COMBINING SEMI-PHYSICAL AND NEURAL NETWORK MODELING: AN EXAMPLE OF ITS USEFULNESS

U. FORSSELL AND P. LINDSKOG

Department of Electrical Engineering, Linköping University, Linköping, SWEDEN.
E-mail: `ufo@isy.liu.se`, `lindskog@isy.liu.se`

Abstract. We illustrate the power of combining semi-physical and neural network modeling in an application example. It is argued that some of the problems related to the use of neural networks, such as high dimensionality of the parameter space and problems with local minima, can be alleviated using this approach.

Key Words. Nonlinear system identification; semi-physical modeling; neural network modeling.

1. INTRODUCTION

System identification as described by, e.g., Ljung (1987) is a well established methodology for designing mathematical models of dynamical systems using input-output data. After experiment design, the problem can be split into two parts: *model structure selection* followed by *parameter estimation*. While various least-squares type of algorithms are predominant for parameter estimation, one has a large spectrum of model structure approaches to choose between.

Physically parameterized modeling (where all physical insight about the system is condensed into the model) is a quite time-consuming procedure that normally requires a lot of prior, which can be more or less hard to acquire. However, such an approach often leads to models that are parsimonious with parameters to estimate, a property that is highly desired in identification.

On the other extreme we have the black box approach, where the models are searched for in a sufficiently flexible model family. Instead of incorporating prior system knowledge, such a procedure uses "size" as the basic structure option, i.e., the models typically involve a large number of parameters so that the unknown function can be approximated without too large a bias (at least in theory). This approach requires much less engineering time but depends heavily on the data quality. For nonlinear systems, *neural networks* (*NNs*) is one out of many possible and reasonable choices within this category.

Between these modeling extremes there is a large zone where important physical knowledge as well as common sense reasoning are used in the identification process, but not to the extent that a fully physically parameterized model is constructed. In this case the regressors (or basis functions) employed are often the result of physical reasoning, while the parameters to estimate usually have little or no direct physical interpretation. This middle zone is sometimes labeled *semi-physical modeling*.

The obvious identification challenge is now to devise general methods that combine the richness and flexibility of, e.g., NNs with the principle of parsimonious, at the same time as the required engineering effort is kept within reasonable limits. In this contribution we discuss a mixed semi-physical and NN modeling procedure equipped with these features. The basic idea is to capture what is actually known about the system using a semi-physical model, and then describe the remaining dynamics using a "small" NN.

In Secs. 2 and 3 we briefly review the basic ideas behind NN and semi-physical modeling, respectively. Taking off from this discussion, Sec. 4 illustrates the benefits of combining these approaches in a rather simple but informative tank level modeling example.

2. NONLINEAR SYSTEM IDENTIFICATION USING NEURAL NETWORKS

In the multi input single output (MISO) case, a rather general prediction model can be written as (Sjöberg et al. (1995))

$$\hat{y}(t|\theta) = g(\varphi(t), \theta) \in \mathbb{R}, \tag{1}$$

where $\varphi(t) \in \mathbb{R}^r$ are the regressors (past inputs and outputs) and $g(\cdot,\cdot)$ is some nonlinear mapping from the regressor space to the output space parameterized by $\theta \in \mathbb{R}^d$. Since NNs are good and versatile function approximators (Haykin, 1994; Kung, 1993) they are well suited for nonlinear system identification problems.

Simply put, one may think of standard (feed-forward, one hidden layer) NNs as function expansions

$$g(\varphi(t), \theta) = \sum_{k=1}^{n} \alpha_k g_k(\varphi(t), \beta_k, \gamma_k), \tag{2}$$

where $g_k(\cdot,\cdot)$ is called a basis function and α_k, β_k and γ_k are weight, scale and position parameters. A standard choice of basis function is the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{3}$$

usually together with a ridge construction

$$g_k(\varphi(t), \beta_k, \gamma_k) = g_k(\beta_k^T \varphi(t) - \gamma_k), \tag{4}$$

thus resulting in the model structure

$$g(\varphi(t), \theta) = \sum_{k=1}^{n} \alpha_k \sigma(\beta_k^T \varphi(t) - \gamma_k). \tag{5}$$

This model structure can now be fit to data using standard algorithms for nonlinear optimization such as the Levenberg-Marquardt algorithm (Fletcher, 1987). In the literature this optimization procedure is often referred to as NN training. A common problem with such iterative optimization algorithms is that of getting trapped in a sub-optimal local minimum. One way to alleviate this difficulty is to repeat the training using different initial parameter values and then accept the best model so obtained.

3. SEMI-PHYSICAL MODELING

The main idea behind semi-physical modeling is to use physical insight about the system so as to to come up with certain nonlinear (and physically motivated) combinations of the raw measurements. These combinations—the new inputs and outputs—are then subjected to standard black-box-like model structures.

More precisely, it is often desirable to have a predictor of the form

$$\hat{y}(t|\theta) = \theta^T \varphi(t), \tag{6}$$

with all nonlinearities appearing in the regressor $\varphi(t)$. Such a linear regression formulation is appealing foremost due to that the parameters θ can be estimated analytically by solving a linear least-squares problem.

The regressors to include in the structure are of course determined on a case by case basis. For example, in order to model the power delivered by a heater element (a resistor), an obvious physically motivated regressor to use would be the squared voltage applied to the resistor. In other and more involved modeling situations the prior is given as a set of unstructured differential-algebraic equations. To then arrive at a model of the form (6) typically requires both symbolic and numeric software tools as is demonstrated in Lindskog and Ljung (1995).

4. APPLICATION EXAMPLE: TANK LEVEL MODELING

In this section we will illustrate how the ideas alluded to in the previous sections can be used to improve the modeling results in a simple laboratory-scale application: the modeling of the water level of the tank depicted in Fig. 1. The identification goal is to explain how the voltage $u(t)$ (the input) affects the water level $h(t)$ (the output) of the tank. All experiments were carried out in MATLAB using the system identification toolbox (Ljung, 1995) and a newly developed NN toolbox (Sjöberg and De Raedt, 1997).

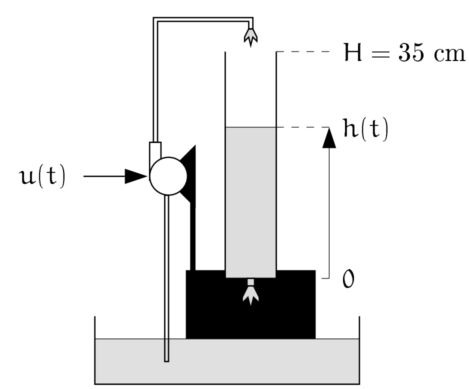

Fig. 1. Schematic picture of the tank system.

4.1. Black Box Modeling

To begin with, standard linear ARX and sigmoidal NN models (structure (5)) were fit to an estimation data set of 1000 input-output samples. The simulation result (given 1000 validation samples) of the "best" linear ARX model found is shown in Fig. 2.

In this particular case, the accepted ARX model include two parameters only:

$$\hat{h}(t|\theta) = \theta_1 h(t-1) + \theta_2 u(t-1). \tag{7}$$

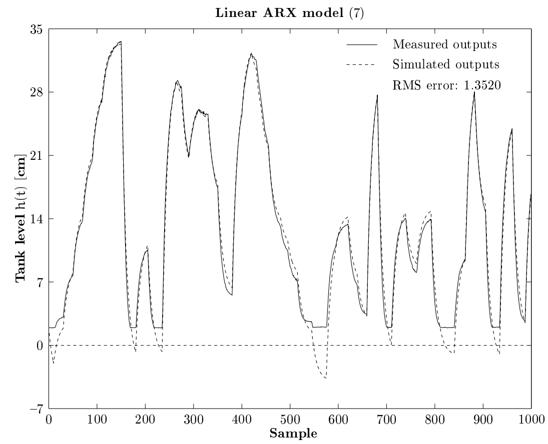

Fig. 2. Simulation of the linear ARX model (structure (7)) using validation data.

As can be seen in Fig. 2, the fit between the simulated and the measured outputs is quite good with the model output tracking the true output most of the time. However, notice that the simulated tank level is sometimes negative. This is of course a nontrivial complication if we are going to use the model to study the behavior of the real system, and we are thus forced to reject this model. In fact, all ARX models we tried had this defect and, perhaps more surprisingly, we were not able to find a NN model, with delayed inputs and outputs as regressors, that did not show this flaw either. Here the main problem seems to be to find good parameter values to avoid getting trapped in a local minimum.

4.2. Semi-Physical Modeling

To try to overcome this problem we next turned to semi-physical modeling. It is actually possible to say quite a lot about how $h(t)$ changes as a function of the inflow using physical reasoning. Let A denote the cross-sectional area of the tank, let a be the area of the outflow aperture and, as usual, let g denote the gravitational constant. When a is small, Bernoulli's law states that the outflow can be approximated by

$$q_{\text{out}}(t) = a\sqrt{2gh(t)}. \tag{8}$$

The rate of change of the amount of water in the tank at time t is equal to the inflow (proportional to $u(t)$) minus the outflow, i.e.,

$$\begin{aligned}\frac{d}{dt}(Ah(t)) &= q_{\text{in}}(t) - q_{\text{out}}(t)\\ &= ku(t) - a\sqrt{2gh(t)}\end{aligned} \tag{9}$$

Discretizing this equation using a simple Euler approximation gives

$$h(t+1) = h(t) - \frac{Ta\sqrt{2g}}{A}\sqrt{h(t)} + \frac{Tk}{A}u(t), \tag{10}$$

where T is the sampling period. After reparameterization, this structure can be expressed as

$$\begin{aligned}\hat{h}(t|\theta) = \theta_1 h(t-1) + \theta_2\sqrt{h(t-1)}\\ + \theta_3 u(t-1).\end{aligned} \tag{11}$$

This is a linear regression model of the form (6), and thus optimal parameter values can be found by solving a standard linear least-squares problem. A simulation of the so estimated model is shown in Fig. 3.

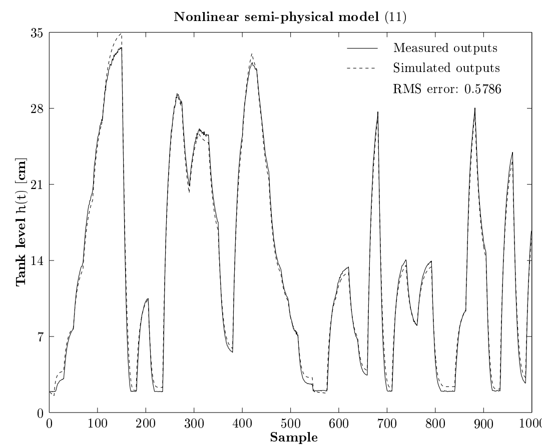

Fig. 3. Simulation of the semi-physical model (structure (11)) using validation data.

The RMS error of this model is as low as 0.5786 and, more importantly, the simulated output is never negative, which indicates that the model is physically sound.

4.3. Combined Semi-Physical and Neural Network Modeling

Even though the low-complexity semi-physical model shows such a good fit it is actually possible to do even better by combining the semi-physical model with a NN model in parallel. The proposed setup for such a blended model is shown in Fig. 4. As is indicated, the semi-physical model is kept fixed while the NN (the parameters θ_{nn}) is trained. Notice also that both sub-models work with the same regressors $\varphi(t)$.

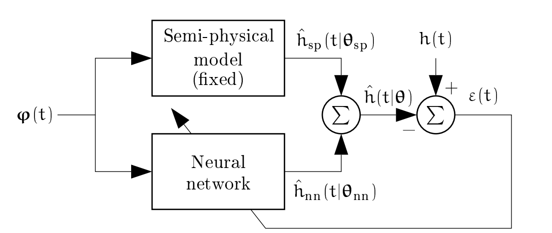

Fig. 4. Neural network trained using the residuals $\varepsilon(t)$ while keeping the semi-physical model fixed.

The accepted overall model has 22 parameters and the RMS error from simulation was as low as 0.2128, which is less than half of that obtained with the semi-physical model alone. Fig. 5 shows that the simulated outputs are virtually impossible to distinguish from the measured ones.

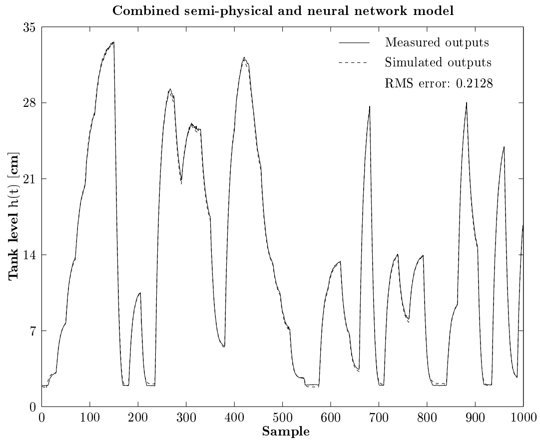

Fig. 5. Simulation of the mixed semi-physical and neural network model using validation data.

By fixing the semi-physical model while training the NN, the NN will try to model the residuals from the semi-physical model. This means that the NN will try to pick up any additional dependencies between the regressors that could not be explained by the semi-physical model, e.g., effects of whirls and air bubbles in the tank. We found this idea to be very fruitful, especially since practically no extra effort was needed to construct the combined model, given the semi-physical one. Furthermore, since the semi-physical model is fixed while training the NN the result on estimation data will be at least as good as with the semi-physical model alone, even if the size of the NN is very small. We also found that the problems with local minima (resulting in negative tank levels) were effectively taken care of by this approach.

Compared to the straightforward idea of fitting NN models directly to data, we have thus gained two things. First, the size and actual configuration of the NN is less critical and, secondly, the risk for getting stuck in a physically undesired local minimum is reduced.

A simpler variant of this idea has been proposed by several authors, e.g., Ljung et al. (1996), with the difference being that a linear model is used instead of a semi-physical one. However, we have not yet been able to obtain a physically sound model this way. Another idea put forward in Lindskog and Sjöberg (1995) is to use the regressors suggested by the semi-physical procedure and feed these into one single neural network, but again this has not lead to a significant improvement. As before, the main problem seems to be that the training algorithm gets trapped in a local minimum.

5. CONCLUSIONS

We have outlined an identification procedure and seen an example of how physical insight and semi-physical modeling can be successfully combined with black box NN modeling. While the semi-physical part captures what is actually known about the system, the NN is responsible for describing the system's unknown features. To a certain extent this made it possible to combine the respective advantages with these approaches: the model is rather parsimonious with parameters but still very flexible at the same time as the engineering effort is reasonably low.

6. REFERENCES

Fletcher, R. (1987). *Practical Methods of Optimization.* John Wiley & Sons.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* Macmillan.

Kung, S.Y. (1993). *Digital Neural Networks.* Prentice-Hall.

Lindskog, P. and L. Ljung (1995). Tools for semi-physical modelling. *Int. J. of Adapt. Control Signal Process.*, **9**(6), 509–523.

Lindskog, P. and J. Sjöberg (1995). A comparison between semi-physical and black-box neural net modeling: a case study. In: *Proc. Int. Conf. Eng. App. Artificial Neural Networks*, (A. B. Bulsari and S. Kallio, Eds.), pp. 235–238. Otaniemi/Helsinki, Finland.

Ljung, L. (1987). *System Identification: Theory for the User.* Prentice-Hall.

Ljung, L. (1995). *System Identification Toolbox – User's Guide.* The MathWorks, Inc.

Ljung, L., J. Sjöberg, and H. Hjalmarsson (1996). On neural network model structures in system identification. In: *Identification, Adaptation, Learning*, (S. Bittanti and G. Picci, Eds.), Vol. 153 of *Computer and Systems Sciences*, pp. 366–399. Springer.

Sjöberg, J. and P. De Raedt (1997). Nonlinear system identification: a software concept and examples. *Submitted to SYSID'97*. Fukuoka, Japan.

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, **31**(12), 1691–1724.