

7-25-2012

# Self-organizing input space for control of structures

Simon Laflamme

*Iowa State University*, laflamme@iastate.edu

J.J.E. Slotine

*Massachusetts Institute of Technology*

J.J. Connor

*Massachusetts Institute of Technology*

Follow this and additional works at: [http://lib.dr.iastate.edu/ccee\\_pubs](http://lib.dr.iastate.edu/ccee_pubs)

 Part of the [Civil Engineering Commons](#), [Construction Engineering and Management Commons](#), [Environmental Engineering Commons](#), and the [Structural Engineering Commons](#)

The complete bibliographic information for this item can be found at [http://lib.dr.iastate.edu/ccee\\_pubs/76](http://lib.dr.iastate.edu/ccee_pubs/76). For information on how to cite this item, please visit <http://lib.dr.iastate.edu/howtocite.html>.

---

This Article is brought to you for free and open access by the Civil, Construction and Environmental Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Civil, Construction and Environmental Engineering Publications by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

---

# Self-organizing input space for control of structures

## Abstract

We propose a novel type of neural networks for structural control, which comprises an adaptive input space. This feature is purposefully designed for sequential input selection during adaptive identification and control of nonlinear systems, which allows the input space to be organized dynamically, while the excitation is occurring. The neural network has the main advantages of (1) automating the input selection process for time series that are not known *a priori*; (2) adapting the representation to nonstationarities; and (3) using limited observations. The algorithm designed for the adaptive input space assumes local quasi-stationarity of the time series, and embeds local maps sequentially in a delay vector using the embedding theorem. The input space of the representation, which in our case is a wavelet neural network, is subsequently updated. We demonstrate that the neural net has the potential to significantly improve convergence of a black-box model in adaptive tracking of a nonlinear system. Its performance is further assessed in a full-scale simulation of an existing civil structure subjected to nonstationary excitations (wind and earthquakes), and shows the superiority of the proposed method.

## Disciplines

Civil Engineering | Construction Engineering and Management | Environmental Engineering | Structural Engineering

## Comments

This is a manuscript from an article from Smart Materials and Structures, 21(115015)2012; 1-16. Doi: [10.1088/0964-1726/21/11/115015](https://doi.org/10.1088/0964-1726/21/11/115015). Posted with permission.

# Self-Organizing Input Space for Control of Structures

S. Laflamme<sup>1</sup>, J.-J.E. Slotine<sup>2</sup> and J.J. Connor<sup>3</sup>

<sup>1</sup>Dept. of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, IA, 50011

<sup>2</sup>Nonlinear Systems Laboratory, Dept. of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139

<sup>3</sup>Dept. of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139

E-mail: laflamme@iastate.edu

**Abstract.** We propose a novel type of neural networks for structural control, which comprises an adaptive input space. This feature is purposefully designed for sequential input selection during adaptive identification and control of nonlinear systems, which allows the input space to be organized dynamically, while the excitation is occurring. The neural network has the main advantages of 1) automating the input selection process for time series that are not known a priori; 2) adapting the representation to nonstationarities; and 3) using limited observations. The algorithm designed for the adaptive input space assumes local quasi-stationarity of the time series, and embeds local maps sequentially in a delay vector using the embedding theorem. The input space of the representation, which in our case is a wavelet neural network, is subsequently updated. We demonstrate that the neural net has the potential to significantly improve convergence of a black-box model in adaptive tracking of a nonlinear system. Its performance is further assessed in a full-scale simulation of an existing civil structure subjected to nonstationary excitations (wind and earthquakes), and shows the superiority of the proposed method.

PACS numbers: 05.45.-a, 45.80.+r, 46.40.-f, 46.40.Ff, 84.35.+i, 87.19.lr

Submitted to: *SMS*

## 1. Introduction

An effective mitigation strategy for structural systems subjected to natural (e.g., wind and earthquake loads) and manmade (e.g., blast) hazards is to include semi-active and active structural control mechanisms. Despite the advancements of this technology, broad implementation of those control mechanisms for effective structural hazard mitigation is rare [1]. This is due to a lack of acceptability, impeded by the lack of applicability of control schemes [2, 3, 4]. Realistically, control solutions applied to civil structures need to rely on low power actuation and robust controllers [5, 6, 7]. To address the control problem, several semi-active control devices capable of large damping forces using limited power have been proposed in the literature [8, 9, 10]. On the controller perspective, the field of civil engineering provides numerous challenges which are inherent to the size of the controlled plant [11, 12, 13]: 1) the dynamics of civil structures contains large parametric and non-parametric uncertainties; 2) controllers only have access to limited measurements; 3) testing and training of controllers is difficult due to the unavailability of input-output data sets; and 4) controllers have an immediate performance requirement upon the occurrence of natural hazards. The structural control problem is analogous to a child learning his or her motor functions, trying different balances until stability is attained. Given these control challenges, a solution is to implement sequential adaptive controllers. These controllers have the advantage of adapting to parametric and non-parametric uncertainties, and can be trained while an excitation is occurring, without pre-training. The issue of limited measurements and immediate performance requirements can be addressed in the controller design.

Numerous control solutions have been proposed to control uncertain systems, including robust controllers [14, 15, 9, 16], fuzzy logic [17, 18, 19], and neural networks [20, 21, 22]. Among these solutions, neural networks have notably gained significant popularity due to their universal approximation capability [23, 24], and are promising self-contained solutions for sequential adaptive control. There exist several applications of sequential adaptive neurocontrollers to the field of structural control. For instance, Zhou *et al.* [25] used adaptive fuzzy control for a nonlinear base isolation system equipped with a magnetorheological (MR) damper. Lee *et al.* [26] developed a semi-active neurocontroller for a base-isolation system controlled with an MR damper, where the neural network was updated using a cost function. Lee *et al.* [27] described an adaptive modal neurocontroller for a structure equipped with an MR damper. Suresh *et al.* [28] proposed an adaptive mapping scheme that uses Gaussian radial functions to control base-isolation of nonlinear buildings equipped with an actuator. Laflamme & Connor [29] presented an inverse neurocontroller, which nodes sequentially adapt to achieve optimal semi-active control, using a sliding controller and adaptive learning rates. A modified version of the controller [30] includes an enhanced robustness in the adaptation laws and uses wavelets instead of Gaussian radial functions for a better functional

localization.

Often, the adaptive controller representations discussed in the literature assume an hyperspace in which the representation is defined, overlooking the input space selection procedure. Input selection for intelligent controllers is a fundamental design task which may influence computation time, adaptation speed, effects of the curse of dimensionality, understanding of the representation, and model complexity [31, 32, 33, 34]. Some methods have been proposed for input selection, including the filter methods, where the input selection is independent of the black-box model [35], the wrapper methods, where the results from the black-box model are used to rank and select the inputs [36], and the embedding methods, where selected inputs are used for adapting the representation [37]. There exist applications of these methods to neural networks with the specific goal of automating the process of input selection [38, 39, 34, 40, 41]. However, those methods are traditionally applied offline, necessitating pre-training of control algorithms to evaluate the performance of each input space at attaining a given performance objective.

The embedding method is the foremost applicable input selection technique to sequential adaptive control. Several popular algorithms for embedding dynamic systems have emerged from the celebrated Takens embedding theorem [42]. The theorem states that the phase-space of an autonomous system can be reconstructed topologically using a vector formed with a number of delayed measurements from a single state. In other words, there exists a set of inputs from limited observations that can represent the system dynamics. The embedding theorem has been extended to a general class of nonautonomous systems with deterministic forcing [43], state-dependent forcing [44], and stochastic forcing [45]. Applications of the embedding theorem to the field of structural engineering are limited; they appear to be unique to structural health monitoring [46, 47, 48].

Nevertheless, the delay embedding theorem has been applied in many fields for model prediction, system identification, and control. More specifically, Cao *et al.* [49] introduced wavelet neural networks (WNN) with embedded inputs for chaotic time series prediction. Principe *et al.* [50] used local nonlinear embedding maps with a self-organizing mapping neural network, with application to system identification and control. Plagianakos & Tzanaki [51] used a neural network to predict an earthquake excitation selecting inputs based on the embedding theorem. Walker *et al.* [52] utilized the same strategy to design a radial-basis model for modeling of an electronic circuit with dynamic effects. Zolock & Greif [53] used a neural network a delay vector embedding for inputs to predict wheel/rail responses of rail vehicles. Remark that all of those applications used the embedding theorem with prior training, by pre-processing the underlying time series.

This paper proposes a sequential adaptive solution for selecting the input space. This creates a novel type of neural network for identification and control of unknown systems for which pre-training is not available. The adaptive neural network is a single-layer wavelet neural network (WNN), using a self-organizing mapping architecture for its hidden layer [54].

Similar to the hidden layer, the input layer also has the capacity to add and prune nodes. The proposed algorithm that selects the input space is termed Self-Organizing Inputs (SOI) algorithm, and is based on Takens embedding theorem. The SOI algorithm determines, at each time step, the input variables that capture the essential dynamics of the system, and uses the information to smoothly adapt the input space. As it will be demonstrated, the proposed SOI-WNN is promising at identification and control of unknown systems, and has the substantial benefits of 1) automating the input selection process for time series that are not known a priori; 2) adapting the representation to nonstationarities; and 3) using limited observations. Examples of applications include uncertain systems evolving in unknown environments, such as civil structures and control of wind turbine blades. Here, the SOI-WNN is applied to semi-active control of a civil structure subjected to unknown and nonstationary excitations (wind and earthquakes).

The paper is organized as follows. Section 2 describes the general controller architecture, with an emphasis on the WNN that will be utilized with the SOI algorithm. Section 3 presents the novel SOI algorithm, where we discuss the algorithm used for the sequential selection and organization of the input space, along with its implications on the hidden layer. Section 4 derives the adaptation rules and shows stability of the control rule. Section 5 discusses selection of the non-adaptive parameters for the proposed neurocontroller to guide future applications. Section 6 verifies the SOI-WNN behavior on tracking of a low-dimensional synthetic example. Section 7 simulates the SOI-WNN on a full-scale structure equipped with semi-active dampers, and subjected to wind and earthquake excitations. Section 8 discusses the results and their implications, and concludes the paper.

## 2. Controller Architecture

The controller representation is constructed using a single-layer feedforward WNN, selected due to its quick and universal approximation capability [24, 30]. The WNN is used with an automatic organization of the input space executed by the SOI algorithm, which constitutes the proposed SOI-WNN controller. The general architecture of the resulting SOI-WNN is described in this section, derived for nonlinear and nonautonomous controlled system of the type:

$$\begin{aligned}
 x(k+1) &= f_x(x(k), u(k), k) \\
 u(k+1) &= f_u(y(k), u(k), k) \\
 y(k) &= f_y(x(k), u(k), k)
 \end{aligned} \tag{1}$$

where  $x$  denotes the state,  $u$  the input,  $y$  the observation,  $f$  the nonlinear functions, and  $k$  the discrete time steps.

Figure 1 shows a representation of the controller for specific applications to semi-active structural control. The structure is subjected to an external forcing (excitation) and forces

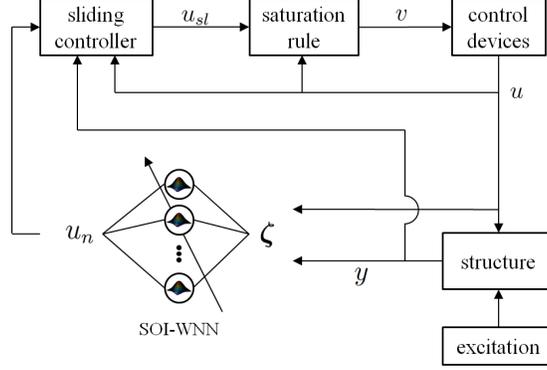


Figure 1: Block diagram of the closed-loop control system.

from the control devices. The dynamic states  $\mathbf{y}$  and the control forces  $\mathbf{u}$  are fed in the adaptive SOI-WNN under the input vector  $\boldsymbol{\zeta}$  to obtain the neurocontrol force  $u_n$ . A sliding controller modifies  $u_n$  to account for the large region of force unreachability arising from the semi-active device, and the resulting force  $u_{sl}$  is used to determine the voltage  $v$  in the control device, which governs the final control force  $u$ .

The single-layer feedforward WNN is composed of *mexican hat* wavelets  $\phi$ :

$$\phi(\boldsymbol{\zeta}) = \left(1 - \frac{\|\boldsymbol{\zeta} - \boldsymbol{\mu}\|^2}{\sigma^2}\right) \exp^{-\frac{\|\boldsymbol{\zeta} - \boldsymbol{\mu}\|^2}{\sigma^2}} \quad (2)$$

where  $\boldsymbol{\zeta}$  is the input vector,  $\boldsymbol{\mu}$  and  $\sigma$  are the centers and bandwidths of the functions respectively, and  $\|\cdot\|^2$  is the 2-norm. Figure 2a illustrates the single-layer WNN, where the input space is fed in  $h$  wavelets  $\phi$ , and the  $i^{\text{th}}$  neurocontroller output  $u_{n,i}$  is constructed from the summation of the weighted nodal magnitudes:

$$u_{n,i} = \sum_j^h \gamma_{j,i} \phi_j(\boldsymbol{\zeta}) \quad (3)$$

Figure 2b shows the representation of the two-dimensional mexican hat wavelet:

$$y(x, \dot{x}) = 2 \left[1 - \left(\frac{(x_1 - 0.1)^2}{0.1^2} + \frac{(x_2 - 0.05)^2}{0.05^2}\right)\right] \exp^{-\left(\frac{(x_1 - 0.1)^2}{0.1^2} + \frac{(x_2 - 0.05)^2}{0.05^2}\right)} \quad (4)$$

with center  $\boldsymbol{\mu} = [0.1, 0.05]$ , bandwidth  $\boldsymbol{\sigma} = [0.1, 0.05]$ , and weight  $\gamma = 2$ . Note that the role of the SOI algorithm (described in the next section) is to defines the size of the input vector  $\boldsymbol{\zeta}$  along with its content.

The neural output in (3) is taken as optimal. In an adaptive mechanism, the neuro-output is an estimated force  $\hat{\mathbf{u}}_n$ :

$$\hat{u}_{n,i} = \sum_j^h \hat{\gamma}_{j,i} \hat{\phi}_j(\boldsymbol{\zeta}) \quad (5)$$

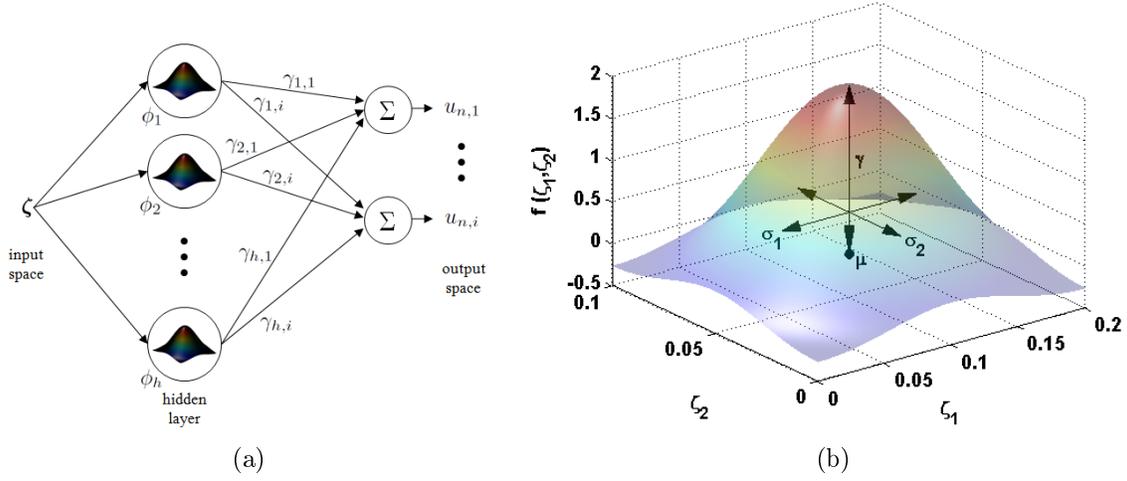


Figure 2: a) Wavelet neural network; and b) *mexican hat* wavelet with weight  $\gamma$ , center  $\mu$ , and bandwidths  $\sigma_1, \sigma_2$ , in function of the input space  $\zeta = \zeta_1, \zeta_2$ .

where the hat denotes an estimation. Also, in semi-active control, the control force (5) is further modified by a sliding controller to account for the large force unreachability of the control devices. Consider the following force adaptation regions for the sliding controller:

$$\begin{aligned}
 \mathcal{C}_d &= \{|\tilde{u}_i| \leq \alpha_d u_b \mid \alpha_d \in [0, 1], u_b, \tilde{u} \in \mathbb{R}\} \\
 \mathcal{C}_t &= \{|\tilde{u}_i| \leq \alpha_t u_b \mid \alpha_t \in [0, 1], u_b, \tilde{u} \in \mathbb{R}\} \\
 \mathcal{C} &= \{|\tilde{u}_i| \leq u_b \mid u_b, \tilde{u} \in \mathbb{R}\}
 \end{aligned} \tag{6}$$

where  $\mathcal{C}$  is the bounded set of neuro-outputs,  $\mathcal{C}_d$  is confined within the device reachability,  $\mathcal{C}_t$ , is the transition region between  $\mathcal{C}_d$  and  $\mathcal{C}$ , such that  $\mathcal{C} \supset \mathcal{C}_t \supset \mathcal{C}_d$ ,  $u_b$  is a bound on the admissible error  $\tilde{u}_i$ :

$$\tilde{u}_i = u_{sl,i} - u \tag{7}$$

and  $\alpha_d, \alpha_t$  are used-defined constants with  $\alpha_d \leq \alpha_t$ . The control law (its stability is shown in Section 4) is modified with respect to the adaptation region:

$$u_{sl,i} = (1 - m_{b,i})(u_{n,i} - \beta \text{sat}(s_i/\Psi)) - m_{b,i} u_{\max} \text{sat}(s_i/\Psi) \tag{8}$$

where the term  $m_{b,i} u_{\max} \text{sat}(s_i/\Psi)$  is the sliding component used to bring the system back inside  $\mathcal{C}_d$ ,  $u_{\max}$  is the maximum force output,  $\text{sat}$  is the saturation function with the scaling parameter  $\Psi$  for the sliding surface  $s_i = \mathbf{P}\mathbf{X}$  representing a weighted error of the regulatory

control problem,  $\mathbf{P}$  is a user-defined vector to be discussed later,  $\mathbf{X}$  is the state vector,  $\beta$  is a positive constant, and  $m_{b,i}$  takes the value:

$$\begin{aligned} m_{b,i} &= 0 && \text{if } \tilde{u} \in \mathcal{C}_d \\ m_{b,i} &= 1/(1 + \exp^{-b_1(|\tilde{u}| - (\alpha_i + \alpha_d)u_b/2)}) && \text{if } \tilde{u} \in \mathcal{C}_t - \mathcal{C}_d \\ m_{b,i} &= 1 && \text{if } \tilde{u} \in \mathcal{C} - \mathcal{C}_t \end{aligned} \quad (9)$$

where  $b_1$  is a positive constant, and  $m_{b,i}$  for  $\tilde{u} \in \mathcal{C}_t - \mathcal{C}_d$  is a sigmoid function. A voltage  $v_i$  is selected to reach  $u_{sl,i}$ :

$$\begin{aligned} v_i &= V_{\max} && \text{if } |u_{sl,i}| > |u_i| \text{ and } \text{sgn}(u_{sl,i}) = \text{sgn}(u_i) \\ v_i &= 0 && \text{otherwise} \end{aligned} \quad (10)$$

where  $V_{\max}$  is the maximum voltage input, and  $\text{sgn}$  is the signum (or sign) function:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (11)$$

The SOI-WNN is designed to be self-organizing and self-adaptive. Self-organizing refers to the capacity of the SOI-WNN to organize its internal architecture. Self-adapting refers the capacity of the SOI-WNN to adapt internal parameters, such as nodal weights, centers, and bandwidths. The self-organizing and self-adapting features of the SOI-WNN are described in the next sections.

### 3. Self-Organizing Feature

The SOI-WNN is designed to self-organize both its input space and its hidden layer. The self-organization of the input space is executed by the SOI algorithm, which is the central feature of this paper. The SOI algorithm is thoroughly presented in the upcoming subsection. The hidden layer is organized following Kohonen self-organizing mapping (SOM) theory [54], and is summarized in the subsequent subsection.

#### 3.1. Input Space

The proposed SOI algorithm has been designed for sequentially organizing the input space of the WNN. The algorithm parameterizes the unknown dynamic system using the time series response of a single observation based on Takens embedding theorem. Subsequently, the dimension of the input space is adapted smoothly, along with the time lag between observations, using the assumption that the new inputs represent the essential dynamics of the unknown system. The objective is to obtain a more efficient representation for the dynamic system by selecting, at each time step, the inputs that contain a sufficient representation of the current system states.

Takens embedding theorem is applicable to reconstruct the state-space of an autonomous dynamic system. The theorem states that the phase-space of the dynamic system in a topological space  $\mathcal{M}$  can be reconstructed from a vector  $\boldsymbol{\nu}$ , termed delay vector, of dimension  $d$ , built from the observations  $y(k) = f_y(x(k))$  delayed by a time factor  $\tau$ :

$$\begin{aligned}\boldsymbol{\nu}(k) &= [y(k) \ y(k - \tau) \ y(k - 2\tau) \ \dots \ y(k - (d - 1)\tau)] \\ &= \Phi(x(k))\end{aligned}\tag{12}$$

where  $\Phi : \mathcal{M} \rightarrow \mathbb{R}^d$ ,  $\tau = a\Delta t$ , with  $\Delta t$  being the sampling rate and  $a$  is a positive integer. Figure 3 illustrates the principle. Time series measurements (figure 3b) of a single state are taken from an unknown system (figure 3a). Those measurements are used to reconstruct a topologically equivalent (diffeomorphic) representation (figure 3c) using  $\boldsymbol{\nu}$ . In other words, a one-to-one map exists between the reconstructed and the unknown systems.

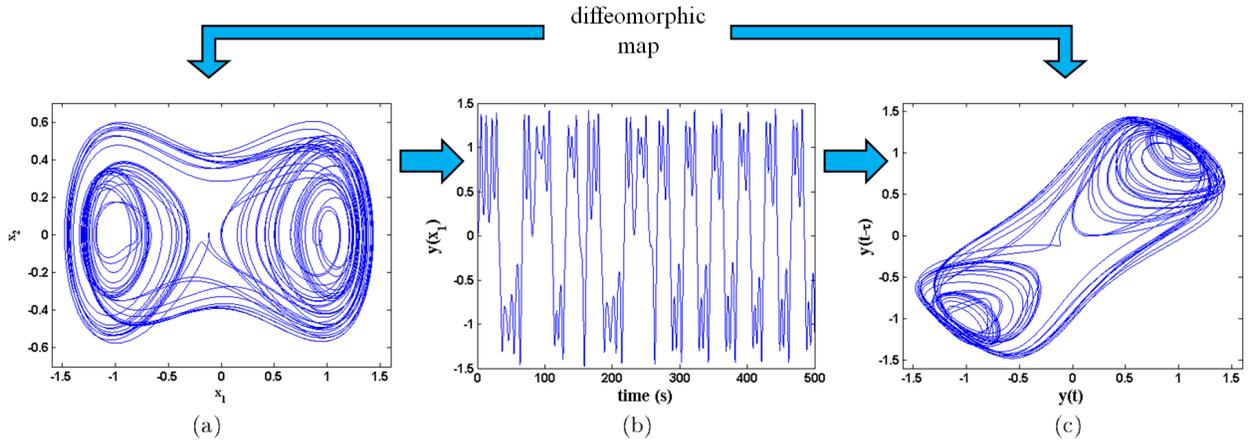


Figure 3: Illustration of Takens embedding theorem. a) Phase-space of the unknown system to be reconstructed (here a 2-dimensional representation of a Duffing system  $y = h(x_1, x_2, x_3)$ ); b) time series measurements  $\mathbf{y}$  are taken from a single state ( $y(x_1) = x_1$  of the unknown system); and c) the unknown system phase-space can be diffeomorphically reconstructed, here in a 2-dimensional phase-space constructed from  $\mathbf{y}$  with a time delay  $\tau$ .

Consider an autonomous form of  $f_x$  in (1):  $x(k + 1) = f_x(x)$ . The function can be represented by  $\tilde{f}_x$ :

$$\tilde{f}_x = \Phi \circ f_x \circ \Phi^{-1}\tag{13}$$

where  $\circ$  denotes function composition. Applying (12) to (13):

$$\begin{aligned}
\tilde{f}_x(\boldsymbol{\nu}(k)) &= \Phi \circ f_x \circ \Phi^{-1}(\boldsymbol{\nu}(k)) \\
&= \Phi \circ f_x \circ \Phi^{-1}(\Phi(x(k))) \\
&= \Phi \circ f_x(x(k)) \\
&= \Phi(x(k+1)) \\
&= \boldsymbol{\nu}(k+1)
\end{aligned} \tag{14}$$

demonstrates that the future delay vector  $\boldsymbol{\nu}(k+1)$  can be predicted using the current delay vector  $\boldsymbol{\nu}(k)$ , or in other words, the future observation  $y(k+1)$  of the state  $x$  can be predicted from a topologically equivalent phase-space constructed with  $\Phi(x(k+1))$ , given that  $\boldsymbol{\nu}$  is formed with an appropriate time delay  $\tau$  and embedding dimension  $d$ . The choice for  $\tau$  and  $d$  will be discussed later. Remark that the algorithm is not concerned about predicting observations; (14) is used to show that a delay vector can contain enough information to predict a dynamic system, or to represent the essential dynamics. Thus, the algorithm takes the delay vector  $\boldsymbol{\nu}$  that would parameterize the reconstructed phase-space, and sequentially adapts the WNN input space  $\zeta(\tau_\zeta, d_\zeta)$ , smoothly, to reach the required delay vector  $\boldsymbol{\nu}(k)$ , using:

$$\begin{aligned}
\tau_\zeta(k+1) &= \tau_\zeta(k) + \text{sgn}(\tau(k+1) - \tau(k)) \\
d_\zeta(k+1) &= d_\zeta(k) + \text{sgn}(d(k+1) - d(k))
\end{aligned} \tag{15}$$

where subscript  $\zeta$  indicates that the parameters are used for populating the input vector  $\zeta$ . Additionally, when  $d_\zeta$  is modified ( $d_\zeta(k+1) \neq d_\zeta(k)$ ), the modified wavelet bandwidths  $\boldsymbol{\sigma}_j$  are adapted smoothly using:

$$\boldsymbol{\sigma}_{\text{mod}} - (\boldsymbol{\sigma}_{\text{mod}} - \boldsymbol{\sigma}_j)m_{c,j}(k) \tag{16}$$

where  $\boldsymbol{\sigma}_{\text{mod}}$  is a vector of large constants,  $\boldsymbol{\sigma}_j$  is the target bandwidths when a dimension is added or the bandwidths to be removed when a dimension is decreased, and  $m_{c,j}(k)$  takes the values:

$$\begin{aligned}
m_{c,j}(k) &= \frac{1}{1 + \exp^{-c_1(k-k_{\text{mod}}-c_2)}} && \text{if } d_\zeta \text{ is increased} \\
m_{c,j}(k) &= \frac{\exp^{-c_1(k-k_{\text{mod}}-c_2)}}{1 + \exp^{-c_1(k-k_{\text{mod}}-c_2)}} && \text{if } d_\zeta \text{ is decreased}
\end{aligned} \tag{17}$$

where  $c_1, c_2$  are positive constants, and  $k_{\text{mod}}$  is the time step at which  $d_\zeta$  has been modified. A dimension is removed from the representation once the bandwidths fall beyond a threshold. Remark that (15) restrains the changes in the input space parameters to unity, which ensures robustness of the representation, as  $y(k-\tau) \simeq y(k-\tau \pm 1)$  and new dimensions are added smoothly using (17).

Despite that the theory discussed above applies to autonomous systems, the embedding theorem has been extended to non-autonomous stationary systems, where it can be shown that the delay vector needs to also include the system inputs  $u$  [45]:

$$\begin{aligned} \boldsymbol{\nu}(k) &= [y(k) \ y(k - \tau) \ y(k - 2\tau) \ \dots \ y(k - (d - 1)\tau) \ u(k) \ u(k - \tau) \ u(k - 2\tau) \\ &\quad \dots \ u(k - (d - 1)\tau)] \\ &= \Phi(x(k), u(k)) \end{aligned} \tag{18}$$

where  $\Phi : \mathcal{M} \rightarrow \mathbb{R}^{2d}$ .

Here, the dynamic system of interest (1) is nonstationary due to the adaptive control rule and the nature of the excitation. To cope with the problem of nonstationarity, the state dynamics  $f_x$  is taken as a series of maps of dimension  $n$ , where each map is assumed to be quasi-stationary. This assumption of local quasi-stationarity will be verified later. A sliding window of size  $n$  is used, which returns the observations  $\mathbf{y}$  on the local dynamics at step  $k$ :

$$\mathbf{y}(k) = [y(k) \ y(k - 1) \ y(k - 2) \ \dots \ y(k - (n - 1))] \tag{19}$$

It follows that the delay vector is allowed to be nonstationary.

The embedding theorem is applicable given that  $\boldsymbol{\nu}$  is constructed using appropriate values for  $\tau$  and  $d$ . The SOI algorithm uses conventional techniques for the determination of these parameters. The time delay  $\tau$  is computed at each time step using the mutual information (MI) method based on Shannon's information theory [55], and the embedding dimension is selected using the false nearest neighbor (FNN) method using the algorithm presented in [56]. The applications of these methods in discrete time steps are summarized in what follows.

*3.1.1. Mutual Information Test* The objective of the MI test is to find an appropriate time delay  $\tau$  for representing the essential dynamics of a system. Take the 2-dimensional system from figure 3 for instance. If one tries to reconstruct the phase space using  $y(k)$  and  $y(k - \tau)$ , a small value for  $\tau$  will collapse the phase-space to a 45 degree line. Qualitatively,  $\tau$  needs to be augmented to unfold the phase-space until enough information from the dynamics can be extracted. If  $\tau$  is too high, the phase-space would have unfolded too much, and information will be lost. Figure 4 illustrates the principle for the unknown system represented in figure 3. The phase-space unfolds between  $\tau = 0.5$  sec and  $\tau = 1.4$  sec, and starts folding over itself at  $\tau = 2$  sec.

The MI test is a technique based on Shannon's information theory to quantitatively select  $\tau$ . The test measures the average information gained from a new measurement, or how well can the estimation on the outputs  $\hat{\mathbf{y}}(k + \tau)$  be computed given the measurements  $\mathbf{y}(\tau)$ . A high probability of estimating  $\hat{\mathbf{y}}$  from  $\mathbf{y}$  signifies that little new information is

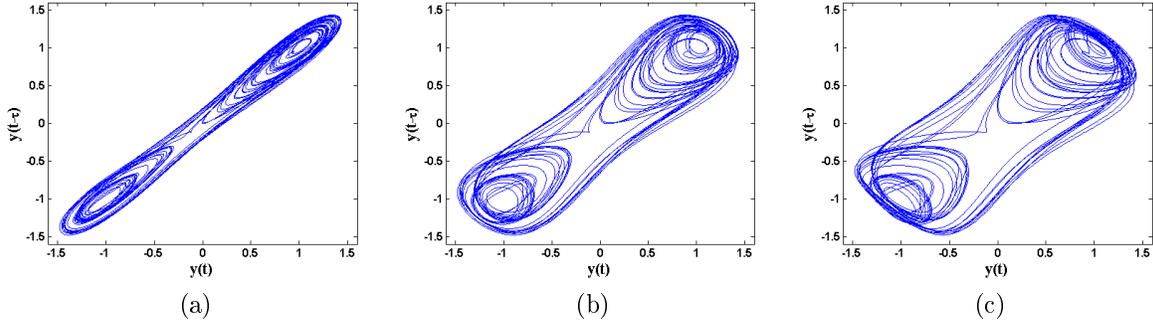


Figure 4: Unfolding the state-space in function of  $\tau$ . a)  $\tau = 0.5$  sec; b)  $\tau = 1.4$  sec; and c)  $\tau = 2$  sec.

contained in the measurements, or that  $\tau$  should be increased. Fraser and Swinney [55] presented the theory for MI test in terms of discrete probabilities:

$$\text{MI}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_{i=1}^n p\hat{y}_i \log_2 p\hat{y}_i - \sum_{j=1}^n py_j \log_2 py_j + \sum_{i=1}^n \sum_{j=1}^n p\hat{y}_i y_j \log_2 p\hat{y}_i y_j \quad (20)$$

where  $\hat{\mathbf{y}}$  and  $\mathbf{y}$  are two sets of  $n$  observations. The first local minima of the MI test gives the optimal time delay, while subsequent minima correspond to a system that has exceedingly unfolded. The computation of (20) is conducted in the SOI algorithm by classifying the last  $n$  observations in a pre-defined number of bins  $\text{MI}_{\text{bin}}$ .

*3.1.2. False Nearest Neighbor Test* The objective of the FNN test is to determine whether the phase-space of a dynamic system is represented in sufficient dimensions. To determine if a dimension is acceptable, the distances between Euclidean neighbors are computed in a given dimension  $d$ , and the distances between the same neighbors recomputed in a new space of dimension  $d + 1$ . If a Euclidean distance increases too much, a false neighbor is detected. Therefore, dimensions are added until the number of false neighbors falls below a threshold. Note that this technique is difficult to apply in dynamic systems with forcing, as there exists numerous crossings in the phase-space; detected false neighbors could actually be true neighbors [47]. In that case, the threshold for the number of false neighbors is augmented.

The application of the FNN test [56] consists of computing the nearest  $r^{\text{th}}$  neighbors in the phase-space from a point  $y(m)$ ,  $1 \leq m \leq n$ , in dimensions  $d$  and  $d+1$ , giving the distance matrices  $R_d(m, r)$  and  $R_{d+1}(m, r)$  between neighbors  $y(m)$  and  $y^{(r)}(m)$ . False neighbors are detected if:

$$\left| \frac{R_{d+1}^2(m, r) - R_d^2(m, r)}{R_d^2(m, r)} \right| > R_{\text{tol}} \quad (21)$$

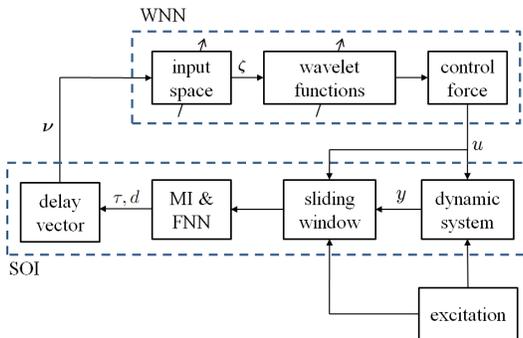


Figure 5: Block diagram of the proposed SOI-WNN.

where  $R_{\text{tol}}$  is a threshold. The authors in [56] also added a second condition to ensure that nearest neighbors are sufficiently close to each other:

$$\frac{R_{d+1}(m)}{R_A} > R_{A,\text{tol}} \quad (22)$$

with:

$$R_A^2 = \frac{1}{n} \sum_{m=1}^n (y(m) - \bar{y})^2$$

where  $\bar{y}$  is the arithmetic average of  $y$ , and  $R_{A,\text{tol}}$  is a threshold. An embedding dimension  $d$  is found when the number of false neighbors fall below the threshold  $R_{\text{num}}$ .

*3.1.3. SOI Algorithm* The proposed SOI algorithm sequentially:

- (i) applies (20) on the last  $n$  observations in the search space  $[\tau(k-1) - 1, \tau(k-1) + 1]$  to find  $\tau(k)$ .
- (ii) applies (21) and (22) using  $\tau(k)$  on the last  $n$  observations in the search space  $[d(k-1) - 1, d(k-1) + 1]$  to find  $d(k)$ .
- (iii) adapts the input vector  $\zeta(k)$  using (15).

Figure 5 summarizes the SOI algorithm integrated to the WNN, which is the proposed SOI-WNN. In the figure, the SOI algorithm selects values of  $\tau$  and  $d$  using the MI and FNN methods on the last  $n$  state and input observations from the sliding window. In the case where the excitation is not observable, the delay vector may be constructed by overembedding the state observation  $\mathbf{y}$  [57]. A delay vector  $\boldsymbol{\nu}$  is constructed and becomes the objective input space. The actual input space  $\zeta$  of the WNN is adapted smoothly based on  $\boldsymbol{\nu}$ , and a new forcing  $u$  is computed.

### 3.2. Hidden Layer

The hidden layer is organized sequentially following Kohonen's SOM theory [54]. A new node is added if:

$$\begin{aligned} \|\zeta - \boldsymbol{\mu}\|^2 &\geq d_{\min} \quad \text{and} \\ s &\geq s_{\text{all}} \end{aligned} \tag{23}$$

where  $d_{\min}$  and  $s_{\text{all}}$  are the thresholds for the minimum nodal distance to the closest node and minimum allowable error, respectively. Once a node is added, the parameters of the new node  $j$  are set to:

$$\begin{aligned} \gamma_j &= s, \quad |\gamma_j| \leq u_b \\ \boldsymbol{\mu}_j &= \zeta \\ \boldsymbol{\sigma}_j &= \lambda \|\zeta - \boldsymbol{\mu}\|^2 \leq \frac{2}{d_j \log \boldsymbol{\sigma}_{\text{mod}}} \log \left( \frac{|u_b| C_t}{\lambda} \right) \end{aligned} \tag{24}$$

where  $\lambda$  is the network resolution. The inequalities represent bounds on the parameters to prevent the addition of nodes with unrealistic functions [58], where  $d_j$  is the dimension of the  $j^{\text{th}}$  wavelet.

Lastly, nodes can be pruned from the network. A node is pruned if its relative weight with respect to the largest nodal weight falls below the threshold  $\gamma_{\min}$  for a given number of consecutive time steps  $\gamma_{\text{num}}$ .

## 4. Self-Adapting Feature

The SOI-WNN parameters from (5) are continuously adapted toward optimality. Given the nonstationary excitation and the integration of semi-active devices in the closed-loop system, the adaptive control rule is not expected to converge. This section derives the parameter adaptation rules. For clarity, the derivation is specialized for a scalar force output (a single control device). The adaptation rules can be easily extended to several control devices using matrix notation. Nevertheless, it might be useful in applications to consider decentralized controllers, for which the scalar notation applies. The derivation shown in this section is adapted from [30].

Consider an estimated neurocontrol force  $\hat{u}_n$ :

$$\hat{u}_n = \hat{\boldsymbol{\gamma}}^T \hat{\boldsymbol{\phi}} \tag{25}$$

with the control rule (8) rewritten here in a scalar version and in terms of (25):

$$u_{sl} = (1 - m_b) \left( \hat{u}_n - \beta \cdot \text{sat} \left( \frac{s}{\Psi} \right) \right) - m_b u_{\text{max}} \text{sat} \left( \frac{s}{\Psi} \right) \tag{26}$$

Using the control rule (8) in the state-space representation:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}_u u + \mathbf{B}_g a_g + \mathbf{B}_w \mathbf{w} \quad (27)$$

with:

$$\mathbf{A} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix}$$

$$\mathbf{B}_u = \begin{bmatrix} \mathbf{O} \\ -\mathbf{M}^{-1}\mathbf{F} \end{bmatrix} \quad \mathbf{B}_g = \begin{bmatrix} \mathbf{O} \\ -\mathbf{E} \end{bmatrix} \quad \mathbf{B}_w = \begin{bmatrix} \mathbf{O} \\ -\mathbf{M}^{-1}\mathbf{H} \end{bmatrix}$$

where  $\mathbf{A} \in \mathbb{R}^{2\text{dof} \times 2\text{dof}}$  is the state-space matrix,  $\mathbf{B}$  is the control force incidence vector with subscripts  $u$ ,  $g$ , and  $w$  referring to actuation, ground excitation, and wind inputs respectively, with  $\mathbf{B}_u \in \mathbb{R}^{2\text{dof} \times a}$ ,  $\mathbf{B}_g \in \mathbb{R}^{2\text{dof} \times 1}$ ,  $\mathbf{B}_w \in \mathbb{R}^{2\text{dof} \times 2\text{dof}}$ ,  $\mathbf{u} \in \mathbb{R}^{a \times 1}$  is the control force input,  $a_g$  is the scalar ground excitation input, and  $\mathbf{w} \in \mathbb{R}^{2\text{dof} \times 1}$  is the wind excitation input,  $\mathbf{X} \in \mathbb{R}^{2\text{dof} \times 1}$  is the state vector,  $\mathbf{x} \in \mathbb{R}^{\text{dof} \times 1}$  is the displacement vector,  $\dot{\mathbf{x}} \in \mathbb{R}^{\text{dof} \times 1}$  is the velocity vector, dof is the number of degrees-of-freedom,  $a$  is the number of control devices ( $a = 1$  in this section),  $\mathbf{I} \in \mathbb{R}^{\text{dof} \times \text{dof}}$  is the identity matrix,  $\mathbf{O}$  are compatible zero matrices,  $\mathbf{M} \in \mathbb{R}^{\text{dof} \times \text{dof}}$ ,  $\mathbf{C} \in \mathbb{R}^{\text{dof} \times \text{dof}}$ , and  $\mathbf{K} \in \mathbb{R}^{\text{dof} \times \text{dof}}$  are the mass, damping, and stiffness matrices respectively,  $\mathbf{F} \in \mathbb{R}^{\text{dof} \times a}$  is the control forces location matrix,  $\mathbf{E} \in \mathbb{R}^{\text{dof} \times 1}$  is a vector of ones,  $\mathbf{H} \in \mathbb{R}^{\text{dof} \times \text{dof}}$  is the wind excitations location matrix, and using the control force error (7) rearranged in terms of estimated forces:

$$u = u_{sl} - \tilde{u} + \epsilon \quad (28)$$

where  $\epsilon$  is the force estimation error, the state error  $\dot{\mathbf{e}}$  between the actual states  $\mathbf{X}$  and desired states  $\mathbf{X}_d$  is written:

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{X}} - \dot{\mathbf{X}}_d \\ &= \mathbf{A}\mathbf{e} + \mathbf{B}_u(u_{sl} - \tilde{u} + \epsilon - u) \\ &= \mathbf{A}\mathbf{e} + \mathbf{B}_u \left( (1 - m_b) \left( \hat{u}_n - \beta \cdot \text{sat} \left( \frac{s}{\Psi} \right) \right) - m_b u_{\max} \text{sat} \left( \frac{s}{\Psi} \right) - \tilde{u} + \epsilon - \gamma^T \phi \right) \end{aligned} \quad (29)$$

Take the following Lyapunov candidate comprising the sliding surface [59]:

$$V = \frac{1}{2} [s^2 + \tilde{\gamma}^T \Gamma_\gamma^{-1} \tilde{\gamma} + \tilde{\phi}^T \Gamma_\phi^{-1} \tilde{\phi}] \quad (30)$$

where  $\Gamma_\gamma^{-1}$  and  $\Gamma_\phi^{-1}$  are positive definite diagonal matrices representing learning parameters, and the tilde denotes the error between the estimated and real values ( $\tilde{\gamma} = \hat{\gamma} - \gamma$ ;  $\tilde{\phi} = \hat{\phi} - \phi$ ). It follows that (30) is positive definite and contains all time varying parameters. Neglecting

the higher order term and specializing for the case where  $s > \Psi$ , the time derivative of (30) is:

$$\begin{aligned}
\dot{V} &= s\mathbf{P}\mathbf{A}e + s\mathbf{P}\mathbf{B}(\hat{\gamma}^T \tilde{\phi} + \tilde{\gamma}^T \hat{\phi} - m_b \hat{\gamma}^T \hat{\phi}) + \tilde{\gamma}^T \Gamma_\gamma^{-1} \dot{\hat{\gamma}} + \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} + \tilde{\gamma}^T \Gamma_\gamma^{-1} \tilde{\gamma} \\
&\quad + \tilde{\phi}^T \Gamma_\phi^{-1} \tilde{\phi} + s\mathbf{P}\mathbf{B}\epsilon - s\mathbf{P}\mathbf{B}\tilde{u} - (1 - m_b)|s|\mathbf{P}\mathbf{B}k - s\mathbf{P}\mathbf{B}m_b u_{max} \text{sat}\left(\frac{s}{\Phi}\right) \\
&= e^T \mathbf{P}^T \mathbf{P} \mathbf{A} e + \tilde{\phi}^T \left( (1 - m_b) \hat{\gamma}^T \mathbf{B}^T \mathbf{P}^T s + \Gamma_\phi^{-1} \dot{\hat{\phi}} \right) \\
&\quad + \tilde{\gamma}^T \left( (1 - m_b) \hat{\phi}^T \mathbf{B}^T \mathbf{P}^T s + \Gamma_\gamma^{-1} \dot{\hat{\gamma}} \right) - s\mathbf{P}\mathbf{B}(\tilde{u} - \epsilon) - (1 - m_b)|s|\mathbf{P}\mathbf{B}k \\
&\quad + \tilde{\xi}^T \dot{\Gamma}_\xi^{-1} \tilde{\xi} - \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} - s\mathbf{P}\mathbf{B}m_b u_{max} \text{sat}\left(\frac{s}{\Phi}\right)
\end{aligned} \tag{31}$$

with:

$$\tilde{\xi} = \begin{bmatrix} \tilde{\gamma} \\ \tilde{\phi} \end{bmatrix}, \Gamma_\xi = \begin{bmatrix} \Gamma_\gamma & 0 \\ 0 & \Gamma_\phi \end{bmatrix}$$

The *tilde* denotes the error between the optimal and estimated parameters, and  $\xi$  represents aggregation of parameters  $\gamma$  and  $\phi$ . By choosing the following adaptation laws:

$$\begin{aligned}
\dot{\hat{\gamma}} &= -(1 - m_c)(\Gamma_\gamma \hat{\phi}) \mathbf{B}^T \mathbf{P}^T s \\
\dot{\hat{\phi}} &= -(1 - m_c)(\Gamma_\phi \hat{\gamma}) \mathbf{B}^T \mathbf{P}^T s \\
\dot{\Gamma}^{-1} &= -s^2 \mathbf{I}
\end{aligned} \tag{32}$$

where  $\mathbf{I}$  is an identity matrix to populate  $\dot{\Gamma}^{-1}$ , equation (31) becomes:

$$\begin{aligned}
\dot{V} &= e^T \mathbf{P}^T \mathbf{P} \mathbf{A} e - s\mathbf{P}\mathbf{B}(\tilde{u} - \epsilon) - (1 - m_b)|s|\mathbf{P}\mathbf{B}k - \tilde{\xi}^T (s^2 \mathbf{I}) \tilde{\xi} - \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} \\
&\quad - s\mathbf{P}\mathbf{B}m_b u_{sl}
\end{aligned} \tag{33}$$

Choosing  $k = u_b$ , where  $u_b$  is a known bound (also positive) on  $\tilde{u}$ , (33) can be rewritten as:

$$\begin{aligned}
\dot{V} &= e^T \mathbf{P}^T \mathbf{P} \mathbf{A} e - s\mathbf{P}\mathbf{B}(\tilde{u} - \epsilon) - (1 - m_b)|s|\mathbf{P}\mathbf{B}k - \tilde{\xi}^T (s^2 \mathbf{I}) \tilde{\xi} - \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} \\
&\quad - s\mathbf{P}\mathbf{B}m_b u_{max} \text{sat}\left(\frac{s}{\Phi}\right)
\end{aligned} \tag{34}$$

Using (9), (34) can be rewritten:

$$\begin{aligned}
\dot{V} &= e^T \mathbf{P}^T \mathbf{P} \mathbf{A} e - s\mathbf{P}\mathbf{B}(\tilde{u} - \epsilon) - |s|\mathbf{P}\mathbf{B}u_b - \tilde{\xi}^T (s^2 \mathbf{I}) \tilde{\xi} - \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} \quad \text{if } \tilde{u} \in C_d \\
\dot{V} &= e^T \mathbf{P}^T \mathbf{P} \mathbf{A} e - s\mathbf{P}\mathbf{B}(\tilde{u} - \epsilon) - |s|\mathbf{P}\mathbf{B}u_{max} - \tilde{\xi}^T (s^2 \mathbf{I}) \tilde{\xi} - \tilde{\phi}^T \Gamma_\phi^{-1} \dot{\hat{\phi}} \quad \text{if } \tilde{u} \in C - C_t
\end{aligned} \tag{35}$$

The first term in (35) is negative semi-definite as the state-space matrix  $\mathbf{A}$  is inherently stable for civil structures. The third term is bigger than the second term for  $\tilde{u} \in C_d$  and is as negative as possible for  $\tilde{u} \in C - C_t$ , and the fourth term is negative definite. The last term in (35) is not necessarily negative-definite, and represents the trade-off in using adaptive

wavelets. Assuming that this last term is smaller than the addition of the first four terms (in absolute value), the error will converge to zero [60].

One can note that the adaptation rules (32) are in function of the unknown matrix  $\mathbf{B}$ . However, it can be assumed that the magnitude of  $\mathbf{B}$  can be roughly evaluated and that its signs (directions of applied forces) are known. Thus,  $\mathbf{B}^T \mathbf{P}^T$  can be incorporated in the learning rate, and the adaptation rules can be written in discrete form:

$$\begin{aligned}
\hat{\alpha}_j(t+1) &= \hat{\alpha}_j(t) - \Delta(1 - m_b)\Gamma_{\alpha_j}\hat{\phi}_j\text{sgn}(\mathbf{B}^T \mathbf{P}^T)\mathbf{s} \\
\hat{\mu}_{j,k}(t+1) &= \hat{\mu}_{j,k}(t) - \Delta(1 - m_b)\Gamma_{\mu_{j,k}}\hat{\alpha}_j \\
&\quad \cdot \left( \frac{1}{\sigma_j^4} \exp \frac{-\|\boldsymbol{\nu} - \mu_j\|^2}{\sigma_j^2} (4\sigma_j^2(\nu_k - \mu_{j,k}) - 2\|\boldsymbol{\nu} - \mu_j\|^2(\nu_k - \mu_k)) \right) \text{sgn}(\mathbf{B}^T \mathbf{P}^T)\mathbf{s} \\
\hat{\sigma}_j(t+1) &= \hat{\sigma}_j(t) - \Delta(1 - m_b)\Gamma_{\sigma_j}\hat{\alpha}_j \\
&\quad \cdot \left( \frac{1}{\sigma_j^5} \exp \frac{-\|\boldsymbol{\nu} - \mu_j\|^2}{\sigma_j^2} (4\sigma_j^2\|\boldsymbol{\nu} - \mu_j\|^2 - 2\|\boldsymbol{\nu} - \mu_j\|^4) \right) \text{sgn}(\mathbf{B}^T \mathbf{P}^T)\mathbf{s}
\end{aligned} \tag{36}$$

where subscript  $k$  is the dimension of the neuron. (36) is the discrete adaptation law used for the simulations.

## 5. Parameters Selection

Parameter selection is a fundamental task in designing neural networks. In structural control, it is not uncommon that non-adaptive parameters are tuned until performance satisfaction is attained. To remain consistent with the proposed application of the controller, one cannot rely on this parameter tuning opportunity. The aim of this section is to provide a discussion on the choice of parameters for the neural network, with the objective to give guidance in the controller design based on limited knowledge of a structure.

Table 1 lists the SOI-WNN main non-adaptive parameters selected for the full-scale simulation presented in Section 7, divided between network objects (inputs, hidden layer, outputs, and adaptation rules). The list is non-exhaustive, but the parameters left out have either little consequences on the performance of the SOI-WNN, or are typically easy to determine. Examples include initial parameters for new nodes and sigmoid functions. To discuss the selection of parameters listed in Table 1, a sensitivity analysis [7] was conducted by subjecting the model of the full-scale structure described in Section 7 to a harmonic excitation tuned at the fundamental frequency. Simulations were ran over 120 sec at a sampling rate of 50 Hz. This section summarizes the main findings.

### 5.1. Inputs

The user-defined parameters related to the input space are associated with the SOI algorithm. The number of bins were selected to be high enough to give good precision in finding  $\tau_\zeta$ ,

Table 1: List of non-adaptive parameters for 1350 kN MFDs.

NN object	parameter class	parameter	value assigned
inputs	lag ( $\tau$ )	# bins (MI test)	20
	dimension ( $d$ )	$R_{\text{tol}}$	15
		$R_{A,\text{tol}}$	2
	window size	$n$	100
dynamic states	$y, u$	$y(\ddot{x}_i), u_i$	
hidden layer	min. nodal distance <sup>1</sup>	$\eta$	0.025
	min. error <sup>1</sup>	$\ \mathbf{P}\mathbf{x}\ _{\min}$	0.025
	network resolution	$\lambda$	$100\ \mathbf{P}\mathbf{x}\ _{\min}$
	pruning	% weight	2%
		# flags	50
outputs	SMC	$\mathcal{C}$	$1/6 u_b$
		$\mathcal{C}_t$	50% $u_b$ kN
		$u_b$	$2u_{\max}$
adaptation	BP error	$\mathbf{P}$ (wind)	parabolic
		$\mathbf{P}$ (earthquake)	constant
	adaptation weights <sup>1</sup>	$\Gamma_\mu$	0.001
		$\Gamma_\sigma$	0.1
		$\Gamma_\gamma$	1000

<sup>1</sup>for inputs normalized to a magnitude of  $10^{-1}$

but low enough to prevent computation time from dramatically increasing. Value around 20 showed to perform well. The distance measures for the FNN tests,  $R_{\text{tol}}$  and  $R_{A,\text{tol}}$ , are more difficult to evaluate, due to the numerous crossings that exist in the phase-space. The thresholds were arbitrarily assigned using high values to account for these crossings. The selection of the window size  $n$  can have dramatic consequences on the network performance and computation time. The size  $n = 100$  performed well for all of the simulations. The sensitivity of the window size  $n$  will be discussed and demonstrated in Section 6.

The dynamic states selected for constructing the  $i^{\text{th}}$  delay vector are the accelerations and the force inputs at floor  $i$ . The multivariate observations are scaled to a comparable magnitude ( $10^{-1}$ ) to prevent some states from having higher importance in the representation. Scaling multivariate representations to comparable magnitudes lead to numerically more stable representations with improved convergence [61].

### 5.2. Hidden Layer

In the hidden layer, the minimum nodal distance  $\eta$ , minimum errors  $e = \|\mathbf{P}\mathbf{x}\|_{\min}$ , and network resolution  $\lambda$  have a direct consequence on the network size and performance. The sensitivity analysis has been performed for the normalized inputs, and the SOI-WNN performance showed to be stable around the selected values, with a limited impact on the network density (size). Pruning parameters did not show to influence the network performance. Those parameters are used to give the controller the capacity to forget control rules, useful for nonstationary systems and impulsive excitations.

### 5.3. Outputs

The outputs of the SOI-WNN are the required forces sent to the control devices. The main non-adaptive parameters are related to the sliding controller and consist of the adaptation regions  $\mathcal{C}$  and  $\mathcal{C}_t$ . For the simulation,  $\mathcal{C}$  has been designed to allow an error on  $\tilde{u}$  of  $1/6 u_b$ , and  $\mathcal{C}_t$  to allow an error on  $\tilde{u}$  of  $1/2 u_b$ , with  $u_b$  equal to twice the damper maximum force, since the force ranges over  $\pm u_{\max}$ .

The sensitivity analyzes performed on  $\mathcal{C}$  and  $\mathcal{C}_t$  show that low values for  $\mathcal{C}$  increase performance, but at the cost of adaptation speed, as expected. The value  $1/6 u_b$  appears to be an appropriate trade-off. The mitigation performance rapidly increases with increasing  $\mathcal{C}_t$  to stabilize after  $1/2 u_b$ .

### 5.4. Adaptation

Decreasing parabolic weights (with increasing height) for the sliding surface  $\mathbf{P}$  are recommended in the case of wind excitations, due to the decreasing control reachability of semi-active control devices. In the case of earthquake excitations, constant sliding surface weights are preferred over parabolic, due to the increasing importance of mitigating interstorey displacements. The matrix  $\mathbf{P}$  is divided between displacement weights (subscript disp) and velocity weights (subscript vel):

$$\Delta\mathbf{P} = [\Delta_{\text{disp}}\mathbf{P}_{\text{disp}}|\Delta_{\text{vel}}\mathbf{P}_{\text{vel}}] \quad (37)$$

where  $\Delta$  is a scalar weight on each sub-matrix, and values within the sub-matrices are taken as  $P_{i,j} \in [-1, 1]$  at the  $i^{\text{th}}$  device location for local measurements (positive value for the top floor and of opposite sign for the bottom floor to create an interstorey state), and zero otherwise. Consequently,  $\Delta$  becomes control weights, and can be user-defined depending on the mitigation goals. For the wind simulation,  $\Delta_{\text{disp}} = 10$  and  $\Delta_{\text{vel}} = 1$ , and for the earthquake simulations,  $\Delta_{\text{disp}} = \Delta_{\text{vel}} = 1000$ . Remark that the earthquake control weights are orders of magnitude higher than the weights utilized for wind mitigation. This enforces a quick adaptation of the controller by overcompensating learning rates, as the adaptation parameters  $\Gamma$  are directly coupled to the choice of  $\Delta$ . During the learning stage, control laws

learned during impulse excitations will create nodes in a sparse region of the representation. These nodes are quickly pruned by the adaptive laws, allowing the controller to forget what could be unrealistic control rules. For the learning rates, sensitivity analyzes showed that values  $\Gamma_u = 0.001, \Gamma_\sigma = 0.1, \Gamma_\gamma = 1000$  led to good performance with respect to mitigation and network resolution.

## 6. Synthetic Example

The proposed SOI-WNN is simulated for tracking the sinusoidal reference signal  $y^*(t) = 0.02 \sin t$  from the following nonlinear equation:

$$y(x, \dot{x}, u) = \left[ 1 - \left( \frac{(x - 0.1)^2}{0.1^2} + \frac{(\dot{x} - 0.05)^2}{0.05^2} \right) \right] \exp^{-\left( \frac{(x - 0.1)^2}{0.1^2} + \frac{(\dot{x} - 0.05)^2}{0.05^2} \right)} + u \quad (38)$$

where the excitation input is  $x = 0.2 \sin 5t$ , and  $u$  is the control input. We have selected this low-dimensional arbitrary example, a particle traveling on a wavelet, because of the stationarity of the excitation and reference signals, which isolates the source of nonstationarity in the adaptive representation. In addition, the stationarity of both signals allows us to pre-process their time series using the embedding theorems in order to determine the fixed inputs that would appropriately represent their dynamics. Lastly, by using a periodic excitation, it is possible to let the neurocontroller converge to a given control rule. Note that the simulations in the next section uses nonstationary excitations.

A delay is induced in the actuator using the following dynamics:  $\dot{u}_{\text{act}} = -\eta(u_{\text{act}} - u_n)$ , where  $\eta$  is a voltage delay taken as  $\eta = 20 \text{ s}^{-1}$  to be consistent with the actuator dynamics taken in [62]. A sliding window size of  $n = 100$  time steps is selected. The choice of  $n$  will be discussed later in this section. The SOI algorithm is compared against three cases of fixed neural inputs built using:

- $\tau = 31$  and  $d = 2$ , which are parameters obtained from pre-processing the time series of the excitation signal, without forcing ( $u = 0$ ).
- $\tau = 8$  and  $d = 2$ , which are parameters obtained from pre-processing the time series of the reference signal.
- $\tau = 31$  and  $d = 8$ , which are the optimal fixed input parameters obtained within the search space  $\tau = [1, 40]$  and  $d = [1, 10]$  while simulating the system with forcing.

Note that values obtained for  $\tau$  are coincidentally the same for the first and third case, and that a dimension of 2 was expected for the first and sec case due to the low complexity of both signals. The large embedding dimension for the third scenario can be explained by a more complex phase-space that counts several crossings once the system (38) includes the forcing  $u$ .

Figure 6 shows the time series response of the SOI algorithm versus the optimal fixed parameters. The SOI algorithm (blue straight line) results in a quicker convergence and better tracking results than the optimal fixed-input WNN (black dot-dash line). This is due to the dynamics of the control rule changing with time, for which adapting the input space results in a more efficient representation, as hypothesized. Table 2 shows the RMS error for the four input strategies over a tracking time of 20 sec, along with the average network size (number of hidden nodes). Results from the overall time series show that the SOI gives good performance relative to the fixed input cases, and preserved a lean network size, with a substantial difference compared with the optimal fixed-input strategy  $\tau = 31$ ,  $d = 8$ . The RMS error taken after 5 sec indicates that its convergence is significantly better. Using the same synthetic example, the next subsections investigate the effect of the sliding window size, evaluate the assumption of quasi-stationarity of local, attempt to sequentially identify fixed inputs, and study the performance of the SOI algorithm under noise.

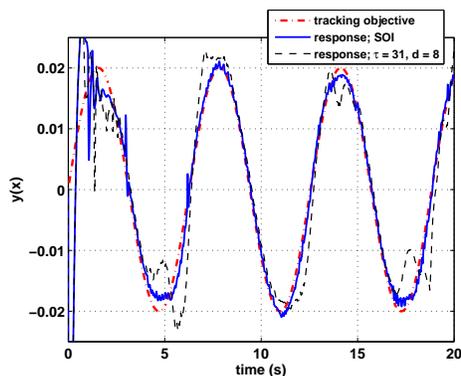


Figure 6: Time series responses. The SOI-WNN (blue straight line) converges more rapidly than the optimal fixed-input WNN (black dot-dash line).

Table 2: RMS error of the controller for various input strategies ( $\times 10^{-5}$ )

		$\tau = 31$	$\tau = 8$	$\tau = 31$
	SOI	$d = 2$	$d = 2$	$d = 8$
over 20 sec	18.9	21.3	23.4	21.1
after 5 sec	3.07	13.2	16.4	12.1
average network size	23.8	25.3	18.6	55.9

### 6.1. Sliding Window Size

Figure 7 shows the RMS error after 5 sec for various values of the sliding window size  $n$ , along with the average computation speed per time step. The performance of the algorithm remains

approximately constant for values greater than 45 time steps, with a slight degradation for larger window sizes. Computation speed is affected negatively for small window sizes, because the controller fails at effectively converging. Once the window size is greater than 45, the computation time augments linearly with increasing  $n$ . Note that the computation speed remains under the sampling rate of 100 Hz for  $45 \leq n \leq 125$ . Simulations were conducted in MATLAB with an Intel i7-2600 3.4 GHz CPU.

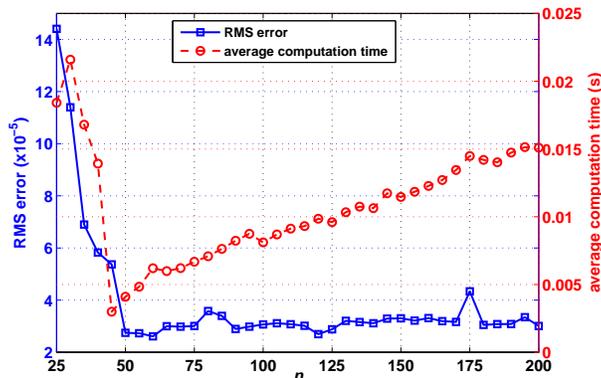


Figure 7: RMS error of the SOI algorithm after 5 sec of simulation, along with the average computation speed per time step for various sliding window sizes

## 6.2. Quasi-Stationarity of Local Maps

To verify the main assumption of quasi-stationarity of local maps, a time-series stationarity index is constructed by determining the change in the control rule within a map. If the change is minimal, then we can write (1) in a stationary way using  $u(k+1) \approx f_u(y(k), u(k))$ . The observations at step  $k$  are taken, and the control force using the control rule at step  $k-n$  computed. The stationarity index is built comparing  $u(k-n)$  and  $u(k)$ , and counts the number of local maps that remained under a given percentage change threshold. Figure 8 graphs the stationarity index as a function of the percentage of change allowed between  $u(k-n)$  and  $u(k)$ , over various time ranges. Results show that 43% of the maps have a change less than 5% over the entire simulation (last 20 sec), which increases to 88% for the last 1 sec. If a change of 10% is allowed, 64% of maps show to be quasi-stationary over the entire simulation, and 91% over the last 5 sec. Results show that the level of quasi-stationarity increases significantly with the convergence of the black-box model. It is estimated that levels of stationary maps above 85% under 10% allowable change satisfy quasi-stationarity, which is met for the last 10 sec of the simulation.

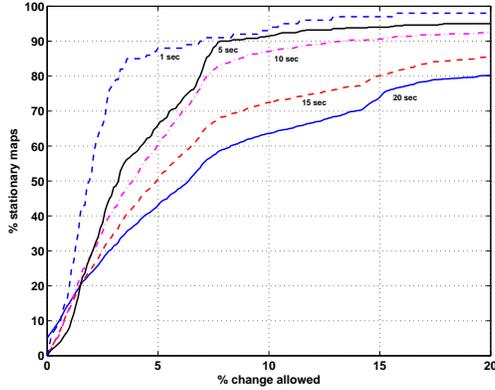


Figure 8: Stationarity index of local maps for the last 1, 5, 10, 15, and 20 sec of the simulation.

### 6.3. Identification of Fixed Inputs

In order to identify fixed (static) inputs for the representation, the SOI algorithm is switched off once the error metric stays below a threshold for a pre-defined number of steps. For this task, the capacity of the network to prune nodes has been relaxed, as we expect needing a denser network to construct an accurate representation of the global dynamics. Figure 9 shows the evolution of the input parameters over time, depicting the self-organizing nature of the input space. The inputs become static after 20 sec, identifying the parameters  $\tau = 12$  and  $d = 2$ . This compares well with the pre-processed values of the controlled time-series aforementioned to be  $\tau = 8$  and  $d = 2$ , as the phase-space of the sinusoidal target only marginally unfolds between both time delays. The value for  $d$  is significantly lower than for the optimal fixed inputs strategy ( $d = 8$ ), because the SOI algorithm computes the optimal  $d$  based only on the last  $n$  observations.

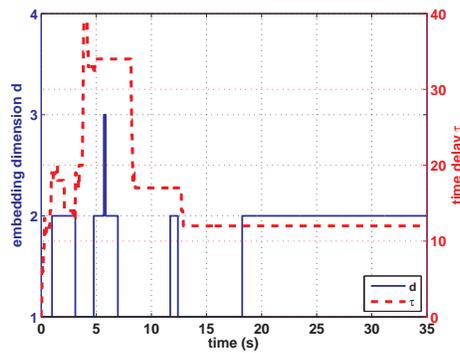


Figure 9: Identification of fixed  $\tau$  and  $d$  for a global representation.

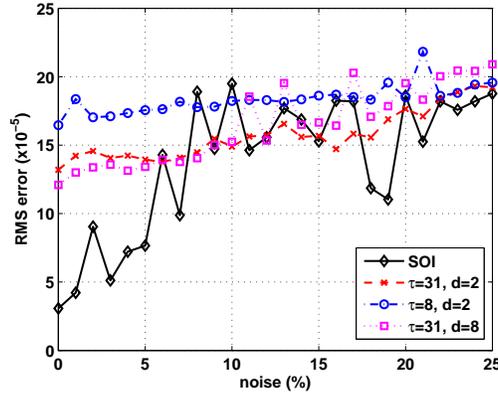


Figure 10: RMS error with respect to noise, after 5 sec ( $\times 10^{-5}$ ).

#### 6.4. Noise

Here, various levels of Gaussian noise have been induced in the observations to study the performance under noise. Figure 10 shows the RMS error for noise ranging from 0% to 25%. The SOI-WNN is capable to significantly outperform any optimal fixed-input strategies for noise under 5%. However, above that level, the relative performance of the algorithm reduces quickly, with a tendency to perform similarly to the fixed-input strategies.

### 7. Full-Scale Simulation

The proposed control algorithm is simulated on a tower located in downtown Boston, Massachusetts, and its performance evaluated for wind and earthquake excitations. This section describes the simulation and shows the results.

#### 7.1. Simulated Structure

The simulated structure is a 39-story office tower located in downtown Boston, Massachusetts. It was built in 1990 with a viscous damping system to mitigate excessive wind vibrations caused by a nearby tower. The design strategy and performance evaluation for the viscous damping system is documented in [63]. The tower comprises two viscous dampers in each direction (X- and Y- direction) every other floor, between floor 5 and 33, for a total of 60 viscous dampers. Figure 11 shows an elevation view of the structure in both directions along with the location of the dampers. Table 3 lists the viscous damper properties. Table 4 compares the fundamental periods of the models with the values reported in [63] from the results of a wind tunnel test.

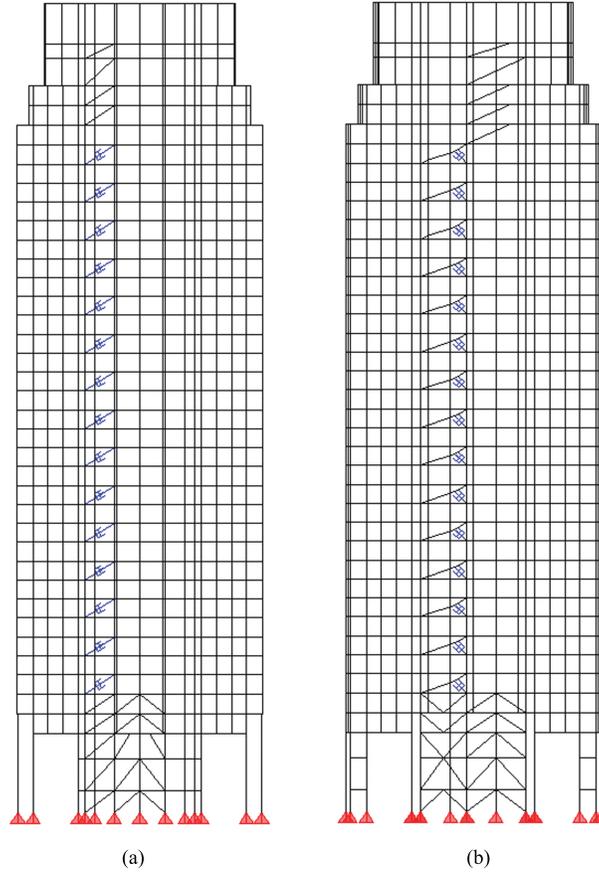


Figure 11: Elevation view of the simulated structure : a) X-direction; and b) Y-direction.

Table 3: Configuration of Viscous Dampers (2 dampers per direction per floor)

	capacity (kN)		number of dampers	
	X-direction	Y-direction	X-direction	Y-direction
below 26 <sup>th</sup> floor	1350	90	22	22
above 26 <sup>th</sup> floor	900	45	8	8

Table 4: Fundamental periods and comparison with values reported in [63] from a wind tunnel testing.

mode	shape	direction	model (sec)	period	
				reported in [63] (sec)	difference (%)
1	X	5.28	5.26	+0.38	
2	Y	5.00	5.00	0.00	
3	$\theta$	3.63	3.65	-0.55	
4	X	2.16	1.92	-12.5	
5	Y	2.07	1.82	-13.7	
6	$\theta$	2.01	1.71	-17.5	

### 7.2. Semi-Active Control Device

In the simulations, the viscous dampers are replaced by variable friction dampers of similar capacity, termed Modified Frictions Devices (MFD) [10]. The semi-active device consists of a drum brake mechanism installed in parallel with a viscous and a stiffness element. The drum brake is equipped with a linear actuator, which force is substantially amplified by the brake self-energizing mechanism. The device can theoretically output a damping force range 0-1350 using a few linear actuators operating on batteries. The MFD was designed to mimic the dynamics of an MR damper. Figure 12 exhibits the idealized dynamics of a 200 kN MFD over various levels of voltage.

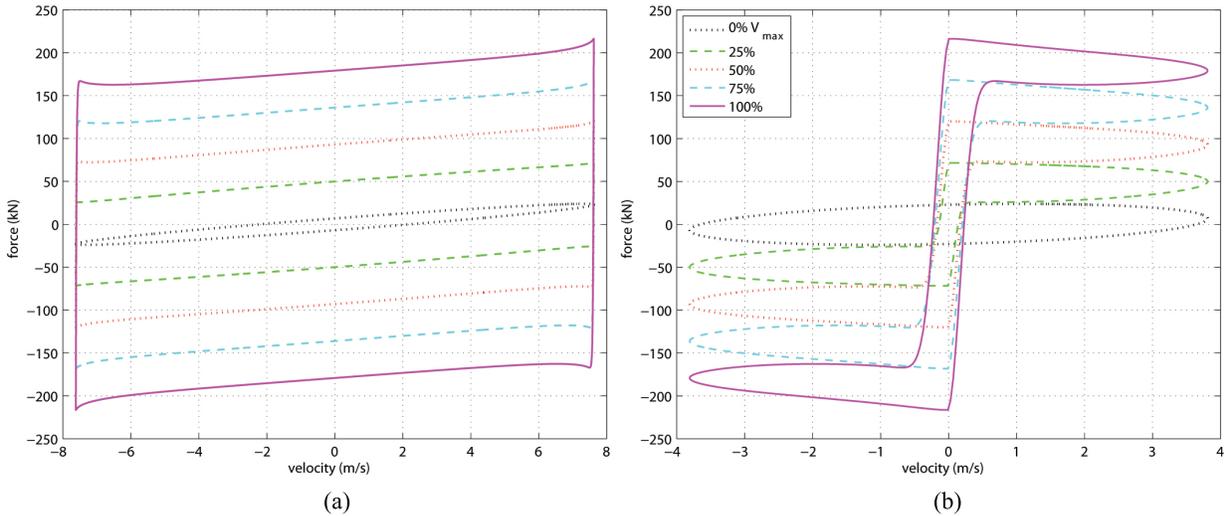


Figure 12: dynamics of the MFD under a 7.62 mm amplitude sinusoidal excitation of 0.5 Hz: a) force-displacement; and b) force-velocity [10].

### 7.3. Performance Criteria

The performance of the SOI-WNN is assessed relative to a benchmark linear quadratic regulator (LQR) designed with full parametric knowledge and using full state feedback. In addition, the SOI-WNN is compared against a WNN controller with fixed-inputs (FI-WNN) that have been optimized a priori, and against the passive-on (ON) case where the device is ideally assumed to operate on maximum voltage. Remark that the LQR controller is designed assuming linearity in the control force. Such design is common in structural control, as the dynamic of civil structures is inherently stable; semi-active systems do not have the capacity to destabilize the controlled plant.

Under the SOI-WNN and FI-WNN, each of the devices is decentrally controlled by a single WNN using local measurements. The only available inputs to the SOI-WNN and

FI-WNN are the acceleration, interstorey displacement, and interstorey velocity of the floor at which the controlled device is installed, in addition to the damping force of the device. The SOI uses the time series of the acceleration to determine the delay vector  $\nu$  and to subsequently adapt the input space  $\zeta$  using data from the acceleration and force states. The interstorey displacements and velocities are used for the sliding controller to compute the sliding surface  $s$ .

Four control objectives are utilized to evaluate the performance of the controller: 1) the maximum acceleration of the 37<sup>th</sup> floor (J1), which corresponds to the highest occupied floor and corresponds with the main control objective in [63]; 2) the maximum floor acceleration (J2); 3) the maximum interstorey displacement (J3); and the average improvement on voltage consumption over the LQR strategy (J4).

#### 7.4. Results - Wind Excitation

To assess the performance of the SOI-WNN at wind mitigation, a nonstationary wind excitation has been selected to match the acceleration results from a wind tunnel test as described in [63]. The performance of the SOI-WNN is compared against 3 fixed-input strategies optimized over the search space  $\tau = [1, 40]$  and  $d = [1, 10]$ :

- $\tau = 16, d = 2$ , the optimized performance for J1.
- $\tau = 4, d = 7$ , the optimized performance for J2.
- $\tau = 40, d = 2$ , the optimized performance for J3.

The index J4 is excluded from the list above because minimizing the index would trivially result in no voltage input. Table 5 shows the relative wind mitigation performance of each controllers with respect to the LQR control strategy. The SOI-WNN outperformed the LQR controller under performance indices J1 and J2, and performed similarly for inter-storey displacement mitigation (J3) consuming approximately 6% more voltage on average (J4). In addition, the SOI-WNN outperforms all of the pre-optimized fixed-input strategies using substantially less voltage (J4). The passive-on case was not effective at mitigating wind acceleration.

#### 7.5. Results - Earthquake Excitations

To assess the performance at impulse-type excitations, 30 different earthquakes of different types and epicentral distances have been simulated. They are listed in Table 6. The excitations have been scaled to 0.12 g to be consistent with the Massachusetts building code.

Table 7 shows the relative earthquake mitigation performance of each controllers with respect to the LQR control strategy. For ease of comparison, the results have been averaged by epicentral distance as it is generally correlated with the level of impulse.

Table 5: Relative wind mitigation performance (J1-J3) and average improvement on voltage consumption over the LQR strategy (J4)

	SOI	fixed-input (FI) strategies			ON
		$\tau = 16$	$\tau = 4$	$\tau = 40$	
		$d = 2$	$d = 7$	$d = 2$	
J1	4.23	-1.41	-12.5	-3.80	-12.8
J2	8.52	0.92	2.36	-0.13	-21.3
J3	0.13	-11.7	-12.0	-5.27	-10.9
J4	-5.95	-32.6	-21.9	-34.0	-138

Here, the SOI-WNN performed similarly to the LQR controller in most cases at acceleration mitigation (J1 and J2) for epicentral distances ranging 0 to 70 km (excluding at-fault). In the case of inter-storey mitigation (J3), the SOI-WNN generally performed better than the LQR in the ranges of 10-50 km. When compared with the FI-WNN, performances are slightly better but comparable for the SOI-WNN. The passive-on case exhibits good mitigation performance for far-field earthquakes in the range of 50-300 km away from the epicenter.

## 8. Discussion & Conclusion

A novel neurocontroller has been presented for semi-active control of civil structures. The controller is an adaptive wavelet neural network, for which the novel feature is a self-adapting input space using the SOI algorithm. The SOI-WNN can sequentially update its input space based on limited and local measurements, while an excitation is occurring, which leads to a more effective representation due to the system nonstationarities.

The synthetic example from Section 6 demonstrated that the SOI algorithm was a powerful feature. For the example, the SOI algorithm resulted in being the best input option to achieve a quick rate of convergence compared against optimized fixed input strategies. This high performance was attained with a good level of stability and acceptable computation time with respect to the sliding window size, and the SOI-WNN was substantially better than any fixed-input strategies over the provided search space for noise levels between 0-5%. Based on the great performance on the low-dimensional synthetic example, the SOI-WNN has been simulated on a full-scale structure equipped with (nonlinear) semi-active control devices, subjected to wind and earthquake excitations.

Results from the wind excitation (Table 5) showed that the SOI-WNN was capable of achieving, at least, the same level of performance as the LQR controller, if not better. It also outperformed all of the fixed-input strategies optimized over the provided search space, using less power. Such high performance is similar to the synthetic example, and may be explained

Table 6: List of the 30 earthquakes simulated

Location	Year	Angle (deg)	Distance (km)	Mag. (RS)
Big Bear City, CA	2003	90	49.3	4.92
Chi-Chi, Taiwan	1999	0	59	7.62
Coalinga, CA	1983	0	30.3	6.36
Coyote Lake, CA	1979	230	10.2	5.74
Denali, Alaska	2002	90	263.6	7.9
Dinar, Turkey	1995	90	at fault	6.4
Duzce, Turkey	1999	90	8	7.14
Erzican, Turkey	1992	90	at fault	6.69
Friuli, Italy	1976	0	49.1	6.5
Gilroy, CA	2002	50	108.1	4.9
Imperial Valley, CA	1940	180	13	7
Irpinia, Italy	1980	0	22.5	6.9
Kern County, CA	1952	111	56	7.36
Kobe, Japan	1995	90	7.1	6.9
Kocaeli, Turkey	1999	0	68.1	7.51
Loma Prieta, CA	1989	170	72.1	6.93
Mammoth Lakes, CA	1980	0	14.3	5.69
Manjil, Iran	1990	66	50	7.37
Michoacan, Mexico	1985	180	250	8.1
Nahanni, Canada	1985	240	at fault	6.76
New Zealand	1987	40	68.7	6.6
Norcia, Italy	1979	90	31.4	5.9
Northridge, CA	1994	90	17.3	6.69
Parkeld, CA	1966	85	9.6	6.19
San Fernando, CA	1971	164	at fault	6.61
San Salvador, El Savador	1986	180	3.7	5.8
San Francisco, CA	1957	10	9.6	5.28
Spitak, Armenia	1988	0	24	6.77
Tabas, Iran	1978	0	1.8	7.35
Victoria, Mexico	1980	45	13.8	6.33

by the low-dimensional response of the structure when subjected to wind excitations, despite of their nonstationary nature.

From such high performance on controlling low dimensional dynamic systems, it could have been hypothesized that the performance of the SOI-WNN would be different

Table 7: Relative earthquake mitigation performance (J1-J3) and average improvement on voltage consumption over the LQR strategy (J4)

Distance (km)	J1			J2			J3			J4		
	SOI	FI	ON									
at fault	-5.97	-4.14	-5.20	-4.29	-2.86	2.37	-8.73	-10.8	-17.4	13.2	6.63	-62.0
]0-10[	-1.30	-2.09	-1.17	-0.11	-0.52	-3.21	-0.20	-0.56	-7.88	5.01	2.57	-85.8
[10-20[	-0.51	-1.69	-1.36	0.71	0.24	0.95	9.84	10.12	-22.4	-6.78	-12.8	-86.4
[20-50[	-0.30	-0.15	-10.5	-2.51	-3.97	-10.5	1.90	1.03	-45.9	8.82	5.56	-88.5
[50-70[	-0.55	-1.07	1.31	-2.95	-3.10	1.84	-10.4	-10.9	-0.49	15.8	11.2	-82.7
[70-300[	-4.84	-4.06	0.43	-1.90	-4.39	3.72	-8.00	-9.77	2.68	20.0	19.9	-62.5

for impulse-type excitations. Simulating the controller to earthquake excitations is an ultimate performance test, because the controller must successfully mitigate an impulse-type excitation without pre-training. In other words, the controller is required to directly learn a control rule lying in a sparse hyperspace. Simulation results (Table 7) show that the SOI-WNN performed similarly to an LQR controller at acceleration and displacement reduction (J1-J3), except for earthquakes at fault (very high impulse) or far-field, but mitigation from the SOI-WNN was typically attained using less voltage (J4). Comparisons against fixed input strategies showed similar vibration reduction capabilities (J1-J3), but here again the SOI-WNN consumed less voltage (J4). Thus, the SOI-WNN was more effective at vibration mitigation than the FI-WNN.

It follows that the proposed controller is a promising method for semi-active structural control. It has substantial benefits compared against other adaptive control approaches: 1) the input selection process is automated for time series that are not known a priori; 2) the representation is adapted to nonstationarities; and 3) it uses limited observations.

## References

- [1] S. Narasimhan and S. Nagarajaiah. A STFT semiactive controller for base isolated buildings with variable stiffness isolation systems. *Engineering structures*, 27(4):514–523, 2005.
- [2] J. Ghaboussi and A. Joghataie. Active control of structures using neural networks. *Journal of Engineering Mechanics*, 121(4):555–567, 1995.
- [3] B. Xu. Neural networks for decentralized control of cable-stayed bridge. *Journal of Bridge Engineering*, 8:229, 2003.
- [4] TT Soong and GP Cimellaro. Future directions in structural control. *Structural Control and Health Monitoring*, 16(1):7–16, 2009.
- [5] M.D. Symans and M.C. Constantinou. Semi-active control systems for seismic protection of structures: a state-of-the-art review. *Engineering Structures*, 21(6):469–487, 1999.
- [6] G. Yang, BF Spencer, JD Carlson, MK Sain, et al. Large-scale mr fluid dampers: modeling and dynamic performance considerations. *Engineering structures*, 24(3):309–323, 2002.

- [7] S. Laflamme. *Control of Large-Scale Structures with Large Uncertainties*. PhD thesis, Massachusetts Institute of Technology, 2011.
- [8] BF Spencer Jr and S. Nagarajaiah. State of the art of structural control. *Journal of Structural Engineering*, 129:845, 2003.
- [9] J.N. Yang, A.K. Agrawal, B. Samali, and J.C. Wu. Benchmark problem for response control of wind-excited tall buildings. *Journal of Engineering Mechanics*, 130:437, 2004.
- [10] S. Laflamme, D. Taylor, M. Abdellaoui-Maane, and J.J. Connor. Modified friction device for control of large-scale systems. *Structural Control & Health Monitoring*, 2011.
- [11] JN Yang, JC Wu, and AK Agrawal. Sliding mode control for nonlinear and hysteretic structures. *Journal of Engineering Mechanics*, 121(12):1330–1339, 1995.
- [12] K. Thai, F. Jabbari, and JE Bobrow. Structural control via semi-active and hybrid control. In *American Control Conference, 1997. Proceedings of the 1997*, volume 1, 1997.
- [13] N. Wongprasert and MD Symans. Numerical evaluation of adaptive base-isolated structures subjected to earthquake ground motions. *Journal of engineering mechanics*, 131:109, 2005.
- [14] S.G. Wang, P.N. Roschke, and HY Yeh. Robust control for structural systems with uncertainties. In *Proceedings of SPIE*, volume 3671, page 206, 1999.
- [15] S.G. Wang, HY Yeh, and P.N. Roschke. Robust control for structural systems with parametric and unstructured uncertainties. *Journal of Vibration and Control*, 7(5):753, 2001.
- [16] WJ Wu and CS Cai. Experimental study of magnetorheological dampers and application to cable vibration control. *Journal of Vibration and Control*, 12(1):67, 2006.
- [17] ZQ Gu and SO Oyadiji. Diagonal Recurrent Neural Networks for MDOF Structural Vibration Control. *Journal of Vibration and Acoustics*, 130:061001, 2008.
- [18] P.Y. Lin and C.H. Loh. Semi-active control of floor isolation system using MR-damper. In *Proceedings of SPIE*, volume 6932, page 69320U, 2008.
- [19] D.A. Shook, P.N. Roschke, P.Y. Lin, and C.H. Loh. Semi-active control of a torsionally-responsive structure. *Engineering Structures*, 31(1):57–68, 2009.
- [20] A. Joghataie and M. Farrokh. Dynamic analysis of nonlinear frames by prandtl neural networks. *Journal of Engineering Mechanics*, 134:961, 2008.
- [21] J. Kim and H.Y. Jeong. A study on the hysteresis, surface temperature change and fatigue life of SM490A, SM490A-weld and FC250 metal materials. *International Journal of Fatigue*, 2009.
- [22] L. Huo, H. Li, and B. Li. Seismic responses prediction of nonlinear building structures based on multi-branch bp neural network. *Advances in Neural Networks–ISNN 2009*, pages 919–928, 2009.
- [23] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
- [24] Q. Zhang, A. Benveniste, and L.T. Hogskola. Wavelet networks. *IEEE transactions on Neural Networks*, 3(6):889–898, 1992.
- [25] L. Zhou and L.X. Wang. Adaptive fuzzy control for nonlinear building-magnetorheological damper system. *Journal of Structural Engineering*, 129:905, 2003.
- [26] H.J. Lee, G. Yang, H.J. Jung, B.F. Spencer, and I.W. Lee. Semi-active neurocontrol of a base-isolated benchmark structure. *Structural Control and Health Monitoring*, 13(2-3):682–692, 2006.
- [27] H.J. Jung, H.J. Lee, S.W. Cho, and I.W. Lee. An Experimental Study of Semiactive Modal Neuro-control Scheme Using MR Damper for Building Structure. *Journal of intelligent material systems and structures*, 19(9):1005–1015, 2007.
- [28] S. Suresh, S. Narasimhan, and N. Sundararajan. Adaptive control of nonlinear smart base-isolated buildings using Gaussian kernel functions. *Structural Control and Health Monitoring*, 15(4):585–603, 2008.
- [29] S. Laflamme and J.J. Connor. Application of self-tuning Gaussian networks for control of civil structures equipped with magnetorheological dampers. In *Proceedings of SPIE*, volume 7288, page 72880M, 2009.

- [30] S. Laflamme, J.-J.E. Slotine, and J.J. Connor. Wavelet network for semi-active control. *Journal of Engineering Mechanics*, 2011.
- [31] R. Sindelar and R. Babuska. Input selection for nonlinear regression models. *Fuzzy Systems, IEEE Transactions on*, 12(5):688–696, 2004.
- [32] G.J. Bowden, G.C. Dandy, and H.R. Maier. Input determination for neural network models in water resources applications. Part 1–background and methodology. *Journal of Hydrology*, 301(1-4):75–92, 2005.
- [33] X. Hong, RJ Mitchell, S. Chen, C.J. Harris, K. Li, and GW Irwin. Model selection approaches for non-linear system identification: a review. *International journal of systems science*, 39(10):925–946, 2008.
- [34] A. da Silva, P. Alexandre, V.H. Ferreira, and R.M.G. Velasquez. Input space to neural network based load forecasters. *International Journal of Forecasting*, 24(4):616–629, 2008.
- [35] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.
- [36] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.
- [37] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [38] DL Yu, JB Gomm, and D. Williams. Neural model input selection for a MIMO chemical process. *Engineering Applications of Artificial Intelligence*, 13(1):15–23, 2000.
- [39] K. Li and J.X. Peng. Neural input selection—A fast model-based approach. *Neurocomputing*, 70(4-6):762–769, 2007.
- [40] J. Tikka. Simultaneous input variable and basis function selection for RBF networks. *Neurocomputing*, 72(10-12):2649–2658, 2009.
- [41] N. Kourentzes and S.F. Crone. Frequency independent automatic input variable selection for neural networks for forecasting. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2010.
- [42] F. Takens. Detecting strange attractors in turbulence. *Dynamical systems and turbulence, Warwick 1980*, pages 366–381, 1980.
- [43] J. Stark. Delay embeddings for forced systems. I. Deterministic forcing. *Journal of Nonlinear Science*, 9(3):255–332, 1999.
- [44] V. Caballero. On an embedding theorem. *Acta Mathematica Hungarica*, 88(4):269–278, 2000.
- [45] J. Stark, DS Broomhead, ME Davies, and J. Huke. Delay embeddings for forced systems. II. Stochastic forcing. *Journal of Nonlinear Science*, 13(6):519–577, 2003.
- [46] L. Moniz, JM Nichols, CJ Nichols, M. Seaver, ST Trickey, MD Todd, LM Pecora, and LN Virgin. A multivariate, attractor-based approach to structural health monitoring. *Journal of Sound and Vibration*, 283(1-2):295–310, 2005.
- [47] LA Overbey, CC Olson, and MD Todd. A parametric investigation of state-space-based prediction error methods with stochastic excitation for structural health monitoring. *Smart Materials and Structures*, 16:1621–1638, 2007.
- [48] E. Monroig. *Detection of Changes in Dynamical Systems by Nonlinear Time Series Analysis*. PhD thesis, University of Tokyo, 2009.
- [49] L. Cao, Y. Hong, H. Fang, and G. He. Predicting chaotic time series with wavelet networks. *Physica D: Nonlinear Phenomena*, 85(1-2):225–238, 1995.
- [50] JC Principe, L. Wang, and MA Motter. Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control. *Proceedings of the IEEE*, 86(11):2240–2258, 1998.
- [51] VP Plagianakos and E. Tzanaki. Chaotic analysis of seismic time series and short term forecasting

- using neural networks. 2001.
- [52] DM Walker, NB Tuffiaro, and P. Gross. Radial-basis models for feedback systems with fading memory. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(9):1147–1151, 2001.
  - [53] J. Zolock and R. Greif. A Methodology for the Modeling of Forced Dynamical Systems From Time Series Measurements Using Time-Delay Neural Networks. *Journal of Vibration and Acoustics*, 131:011003, 2009.
  - [54] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
  - [55] A.M. Fraser and H.L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, 1986.
  - [56] M.B. Kennel, R. Brown, and H.D.I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411, 1992.
  - [57] E. Monroig, K. Aihara, and Y. Fujino. Modeling dynamics from only output data. *Physical Review E*, 79(5):56208, 2009.
  - [58] M. Cannon and J.J.E. Slotine. Space-frequency localized basis function networks for nonlinear system estimation and control. *Neurocomputing*, 9(3):293–342, 1995.
  - [59] J.J.E. Slotine and JA Coetsee. Adaptive sliding controller synthesis for non-linear systems. *International Journal of Control*, 43(6):1631–1651, 1986.
  - [60] J.J.E. Slotine and W. Li. *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991.
  - [61] M. Nørgaard, O. Ravn, NK Poulsen, and LK Hansen. *Neural networks for modelling and control of dynamic systems: A practitioner’s handbook*. Springer Verlag, 2000.
  - [62] S. Laflamme, J.J.E. Slotine, and J.J. Connor. Wavelet Network for Semi-Active Control. *J. of Engineering Mechanics*, 137(7).
  - [63] R.J. McNamara and D.P. Taylor. Fluid viscous dampers for high-rise buildings. *The Structural Design of Tall and Special Buildings*, 12(2):145–154, 2003.