2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Identification of linear systems using output measurements with only two possible values

Bruno Depraetere[1], Gregory Pinte[2] and Jan Swevers[1]
[1]: PMA, Faculty of Mechanical Engineering, Katholieke Universiteit Leuven,
[2]: Flanders Mechatronics Technology Centre (FMTC)
Celestijnenlaan 300 B[1],D[2] B-3001, Leuven, Belgium.
bruno.depraetere@mech.kuleuven.be

Abstract— **Classical identification cannot be applied when no output measurements are available. In many situations however, discrete information on the unmeasured outputs can still be obtained and used to identify the underlying dynamics. An example is a moving object where an optical sensor can detect whether or not is in the sensors line of sight but whose position is not measured. Using these discrete data sources to estimate a model for the underlying dynamics is equivalent to the estimation of the linear parameters of a Wiener system, which has a known but non-invertible static non-linearity with two output levels. Techniques are derived to perform this estimation, using sequential quadratic programming to minimize a least squares goal function. Simulations are used to validate the proposed approach, yielding good convergence of the linear model parameters to their targets and a high prediction accuracy for the unmeasured variable of the Wiener system.**

keywords: identification, wiener model, gauss-newton, least squares, sequential quadratic programming, discrete sensor data, linear model estimation

I. Introduction

Most existing identification techniques employ measurements of the inputs and outputs of a system in order to estimate their relation. Typically, these measurements are made at discrete, closely sampled time intervals and with small quantization errors, such that these effects can be neglected and the measurements can be considered quasi-continuous. In many applications however it is too costly or impractical to install sensors measuring the output. Despite the lack of continuous output measurements, discrete information on the output can sometimes still be derived. An example is a moving object whose position is not measured, but where it can be detected once it passes a specific point. This can be done by an optical sensor or even indirectly due to a change in the behavior of the rest of the application.

In this paper, this discrete information will be used to estimate a model of the underlying dynamics. The resulting model can be used to gain insight into the process or for modelbased control purposes. An alternative approach to control such a system would be to use modelfree learning algorithms such as reinforcement learning. It then becomes possible to experimentally determine the best parameters of a parameterized input signal, such that for example the output passes by the sensor as close as possible to a given point in time. A drawback of these modelfree techniques is that they are not able to predict the output for another input or to adapt the control to perform a similar operation, as they require repetitiveness in order to learn.

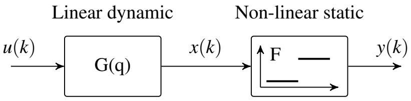

Fig. 1. Considered system consisting of a linear and a non-linear static part.

If the system to be modeled is linear, estimating the model parameters based on discrete information is equivalent to finding the parameters of the linear part of the Wiener system shown in Fig 1. In this system, the non-linear static part is known and represents the discrete measurement $y(k)$ of the unmeasured output $x(k)$ of the linear system, with

$$y(k) = F(x(k)), \tag{1}$$

where $F : \mathbb{R} \to \mathbb{R}$ is a function that equals 0 as long as $x(k)$ does not exceed a given, known threshold, and equals 1 once $x(k)$ does exceed it. Here, the threshold is arbitrarily fixed at 1, so that F is defined as

$$F(x(k)) = \begin{cases} 0 & x(k) < 1, \\ 1 & x(k) \geq 1. \end{cases} \tag{2}$$

In these equations, $x(k)$ is the unmeasured output of the linear system. It can be related to the input $u(k)$ using the linear systems' transfer function $G(q)$, as

$$x(k) = G(q)u(k), \tag{3}$$

with q^{-1} the delay operator such that $q^{-1}u(k) = u(k-1)$. The goal for the estimation is then to find a discrete-time linear model $\hat{G}(k)$ that approximates the behavior of the linear system $G(q)$ as closely as possible.

Plenty of literature is available on the identification of Wiener models, but the assumption is usually made that the nonlinearity is invertible. This is not the case for the specific function F considered in this paper, which is a non-invertible function with only two possible outputs. The number of papers that consider non-invertible non-linearities is very limited. In the majority of these, polynomial parameterizations are used for the non-linearities [1], [2], but these

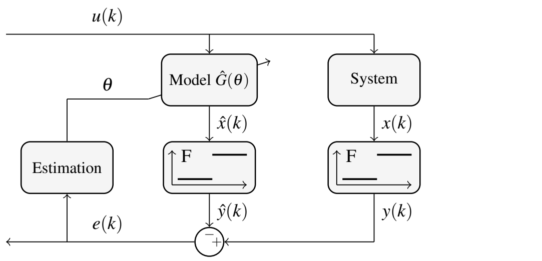

Fig. 2. Proposed estimation algorithm, with the model parameters θ chosen to minimize the difference between the predicted $\hat{y}(k)$ and measured $y(k)$.

functions are smooth and fail to accurately describe F as given by (2). Some other papers describe algorithms for non-invertible non-linearities with different parameterizations, like piecewise-linear functions, dead zones, preload, etc [3], [4], [5]. Even though they manage to describe discontinuous functions they are, to the author's knowledge, still not able to describe the F considered in this paper.

When only the linear part has to be identified and the non-linear static function F is non-invertible but known, some alternative techniques can be employed. For a first alternative, consider the specific model structure as a combination of a linear dynamic part with noise added to its output. It then follows from [6] that a Gaussian input $u(k)$ results in a non-parametric best linear approximation that asymptotically converges to the underlying dynamics of $G(q)$. This procedure would require long measurements due to the low signal to noise ratio, but it can be used to get an initial idea of the dynamics and to initialize other methods.

Another alternative is to exploit the knowledge of F and focus on the discrete nature of the output $y(k)$, which is the approach taken in this paper. The suggested estimation algorithm is illustrated in Fig. 2. It consists of comparing the measured outputs $y(k)$ to some values $\hat{y}(k)$, predicted using the known F, and selecting the model parameters θ such that the difference $y(k) - \hat{y}(k)$ is minimized [7]. Comparing these two discrete functions yields a discontinuous cost function that can not be handled using the standard least squares formulation. Some modifications are therefore introduced to make the cost function continuous, once again exploiting the knowledge of F. Two separate cases are discussed. The first deals with a long measurement, with $x(k)$ passing the threshold many times causing many switches in the output $y(k)$. The second considers a sequence of short measurements with only 1 switch in each measurement, making the estimation applicable to many practical situations.

The remainder of this work is structured as follows. First, in section II the methodology to estimate the linear model parameters is discussed and several estimation algorithms are derived. These techniques are then applied to a simulation example in section III as a proof of concept and illustration of the estimation process. The main results of this work are then summarized in section IV, while some suggestions and plans for future research are given in section V.

II. Estimation methodology

In this section the methodology used to estimate the linear model parameters is derived. In II-A, this is first done for the case where a single, long measurement of in- and outputs is available, while the needed adjustments to combine information from several short measurement are discussed in II-B.

Initially, the techniques are discussed assuming the correct model structure is selected and no noise is present, which allows convergence to the real parameters. In II-C the effect of a model structure mismatch is investigated and the influence of noise is discussed.

A. Estimation based on a single measurement

Consider first a measurement of inputs $u(k)$ and outputs $y(k)$. Using this data, a model for the linear system $G(q)$ can be estimated reliably in case some requirements are met. First of all, there needs to be a sufficient number of changes in the output $y(k)$, such that enough information is present to estimate the model parameters. Furthermore, the input $u(k)$ has to be chosen such that it persistently excites all the dynamics of the system, which is a condition also found when directly identifying linear systems [7]. [1]

Once these conditions are met, the estimation should ideally be done by minimizing the prediction error on the output of the linear part of the Wiener model, by comparing the real values of $x(k)$ with those of the predicted $\hat{x}(k)$. However, since $x(k)$ is not measured due to a lack of sensors, the values of the measured output $y(k)$ have to be compared to the predicted values of $\hat{y}(k)$ instead. These can be predicted since the non-linear static function F is known, using the relation

$$\hat{y}(k) = F(\hat{x}(k)). \tag{4}$$

The predicted variables $\hat{x}(k)$ can in turn be found as

$$\hat{x}(k) = \hat{G}(q)u(k), \tag{5}$$

with $\hat{G}(q)$ an estimated discrete-time linear model. This model can be represented as a transfer function

$$\hat{G}(q) = \frac{\hat{B}(q)}{\hat{A}(q)}, \tag{6}$$

with numerator $\hat{B}(q)$ and denominator $\hat{A}(q)$ defined as

$$\hat{B}(q) = \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2} + \ldots + \hat{b}_{n_b} q^{-n_b}, \tag{7}$$

$$\hat{A}(q) = 1 + \hat{a}_1 q^{-1} + \hat{a}_2 q^{-2} + \ldots + \hat{a}_{n_a} q^{-n_a}. \tag{8}$$

The coefficients of $\hat{B}(q)$ and $\hat{A}(q)$ are combined into θ as

$$\theta = [b_1 \ b_2 \ \ldots \ b_{n_b} \ a_1 \ a_2 \ \ldots a_{n_a}], \tag{9}$$

such that θ fully represents the model (6), and allows prediction of $\hat{x}(k)$ for a given input and through (4) also of $\hat{y}(k)$. The goal is then to find the parameters θ such that the predicted outputs $\hat{y}(k)$ match the measured outputs $y(k)$

[1] The requirements given have not been analytically derived, and further work on this is planned in the future. The minimum required number of changes in the output $y(k)$ for example has not been investigated yet.

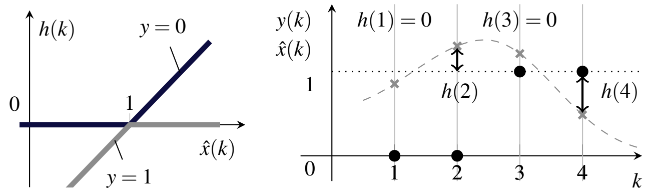

Fig. 3. Function $h(k)$ used to make the cost function continuous. On the left, the profile of $h(k)$ is shown for the value of the measured output $y(k)=0$ (black) and $y(k)=1$ (gray). On the right, $\hat{x}(k)$ (gray ×) and $y(k)$ (black •) are shown as well as the threshold (black dotted line). For $k=2$ and $k=4$, $\hat{x}(k)$ is on the wrong side of the threshold, and for those cases it is indicated how $h(k)$ is calculated, while $h(k)=0$ for $k=1$ and $k=3$.

as closely as possible and a good approximation of the real system is obtained. This means θ can be found by solving

$$\min_{\theta} \quad \| \mathbf{y}-\hat{\mathbf{y}}(\theta,\mathbf{u}) \|_2^2, \tag{10}$$

where

$$\mathbf{y} = [y(1)\ y(2)\ \dots\ y(N)]^{\mathrm{T}} \tag{11}$$

$$\text{and} \quad \hat{\mathbf{y}}(\theta,\mathbf{u}) = [\hat{y}(1)\ \hat{y}(2)\ \dots\ \hat{y}(N)]^{\mathrm{T}} \tag{12}$$

contain the N measured and predicted outputs respectively.

Due to the discrete nature of the output variables, solving (10) is not straightforward. A small change of θ can still yield exactly the same predicted outputs $\hat{\mathbf{y}}(\theta,\mathbf{u})$ and the same cost. On the other hand, a small change in θ can also lead to a large change in the cost, if one of the values $\hat{x}(k)$ suddenly passes the threshold. This is because the cost function is discontinuous and non-differentiable, which makes it impossible to use standard local optimization techniques [8]. This problem can be solved by using global optimizers that don't impose any conditions on the cost function, like genetic algorithms [8], [9]. It is however expected that these techniques would require a long convergence period. Since one of the plans for future developments is to derive a recursive estimator to regularly update the model parameters online, this long convergence process could becomes an issue. It is therefore chosen to solve the problem of the discontinuous cost function by adapting problem (10) such that the cost becomes continuous and local optimization techniques can be applied.

The cost function of (10) can be adapted in many different ways. A simple adaptation that still yields good results consists of replacing the prediction error $y-\hat{y}(\theta,u)$ in (10) by a function $h:\mathbb{R}\to\mathbb{R}$ defined as

$$h(k) \;=\; \begin{cases} 0 & y(k)=\hat{y}(k), \\ \hat{x}(k)-1 & \text{otherwise.} \end{cases} \tag{13}$$

This function $h(k)$, shown in Fig. 3, essentially sets the cost to 0 when no prediction error is present on $\hat{y}(k)$ and penalizes the difference between the predicted value of $\hat{x}(k)$ and the threshold whenever there is a prediction error. Using $h(k)$, the parameters θ can be found by solving

$$\min_{\theta} \quad \| \mathbf{h}(\theta,\mathbf{u}) \|_2^2, \tag{14}$$

where $\mathbf{h}(\theta,\mathbf{u})$ is defined as $\mathbf{h}(\theta,\mathbf{u}) = [h(1)\ h(2)\ \dots\ h(N)]^{\mathrm{T}}$. In theory, the global solutions for problems (10) and (14) can be different. In (10) penalties are aplied whenever the predicted discrete value of $\hat{y}(k)$ is incorrect, while in (14) a measure is used of how large the error is on $\hat{x}(k)$ with respect to the expected value. When the input signal is sufficiently long and enough information is present in the in- and output signals, the global solutions of (10) and (14) are however expected to lie close to each other, as models yielding low costs according to (10) also yield low costs according to (14) and vice versa.

The optimization problem (14) is solved using the Gauss-Newton algorithm, which is a type of sequential quadratic programming (SQP) [10]. In this method, $\mathbf{h}(\theta,\mathbf{u})$ is linearized around the latest estimate of θ_i, such that

$$\mathbf{h}(\theta,\mathbf{u}) \approx \mathbf{h}(\theta_i,\mathbf{u}) + \varphi(\theta-\theta_i) = \mathbf{h}(\theta_i,\mathbf{u}) + \varphi\delta\theta, \tag{15}$$

where φ is the $N \times (n_b+n_a)$ Jacobian matrix containing the first order derivatives of the components of $\mathbf{h}(\theta,\mathbf{u})$ with respect to the parameters θ, calculated at the current estimate θ_i. The optimization criterion then becomes

$$\begin{aligned} \| \mathbf{h}(\theta,\mathbf{u}) \|_2^2 &= \mathbf{h}(\theta,\mathbf{u})^{\mathrm{T}}\mathbf{h}(\theta,\mathbf{u}) & (16)\\ &\approx (\mathbf{h}(\theta_i,\mathbf{u})+\varphi\delta\theta)^{\mathrm{T}}(\mathbf{h}(\theta_i,\mathbf{u})+\varphi\delta\theta) & (17)\\ &\approx \mathbf{h}(\theta_i,\mathbf{u})^{\mathrm{T}}\mathbf{h}(\theta_i,\mathbf{u}) + 2\delta\theta^{\mathrm{T}}\varphi^{\mathrm{T}}\mathbf{h}(\theta_i,\mathbf{u}) & \\ &\qquad + \ \delta\theta^{\mathrm{T}}\varphi^{\mathrm{T}}\varphi\delta\theta & (18) \end{aligned}$$

After derivation, this yields the optimal solution as

$$\delta\theta \;=\; -(\varphi^{\mathrm{T}}\varphi)^{-1}\varphi^{\mathrm{T}}\mathbf{h}(\theta_i,\mathbf{u}), \tag{19}$$

which can be used to update the model parameters and find θ_{i+1}. At the new estimate, $\mathbf{h}(\theta,\mathbf{u})$ is linearized again and the entire process is repeated. Convergence of this algorithm to a local minimizer can be guaranteed [10] if an appropriate line search method is introduced, such that θ_{i+1} is found as

$$\theta_{i+1} = \theta_i + t\delta\theta. \tag{20}$$

The factor t needs to be included to ensure a sufficient decrease of the cost function, and its value can be found using a backtracking algorithm [10]. This line search method is also used to include constraints on the allowed variation of the model parameters, by setting an upper bound for t.

While the optimization problem (14) is continuous, it is not differentiable such that the SQP can struggle to find the true optimum but ends up near it. Subgradient methods can then be applied to overcome this [11]. Another option is to reformulate the functions on the left of 3 in a similar manner as the reformulation of L_1 minimization [12], resulting in a continuous problem by adding help variables and constraints. Since both solutions increase the required calculations, the SQP can be used first to quickly get close to the optimum, followed by a few iterations of the slower methods.

Only convergence to a local minimum can be guaranteed, so attention must be paid to the initialization of the model parameters. When possible, a good first estimate should be used, close to the optimal solution, obtained for example from physical insight. Alternatively, the algorithm can be started from several different initial estimates, each converging to a local minimum after which the best one is selected.

B. Estimation based on a sequence of measurements

The estimation algorithm needs to be adapted if there is no single measurement with many changes in the output, but instead a sequence of shorter measurements is available. For many applications, data of this type can easily be obtained or is already available. A further restriction is that each measurement contains only a single change in the output. This can be the case when no reliable information about the state is known after the first change of the output, for example when a flawed detection procedure only allows to detect a switch from 0 to 1 but not from 1 to 0. Using data from a single measurement would not allow a consistent identification, so the information of M of these measurements has to be combined. There is thus a sequence of M measurements available, each with N_i samples and a single change of $y(k)$ at the last sample of each measurement. Some additional requirements need to be fulfilled in order to perform a consistent identification. First, the input signals for the different measurements have to persistently excite all the dynamics. In addition, it is required that there is sufficient variation between the inputs of the different measurements to avoid trivial solutions.

The parameters θ are found by solving an optimization problem with a cost function that is found in a similar manner as in II-A. It is possible however to exploit the knowledge that each measurement contains exactly 1 change, and that it occurs at the last sample. To do so, the difference between $\hat{x}(k)$ and the threshold around the transition is used to calculate some criterium, and θ is found by minimzing the sum of these criteria evaluated for the different measurements. This criterium only considers the last two samples of each measurement, and is given by a continuous function $g : \mathbb{R}^N \to \mathbb{R}$ defined as

$$g = \begin{cases} h(N_i) & \hat{y}_i(N_i) \neq y_i(N_i), \\ h(N_i - 1) & \text{otherwise}, \end{cases} \qquad (21)$$

with $h(k)$ as defined in section II-A This function $g(\theta, \mathbf{u}_i, \mathbf{y}_i)$ is illustrated in Fig. 4. If the output at the last sample $\hat{y}(N_i)$ is predicted incorrectly, the difference between $\hat{x}(N_i)$ and the threshold is penalized as $h(N_i)$. Otherwise, if the last sample is predicted correctly, the cost is found by comparing $\hat{x}(N_i - 1)$ to the threshold, using $h(N_i - 1)$. t By defining

$$\mathbf{g} = [g(\theta, \mathbf{u}_1, \mathbf{y}_1)\ g(\theta, \mathbf{u}_2, \mathbf{y}_2)\ \ldots\ g(\theta, \mathbf{u}_M, \mathbf{y}_M)]^{\mathrm{T}}, \qquad (22)$$

the optimization problem to find the model parameters θ becomes

$$\min_{\theta} \quad \| \mathbf{g}(\theta, \mathbf{u}_i, \mathbf{y}_i) \|_2^2, \qquad (23)$$

which can be solved in a similar manner as (14) in II-A.

C. Effects of model structure mismatch and noise

It has been observed that the estimation becomes more difficult when the model structure is chosen incorrectly. When the orders are too low such that there are insufficient degrees of freedom, many local minima are typically obtained with poor model quality since perfect prediction becomes impossible. When the orders are too high the correct model can still be generated but there are too many degrees of freedom. In this situation it is often possible to get good prediction in some of the minima, but there are typically a large amount of suboptimal local minima as well. In both situations the choice of the initial θ becomes essential in order to find a good local minimum with good prediction quality. As stated before in section II-A, this problem can be solved by starting with a good initial estimate or by trying several different initial estimates.

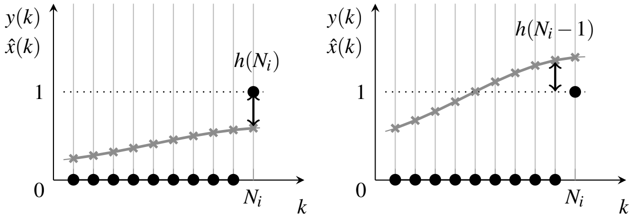

Fig. 4. Behavior of function $g(k)$ used to make the cost function for multiple runs continuous, determined by the values of $\hat{x}(k)$ (gray ×) and $y(k)$ (black •) with respect to the threshold (black dotted). On the left, the last sample is incorrectly predicted such that the cost is calculated as $h(N_i)$, comparing $\hat{x}(N_i)$ to the theshold. On the right, the last sample is predicted correctly, such that the cost is given by $h(N_i - 1)$, by comparing $\hat{x}(N_i - 1)$ to the theshold at the but last sample.

Measurement noise on the output $y(k)$ does not have to be considered since this is a signal with 2 discrete levels. There can however be many sources of noise entering the system, all leading to some variation on $x(k)$ and thus also affecting the output $y(k)$. Unless these sources are known and they can be included in the model, not much can be done other than to make sure the estimation is robust. One simple robustifying alteration is to adapt $h(k)$ to include a deadband [12]. This way $h(k)$ only differs from 0 if the amplitude of the difference between $\hat{x}(k)$ and the threshold is larger than a given value ε. Hence, small prediction errors $< \varepsilon$ on $\hat{x}(k)$ that can occur due to noise are not penalized.

III. Simulation results

Now the methods derived in section II are validated by estimating the model parameters of a simulated system, considered unknown to the estimation procedures. Like in II, this is first done for a single long measurement and then for a sequence of shorter measurements, always selecting the input such that the binary output data is informative. Also like in II, it is first assumed that no noise is present, and afterwards the effects of adding noise are investigated. The case of model structure mismatch is omitted, as this can be solved by a thorough model selection procedure or by trying several structures.

To validate the estimation algorithm derived in section II-A, it is used to estimate the parameters of a discrete-time linear model with the sampling time set at $T_s = 0.001$s. The frequency response functions (FRF) of this simultion

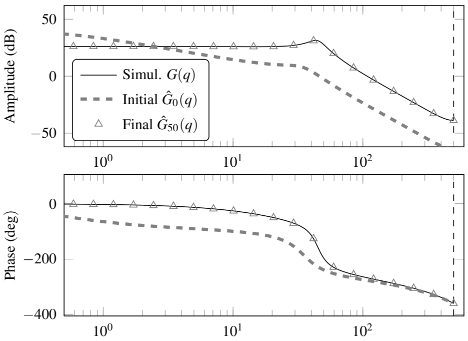

Fig. 5. Frequency response functions of the simulation model (black), as well as the model used to initialize the estimation procedures (gray dashed) and the estimated model obtained after 50 iterations (△).

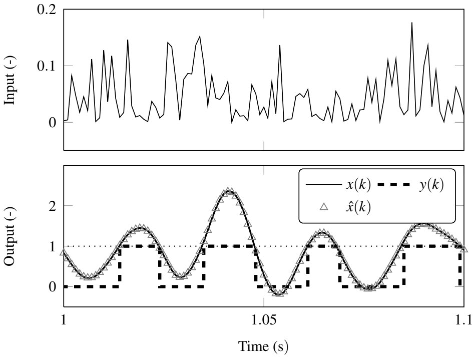

Fig. 6. Part of signals used for the estimation. The top figure shows the input $u(k)$ while the bottom one shows the continuous output $x(k)$ (black), the discrete output $y(k)$ (black dashed) and the predicted $\hat{x}(k)$ (gray △).

model is shown in Fig. 5, and the parameters of its transfer functions $G(q)$ are given in the first row of table I. In order to estimate these parameters, an input of length 5 s (N=5000 samples) is applied. It is made up of random noise with an additional DC component to ensure the output varies around the threshold, as illustrated in Fig. 6, where a part of the input and its corresponding output are shown. The input $u(k)$ and the outputs $y(k)$ are then used to estimate the parameters, starting from the initial set of model parameters indicated by the dashed line in Fig. 5. The middle row in table I gives the resulting parameters, obtained after 50 iterations of the estimation procedure. A good correspondence can be observed between the estimated and the real parameters.

TABLE I

COMPARISON OF SIMULATION MODEL PARAMETERS AND ESTIMATED VALUES, EVALUATED AFTER 50 ITERATIONS.

	Parameters					
	b_1	b_2	b_3	a_1	a_2	a_3
Sim. model	0.042	0.16	0.037	-2.69	2.48	-0.78
Est. ($\sigma = 0$)	0.042	0.16	0.037	-2.69	2.48	-0.78
Est. ($\sigma = 0.05$)	0.041	0.16	0.037	-2.69	2.49	-0.78

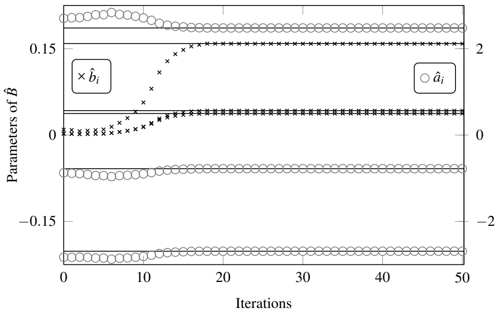

Fig. 7. Evolution of the estimated model parameters $\hat{b}_i$ (×) and $\hat{a}_i$ (○) with respect to the true parameters (solid lines), as a function of the iterations.

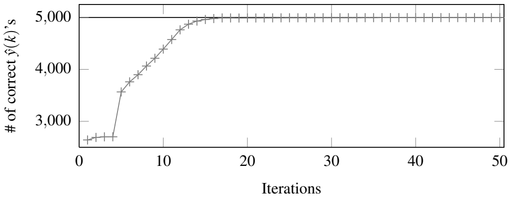

Fig. 8. Evolution of the number of correctly predicted values $\hat{y}(k)$, as a function of the iterations, with 5000 the total number of samples.

To illustrate the convergence process, the estimation is investigated in some more detail. Fig. 7 shows the evolution of the estimated parameters as a function of the iterations. All variables converge towards the true values of the simulation model. As a result, prediction of the unmeasured values $x(k)$ becomes possible. Fig. 6 illustrates this, showing a close match between the predicted values $\hat{x}(k)$ (gray △) and the real outputs $x(k)$ (black solid), despite the lack of measurements for this variable. Since $x(k)$ can be predicted accurately, the same holds for $y(k)$. Fig. 8 illustrates this, showing the evolution of the number of correctly predicted outputs as a function of the iterations. Initially, just over half the samples are predicted correctly, which is to be expected even from a random guess. After the first few iterations this number quickly increases and after 50 iterations all but 3 samples are predicted correctly.

To evaluate the performance in the presence of process noise zero-mean Gaussian noise with a standard deviation of $\sigma = 0.05$ is added to the output $x(k)$, leading to a different sequence of observed outputs $y(k)$. The same estimation procedure is then repeated, starting from the same initial model parameters. Due to the noise, the number of correctly predicted outputs $y(k)$ is lower, with 208 samples predicted incorrectly after 50 iterations and this number not decreasing any further. This is to be expected however as imperfect prediction would be obtained even when the true system parameters are used. The more important result is that the model parameters do converge to their true targets or values very close to them, as indicated in the bottom row of table I.

To validate the algorithm derived in section II-B for a sequence of short measurements, the same simulation model is used and the same initial conditions are selected. The input signal of 5 s is however replaced by a sequence of 100 short measurements. Fig. 9 shows the inputs and the corresponding outputs for two examples of such measure-

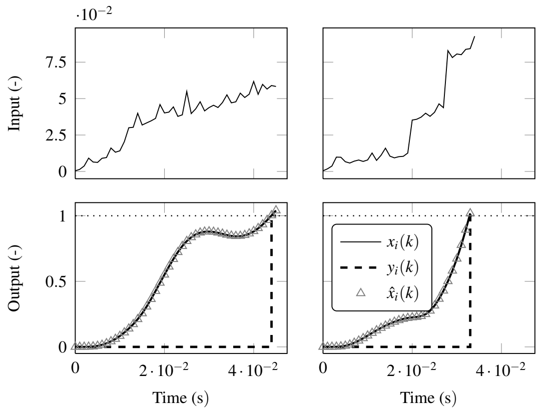

Fig. 9. Two of the signals used for the estimation based on a sequence of measurements. The top figures show the input $u_i(k)$ while the bottom figures show the corresponding outputs; the continuous output $x_i(k)$ (black), the discrete output $y_i(k)$ (black dashed) and the predicted $\hat{x}_i(k)$ (gray $\triangle$).

TABLE II

COMPARISON OF SIMULATION MODEL PARAMETERS AND VALUES ESTIMATED USING $M = 100$ SHORT MEASUREMENTS, EVALUATED AFTER 50 ITERATIONS.

	Parameters					
	b_1	b_2	b_3	a_1	a_2	a_3
Sim. model	0.042	0.16	0.037	-2.69	2.48	-0.78
Est. ($\sigma = 0$)	0.042	0.16	0.037	-2.69	2.48	-0.78
Est. ($\sigma = 0.05$)	0.042	0.16	0.037	-2.69	2.47	-0.77

ments. These inputs signals are obtained as Gaussian noise with the addition of some random ramps and steps to ensure the output eventually passes the threshold. Since only the data up to the first measured change in $y_i(k)$ is considered, each input signal leads to a switch in $y_i(k)$ at the last sample N_i, which can be different for each measurement. These 100 inputs $u(k)$ and outputs $y(k)$ are then used to perfrom the estimation. Table II gives the estimated parameters after 50 iterations, showing convergence of the parameters to their true values. As another indicator of the performance, Fig. 9 also shows the predicted values of $\hat{x}(k)$ (gray $\triangle$), which lie very close to the real values of $x(k)$ (black solid).

In a final test the same type of noise is added to the output $x(k)$ of the 100 measurements, and the estimation is started with the same initial parameters. The estimated model parameters after 50 iterations are given at the bottom of table II, once again showing convergence to the true parameters or values close to them.

IV. CONCLUSIONS

A method is proposed for estimating the linear model parameters of Wiener systems with known non-invertible discrete static non-linearities. These methods can be used whenever sensors are lacking but discrete information like the passing of a threshold can be obtained. Two algorithms are derived to estimate the model parameters by minimizing the difference between the measured discrete outputs and the predicted outputs. The first can be used when a long measurement is available, with many changes in the discrete output. The second can be used when many short measurements are available, with only one change in the discrete output per measurement, as is often the case in practical situations. Both methods are illustrated using a simulation example, yielding good performance with the parameters converging to the correct values and a high prediction accuracy, even for the unmeasured outputs of the Wiener system's linear part.

V. OUTLOOK

One aspect that will be investigated in the future is the design of the input signals used to excite the system. Open questions are what types of excitation signals provide the best results and what the minimum requirements are to get consistent models. Another planned development is the derivation of a recursive algorithm. Starting from the algorithm for a sequence of measurements, the goal is to obtain an algorithm that can run online and update the model parameters every time a new measurement is available. Finally, it is our goal to perform an experimental validation of the developed estimation techniques in the near future.

ACKNOWLEDGEMENTS

This work has been carried out within the framework of projects IWT-SBO 80032 (LeCoPro) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) and G.0422.08 of the Research Foundation - Flanders (FWO - Vlaanderen). This work also benefits from K.U.LeuvenBOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC) and from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office.

REFERENCES

[1] S. Lacy and D. Bernstein, "Identification of FIR Wiener systems with unknown, noninvertible, polynomial nonlinearities," *Proceedings of the 2002 American Control Conference*, pp. 893–898, 2002.

[2] M. J. Korenberg and I. W. Hunter, "Two methods for identifying Wiener cascades having noninvertible static nonlinearities." *Annals of biomedical engineering*, vol. 27, no. 6, pp. 793–804, 1999.

[3] H.-F. Chen, "Recursive Identification for Wiener Model With Discontinuous Piece-Wise Linear Function," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 390–400, Mar. 2006.

[4] J. Vörös, "Parameter identification of Wiener systems with discontinuous nonlinearities," *Systems & Control Letters*, vol. 44, no. 5, pp. 363–372, Dec. 2001.

[5] R. Pupeikis, "On the Identification of Wiener Systems Having Saturation-like Functions with Positive Slopes," *INFORMATICA*, vol. 16, no. 1, pp. 131–144, 2005.

[6] J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain, "Identification of linear systems with nonlinear distortions," *Automatica*, vol. 41, no. 3, pp. 491–504, 2005.

[7] L. Ljung, *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, December 1998.

[8] K. Kristinsson and G. Dumont, "System identification and control using genetic algorithms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 5, pp. 1033–1046, 1992.

[9] Y.-c. Lu, M.-h. Chang, and T.-j. Su, "Wiener Model Identification using Genetic Programming," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II*, vol. II, 2008, pp. 19–21.

[10] J. Nocedal and S. J. Wright, *Numerical Optimization (2nd Edition)*. Springer, 2006.

[11] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*. Springer-Verlag, 1985.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.