Semi-physical neural modeling for linear signal restoration

Laurent Bourgois, Gilles Roussel, Mohammed Benjelloun

Semi-physical neural modeling for linear signal restoration

Laurent Bourgois[a,*], Gilles Roussel[b], Mohammed Benjelloun[b]

[a] *SUPELEC E3S, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette Cedex, France*
[b] *LISIC, 50 rue Ferdinand Buisson, 62228 Calais Cedex, France*

Abstract

This paper deals with the design methodology of an Inverse Neural Network (INN) model. The basic idea is to carry out a semi-physical model gathering two types of information: the *a priori* knowledge of the deterministic rules which govern the studied system and the observation of the actual conduct of this system obtained from experimental data. This hybrid model is elaborated by being inspired by the mechanisms of a neuromimetic network whose structure is constrained by the discrete reverse-time state-space equations. In order to validate the approach, some tests are performed on two dynamic models. The first suggested model is a dynamic system characterized by an unspecified *r*-order Ordinary Differential Equation (ODE). The second one concerns in particular the mass balance equation for a dispersion phenomenon governed by a Partial Differential Equation (PDE) discretized on a basic mesh. The performances are numerically analyzed in terms of generalization, regularization and training effort.

Keywords: Semi-physical modeling, inverse problem, neural network, model fusion

1. Introduction

Many applications require data inversion. Inverse problems or signal restoration are solved by the inversion of a forward representation which models the actual conduct of the studied system. There are several techniques that can be used to realize this inversion such as variational method, criterion optimization, inverse filtering, analytical solution from forward model, *etc.*. All these different methods depend upon a mathe-

*Corresponding author. Tel: +33 169 851 397; Fax: +33 169 851 429.
Email address: laurent.bourgois@supelec.fr (Laurent Bourgois)

matical description of the real behavior of the system. According to how much *a priori* information is available, it is possible to carry out either a knowledge-based (white-box) model based on the physical, chemical, biological or sociological principles, or an empirical (black-box) model based on the *a priori* choice of a well-suited analytical function followed by a data identification procedure. Of course, the quality of the restoration by data inversion depends on the observation noise, on the model accuracy, and on the inversion method. However, it is usually difficult to find an analytical solution when the system is quite complex, often non-linear and time-dependent. Such complex or imprecise system can be modeled by combining knowledge on the physical laws and data measured during system operation. This model is named semi-physical or gray-box concept. Although this approach is usually reserved for forward modeling, the idea consists in carrying out a semi-physical inverse neural network model gathering physical knowledge of an inverse relaxed mechanistic model and data accumulated during a statistical learning phase. Thus, a robust INN model is ensured using *a priori* knowledge on the physical laws which govern the system. With the help of this INN model, we propose a technique having not only a faculty of learning and adaptability, but also a good efficiency relative to inverse problem difficulties. In order to test the method, we have studied the deconvolution problem by examining a linear model defined by an ODE and a linear spatio-temporal model governed by a PDE.

Establishing a robust white-box model within the meaning of exhaustiveness compared to the variations of context is often tricky to express for several reasons. One needs a perfect expertise to enumerate all the physical laws and influential variables brought into play. Besides, an exhaustive spatial and temporal system description is also required. However, even if the previous stage is completed, some parameters may not be measured or precisely known. It is then advisable to estimate these parameters starting from observable data. Once the physical model has been fixed, it is endowed with a good robustness.

A black-box model is a behavior model particularly well-suited for complex system representation (Sjöberg et al., 1995), but which does not take into account any *a priori* information. Many standard process forms which present system's input-output relation starting from experimental data can be considered as black-box models: ARMA,

ARMAX (Ljung, 1999), NARMAX, Box-Jenkins (Box et al., 1994), NOE (Nonlinear Output Error), *etc.*. Another approach based on classical neural networks does not specify a mathematical form but rather a neural design which is more suited to the system dynamics. One of the main advantages of neural networks is their great adaptability to static, dynamic, linear or nonlinear functions, thanks to the universal approximation property (Sontag, 1997). Moreover, neural networks have been successfully used to nonlinear dynamic systems modeling. The form of usual nonlinear activation functions (*e.g.* sigmoid activation functions) results in parsimonious estimation, *i.e.* weak residual error with a minimum number of parameters (Barron, 1993). Nevertheless, black-box models are often less parsimonious than knowledge-based ones. Indeed, the mathematical functions used to describe white-box models are more accurate and minimize output errors in absence of noise.

Between the two models previously exposed, the gray-box model has emerged as an important tool during the past two decades. This approach has been termed gray-box modeling (Duarte et al., 2004; Beghi et al., 2007), hybrid modeling (Zorzetto et al., 2000) or semi-physical modeling (Lindskog & Ljung, 1995) in the literature. A paper by Leifsson et al. (2008) distinguishes two types of approach: serial and parallel gray-box modeling. These two patterns differ in the manner in which they combine black-box and white-box models. Serial gray-box modeling makes a numerical separation between the known and the unknown physical part of the system (Nelles, 2001), whereas parallel gray-box modeling introduces a kind of competition between black-box and white-box models. Generally, black-box model corrects the predicted outputs of the white-box model. Between these two methods, there is another approach which is much more closer to the notion of model fusion. This approach consists in modifying the design of a recurrent neural network (Oussar & Dreyfus, 2001; Ploix & Dreyfus, 1997). The idea is to design a recurrent neural network using engineer's knowledge on the fundamental laws which govern the system. In this case, *a priori* information is based on the network design. One or more degrees of freedom (*e.g.* additional neurons) may also be added to help the network successfully adapt to the ignored parts of the system (Oussar & Dreyfus, 2001). Process measurements are then used to learn the network. The recall phase then supplies predicted output values in real-

time (Krasnopolsky & Fox-Rabinovitz, 2006). Other approaches have been proposed by Cherkassky et al. (2006). They consist in carrying out the emulation of physically-based process models using neural network training starting from white-box model simulations. Semi-physical or gray-box modeling has often been used in the case of forward models. This type of model fulfills at the same time precision requirements, robustness and parsimony of the knowledge-based models, and also possesses the faculty of training and adaptability. Our idea consist in being inspired by such a concept in order to apply it in the case of inverse problems.

2. Inverse neural modeling

2.1. Principle

The objective of many applications such as inverse problems in meteorology, tomography, software sensor, deconvolution or open-loop control system is to realize the inversion of a physical model. It generally consists in estimating non-measurable parameters or inputs starting from the measurable observations and *a priori* information about the system. There are several numerical ways to deal with this problem such as state-space transformations (*e.g.* Laplace, Fourier, *etc.*), forward state-space model discretization followed by a matrix inversion, or the definition of a performance function to minimize (Groetsch, 1993; Tarantola, 1987).

Our proposed additional objective is to realize the inverse model training. Some ideas for forward and inverse model training in physical measurement applications have been proposed by Krasnopolsky & Schillerb (2003). Learning phase consists in weight estimation by backpropagation. The coefficients are then adjusted to move the network outputs closer to the desired inputs (figure 1).

In recall phase, the network estimates the input sequences by supposing that the real model does not evolve any more after the last training (figure 2). Implicitly, this method looks like the error propagation through the adjoint network.

2.2. Regularization

Inverse problems are often ill-posed in the Hadamard sense (Groetsch, 1993). They can present an absence of solution, multiple solutions, or an unstable solution. To

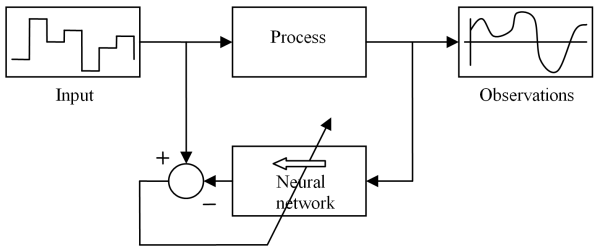

Figure 1: Training phase of the INN model. A (noisy) synthetic output signal is simulated starting from the input signal, and introduced in the neural network input. The coefficients are then adjusted to move the network outputs closer to the desired inputs.

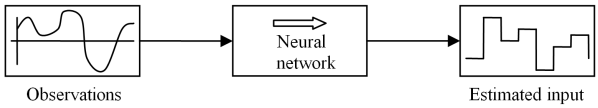

Figure 2: Recall phase of the INN model. The network estimates the input sequences by supposing that the real model does not evolve any more after the last training.

transform ill-posed problems into well-conditioned ones, it is necessary to add *a priori* knowledge on the system before inversion. There are two approaches which differ according to the type of *a priori* knowledge introduced. The first procedure employs regularization methods based on deterministic information (Thikhonov & Arsenin, 1977). The second strategy considers techniques based on probabilistic information such as Bayesian methods (Marroquin et al., 1987; Demoment, 1989) or maximum entropy methods (Mohammad-Djafari et al., 2002).

But, can we discuss the regularization problem in the case of the INN model ? Let us underline that a neural network always provides an output, regardless of the appropriateness of the input, due to its autoassociative memory property. That answers the two main difficulties of ill-posed inverse problems, even if the suggested solution can prove to be false. In addition, regularization during training phase improves generalization with respect to the set of examples. It avoids the problem of overfitting which results in an instability. It is also remarkable that early stopping procedure, *i.e.* stopping the gradient descent before learning process reaches the optimal solution on the training set, supplies solutions with smaller generalization error. Besides, some

Bayesian techniques have been developed to adjust the regularization coefficients of the performance function (MacKay, 1992). This confirms our opinion to use the neural network like an inverse model.

3. Design of a semi-physical inverse neural network model

The construction of a gray-box forward neural network model is generally performed in three steps:

Step 1: Discrete-time neural network design from the knowledge-based model;

Step 2: Training of the semi-physical forward neural network model from knowledge-based simulations in order to obtain appropriate initial values;

Step 3: Training of the semi-physical neural model from experimental data.

The knowledge-based model is usually represented in the form of a set of coupled, differential, partial differential, algebraic and sometimes nonlinear equations. The starting model can be described by the standard state-space form:

$$\begin{cases} \dfrac{dx}{dt} &= f\left[x(t), u(t)\right] \\ y(t) &= g\left[x(t)\right] + b(t) \end{cases} \tag{1}$$

Where x is the state variable vector, y is the output vector, u is the control input vector and b corresponds to the noise vector. The vector functions f and g are known, but they may also be partially known or inaccurate. In black-box neural modeling, functions f and g are approximated during the training step from experimental data. In gray-box neural modeling, those functions are described by their analytical form and implemented as neural models with some fixed parameters. Other unknown parameters are computed during the training step from experimental data.

The discretized equations of the neural model can be written under the canonical form (2), where φ^{NN} corresponds to the transition vector function, Ψ^{NN} represents the output vector function and $b(n)$ is the output noise at time instant n. Since the output noise only appears in the observation equation, it does not have any influence on system

dynamics.

$$\begin{cases} x(n+1) &= \varphi^{NN}[x(n),u(n)] \\ y(n) &= \psi^{NN}[x(n)]+b(n) \end{cases} \tag{2}$$

Figure 3 represents the graphical form of the forward neural state-space model.

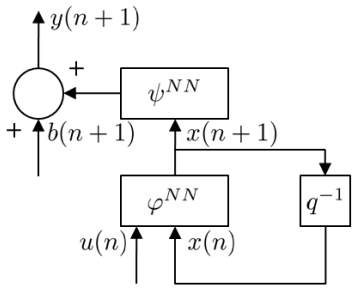

Figure 3: Forward neural state-space model. The q^{-1} operator stands for one T sample time delay.

Similarly, we have carried out the semi-physical INN model by adding an inversion step before the training. The reverse-time equation design has consisted in the expression of $u(n)$ according to the noisy observation vector $y_{obs}(n)$. Then, the state variables at time instant n have been extracted to obtain a new system, according to the state variables at time instant $n+1$.

Consequently, the INN model can be described by the canonical form (3), where φ_I^{NN} corresponds to the reverse-time transition vector function and Ψ_I^{NN} represents the restoring vector function of the input.

$$\begin{cases} x(n) &= \varphi_I^{NN}[x(n+1),y_{obs}(n)] \\ u(n) &= \psi_I^{NN}[x(n+1),y_{obs}(n)] \end{cases} \tag{3}$$

Figure 4 represents the graphical form of the inverse neural state-space model.

4. Inversion of a semi-physical ODE model

In the first part of this section, we obtain the canonical form of the inverse model which refers to (3) in the case of a dynamic system characterized by a r-order ODE. In the second part, we present a study concerning an illustrative second order example.

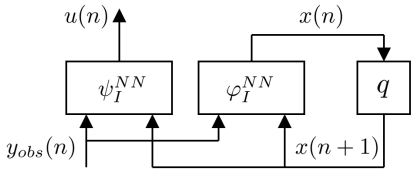

Figure 4: Inverse neural state-space model. The output vector $u(n)$ of this inverse model corresponds to the system input. The q operator stands for one T sample time forward. Practically, the q operator is replaced by the q^{-1} operator which stands for one T sample time delay, and the noisy observation vector $y_{obs}(n)$ is presented in reverse-time at the input of the network to preserve causality.

Some promising results about semi-physical ODE models have already been developed by Bourgois et al. (2007b).

4.1. General case study: an r-order ODE without input derivative

Let us consider a continuous, mono input and mono output system governed by an ordinary differential equation:

$$a_r \frac{d^r y}{dt^r} + a_{r-1} \frac{d^{r-1} y}{dt^{r-1}} + \quad \cdots \quad + a_1 \frac{dy}{dt} + a_0 y \quad = \quad c_1 u(t) \tag{4}$$

The corresponding continuous state-space form is given by:

$$\left\{ \begin{array}{lcl} \dfrac{dx(t)}{dt} & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + b(t) \end{array} \right. \tag{5}$$

And the state-space matrices A, B and C are worth:

$$A \;=\; Comp(P), \quad B^T \;=\; \left[\begin{array}{cccc} 0 & \cdots & 0 & \dfrac{c_1}{a_r} \end{array} \right], \quad C \;=\; \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \end{array} \right]$$

Here, $Comp(P)$ is the companion matrix of the monic polynomial obtained starting from (4) and defined by $P(q) = \dfrac{a_0}{a_r} + \dfrac{a_1}{a_r} q + \cdots + \dfrac{a_{r-1}}{a_r} q^{r-1} + q^r$. By choosing the explicit Euler method and supposing the sampling period T such as $t = nT$, the equation (5) leads to the discrete-time state-space form (6):

$$\left\{ \begin{array}{lcl} \dfrac{x(n+1) - x(n)}{T} & = & Ax(n) + Bu(n) \\ y(n) & = & Cx(n) + b(n) \end{array} \right. \Longleftrightarrow \left\{ \begin{array}{lcl} x(n+1) & = & Fx(n) + Gu(n) \\ y(n) & = & Hx(n) + b(n) \end{array} \right. \tag{6}$$

The new state-space matrices are expressed by $F = TA + I_r$, $G = TB$ and $H = C$. Here, I_r is the identity matrix with $dim(I_r) = dim(F) = r \times r$, $dim(G) = r \times 1$ and $dim(H) = 1 \times r$.

By gathering the equations (A.10) and (A.12) of the demonstration of the Appendix A, we have carried out the reverse-time state-space equation system (7) which fits to the canonical form (3):

$$\left\{ \begin{array}{rcl} x(n) & = & F_I x(n+1) + G_I [y(n) - b(n)] \\ u(n) & = & H_I x(n+1) + I_I [y(n) - b(n)] \end{array} \right. \tag{7}$$

Where the reverse-time state-space matrices are worth:

$$F_I = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{1}{T} & 0 & & \\ \vdots & & \ddots & \\ -\left(-\frac{1}{T}\right)^{r-1} & & \frac{1}{T} & 0 \end{bmatrix}, G_I = \begin{bmatrix} 1 \\ -\frac{1}{T} \\ \vdots \\ \left(-\frac{1}{T}\right)^{r-1} \end{bmatrix}$$

$$H_I = \begin{bmatrix} 0 & \cdots & 0 & \frac{a_r}{Tc_1} \end{bmatrix} + \begin{bmatrix} \frac{a_0}{c_1} & \cdots & \frac{a_{r-2}}{c_1} & \frac{1}{Tc_1}(a_{r-1}T - a_r) \end{bmatrix} F_I$$

$$I_I = \begin{bmatrix} \frac{a_0}{c_1} & \cdots & \frac{a_{r-2}}{c_1} & \frac{1}{Tc_1}(a_{r-1}T - a_r) \end{bmatrix} G_I$$

4.2. Study of a second order ODE model

We have studied the deconvolution problem for a linear model governed by an ordinary differential equation in order to test the method. Let us suppose a system represented by the differential equation:

$$\frac{d^2y}{dt^2} + 2\xi\omega_n \frac{dy}{dt} + \omega_n^2 y = c_1 u(t) \tag{8}$$

This second order ordinary differential equation may be either the representation of a mechanical system (*e.g.* mass, spring, shock absorber, *etc.*) or the representation of an electrical one (*e.g.* RLC filter) excited by a time-dependent input $u(t)$. The damping parameter ξ, the natural pulsation ω_n, and the static gain c_1 are not *a priori* known in this physical model. By referring to the relation (5), the model can be represented by

the following state-space system:

$$\begin{cases} \dfrac{dx(t)}{dt} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ c_1 \end{bmatrix} u(t) \\ \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + b(t) \end{cases} \tag{9}$$

The discrete-time state-space matrices F, G and H of the relation (6) are expressed by:

$$F = \begin{bmatrix} 1 & T \\ -\omega_n^2 T & 1-2\xi\omega_n T \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ Tc_1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

By referring to the system (7), we have finally obtained the inverse state-space model:

$$\begin{cases} x(n) &= \begin{bmatrix} 0 & 0 \\ \dfrac{1}{T} & 0 \end{bmatrix} x(n+1) + \begin{bmatrix} 1 \\ -\dfrac{1}{T} \end{bmatrix} [y(n) - b(n)] \\ \\ u(n) &= \begin{bmatrix} \alpha & \beta \end{bmatrix} x(n+1) + \gamma [y(n) - b(n)] \end{cases} \tag{10}$$

Where the parameters α, β and γ are worth:

$$\alpha = \frac{2\xi\omega_n T - 1}{T^2 c_1}, \quad \beta = \frac{1}{Tc_1}, \quad \gamma = \frac{(\omega_n T)^2 + (1 - 2\xi\omega_n T)}{c_1 T^2}$$

Of course, this non-causal system can be implemented only if the state variables at time instant $n+1$ are known before the calculation of state variables at time instant n. Inverse problems are more familiar with this concept. It is the case during the input sequence restoration at the initial time instant. In the reconstructed input, the observation noise $b(n)$ now appears as a correlated noise and is also amplified by the real γ. Let us underline that the reverse-time system remains stable for any T since the eigenvalues of the state-space matrix are all null for this example. The INN model of the figure 5 is carried out starting from the relation (10). Here, the activation functions f are all linear. Besides, even if the sampling period T is generally known, the physical parameters c_1, ξ and ω_n may be imprecise, or completely unknown. The degrees of freedom may relate to these coefficients.

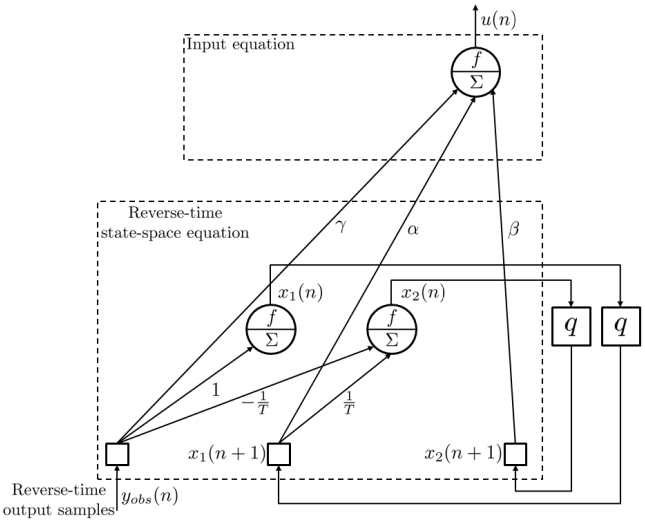

Figure 5: Second order INN model representation. The output vector $u(n)$ of this inverse model corresponds to the system input. The design fits into the reverse-time state-space equation system. The neural network is not fully connected and the activation functions f are linear. The q operator stands for one T sample time forward. Practically, the q operator is replaced by the q^{-1} operator which stands for one T sample time delay, and the noisy observation vector $y_{obs}(n)$ is presented in reverse-time at the input of the network to preserve causality.

5. Study of a Dispersion Model

In this section we test the previous method on an atmospheric pollutant dispersion model governed by a partial differential equation in order to fulfill the pollution source deconvolution and the receptor concentration estimation.

5.1. Atmospheric Pollutant Dispersion Modeling

We develop and increase the results we have obtained in Bourgois et al. (2007a). Let us suppose a system represented by the following PDE (Turner, 1994):

$$\begin{aligned} \frac{\partial x(\overrightarrow{p},t)}{\partial t} &= D(\overrightarrow{p},t)\left(\frac{\partial^2 x(\overrightarrow{p},t)}{\partial \overrightarrow{p}^2}\right) - \overrightarrow{V}(\overrightarrow{p},t)\left(\frac{\partial x(\overrightarrow{p},t)}{\partial \overrightarrow{p}}\right) \\ &- Kx(\overrightarrow{p},t) + \Gamma(x(\overrightarrow{p},t)) + \sum_{i=1}^{n_s} u(s_i,t)\delta(\overrightarrow{p} - \overrightarrow{s_i}) \end{aligned} \tag{11}$$

- $x(\overrightarrow{p},t)$ is the concentration (in $g.m^{-3}$) at a receptor location $\overrightarrow{p} = (p_1, p_2, p_3)$ at time t in the referential $(O, \overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$. It comes from the air dispersion of

n_s pollutant sources of intensity $u(s_i,t)$ at the position $\overrightarrow{s_i} = (s_{(i,1)}, s_{(i,2)}, s_{(i,3)})$, inside a bounded open domain Ω of dimension $l \times L \times H$;

- D is the diffusion tensor (in $m^2.s^{-1}$) defined by its diagonal elements $d_i(\overrightarrow{p},t)$;
- $\overrightarrow{V}(\overrightarrow{p},t) = (v_1(\overrightarrow{p},t), v_2(\overrightarrow{p},t), v_3(\overrightarrow{p},t))^T$ is the wind speed field (in $m.s^{-1}$), responsible for the $3D$ transport;
- K is the reaction coefficient of a first order chemical transformation;
- $\Gamma(x)$ appears when the chemical species presents nonlinear reactions;
- δ represents the Dirac function.

The observatory is configured by a network of n_c sensors at the positions $\overrightarrow{c_i} = (c_{(i,1)}, c_{(i,2)}, c_{(i,3)})$. To simplify the presentation, we have chosen to present the method in the one-dimensional case. By projecting on $O\overrightarrow{i}$, choosing the explicit Euler method and supposing the sampling period T such as $t = nT$ and the spatial sampling step Δ_{p_1} such as $p_1 = k\Delta_{p_1}$, we have obtained the recurrent equation (12):

$$\begin{aligned} x(k,n+1) &= m_1(k,n)x(k+1,n) + m_2(k,n)x(k,n) + m_3(k,n)x(k-1,n) \\ &+ T\Gamma(x(k,n)) + T\sum_{i=1}^{n_s} u(s_i,n)\delta(k - s_{(i,1)}) \end{aligned} \tag{12}$$

Where the parameters $m_1(k,n)$, $m_2(k,n)$ and $m_3(k,n)$ are worth:

$$\begin{cases} m_1(k,n) &= \dfrac{Td_1(k,n)}{(\Delta p_1)^2} - \left(\dfrac{1 - sgn(v_1(k,n))}{2}\right)\left(\dfrac{Tv_1(k,n)}{\Delta p_1}\right) \\ m_2(k,n) &= 1 - KT - sgn(v_1(k,n))\left(\dfrac{Tv_1(k,n)}{\Delta p_1}\right) - \dfrac{2Td_1(k,n)}{(\Delta p_1)^2} \\ m_3(k,n) &= \dfrac{Td_1(k,n)}{(\Delta p_1)^2} + \left(\dfrac{1 + sgn(v_1(k,n))}{2}\right)\dfrac{Tv_1(k,n)}{\Delta p_1} \end{cases}$$

Here, *sgn* corresponds to the sign function. The equation (12) characterizes the deconvolution mask and presents a linear part according to the coefficients $m_1(k,n)$, $m_2(k,n)$ and $m_3(k,n)$. By supposing $M = \left\lfloor \dfrac{l}{\Delta_{p_1}} \right\rfloor + 1$ meshes on one dimension, $x(n) =$

$\left[\begin{array}{cccc} x(1,n) & \cdots & x(M,n) \end{array}\right]^T$ and $u(n) = \left[\begin{array}{cccc} u(s_1,n) & \cdots & u(s_{n_s},n) \end{array}\right]^T$, we have obtained the forward state-space equation (13):

$$x(n+1) \quad = \quad Fx(n) + Gu(n) + T\Gamma(x(n)) \tag{13}$$

The tridiagonal matrix F of size $dim(F) = M \times M$ takes the form:

$$F \quad = \quad \left[\begin{array}{ccccc} m_2(1,n) & m_1(1,n) & 0 & \cdots & 0 \\ m_3(2,n) & m_2(2,n) & m_1(2,n) & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & & & m_1(M-1,n) \\ 0 & \cdots & 0 & m_3(M,n) & m_2(M,n) \end{array}\right]$$

The matrix G of size $dim(G) = M \times n_s$ is worth:

$$G \quad = \quad T \left[\begin{array}{cccc} \delta(1-s_{(1,1)}) & \delta(1-s_{(2,1)}) & \cdots & \delta(1-s_{(n_s,1)}) \\ \delta(2-s_{(1,1)}) & \delta(2-s_{(2,1)}) & & \\ \vdots & & \ddots & \\ \delta(M-s_{(1,1)}) & & & \delta(M-s_{(n_s,1)}) \end{array}\right]$$

Let $y(n) = \left[\begin{array}{cccc} y(1,n) & \cdots & y(n_c,n) \end{array}\right]^T$ and $b(n) = \left[\begin{array}{cccc} b(1,n) & \cdots & b(n_c,n) \end{array}\right]^T$. In equation (6) characterizing the observations, the placing matrix H of the n_c sensors of size $dim(H) = n_c \times M$ is expressed by:

$$H \quad = \quad \left[\begin{array}{cccc} \delta(1-c_{(1,1)}) & \delta(2-c_{(1,1)}) & \cdots & \delta(M-c_{(1,1)}) \\ \delta(1-c_{(2,1)}) & \delta(2-c_{(2,1)}) & & \\ \vdots & & \ddots & \\ \delta(1-c_{(n_c,1)}) & & & \delta(M-c_{(n_c,1)}) \end{array}\right]$$

The term $b(i,n) = b_{mod}(i,n) + b_{mes}(i,n)$ is a random vector, Gaussian centered $b(i,n) \sim \mathcal{N}(0,\sigma^2)$, of unknown variance σ^2, modeling the general uncertainty of the observations. It groups together model errors $b_{mod}(i,n)$ (phenomenon and wind fields uncertainty) and measurement uncertainty $b_{mes}(i,n)$ resulting from sensors or measurement environment.

5.2. Study Assumptions

We have considered a basic mesh to reproduce, constituted by three nodes or neurons. We have supposed there is only one source of flow $u(n)$ in this mesh, at the level of the central node. A sensor is positioned at the level of a lateral node. Wind speed is supposed to be constant in time, and the term of nonlinearity $\Gamma(y)$ is considered to be insignificant. This choice has been done in order to confirm the method in a linear case. Only linear case will be considered in this study. For this basic mesh, the matrices F, G and H are worth:

$$F = \begin{bmatrix} m_2(1,n) & m_1(1,n) & 0 \\ m_3(2,n) & m_2(2,n) & m_1(2,n) \\ 0 & m_3(3,n) & m_2(3,n) \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ T \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$$

The reverse-time equation design has consisted in the expression of the flow $u(n)$ according to the sensor observation. Then, the state variables at time n have been extracted to obtain a new system, according to the state variables at time $n+1$. We have thus carried out the reverse-time state-space equation system (7) where the inverse state-space matrices are expressed by:

$$F_I = \begin{bmatrix} \frac{1}{m_2(1,n)} & 0 & -\frac{m_1(1,n)}{m_2(1,n)m_3(3,n)} \\ 0 & 0 & \frac{1}{m_3(3,n)} \\ 0 & 0 & 0 \end{bmatrix}, \quad H_I = \begin{bmatrix} -\frac{m_3(2,n)}{Tm_2(1,n)} \\ \frac{1}{T} \\ \zeta \end{bmatrix}^T$$

$$G_I = \begin{bmatrix} \frac{m_1(1,n)m_2(3,n)}{m_2(1,n)m_3(3,n)} & -\frac{m_2(3,n)}{m_3(3,n)} & 1 \end{bmatrix}^T, \quad I_I = \frac{\eta - \kappa - \nu}{Tm_2(1,n)m_3(3,n)}$$

The parameters ζ, η, κ and ν are worth:

$$\zeta = \frac{m_1(1,n)m_3(2,n) - m_2(2,n)m_2(1,n)}{Tm_2(1,n)m_3(3,n)}, \qquad \kappa = m_1(2,n)m_2(1,n)m_3(3,n)$$

$$\eta = m_2(1,n)m_2(2,n)m_2(3,n), \qquad \nu = m_1(1,n)m_2(3,n)m_3(2,n)$$

The INN model is then carried out starting from the previous reverse-time state-space equation system (figure 6). But, even if previous results provide accurate coefficients, we do not need them to design the shape of the INN model. One only needs

to know the structure, *i.e.* the location of non-zero values. Indeed, the non-zero coefficients define the remaining connections symbolized by arrows in figure 6. The corresponding weights (degrees of freedom) are then estimated during the training. Here, the activation functions f are linear. However, neural networks have been successfully used to nonlinear dynamic systems modeling. Indeed, the form of the usual nonlinear activation functions (*e.g.* sigmoid activation functions) results in more parsimonious approximation, *i.e.* the same residual error with less number of parameters (Barron, 1993).

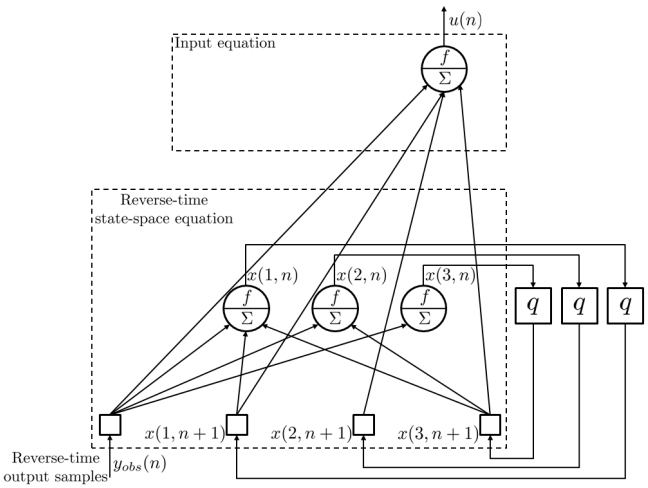

Figure 6: INN model representation of a basic mesh of the discrete-time dispersion model where wind and dispersion parameters remain constant. The output vector $u(n)$ of this inverse model corresponds to the system input. The design fits into the reverse-time state-space equation system. The neural network is not fully connected and the activation functions f are linear. The q operator stands for one T sample time forward. Practically, the q operator is replaced by the q^{-1} operator which stands for one T sample time delay, and the noisy observation vector $y_{obs}(n)$ is presented in reverse-time at the input of the network to preserve causality.

5.3. Study of Causality and Stability

The problem of causality have been raised at two levels:

- During the error calculation associated with each training example and during the recall phase, we have truncated all the sequences by deleting the $r-1$ first

samples because of the unknown initial conditions (r being the system order);

- During numerical simulations, the simulated data have been rearranged before the training to obtain reverse-time sequences (the first element has become the last one, *etc*). The q operator has assumed the role of q^{-1} operator which stands for one T sample time delay to ensure causality is not violated.

This study have led us to treat stability conditions in two times:

- During the training phase, data are simulated starting from the forward state-space model. It has been necessary to check the stability of the simulation model. The stability is ensured if and only if the spectral radius $\rho(F) < 1$;
- On the other hand, it has been advisable to know the behavior of the inverse state-space model in term of stability. The stability is ensured if and only if $\rho(F_I) < 1$.

However, the matrices F and F_I being essentially composed of fixed physical coefficients, the only adjustable parameter is the sampling period T. Thus, for invariant simulation parameters, we have studied the spectral radius evolution of the matrices F and F_I according to T (figure 7).

The inverse state-space model stability zone is totally antagonist with the forward state-space model one. For non-minimum phase system, it is then not possible to find a sampling period which ensures forward and inverse model stability. Consequently, we have chosen a sampling period T such as $\rho(F) < 1$ to ensure the simulation model stability and to remain faithful to the reality. Of course, this choice is unfavorable to the inverse state-space model stability but does not have any influence on the inverse state-space neural model which remains stable.

6. Results

The goal of this section is to check the assumptions of awaited quality concerning the gray-box INN model in term of robustness with respect to an unknown input from the training base, in term of robustness with respect to the noise on the output (*i.e.*

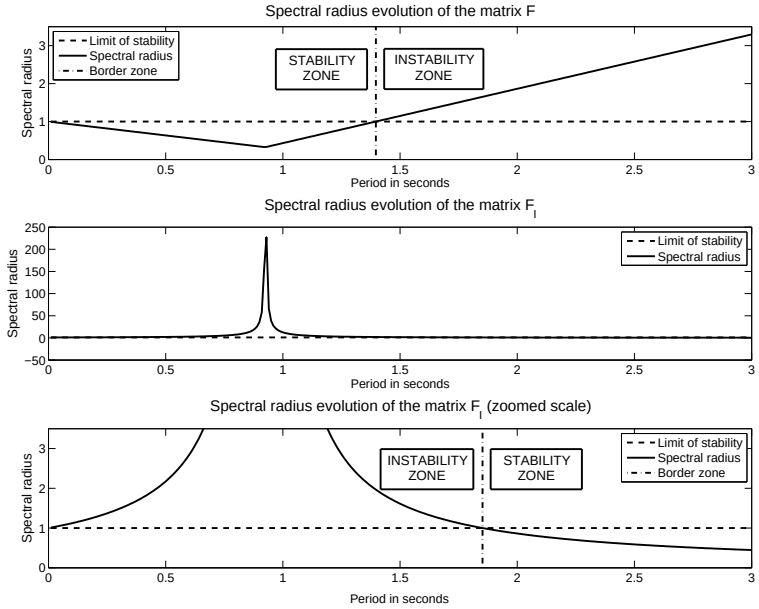

Figure 7: Spectral radius evolution according to T : a) Forward state-space model stability zones, b) & c) Reverse-time state-space model stability zones. The inverse state-space model stability zone is totally antagonist with the forward state-space model one.

the regularizing effect), and in term of gain about the training effort. For that, the semi-physical INN model has been compared to a traditional black-box INN model.

6.1. Networks design

The black-box INN model is a fully connected Elman network. In the case of the ODE model, the network is constituted by two linear neurons on its recurrent layer and one linear neuron on its output layer. For the PDE dispersion model, the recurrent layer possesses three linear neurons. After being randomly initialized, all the synaptic weights and biases are left free during the whole training. Figure 8 represents a classical design of a two layer Elman network. We have called $IW_{i,j}$, the weight matricies connected to inputs and $LW_{i,j}$ weight matrices coming from layer outputs . The subscript indices i and j have been used to identify the source (second index) and the destination (first index) for the various weights. Here, b_1 and b_2 correspond to the

biases.

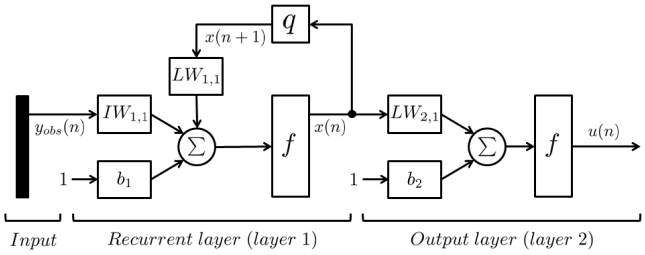

Figure 8: Classical design of a two layer Elman network. We adopt the MATLAB convention for weight matrices and biases. The activation functions f are linear. The q operator stands for one T sample time forward. Practically, the q operator is replaced by the q^{-1} operator which stands for one T sample time delay, and the noisy observation vector $y_{obs}(n)$ is presented in reverse-time at the input of the network to preserve causality.

The gray-box INN model is designed starting from the previous black-box model and modified to obtain the inverse neural structure of figure 5 (ODE case) or figure 6 (PDE case). For that, we have connected the input layer to the output layer, added a delay between the two layers, and some values in the weight matrix $LW_{1,1}$ have been forced to be null to delete corresponding connections. No neuron has been added. The remaining coefficients are left free during the whole training. Figure 9 represents the gray-box network.

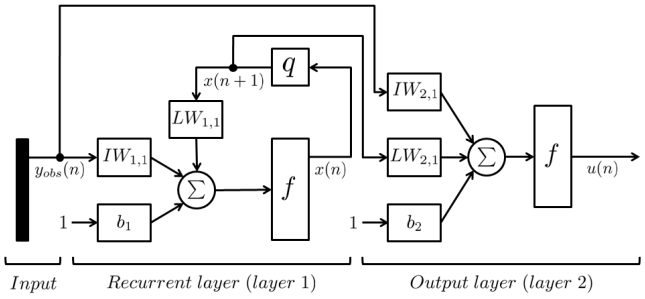

Figure 9: Semi-physical INN model design. We adopt the MATLAB convention for weight matrices and biases. The activation functions f are linear. The q operator stands for one T sample time forward. Practically, the q operator is replaced by the q^{-1} operator which stands for one T sample time delay, and the noisy observation vector $y_{obs}(n)$ is presented in reverse-time at the input of the network to preserve causality.

6.2. Numerical simulations

In the case of the ODE model, we have chosen a damping parameter $\xi = 0.9$, a natural pulsation $\omega_n = 5\ rad.s^{-1}$, a static gain $c_1 = 30$ and a sampling period $T = 0.05$ s. Let us underline that this choice of parameters ensures for the matrix F of the system (6) a spectral radius lower than 1. The forward state-space model stability is then guaranteed. For the PDE dispersion model, we have fixed a spatial sampling step $\Delta p_1 = 5\ m$, a wind speed field such as $v_1(1,n) = 5\ m.s^{-1}$, $v_1(2,n) = 5\ m.s^{-1}$ and $v_1(3,n) = 4\ m.s^{-1}$, a diffusion tensor such as $d_1(1,n) = 1\ m^2.s^{-1}$, $d_1(2,n) = 2\ m^2.s^{-1}$ and $d_1(3,n) = 2\ m^2.s^{-1}$, and a chemical reaction coefficient $K = 0$. For the reasons previously exposed, we have set a sampling period $T = 0.2\ s$, ensuring the simulation model stability. The two INN models have been subjected to a learning with pseudo-experimental noisy data.

To construct the set of training, we have generated four short random input sequences of length $N = 50$ samples. These signals are step functions resulting from the product of an amplitude level A_e by a Gaussian law of average μ_e and variance σ_e^2. The period T_e is adjustable and characterizes the changes of states. By simulating the direct knowledge-based model starting from these input signals, we have obtained four noisy synthetic output signals. The average μ_b, the variance σ_b^2, and the period T_b characterize the noise dynamic. We have fixed $A_e = 1$, $\mu_e = 0$, $\sigma_e^2 = 1$, $\mu_b = 0$ and $T_b = 3T$. Of course, T_e influences the dynamic of the input signals and thus, the dynamic of the noisy synthetic output signals. We have then generated for each input sequence a random value for T_e such as a significant variation of the output signals is visible.

The learning stops if the number of iterations reaches 400 or if the mean squared error (MSE) is lower than 0.001 (ODE case) or 0.005 (PDE case). The error is calculated as the difference between the target output t (the desired input) and the network output $\widehat{t}$ (the estimated input):

$$\text{MSE} \quad = \quad \frac{1}{N}\sum_{k=1}^{N}[t(k) - \widehat{t}(k)]^2 \tag{14}$$

In order to prevent overfitting on the training data, we have memorized all the weight matrices obtained after each epoch with a training signal. We have then kept the weights which give the best performance function. Moreover, early stopping improves

regularization and tends to reduce noise influence, but in this case input restoration errors are more visible at the level of the changes of states (discontinuities). In the worst case, *i.e.* when the minimum MSE value is never reached, the total number of epochs at the end of the training is 1600.

During the test step (recall phase), we have studied the semi-physical contribution in terms of generalization and regularization according to a new test signal. For that, we have generated another long random input sequence of length $N = 400$ samples. The noise variance of the corresponding noisy synthetic output signal is also worth σ_b^2.

To measure the noise influence, we have reproduced the previous protocol for several values of σ_b^2. The signal-to-noise ratio (SNR) of the corresponding synthetic output signals lies between plus infinity (absence of noise) and 10 dB. Sometimes, the back-propagation algorithm may converge to unsatisfactory local minima, and may not be able to find weights that minimize the error during the training phase. This may cause unstable network outputs and high MSE. Consequently, we have chosen to repeat each test one thousand times and to calculate the average performances of the two INN models. Since each test is realized with new random signals, we have used the normalized mean squared error (NMSE):

$$\text{NMSE} \quad = \quad \frac{\text{MSE}}{\frac{1}{N-1}\sum_{k=1}^{N}\left[t(k)-\frac{1}{N}\sum_{k=1}^{N}t(k)\right]^2} \tag{15}$$

Here, the denominator corresponds to the unbiased variance of the desired input.

Moreover, we have also computed for each experiment the percentage of the output variation that is explained by each model:

$$fit \quad = \quad 100\left(1-\frac{\|\widehat{t}-t\|}{\|\overline{t}-t\|}\right) \tag{16}$$

Where $\|.\|$ represents the Euclidean norm and $\overline{t}$ corresponds to the empirical mean of the desired input.

In the case of the second order ODE model, we have also compared the two INN models with two traditional models: the ARMAX model and the Box-Jenkins model. Of course, these second order models have been designed in reserve-time. In order to estimate parameters, we have applied the MATLAB functions *armax* and *bj* which

minimize a robustified quadratic prediction error criterion with the help of an iterative search algorithm. We have followed the same protocol as previously except the fact that the set of training is constructed by generating one long random input sequence of length $N = 400$ samples. Obviously, the parameter estimation is faster than the neural network training and these models are not compared in term of learning effort. The test step is realized with the same long random input sequence of length $N = 400$ samples which has been used to test the INN models. The NMSE and the fit to the data are evaluated by using the previous equations.

6.3. Modeling Errors and Regularizing Effect

Let us remember that all the considered signals evolve in reverse-time. The estimated input signals obtained without noise in the ODE case are shown in figure 10. Obviously, it deals with a particular example which has been randomly chosen among the thousands available. Similarly, figure 11 gathers results with a SNR of 10 dB.

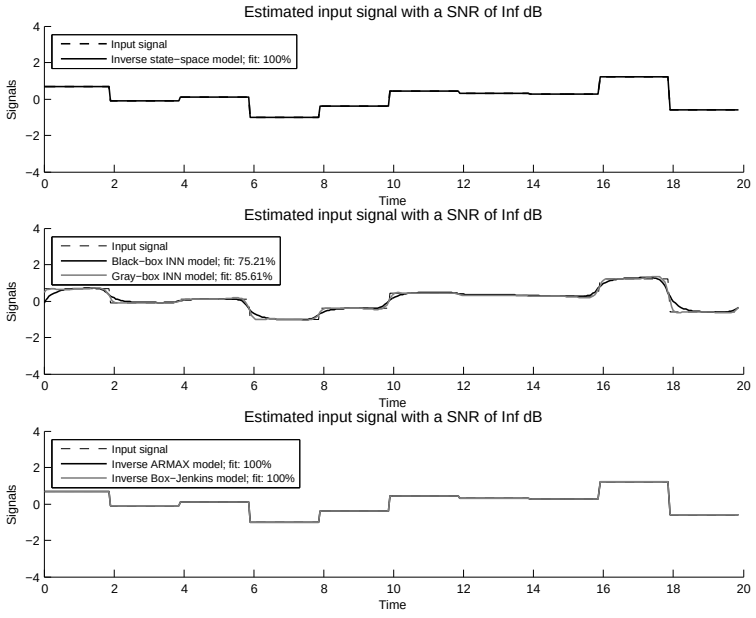

Figure 10: Estimated input signals obtained without noise in the ODE case : a) Inverse state-space model, b) Black-box and gray-box INN models, c) Inverse ARMAX and Box-Jenkins models.

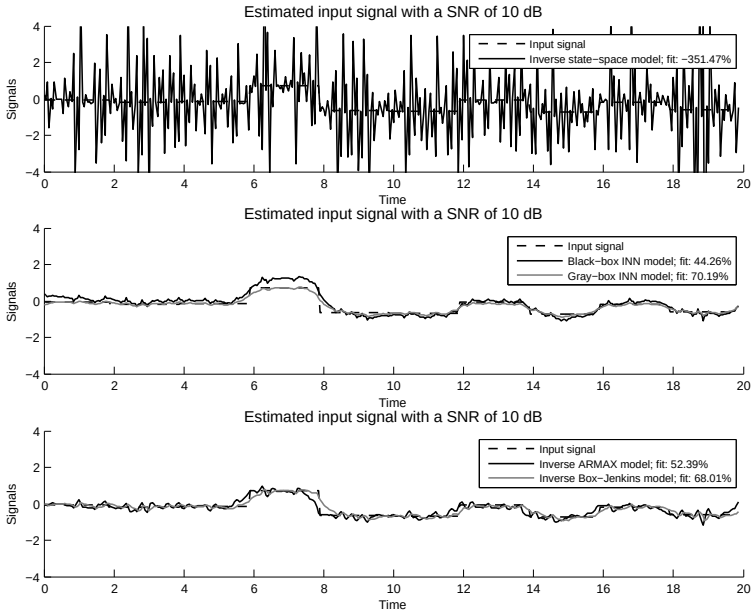

Figure 11: Estimated input signals obtained with a SNR of 10 dB in the ODE case : a) Inverse state-space model, b) Black-box and gray-box INN models, c) Inverse ARMAX and Box-Jenkins models.

Without noise in the training and test sequences, the inverse state-space model provides an excellent fit to the data of 100%. By the same token, the inverse ARMAX and Box-Jenkins models also supply an accurate input restoration (fit: 100%). The semi-physical INN model presents a nearly perfect input signal restoration, except for discontinuous zones (fit: 85.61%). The model does not exactly reproduce the changes of states. The estimated input signal obtained with the black-box INN model is a bit less precise than the gray-box one (fit: 75.21%). With a SNR of 10 dB, the inverse state-space model is largely penalized. Indeed, the noise is amplified and the restoration is incorrect. The black-box INN model also suffers from noisy perturbation and presents a bad fit to the data of 44.26% characterized by a relatively approximative input restoration. Similarly, the inverse ARMAX model provides a naughty fit to the data of 52.39%. For the inverse Box-Jenkins model, restoration errors remain weak and suitable (fit: 68.01%), but there is a slightly noise influence on the estimated input dynamic. It finishes in second position just behind the gray-box INN model which sup-

plies the best fit to the data (fit: 70.19%). For the gray-box model, the noise influence is less visible than for the other models. Table 1 presents results of the average evaluation of the model fit for a signal-to-noise ratio which lies between plus infinity and 10 dB in the ODE case.

	10 dB	20 dB	30 dB	40 dB	∞ dB
Inverse state-space model fit	$< 0\%$	$< 0\%$	61%	88%	100%
Black-box INN model fit	45%	64%	76%	78%	81%
Gray-box INN model fit	71%	79%	81%	82%	86%
Inverse ARMAX model fit	51%	66%	78%	88%	100%
Inverse Box-Jenkins model fit	69%	78%	84%	90%	100%

Table 1: Results of the average evaluation of the model fit according to the SNR in the ODE case.

Figure 12 presents the estimated input signals obtained without noise in the PDE case. Estimated input signals with a SNR of 10 dB are shown in figure 13.

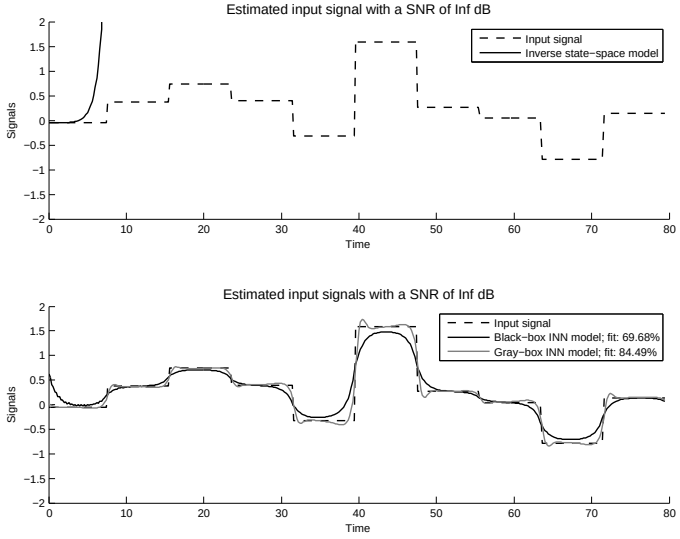

Figure 12: Estimated input signals obtained without noise in the PDE case : a) Inverse state-space model, b) Black-box and gray-box INN models.

Let us bear in mind that in this case, only the forward scheme stability is ensured. Thus without surprise, the inverse state-space model quickly diverges with and without

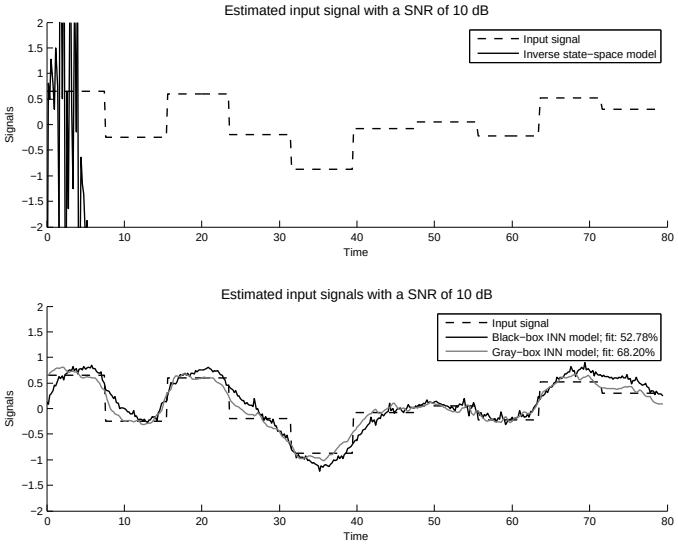

Figure 13: Estimated input signals obtained with a SNR of 10 dB in the PDE case : a) Inverse state-space model, b) Black-box and gray-box INN models.

noise. On the other hand, gray-box and black-box INN models still provide a close fit to the data and supply results which are approximately similar or slightly less good than those obtained in the ODE case. Table 2 presents results of the average evaluation of the model fit for a signal-to-noise ratio which lies between plus infinity and 10 dB in the PDE case.

	10 dB	20 dB	30 dB	40 dB	∞ dB
Black-box INN model fit	51%	59%	61%	66%	73%
Gray-box INN model fit	69%	71%	76%	78%	84%

Table 2: Results of the average evaluation of the model fit according to the SNR in the PDE case.

Figure 14 gathers the average NMSE of the inverse models according to the SNR in the case of the second order ODE model.

Without noise in training and test sequences, the gray-box INN model provides better average performances (NMSE $\simeq 0.02$) than the black-box INN model which is slightly less effective (NMSE $\simeq 0.04$). Of course, the inverse state-space model

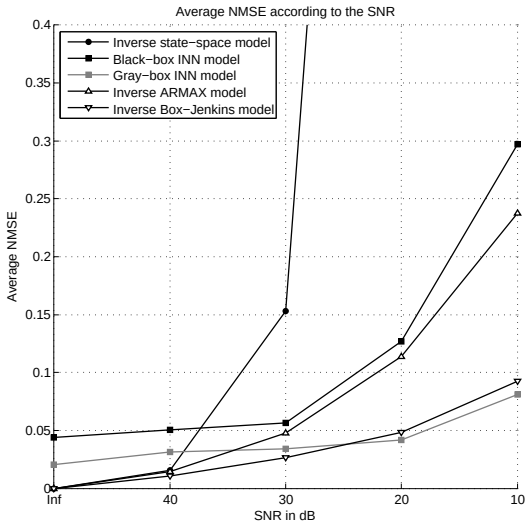

Figure 14: Regularizing effect of the gray-box INN model: average NMSE according to the SNR in the ODE case.

provides accurate results. For the inverse ARMAX and Box-Jenkins models, the average NMSE value tends to be null which means that parameters have been perfectly estimated. When the noise grows, the inverse state-space model is largely penalized, whereas the two INN models are moderately sensitive. The regularizing effect is real. The inverse ARMAX and Box-Jenkins models supply the best performances until about 30 dB. In high noise situation, the black-box INN model and the inverse ARMAX model present approximately the same performances, whereas the gray-box INN model outmatches the inverse Box-Jenkins model and supply the best results in term of robustness with respect to the noise.

As we have previously exposed, the gray-box INN model is achieved by training the weights of a neural network whose structure is constrained by the discrete reverse-time state-space equations. These synaptic weights are adjusted using the backpropagation (gradient descent) iterative procedure whose stopping criterion is defined by a specified error threshold and a predetermined maximum number of iterations. Moreover, regularization by early stopping is also used to avoid the risk of overfitting. Consequently, the weights are determined before they have fully converged and differ from

the coefficients of the reverse-time state-space equations, but ensure the regularization objective. When $\sigma_b^2 = 0$, it is obvious to not accurately retrieve the input signal since the coefficients are slightly different. The average NMSE is then nonzero. On the other hand, in high noise situation, the gray-box INN model is more regularizing than other models. Indeed, the gray-box INN model is more stable after training than the inverse state-space model, which is conform to our expectations. This consists in increasing the sampling period (see figure 7) or decreasing the cutoff frequencies of the inverse model. These cutoff frequencies are then lower than those obtained by identifying the ARMAX or Box-Jenkins models, which look for a perfect fit. These two conventional models are then more sensitive to the measurement noise than the gray-box INN model.

For the PDE dispersion model, the evolution of the NMSE according to the SNR is represented by figure 15.

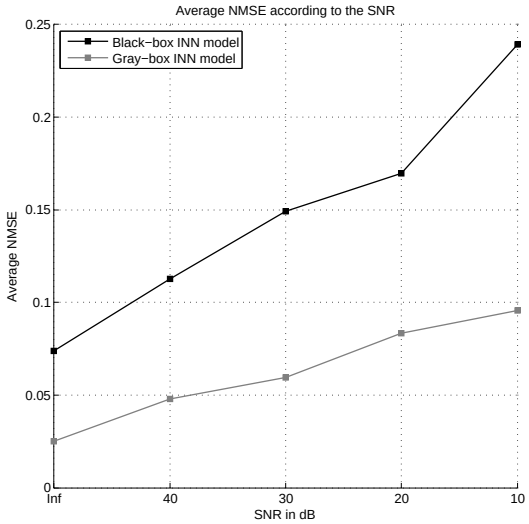

Figure 15: Regularizing effect of the gray-box INN model: average NMSE according to the SNR in the PDE case.

The semi-physical INN model again provides best average performances without noise (NMSE $\simeq 0.03$). Indeed, the black-box neural model is slightly less effective (NMSE $\simeq 0.07$). Since the inverse state-space model fastly diverges, we do not compare its average performance. When the noise grows, the two INN models are mod-

erately sensitive due to the regularizing effect. In addition, having chosen a sampling period T such as $\rho(F_I) < 1$ does not interfere with the INN model. In high noise situation, the two inverse neural models keep the same tendencies.

6.4. Learning Effort

We have compared the product of the NMSE by the number of epochs, *i.e.* the final error amplified by the number of iterations of the training phase. The results obtained with the ODE model are illustrated figure 16, whereas figure 17 gathers those obtained in the case of the PDE dispersion model.

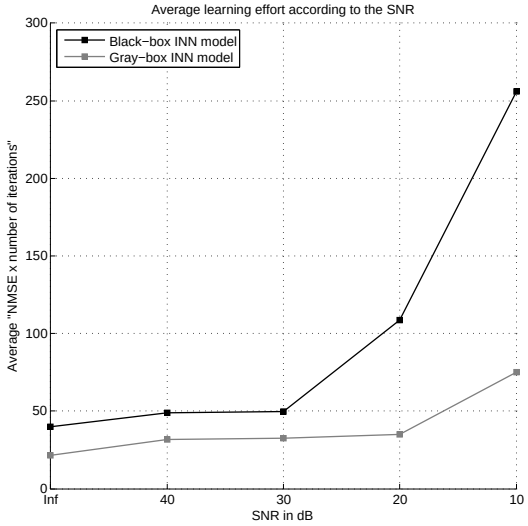

Figure 16: Average learning effort according to the SNR in the ODE case. The learning effort is defined as the product of the NMSE by the number of epochs, *i.e.* the final error amplified by the number of iterations of the training phase.

We note that the gray-box INN model is more effective in term of gain about the training effort in both slight and high noise situation than the black-box INN model. Physical knowledge favors the convergence of the weights so that the behavior approaches the data. The black-box INN model is largely penalized because of its lesser capacity of regularization.

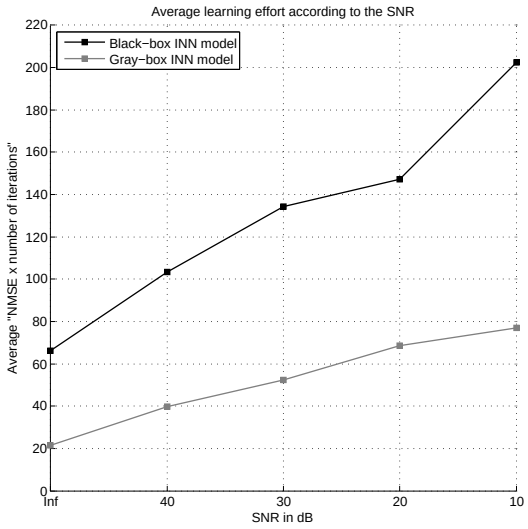

Figure 17: Average learning effort according to the SNR in the PDE case. The learning effort is defined as the product of the NMSE by the number of epochs, *i.e.* the final error amplified by the number of iterations of the training phase.

7. Conclusion

We have proposed an approach to realize an inverse dynamic model resulting from the fusion of statistical training and deterministic modeling. We have chosen to carry out this inverse semi-physical model starting from a recurrent neural network to exploit typical properties of neural algorithms. Indeed, experimental results have shown that neural learning plays the part of statistical regressor and regularization operator. Moreover, input restoration errors are weak. In order to evaluate the semi-physical contribution, the gray-box INN model has been compared with a traditional black-box INN model, with an inverse ARMAX model and with an inverse Box-Jenkins model. The tests realized on a dynamic system characterized by an ODE and on a basic mesh of an atmospheric pollutant dispersion model governed by a PDE have reveal that the semi-physical INN model is more parsimonious than the black-box INN model and presents better performances in term of robustness with respect to the noise than the inverse ARMAX and Box-Jenkins models. Besides, gray-box modeling provides better performances in term of training effort than black-box modeling due to the knowledge

introduced by the deterministic model.

Appendix A. Reverse-time state-space equations system

By considering the relation (6), we have obtained:

$$\begin{cases} \dfrac{x(n+1)-x(n)}{T} &= Ax(n)+Bu(n) \\ y(n) &= Cx(n)+b(n) \end{cases} \tag{A.1}$$

Let us split the matrix A of (A.1) in two parts and let us write:

$$\begin{cases} \dfrac{x(n+1)-x(n)}{T} &= \begin{bmatrix} \overline{A_{r-1}} \\ \underline{A_1} \end{bmatrix} x(n) + \begin{bmatrix} \overline{B_{r-1}} \\ \underline{B_1} \end{bmatrix} u(n) \\ y(n)-b(n) &= Cx(n) \end{cases} \tag{A.2}$$

Where $\overline{A_k}$ (respectively $\overline{B_k}$) is constituted by the k first lines of A (respectively B), and $\underline{A_k}$ (respectively $\underline{B_k}$) is constituted by the k last lines of A (respectively B).

By setting $x(n) = \begin{bmatrix} x_1(n) & x_2(n) & \cdots & x_r(n) \end{bmatrix}^T = \begin{bmatrix} \overline{x_{r-1}}(n) \\ \underline{x_1}(n) \end{bmatrix}$ in (A.2), we have obtained:

$$\begin{bmatrix} \overline{A_{r-1}} \\ \underline{A_1} \end{bmatrix} x(n) = \frac{1}{T}\begin{bmatrix} \overline{x_{r-1}}(n+1) \\ \underline{x_1}(n+1) \end{bmatrix} - \frac{1}{T}\begin{bmatrix} \overline{x_{r-1}}(n) \\ \underline{x_1}(n) \end{bmatrix} - \begin{bmatrix} \overline{B_{r-1}} \\ \underline{B_1} \end{bmatrix} u(n) \tag{A.3}$$

$$x_1(n) = [y(n)-b(n)] \tag{A.4}$$

By remarking that $\overline{A_{r-1}}x(n) = \underline{x_{r-1}}(n)$ and separating (A.3), we have obtained:

$$\underline{x_{r-1}}(n) = \frac{1}{T}\overline{x_{r-1}}(n+1) - \frac{1}{T}\overline{x_{r-1}}(n) \tag{A.5}$$

$$\underline{A_1}x(n) = \frac{1}{T}\underline{x_1}(n+1) - \frac{1}{T}\underline{x_1}(n) - \frac{c_1}{a_r}u(n) \tag{A.6}$$

By concatening (A.4) and (A.5), we have expressed:

$$\begin{bmatrix} x_1(n) \\ \underline{x_{r-1}}(n) \end{bmatrix} = \frac{1}{T}\begin{bmatrix} T\,[y(n)-b(n)] \\ \overline{x_{r-1}}(n+1) \end{bmatrix} - \frac{1}{T}\begin{bmatrix} 0 \\ \overline{x_{r-1}}(n) \end{bmatrix} \tag{A.7}$$

By setting $\boxminus_{r-1} = \begin{bmatrix} T\,[y(n)-b(n)] \\ \overline{x_{r-1}}(n+1) \end{bmatrix}$ in (A.7), we have written more concisely:

$$x(n) \quad = \quad \frac{1}{T}\boxminus_{r-1} - \frac{1}{T}\begin{bmatrix} 0 \\ \overline{x_{r-1}}(n) \end{bmatrix} \tag{A.8}$$

By using a recursive decomposition of (A.8), we have obtained:

$$x(n) \quad = \quad \frac{1}{T}\boxminus_{r-1} - \frac{1}{T}\begin{bmatrix} 0 \\ \frac{1}{T}\boxminus_{r-2} - \frac{1}{T}\begin{bmatrix} 0 \\ \overline{x_{r-2}}(n) \end{bmatrix} \end{bmatrix}$$

And we have finally expressed:

$$x(n) \quad = \quad \frac{1}{T}\boxminus_{r-1} - \frac{1}{T}\begin{bmatrix} 0 \\ \frac{1}{T}\boxminus_{r-2} - \frac{1}{T}\begin{bmatrix} \ddots & \\ & 0 \\ & \frac{1}{T}\boxminus_{1} - \frac{1}{T}\begin{bmatrix} 0 \\ \overline{x_1}(n) \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{A.9}$$

By expanding the expression (A.9), we have obtained:

$$x(n) \quad = \quad -\sum_{i=1}^{r-1}\left(-\frac{1}{T}\right)^{i}\begin{bmatrix} 0 \\ \overline{x_{r-i}}(n+1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\frac{1}{T} \\ \vdots \\ \left(-\frac{1}{T}\right)^{r-1} \end{bmatrix}[y(n)-b(n)]$$

We have thus carried out the reverse-time state-space equation (A.10), where the state-space matrices F_I and G_I depend on the sampling period T:

$$x(n) \quad = \quad F_I x(n+1) + G_I\,[y(n)-b(n)] \tag{A.10}$$

The lower triangular matrix F_I of size $dim(F_I) = r \times r$ and the matrix G_I of size

$dim(G_I) = r \times 1$ are worth:

$$F_I = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{1}{T} & 0 & & \\ \vdots & & \ddots & \\ -\left(-\frac{1}{T}\right)^{r-1} & & \frac{1}{T} & 0 \end{bmatrix}, G_I = \begin{bmatrix} 1 \\ -\frac{1}{T} \\ \vdots \\ \left(-\frac{1}{T}\right)^{r-1} \end{bmatrix}$$

In addition, the relation (A.6) allows us to write:

$$u(n) = \frac{a_r}{Tc_1}\underline{x_1}(n+1) - \frac{a_r}{c_1}\underline{A_1}x(n) - \frac{a_r}{Tc_1}\underline{x_1}(n)$$

By simplifying, we have obtained (A.11):

$$u(n) = \begin{bmatrix} 0 & \cdots & 0 & \frac{a_r}{Tc_1} \end{bmatrix} x(n+1) - \frac{a_r}{c_1}\left[\underline{A_1} + \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{T} \end{bmatrix}\right] x(n) \tag{A.11}$$

By incorporating relation (A.10) in (A.11), we have designed the reverse-time state-space equation (A.12), where the state-space matrices H_I and I_I also depend on the sampling period T:

$$u(n) = H_I x(n+1) + I_I [y(n) - b(n)] \tag{A.12}$$

The matrix H_I of size $dim(H_I) = 1 \times r$ is expressed by (A.13):

$$H_I = \begin{bmatrix} 0 & \cdots & 0 & \frac{a_r}{Tc_1} \end{bmatrix} + \begin{bmatrix} \frac{a_0}{c_1} & \cdots & \frac{a_{r-2}}{c_1} & \frac{1}{Tc_1}(a_{r-1}T - a_r) \end{bmatrix} F_I \tag{A.13}$$

The matrix I_I of size $dim(I_I) = 1 \times 1$, is given by (A.14):

$$I_I = \begin{bmatrix} \frac{a_0}{c_1} & \cdots & \frac{a_{r-2}}{c_1} & \frac{1}{Tc_1}(a_{r-1}T - a_r) \end{bmatrix} G_I \tag{A.14}$$

With the help of the equations (A.10) and (A.12), we have thus carried out the reverse-time state-space equation system which corresponds to the canonical form (3).

References

Barron, A. (1993). Universal approximation bounds for superposition of a sigmoidal function. *IEEE Transactions on Information Theory*, *39*, 930-945.

Beghi, A., Liberati, M., Mezzalira, S., & Peron, S. (2007). Grey-box modeling of a motorcycle shock absorber for virtual prototyping applications. *Simulation Modelling Practice and Theory*, *15*, 894-907.

Bourgois, L., Roussel, G., & Benjelloun, M. (2007a). Inversion of a semi-physical dispersion model. In *Proceedings of the 3rd IFAC Workshop on Advanced Fuzzy and Neural Control*. Valenciennes, France.

Bourgois, L., Roussel, G., & Benjelloun, M. (2007b). Inversion of a semi-physical ODE model. In *Proceedings of the 4th International Conference on Informatics in Control, Automaton and Robotics*. Angers, France.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). *Time series analysis, forecasting and control (3rd edition)*. Prentice Hall.

Cherkassky, V., Krasnopolsky, V. M., Solomatine, D. P., & Valdes, J. (2006). Computational intelligence in earth sciences and environmental applications: issues and challenges. *Neural Networks*, *19*, 113-121.

Demoment, G. (1989). Image reconstruction and restoration: overview of common estimation structures and problems. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*, 2024-2036.

Duarte, B., Saraiva, P. M., & Pantelides, C. C. (2004). Combined mechanistic and empirical modelling. *International Journal of Chemical Reactor Engineering*, *2*.

Groetsch, C. W. (1993). *Inverse problems in the mathematical sciences*. Vieweg.

Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, *19*, 122-134.

Krasnopolsky, V. M., & Schillerb, H. (2003). Some neural network applications in environmental sciences. Part I. Forward and inverse problems in geophysical remote measurements. *Neural Networks*, *16*, 321-334.

Leifsson, L. P., Sævarsdòttir, H., Sigurösson, S. P., & Vésteinsson, A. (2008). Grey-box modeling of an ocean vessel for operational optimization. *Simulation Modelling Practice and Theory*, *16*, 923-932.

Lindskog, P., & Ljung, L. (1995). Tools for semi-physical modeling. *International Journal of Adaptive Control and Signal Processing*, *9*, 509-523.

Ljung, L. (1999). *System identification: theory for the user (2nd edition)*. Prentice Hall.

MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation*, *4*, 415-447.

Marroquin, J., Mitter, S., & Poggio, T. (1987). Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, *82*, 76-89.

Mohammad-Djafari, A., Giovannelli, J. F., Demoment, G., & Idier, J. (2002). Regularization, maximum entropy and probabilistic methods in mass spectrometry data processing problems. *International Journal of Mass Spectrometry*, *215*, 175-193.

Nelles, O. (2001). *Nonlinear system identification*. Springer.

Oussar, Y., & Dreyfus, G. (2001). How to be a gray-box: dynamic semi-physical modeling. *Neural Networks*, *14*, 1161-1172.

Ploix, J. L., & Dreyfus, G. (1997). Early fault detection in a dillatation column: an industrial application of knowledge-based neural modeling. In B. Kappen, & S. Gielen (Eds.), *Neural Networks: best practice in Europe*. World Scientific.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., Hjalmarsson, H., & Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, *31*, 1691-1724.

Sontag, E. D. (1997). Recurrent neural networks: some systems-theoretic aspects. In M. Karny, K. Warwick, & V. Kurkova (Eds.), *Dealing with Complexity: a neural network approach* (pp. 1-12). London: Springer.

Tarantola, A. (1987). *Inverse problem theory, methods for fitting and model parameter estimation*. Elsevier.

Thikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. John Wiley and Sons.

Turner, D. B. (1994). *Workbook of atmospheric dispersion estimates: an introduction to dispersion modeling*. CRC Press.

Zorzetto, L. F. M., Filho, R. M., & Wolf-Maciel, M. R. (2000). Processing modelling development through artificial neural networks and hybrid models. *Computers and Chemical Engineering*, *24*, 1355-1360.