{ur}\ddot{x}_{2r} \quad (4)$$

Considering the characteristics of the springs and dampers, the equation representing the suspension forces balance can be represented by:

$$\begin{aligned}
F_{ms} &= -F_{ksf} - F_{bsf} - F_{ksr} - F_{bsr} - F_f - F_r \\
F_\theta &= l_f\cos(\theta)(F_{ksf} + F_{bsf} + F_f) - l_r\cos(\theta)(F_{ksr} + F_{bsr} + F_r) \\
F_{muf} &= F_{ksf} + F_{bsf} - F_{ktf} + F_f \\
F_{mur} &= F_{ksr} + F_{bsr} - F_{ktr} + F_r
\end{aligned} \quad (5)$$

Substituting Eq. (5) into Eq. (4), it is obtained the state-space system:

$$\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = \dfrac{1}{m_s}\left(-F_{ksf} - F_{bsf} - F_{ksr} - F_{bsr} - F_f - F_r\right) \\
\dot{x}_3 = x_4 \\
\dot{x}_4 = \dfrac{1}{J}\left(l_f\cos(\theta)(F_{ksf} + F_{bsf} + F_f) - l_r\cos(\theta)(F_{ksr} + F_{bsr} + F_r)\right) \\
\dot{x}_5 = x_6 \\
\dot{x}_6 = \dfrac{1}{m_{uf}}\left(F_{ksf} + F_{bsf} - F_{ktf} + F_f\right) \\
\dot{x}_7 = x_8 \\
\dot{x}_8 = \dfrac{1}{m_{ur}}\left(F_{ksr} + F_{bsr} - F_{ktr} + F_r\right)
\end{cases} \quad (6)$$

where: $x_1 = x; x_2 = \dot{x}; x_3 = \theta; x_4 = \dot{\theta}; x_5 = x_{2f}; x_6 = \dot{x}_{2f}; x_7 = x_{2r}; x_8 = \dot{x}_{2r}$.

Tusset *et al*. (2014) used the State Dependence Ricatti Equation (SDRE) control for stabilize the half-car model. The SDRE control is used for the control of systems that present nonlinearities or even chaotic behaviors. However, such control is difficult to experimentally implement, for, to generate the control signal is required to solve, several times, Riccati equation $P(x)$.

$$P(x)A(x)+A^T(x)P(x)-P(x)BR^{-1}B^T P(x)+Q=0 \tag{7}$$

where $P(x)$ is the solution of the Equation of Riccati dependent on the States, used to find the control gain. The inversion, product and scheduling of matrix, in the solution of the matrix Eq. (7), generate an increase of the computational cost and the time required to calculate the control signal. Due to the computational cost and response, the time of the SDRE becomes difficult to experimentally implement it, because for each new input, it is necessary to find the solution of matrix P(x).

The time required to the control acts in the system is important, because if the time for calculating the control is long, the system can lost instability and occurring overturning of the vehicle.

To obtain the control with the control signal calculus and decrease the computation cost, this work proposes to use the artificial neural network (ANN). The Artificial Neural Networks were recurrence neural network called Echo State Neural (ESN). ESN presented nonlinear behavior with a quicker algorithm of training and can be very useful to control nonlinear systems.

After training, the ESN emulates the behavior of SDRE with smaller time required for the calculus of the signal control in relation to the SDRE. The control through the ESN is robustness to the noise. The neural network needs a smaller number of input signal for calculate the control, decreasing the cost of the control and the situation subjected to noise.

2. ARTIFICIAL NEURAL NETWORKS

The main development of this paper is about the building and run an artificial Neural Network, the echo state network, trained by a sample of SDRE controller behavior by determinate input signals. The ANNs are mathematical models inspired in biological Neural Networks that is composed of a great amount of neurons and the connection between them. Each neuron has one output that will have your value determined by its inputs, where the input is (u_n), synaptic weights (W_{kn}^i), activation function (f_s) and outputs (u_l).

For each input (u_n), there is a weight that will multiply it (W_{kn}^i), after multiplied, all the multiplied inputs are summed (x) and then it is passed by an activation function resulting in the output of the neuron (y). The activation function has the objective of "formatting" the output data of the neuron, for example, the output of the neuron can be only 0 or 1, or even -1 to +1 continuously, depending on the neural network architecture used. This can be seen further in Haykin (1994).

The way in which neurons connect between them from the input to the output of the neural network is characterized by neural network architecture. One of the most pervasive and diffused is the MultiLayer Perceptron (MLP). In the case of this network, there is an input layer *u* with *n* inputs, where all these inputs enter in all the neurons in the middle layer. All outputs from the neurons in the middle layer are input from the neurons of the output layer.

The neurons of the output layer define the output *y* of the network. Other intermediate layers may be defined if necessary. An equation that would define an output y for a given input x would be:

$$u_l = f_s\left(\sum_0^K W_{iK}^s\left(f\left(\sum_0^N W_{Kn}^i u_n\right)\right)\right) \tag{8}$$

The weights of the neurons are stored in a matrix $K \times n$, where *n* is the number of inputs and *K* the number of neurons in the middle layer, it is easy to calculate since only the matrix multiplication is used to perform the calculations.

Each different type of neural network architecture is capable to generate an output signal from inputs, however, so that the output of the network is a desired output from an input signal, in addition to the adjustment of how the neurons will be connected in the network, an adjustment of the weights of the network neurons must be performed. This adjustment of network weights is called training, which can be done in several different ways depending on the characteristics of the network, most often through data samples, and there must be a set of inputs associated with a set of desired outputs. The training is usually expressed in an algorithm, and it regulates the weights of the neurons from the data sample (Silva, 2011). For these adaptations of the weights, the NLP network uses a method called error backpropagation algorithm (Rumelhart *et al.*, 1986).

In the neural networks feedforward, there are only connections between neurons in the input to output direction, never in the opposite direction. This characteristic is called *feedforward* differently from the biological neural network where there are feedbacks. These feedback networks are called Recurrent Neural Networks (RNN). RNNs are known for unstable and difficult training neural networks, this is mainly due to feedbacks that generate nonlinear systems, however, in addition, these feedbacks produce temporal dependence, i.e., they are systems that, in a way, store information from the previous inputs, thus generating, what is a very efficient system resolution of complex systems.

2.1 Echo State Network

The Echo State Network was developed by Hebert Jaeger and introduced in Jaeger (2001) and (2008). ESN is a recurrent neural network, where recurrence make it a complex nonlinear system (Lukoševičius *et al.*, 2012). The use of RNN becomes simple with an innovative form of training. It is a very efficient tool to detect the behavior of complex systems. In this work, ESN will detect the behavior of the SDRE controller by controlling a half-car system. With that, the ESN perform the same work of the SDRE controller, however, with less computational resource.

The echo state network is composed of three layers: input, intermediate layer called reservoir computing (RC) and output layer. ESN training occurs only at the output layer, decreasing the time required for training. The training algorithm for ESN is only possible due to the computational reservoir and its properties (Jaeger, 2001).

Figure 1 shows the architecture of the ESN, where the input layer has K inputs, the RC has N neurons interconnected to each other and also connected the output layer with L outputs. There may be connections between the neurons of the output layer and the input layer as Jaeger performed in Jaeger (2008) and from the input layer to the output layer. Connections between the output layer do not characterize an ESN, however a RNN.

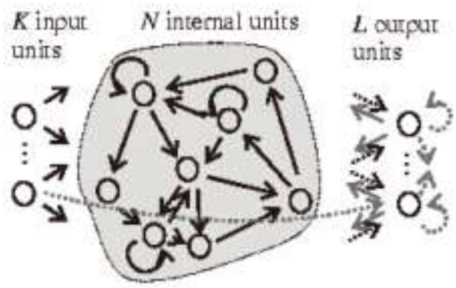

Figure 2. Architecture of the Echo State Network, with three layers: input, intermediate and output. The intermediate layer is reservoir computing.

The weights connections between the N neurons of the RC are represented by a matrix $W \in R^{N \times N}$. This matrix is sparse and randomly generated in the range [0, 1], then each neuron does not connect with all other neurons in the RC and this matrix must have an expectation radius less than 1 and it is a parameter to be regulated, which allows the echo state property, that is, the state of the reservoir must depend only on the input of the data (Lukoševičius *et al.*, 2012).

The *K* inputs must enter the RC through the inputs of the *N* neurons within this, hence each input of each neuron will have a weight to multiply each input. These weights are stored in a matrix $W \in R^{K \times N}$, whose matrix is also randomly generated.

The output values of each neuron of the RC are called *x* state. This state is calculated through the input values plus the feedback values of the RC, this for an ESN without connection of the output layer towards the RC. Mathematically, the state *x* is calculated as:

$$x(t+1) = f\left(Wx(t) + Wu(t+1)\right) \tag{9}$$

where: the term $Wx(t)$ represents the feedback part of the previous state of the reservoir. The term $Wu(t+1)$ represents the new system inputs to the reservoir. The activation function f normally used in the ESN is the hyperbolic tangent, which is required to satisfy the echo state property. At *x*, there will be the status of the current reservoir. This state will be used to calculate the network outputs (Jaeger, 2001).

The output of the network is defined by the value of the L neurons of the output layer, which receives as input the state of the reservoir (*x*). The weights of these links are stored in the matrix $W \in R^{K \times L}$. In this way, the output is defined by the equation:

$$y(t+1) = f\left(W^{out} x(t+1)\right) \tag{10}$$

The only weights that are computed (trained) in the echo state network are the output layer weights that are represented by a W^{out}. This calculation is performed according to a desired **U** and output sample samples **D**. **U** and **D** are related matrices, each row of matrix **U** represents an input **u** that has a relation to the same row of matrix **D**. Rows are sequenced in time, demonstrating the desired temporal behavior.

The inputs are shown to the neural network and thus, the reservoir shows its behavior for these input data. This behavior is a history of states calculated by Eq. (10). With each new entry, the status of the reservoir changes and is characterized by the current entry and the previous entries. This state history must be calculated and stored in a matrix called **M**. In addition to the matrix *M*, to train the ESN, it is needed a matrix *T*, which is the history of all outputs, however before, by the inverse of the function of activation of the output layer:

$$T(t+1) = f^{-1}\left(D(t+1)\right) \tag{11}$$

To calculate the weights Jaeger proposes, among other techniques, to calculate W^{out} using linear regression as follows:

$$W^{out} = M^{\text{inv}}T \tag{12}$$

where: M^{inv} represents the pseudo-inverse matrix of Moore-Penrose of M.

Having the value of the weights in the vector W^{out}, the next step in the neural network is to perform tests normally making the network function as input to the training U samples. The output that the neural network generates is then compared to the training output data D. The difference between the network output and the D values is compared by means of statistical methods, such as the mean square error, and it is possible to calculate the efficiency of the neural network. For a better test it is important to set in the input of the neural network a sequence of data that the network has never seen (set of tests), showing how much the network generalized the problem when receiving the training (Jaeger, 2001).

3. COMPUTATIONAL PROCEDURE

The procedure considered here is described in the flowchart of Figure 2. Initially the half-car system is modelled into state equations, the system is dimensioned, and then an SDRE control is implemented to bring the system to a desired behaviour. Simulating the performed control, the training set is obtained from the SDRE behavior for learning the network. In the next step, an ESN with arbitrary characteristics such as: number of neurons, spectral rays, type of random number generation and degree of sparsity of the dynamics reservoir are generated. Having the neural network formed, it is necessary to train this from the data collected from the half-car behavior and the SDRE control. Once the neural network was trained, tests are performed to capture the behavior of the half-car, using ESN as a control. After the realization of different neural networks, the results are analyzed comparing the efficiency of the neural network in relation to the SDRE control taking into account the control and the computational cost to perform it.

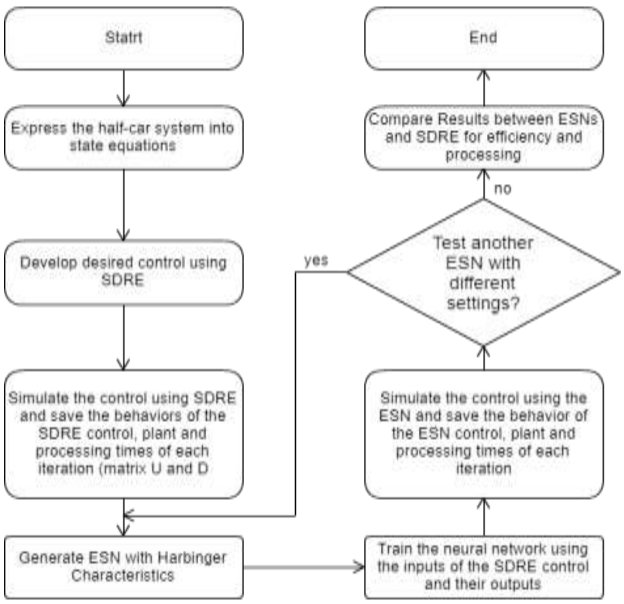

Figure 3. Flowchart computational procedure

3.1 Numerical Simulations

For the numerical simulations, the following parameters will be considered: $m_s = 580[kg]$; $J = 1100[kgm^2]$; $b^l_{sr} = 400[Ns/m]$; $m_{uf} = 40[kg]$; $m_{ur} = 40[kg]$; $l_f = 1[m]$; $l_r = 1.5[m]$; $k^l_{sf} = 150(10^2)[N/m]$; $k^l_{sr} = 150(10^2)[N/m]$; $b^l_{sf} = 400[Ns/m]$; $k^{nl}_{sf} = 150(10^4)[N/m]$; $k^{nl}_{sr} = 150(10^4)[N/m]$; $k_{tf} = 190(10^2)[N/m]$; $k_{tr} = 190(10^2)[N/m]$; $b^y_{sf} = 288[Ns/m]$; $b^y_{sr} = 288[Ns/m]$; $b^{nl}_{sf} = 288[Ns/m]$; $b^{nl}_{sr} = 288[Ns/m]$. The road disturbances are assumed to be sinusoidal: $w_f = \alpha \sin(2\pi\omega(t))$ and $w_r = \alpha \sin(2\pi\omega(t) + \pi)$ and the proposed active controls $F_f = u_f + \tilde{u}_f$; $F_r = u_r + \tilde{u}_r$, where the feedforward control are obtained (Tusset *et al*., 2014):

$$\begin{aligned}
\tilde{u}_f &= \frac{1}{m_s}(k^{nl}_{sf}\Delta_f{}^3 - k^l_{sf} l_f sen(x_3) - b^y_{sf}|\dot{\Delta}_f| + b^{nl}_{sf}\sqrt{|\dot{\Delta}_f|}sign(\dot{\Delta}_f) - b^l_{sf} l_f x_4 \cos(x_3) - k^l_{sf} x_5 - b^l_{sf} x_6) \\
\tilde{u}_r &= \frac{1}{m_s}(k^{nl}_{sr}\Delta_r{}^3 + k^l_{sr} l_r sen(x_3) - b^y_{sr}|\dot{\Delta}_r| + b^{nl}_{sr}\sqrt{|\dot{\Delta}_r|}sign(\dot{\Delta}_r) + b^l_{sr} l_r x_4 \cos(x_3) - k^l_{sr} x_7 - b^l_{sr} x_8)
\end{aligned} \tag{13}$$

and the feedback control are obtained by SDRE control (Tusset *et al*., 2014):

$$\begin{aligned}
u_f &= -R^{-1}B^T P(x) \\
u_r &= -R^{-1}B^T P(x)
\end{aligned} \tag{14}$$

where: $R = 10^{-4}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ and $B = \begin{bmatrix}0 & 0\\ -\frac{1}{ms} & -\frac{1}{ms}\end{bmatrix}$. $P(x)$ is obtained by solving the Riccati equation (7), considering: $A(x) = \begin{bmatrix}0 & 1\\ -\frac{1}{ms}\left(k^l_{sf} + k^l_{sr}\right) + x_3^3 x_2 & -\frac{1}{ms}\left(b^l_{sf} + b^l_{sr}\right) - x_3^3 x_1\end{bmatrix}$ and $Q = 10^3\begin{bmatrix}500 & 0\\ 0 & 1\end{bmatrix}$.

The behavior of the system with control proposed by Tusset *et al*. (2014), can be seen in Fig. 3. The displacement of chassis x_1 and front wheel x_5 are in the Figs. 4a and 4b. In Figs. 4c and 4d are x_3 angle pitch and x_7 is the displacement of rear wheel. To reach this behaviour, the active controller acted in form expressed in Figs. 4e and 4f.

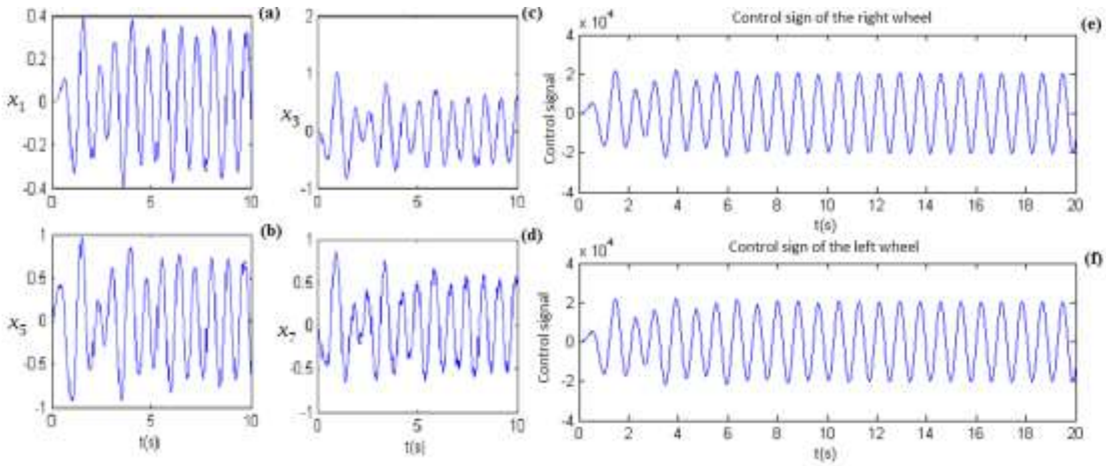

Figure 4. In figure (a), (b), (c) and (d) behavior half-car system when controlled with its active control. (e) and (f) signal to generate by right and left wheel.

3.2 Active Control considering Echo State Network (ESN)

To generate the training data from the neural network then the simulation of the active controller is generated and stored in the U (INPUTS) and D (OUTPUTS) arrays. The matrix U is expressed by (x_2, x_4, x_6 and x_8) and the matrix D by (The control forces are F_f in front and F_r in rear) Figs. 4e and 4f.

Figure 5 shows the performance of the control with the neural network with the training samples comparing the output of the network with the output of the controller for a network of 120 neurons in the reservoir, W-matrix with spectral radius 0.4 and 5% sparsity, using a sequence of 40000 data for the matrices of training.

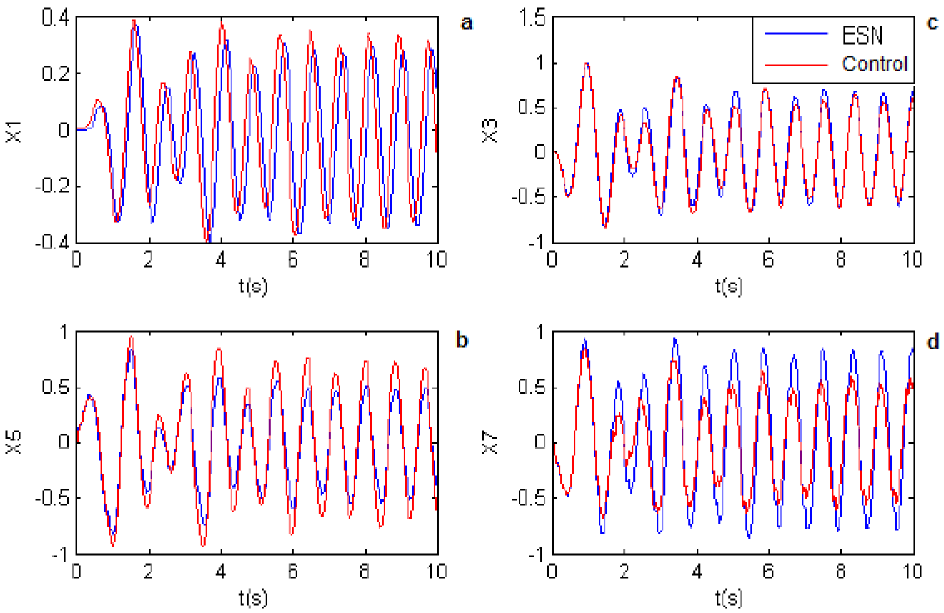

Figure 5. Comparison of the behavior of the half-car system controlled by the ESN controller and the active controller

The control by ESN presented a satisfactory result for control of the displacement of chassis x_1 and front wheel x_5, represented in Figs. 5a and 5b, respectively, and in Figs. 5c and 5d illustrate x_3, angle pitch and x_7, displacement of rear wheel. To compare the computational efficiency of the two controllers, 20 seconds of simulation are considered, 20,000 “spend time” data to calculate the control signal.

In Fig. 6 can be seen those results, where the graph shows the time required for the calculation of each control signal using the SDRE and ESN controller, also has 2 green lines representing the mean of those control signal calculation time. The average time of the ESN is six times smaller than the average time of the SDRE for short time interval.

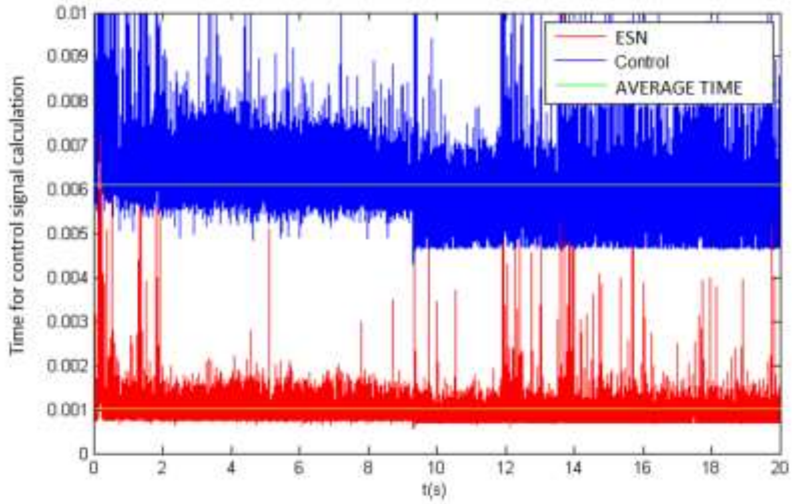

Figure 6. Times to calculate the control signal for ESN 120 neurons and active control

To verify the distribution of time and errors in relation to the average, a histogram was used, where it can be seen in Fig. 7 that the ESN-based controller used much less processing time, with three peak values in approximately 0.7, 0.9 and 1.1 milliseconds. SDRE consumed between 5 and 6 milliseconds, with a peak in approximately 5.5. The variation is lager in the SDRE control, where the times are 5 and 8 milliseconds. For ESN control the variation of time stayed between 0.7 e 1.3 milliseconds.

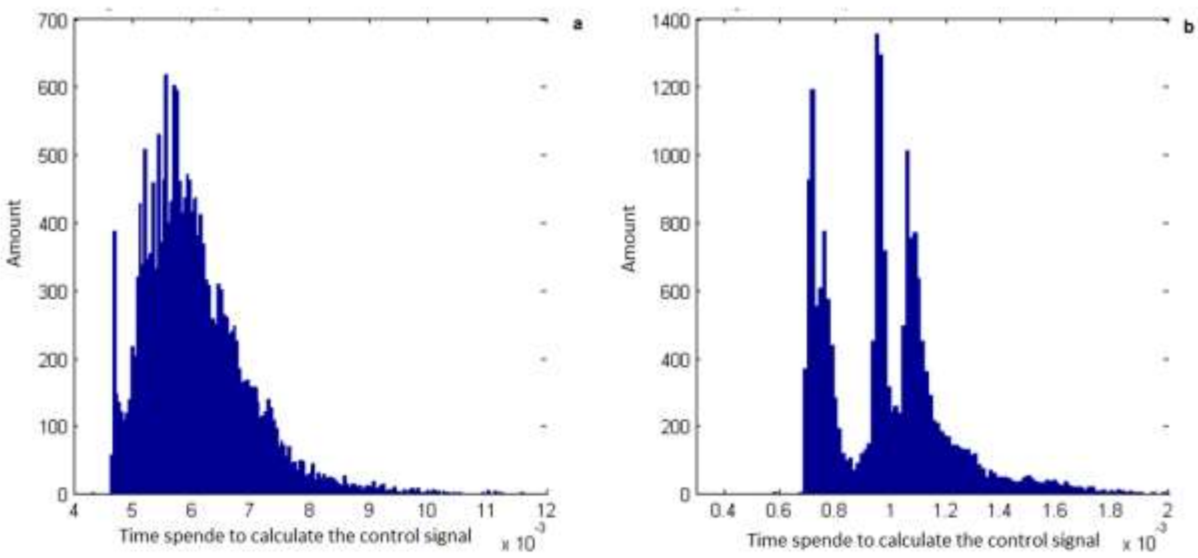

Figure 7. Histogram of the times for the calculation of the control signal. (a) active control. (b) ESN control

Table 1 presents the results of the corporation for controls, considering the RMS, and the maximum absolute for the four states x_1, x_3, x_5 and x_7, considering different amounts of neurons.

Table 1. Comparison of the behavior of the half-car system controlled by the active control and different ESNs with different amounts of neurons in the dynamic reservoir.

	Active Control	ESN 60	ESN 120	ESN 200	ESN 500
RMS (x_1)	0.2348	0.2185	0.2183	0.2151	0.2259
max($\|x_1\|$)	0.3961	0.4168	0.3993	0.4106	0.422
RMS (x_3)	0.4226	0.4535	0.4574	0.4592	0.4572
max($\|x_3\|$)	1.0031	1.0684	1.0044	1.054	0.9949
RMS (x_5)	0.5097	0.4157	0.3687	0.3781	0.3653
max($\|x_5\|$)	0.9575	0.9298	0.8342	0.8851	0.8163
RMS (x_7)	0.3964	0.5096	0.7759	0.5499	0.5712
max($\|x_5\|$)	0.8615	0.9292	0.9362	0.9825	0.9769
Average time to calculate control signal	0.0061	9.10E-04	0.001	0.0015	0.0064
Standard deviation of time to calculate control signal	0.0011	4.53E-04	6.99E-04	8.25E-04	5.57E-04

3.3 Robustness analysis for active control considering Echo State Network (ESN)

In order to test the robustness of the ESN controller, disturbances were induced in the reading of the input data, as if the sensors of the system were not having a good precision. The controller still controls the system through an efficiently way even with 2 and 5 percent disturbances.

This behavior can be seen in Fig. 8 and Tab. 2.

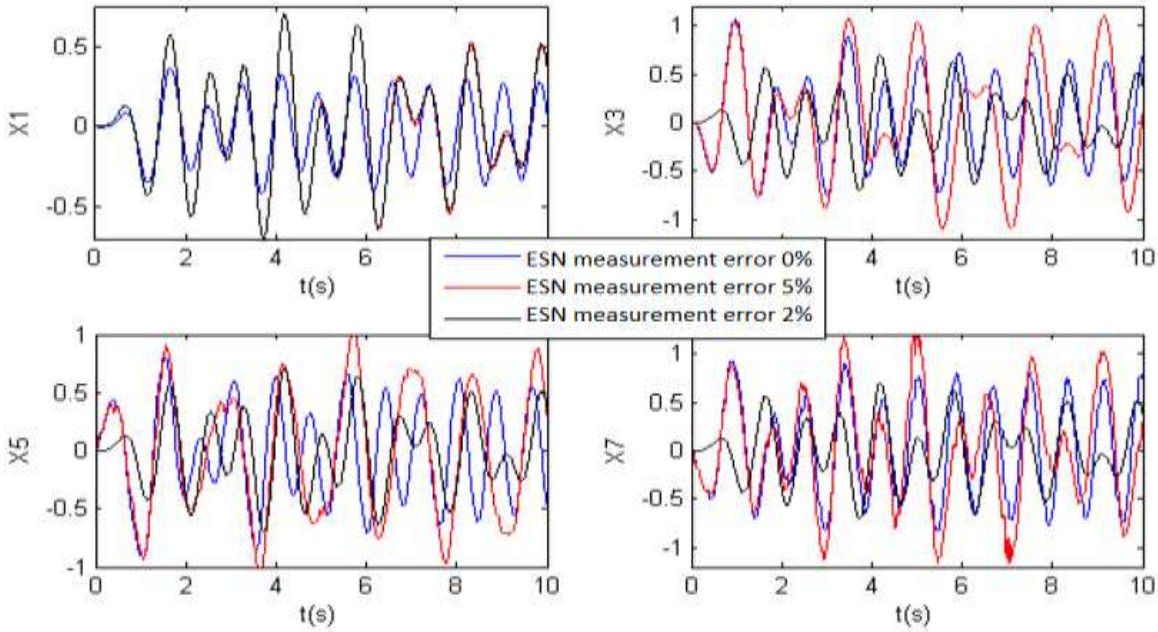

Figure 8. Comparison of the behavior of the ESN controlled half car system with different input signal disturbances

In table 2 is presented the results of the corporation for controls, considering the RMS, and the maximum absolute for the four states x_1, x_3, x_5 and x_7, considering disturbances in the control.

Table 2. Comparison of the behavior of the ESN controlled half car system with different input signal disturbances

	ESN	ESN 2%	ESN 5%
RMS (x_1)	0.219	0.322	0.323
max($\|x_1\|$)	0.417	0.706	0.706
RMS (x_3)	0.454	0.572	0.574
max($\|x_3\|$)	10.684	11.300	10.982
RMS (x_5)	0.416	27.599	0.540
max($\|x_5\|$)	0.930	10.805	10.785
RMS (x_7)	0.510	0.566	0.567
max($\|x_5\|$)	0.929	14.444	14.308

4. CONCLUSION

The results of this work showed that the ESN neural network can do the same as the SDRE-based controller, with greater efficiency by spending less time on half-car control. The neural network requires fewer inputs for the calculation of the control signal, facilitating the experimental implementation of the control for the half-car system. The use of ESN and the SDRE controllers together opens a new possibility for experimental implementation of the SDRE controller and a new application for ESN.

5. REFERENCES

Lukoševičius, M., Jaeger, H. and Schrauwen, B., 2012. “Reservoir Computing Trends”. *KI - Künstliche Intelligenz*, Vol. 26, p. 365–371.

Tusset, A. M., Piccirillo, V., Janzen, F. C., Lenz, W., Lima, J. J., Balthazar, J. M., Brasil, M. R. L. F., 2014. “Suppression of vibrations in a nonlinear half-car model using a magneto-rheological damper”. *Mathematics in Engineering, Science and Aerospace: the transdisciplinary international journal*, Vol. 5, p. 427-443.

Haykin, S., 1994. *Neural Networks - A Comprehensive Foundation*. Macmillan College Publishing Company.

Silva, J. V., 2011. *Fundamentos das Redes Neurais*. Livro Rápido.

Rumelhart, D., Hinton, G. and Williams, R., 1986. *Learning representations by backpropagation errors*. Nature.

Jaeger, H., 2001. *The echo state approach to analysing and training recurrent neural networks*. Fraunhofer Institute for Autonomous Intelligent Systems.
Jaeger, H., 2008. *A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the echo state network approach*. Fraunhofer Institute for Autonomous Intelligent Systems (AIS).

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.