Modélisation d'un processus dynamique à l'aide de réseaux de neurones bouclés. Application à la modélisation de la relation pluie-hauteur d'eau dans un réseau d'assainissement et à la détection de défaillance de capteur.

Modeling of a dynamic process with recurrent (feedback) neural networks. Application to the modeling of the rainfall-water height relationship in an urban sewer system and to sensor failure detection.

Pierre ROUSSEL, Frédérique MONCET, Bruno BARRIEU
Laboratoire d'Electronique
ESPCI 10, rue Vauquelin F75005 PARIS

Arnaud VIOLA
Direction des Eaux et de l'Assainissement Seine Saint-Denis
BP 193 93003 BOBIGNY cedex

RÉSUMÉ

Depuis plusieurs années, des mesures systématiques de pluie et de hauteur d'eau dans les canalisations d'un système d'assainissement sont effectuées dans un département français. Cependant des défaillances possibles des capteurs rendent nécessaire une validation de ces mesures.
Nous avons réalisé un modèle "boîte noire", à l'aide de réseaux de neurones, de la relation pluie-hauteur d'eau dans un bassin versant simple équipé d'un capteur de pluie et d'un capteur de hauteur, afin de nous assurer de la validité des mesures à partir de la comparaison des valeurs de hauteur mesurées et des valeurs prédites par le modèle.

ABSTRACT

Systematic measurements of the rainfall as well as of the resulting water height, flow and velocity in the pipes, have been performed for several years in the sewer system of a French department. Due to possible electrical failures or slow sensor drifts, a validation procedure must be performed in order to guarantee the validity of the stored measurements.
To this end, we have developed a neural dynamic black-box model of the rainfall-water height relationship on a simple urban catchment equipped with a single water gauge and a single rainfall gauge. The validity of the measurements is assessed from the comparison between the measured heights and the corresponding values predicted by the model.

KEYWORDS

Process modeling, Recurrent neural networks, Sewer system, Sensor failure detection.

INTRODUCTION

Neural networks have been used for several years with significant success in order to model nonlinear processes. This is due to their ability to approximate any nonlinear function with a relatively small number of parameters. The neural modeling of static processes is fairly straightforward; by contrast, the modeling of dynamic processes with recurrent neural networks requires more care. It has been shown by Nerrand et al. [Nerrand94] that the desirable training procedure depends on how disturbances interact with the process; this is summarized in section 1. Measurements of the rainfall and water height in the pipe of a sewer system show that the relation between these quantities is nonlinear and dynamic. Therefore, we thus have used feedback neural networks in order to model this relationship, using the framework presented in section 1. The results obtained are presented in section 2, and two statistical characteristics of the modeling error, i.e. of the difference between measured and predicted heights are estimated. When a sensor is out of order, it can either output a constant value; or add a slow drift to the water height. We show in section 3 that these failures lead to changes that can be detected from the statistical characteristics of modeling errors obtained from different dynamic models.

1. MODELING DYNAMIC PROCESSES WITH FEEDBACK NEURAL NETWORKS

Neural networks are parsimonious approximators, i.e. they can approximate any nonlinear function with the required accuracy using a smaller number of parameters than nonlinear models that are linear with respect to their parameters, such as polynomial models for instance.

The output y_i of a single neuron is equal to a nonlinear function f of the weighted sum of its inputs x_j: $y_i = f\left(\sum_j c_{ij} x_j\right)$, where f will be a *tanh* in the following. The models used in this paper are networks with a single hidden layer: the inputs are connected to the neurons of the hidden layer and all neurons of that layer are connected to a single linear output neuron. The values of the parameters of the network are computed by an iterative algorithm that minimizes a cost function based on the squared modeling errors.

When dealing with dynamic processes, one makes the assumption that the output of the process can be expressed as a function of past values of the output. As a consequence, either *measured* outputs (output of the process) or *predicted* outputs (output of the model) may be used as inputs of the model. The optimal model can be derived from an assumption on how disturbances interact with the process, as illustrated below.

We consider a dynamic, stochastic process, i.e. a process whose output h at time k does not depend only on input r at time k, but also on earlier values, and whose output cannot be computed exactly solely from the external inputs up to time k.

We want to predict the value of the output at time k from the values of the controllable inputs and from the past values of the output of the process. . In such applications as water height prediction, the output does not depend only on controllable inputs (here, the water height does not depend on any controllable input since rainfall cannot be controlled), and it is also subject to disturbances. Measured disturbances (such as rainfall) can be considered as non-controllable inputs, whereas non-measured disturbances, whose effect can only be observed on the output, are modeled from a sequence of iid random variables $w(k)$.

We make the assumption that the output of the process at time k is related, through an unknown relationship, to present and past values of the input, and to past values of the output. Moreover, since the output of the process cannot be computed exactly from these values, the unpredictable part of the output value is modeled as the realization of the variable $w(k)$. Thus, the best prediction at time k will lead to a prediction error equal to the realization of $w(k)$. This prediction will be obtained only if we are able to model exactly the predictable part of the process.

Let us illustrate this on some examples.

If we assume that the behavior of the process can be appropriately described by
$h(k) = \square[h(k\text{-}1),..., h(k\text{-}n), r(k), ..., r(k\text{-}m)]+ w(k)$
and if we denote by $h_m(k)$ the *predicted* output of the model, the best predictive model will be such that
$h(k) - h_m(k) = w(k)$
Then the best predictive model is
$h_m(k) = \square[h(k\text{-}1),..., h(k\text{-}n), r(k), ..., r(k\text{-}m)]$ [1]
Thus, $h(k)$ can be predicted with a minimal prediction error equal to $w(k)$ if a *feedforward* network, whose inputs are $h(k\text{-}1),..., h(k\text{-}n), r(k), ..., r(k\text{-}m)$ can be trained to realize function $\square$. Note that relation [1] is an *algebraic* equation: the model is a feedforward network whose inputs are *measured* values of the output and external inputs.

If we now assume that the process is subject to measurement noise, it can be described by
$g(k) = \square[g(k\text{-}1),..., g(k\text{-}n), r(k), ..., r(k\text{-}m)]$
$h(k) = g((k))+ w(k)$
and the best predictive model is
$h_m(k) = g((k) = \square[h_m (k\text{-}1),..., h_m (k\text{-}n), r(k), ..., r(k\text{-}m)]$ [2]

Note that relation [2] is a *recurrent* equation: the model is a recurrent (or feedback) network whose inputs are past outputs *predicted* by the model (feedback inputs) and external inputs, whereas model [1] requires past *measured* values of the output.

This approach is directly related to well known linear models : model [1] corresponds to the equation-error model, and model [2] to the output-error model [Ljung87].

The most general blackbox linear model is known as the ARMAX model, which can be transposed as well to the nonlinear NARMAX model [Léontaritis85]:
If we assume that the process can be appropriately described by
$h(k) = \square[h(k\text{-}1),..., h(k\text{-}n), r(k), ..., r(k\text{-}m), w(k), ..., w(k\text{-}p)]+ w(k)$
the best predictive model is given by
$h_m(k) = \square[h_m (k\text{-}1),..., h_m (k\text{-}n), r(k), ..., r(k\text{-}m), h(k\text{-}1) - h_m (k\text{-}1),..., h(k\text{-}p) - h_m (k\text{-}p)]$ [3]

Thus, depending on the assumptions made on the disturbances, various structures of the predictive model arise, leading to corresponding structures of the neural networks. The most general structure used is illustrated on Figure 1.

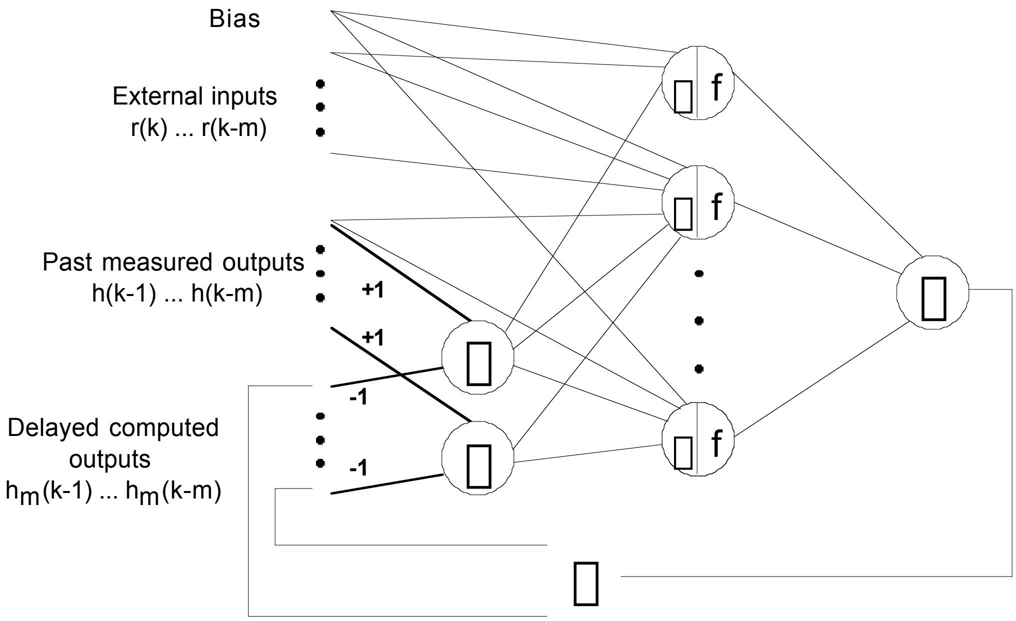

Figure 1. Structure of a neural NARMAX model

2. MODELING THE RAINFALL-WATER HEIGHT RELATIONSHIP

The « Direction de l'Eau et de l'Assainissement » of the French department of Seine Saint-Denis has performed systematic measurements of the rainfalls and of the water heights in the sewer system. Due to sensor failures, these measurements can be erroneous; in order to perform the validation of these measurements automatically, we have developed a black-box model of a simple urban catchment equipped with a single water gauge and a single pipe.

If we consider that the process can be appropriately described by an underlying stochastic model, the best possible modeling error will be obtained if i) the structure (as defined above) of the neural network matches that of the underlying model, ii) the neural network can be made, through training, to approximate function □ with an error that is small as compared to $w(k)$. Under normal operation of the sensors, this minimal modeling error is equal to $w(k)$ and can thus be characterized by a zero mean and an autocorrelation function equal to a Dirac function, since $w(k)$ is independent of the past realizations of w.

Choice of the inputs

As presented above, two kinds of inputs must be considered in the underlying model of the process : the external inputs (the rainfall), and (depending on the disturbances), either past *predicted* water heights or past *measured* water heights. For each of these inputs, one must determine the number of delayed values to be taken into account (parameters n and m). Furthermore, the water height responds to a rainfall with a pure delay □ which must also be taken into account.

The values of these parameters were estimated from the measured values: i) we have considered short rainfalls as being equivalent to a Dirac function, and we have estimated □ to be between 10 and 30 minutes, depending of the value of the waterfall, ii) would the

process be linear, the response could be considered as that of a second-order process. The measurement sampling period is one minute, leading to too large a number of inputs. Furthermore, data are available at irregular time intervals. We thus have chosen to sum the waterfalls on 10-minute intervals from *k*-10 to *k*-59, i.e. 5 external inputs, and 2 feedback inputs to account for the second-order behavior.
Since the effect of disturbances on the process is by no means clear, each of the above predictor structures were tested..

Choice of the training set

Once the structure of the neural network is defined, the values of the parameters are estimated by minimization of a cost function *J*, equal to the sum of squares of the modeling errors on a training set. The neural network will be a good predictor if the training set is representative of the conditions of use of this predictor. We thus have chosen a training period containing various types of rainfalls. The performance of the predictor is then estimated on a test set, consisting of another period of the year. In order to minimize efficiently the training function, we have used a second order iterative algorithm, the Levenberg-Marquardt algorithm. At each step, the coefficients are modified according to the expression $\Delta C = -(\mu I + H)^{-1} \nabla J$, where H is the second derivative matrix of the training function with respect to the coefficients, ∇J is the gradient of J and μ is a coefficient [Bishop95].

Results

The procedure described above has been used to model the rainfall-water height relationship with the following ingredients : i) the training set extended on 3 days, ii) the optimal number of neurons on the hidden layer was found equal to two, iii) the data set was completed by linear interpolation so as to get an output value every minute (output values are recorded only on given conditions by the measurement apparatus and data are not available every minute from these measurements). The modeling error was characterized by its daily average value and by its correlation function on a one-minute interval R(1) estimated on one day as $R(1) = \sum \left(e(t).e(t-1)\right) \Big/ \sum e(t)^2$.

The results are shown on figures 2 and 3 for model [2] (feedback network) and model [3] (NARMAX network). It appears readily that model [3] outperforms model [2] regarding both the daily average of the prediction error and the characteristics of the correlation function value R(1).
Model [2] yields an average error between +2cm and –2cm on the test set, and R(1) remains close to 1. Model [3] exhibits an average error of the order of 0.01 cm and R(1) varies around 1 or 0.2. The high values of R(1) corresponds to situations where the water height remains constant because it does not rain. Thus the modeling error is constant and e(t) = e(t-1), resulting in a value of R(1) equal to 1.

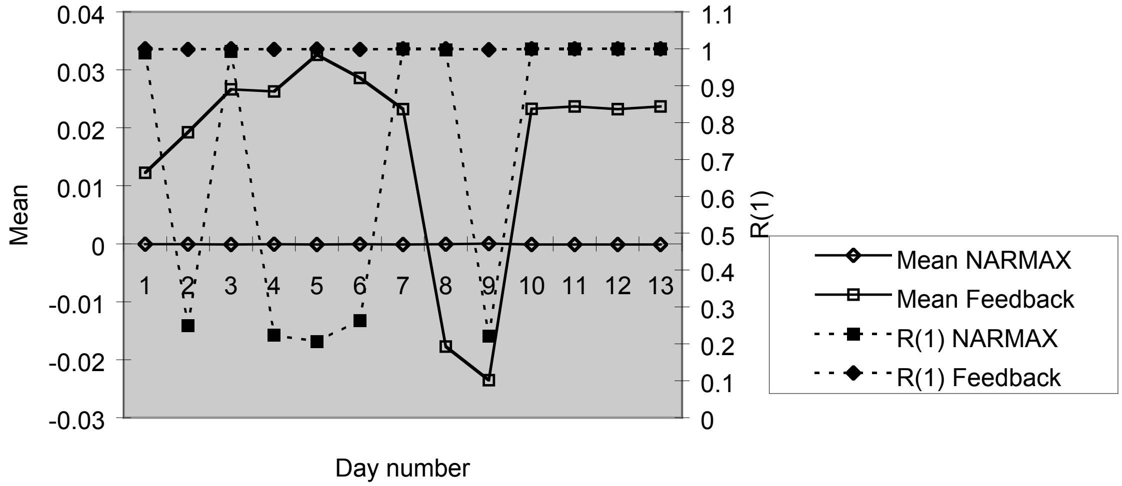

Figure 2. Mean and correlation R(1) of the modeling error for the NARMAX and feedback models.

3. SENSOR FAILURE DETECTION

Sensors may exhibit two types of failure : either they output a constant value, or they add a slow drift to the water height.
In the following, we investigate the influence of erroneous values on the predictions of the model.
We assume that the optimal predictor can be written as a NARMAX model
$h_m(k) = \square[\, h(k\text{-}1),\, h_m\,(k\text{-}1),\, r(k)]$ [4]
and that the prediction error under normal conditions is equal to $w(k)$.
Thus the output of the process is equal to $h(k) = h_m(k) + w(k)$.
We denote the measured output by $h^e(k)$.
The error, i.e. the difference between the predicted output and the measured output, can now be written as $h^e(k)$ - $h_m{}^e(k)$. It differs from that observed under normal conditions in two respects: i) the measured output $h^e(k)$ is erroneous and differs from the true value of the measured quantity; this difference, $h(k)$- $h^e(k)$, is equal to the true value minus a constant or minus a drift depending of the nature of the failure; ii) the predicted output $h_m{}^e(k)$ is computed from an erroneous measured value, $h_m{}^e(k) = \square[\, h^e(k\text{-}1),\, h_m{}^e(k\text{-}1),\, r(k)]$.

This results in changes in the statistical characteristics of the error.
When the sensor outputs a constant value, the average value of the error remains low, but the value of the autocorrelation function R(1) increases significantly, as can be seen on Figure 3.
When the sensor exhibits a slow drift (1cm per day), neither characteristics is significantly modified: the daily average error increases as fast as the drift, as shown on Figure 4. This is due to the facts that the drift is small between two successive measurements, and that the predicted output at time *t* is computed from the measured, erroneous value, at time *t*-1.
In order to fully exhibit this drift, the solution consists in using the feedback model: with this model, the measured output is not used as an input to the neural network, and thus the drift accumulates in the erroneous measured output and not in the modeled output. Despite the fact that the average modeling error is higher with the feedback model than with the

feedforward model, this increasing drift appears clearly on the averaged error, as illustrated on Figure 5.

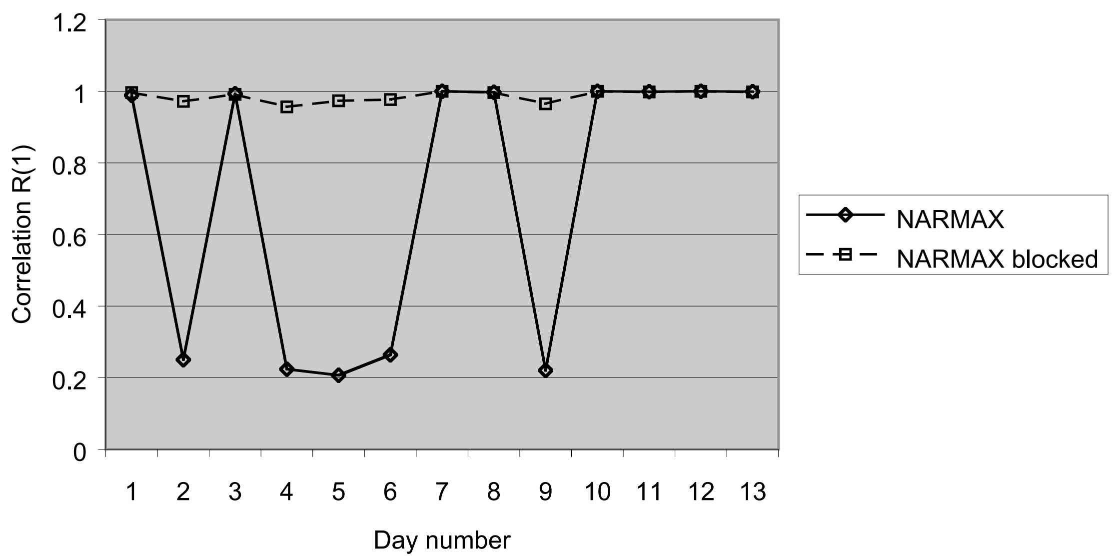

Figure 3. Autocorrelation R(1) of the modeling error for the NARMAX model under normal operation and when the sensor is blocked.

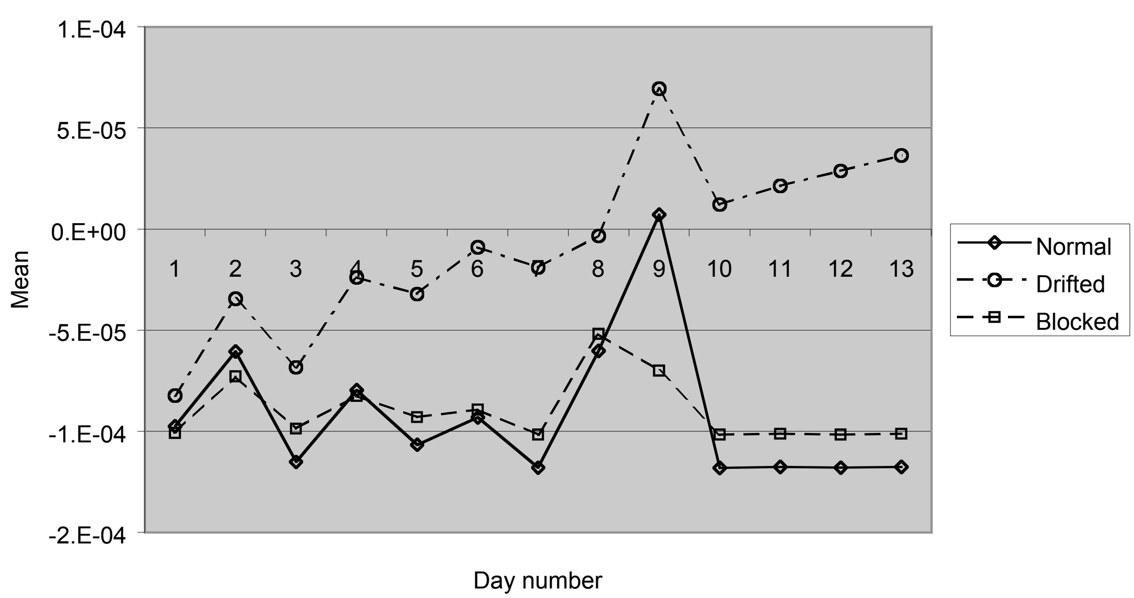

Figure 4. Mean value of the modeling error for a NARMAX model under normal operation, when the sensor is blocked and when it drifts.

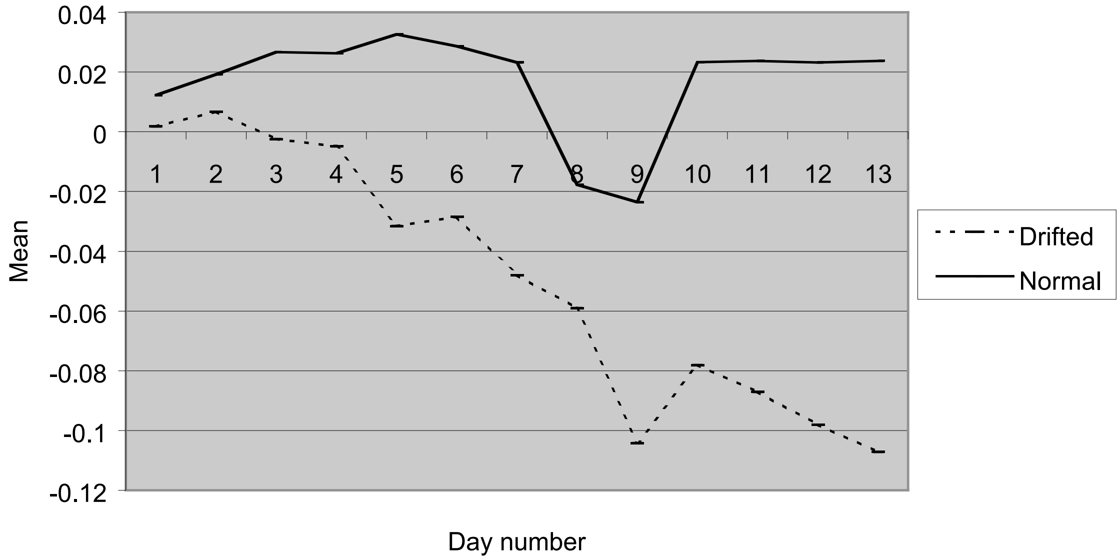

Figure 5. Daily average of the modeling error for the feedback model under normal operation and when the sensor drifts.

Conclusion

We have modeled the rainfall-water height of a simple urban catchment, using feedforward and feedback neural networks. The modeling error obtained with a NARMAX model exhibits a small average modeling error and an autocorrelation function R(1) whose value is low. If the sensor outputs a constant value, when using a NARMAX model, the autocorrelation function is significantly higher and can be used to detect the failure. But, when the sensor exhibits a slow drift, neither characteristics of the error can be used because the erroneous measure is used as an input to the model. One can then use a feedback model, for which the drift accumulates on the error.

This approach can be used for more complex urban catchments, including multiple water height sensors. The information obtained from up-stream sensors can then be used as inputs to the neural model.

[Bishop95] Bishop, C. (1995). Neural Networks for Pattern Recognition. Clarendon Press-Oxford.

[Léontaritis85] Leontaritis I.J., S.A. Billings. (1985). Input-Output parametric methods for non-linear system. Int. J. of Control, vol.41, 311-341.

[Ljung87] Ljung, L. (1987), System Identification: Theory for the User. Prentice Hall.

[Nerrand94] Nerrand, O., P. Roussel-Ragot, D. Urbani, L. Personnaz, G. Dreyfus. (1994). Training Recurrent Neural Networks : Why and How ? An Illustration in Dynamical Process Modeling. IEEE Transactions on Neural Networks, vol.5, pp178-184.