

Calibration of physical models with process data using FIR filtering

Enso Ikonen¹ and István Selek¹

Abstract—Automatic calibration of physical plant models in the context of monitoring and control of industrial processes is considered. A structure integrating a physical model and estimated FIR filters is proposed. In addition, a finite state FIR structure is proposed to complement the calibrated physical model with a data-driven mapping. The approach is illustrated in simulations using the van der Vusse CSTR benchmark.

I. INTRODUCTION

Modeling always serves a purpose. A model can be necessary for designing a model-based controller or state-estimator or optimizing system dynamic behaviour otherwise. Model-based fault detection and isolation, operator training simulators or plant analysis and design are other examples of applications requiring a proper dynamic process model. A modern digital twin gets its essential behavioral contents from a plant model, perhaps linking simulations with on-line data and various services such as data visualization and analytics, what-if analysis or 3D plant animations. In this paper, the focus is on process control.

In the era of IoT and AI, the role of data-driven modeling has been emphasized. Data-driven modeling is characterized by the bias-variance dilemma, related to removal of noise from measurements and faithful representation of the underlying nonlinear mapping. Both problems are impossible to solve using data alone. The impact of noise can be reduced by taking repeated samples, but for a finite data set the uncertainty will remain. If only sampled data is available, there is no information about the mapping between samples (time or space). Consequently interpolation/extrapolation, i.e. generalization, is always based on some additional assumptions.

Yet, there exists a vast literature on proposed data-driven approaches to process modeling and experiences on their application to simulated and/or real-life plants [1][2]. The approaches include a wide variety of various basis-function constructions proposed suitable for process modelling tasks, statistical analytics on conclusions that can be drawn from data, and algorithms providing optimal active exploration, just to mention a few. Probabilistics is not the only possibility to deal with uncertainties, even if -by far- the most popular. Fuzzy reasoning provides also a convenient means to introduce human experimental knowledge in the form of expert rules. In multiple-model approaches a single model is replaced by reasoning over a fleet of models. Modeling can be postponed until the exact conditions are known and local predictions conducted. In this variety of the approaches, the

choice of paradigm appears to depend on the experience of the modeler.

The application of physical model is widespread in many fields of engineering. Physical models provide justified and transparent predictions and explanations for the outcomes, based on mass, momentum and energy balances, laws of mechanics, chemistry, etc. The role of physical models is pronounced in the heavy process industry, where models are extensively used in plant design. In the industrial scale, experimentation and exploration with plants is severely complicated due to costs, slowness of processes and safety issues. However, construction of physical models is resource consuming and tuning of such models to site conditions can be complicated.

This paper considers the problem of identifying a I/O model for a MIMO system, such as

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \quad (1)$$

$$\mathbf{y}(k) = g(\mathbf{x}(k), \mathbf{v}(k)) \quad (2)$$

where $f : R^{n_x} \times R^{n_u} \times R^{n_w} \rightarrow R^{n_x}$ and $g : R^{n_x} \times R^{n_v} \rightarrow R^{n_y}$. For brevity, the sampling time is assumed to be a constant and the important topics of input/output selection and design of sampling are not considered here. The inputs \mathbf{u} are assumed to be known and measured, possible uncertainties in inputs can be expressed as state noise \mathbf{w} . It is not assumed that the states \mathbf{x} are known, measured or estimated. The plant outputs are measured, corrupted by noise \mathbf{v} . Noise characteristics need not be known, smoothness or existence of derivatives of f or g is not assumed. However, it is assumed that the system is time-invariant, or that it varies with time very slowly so that (some kind of) ergodic assumptions hold.

This work proposes an approach based on the use of i) physical modeling, ii) finite impulse response (FIR) dynamics and iii) finite state representation. The main benefits are in that the plant modeling (simulators) can be incorporated into the proposed structure and the parameter estimation in the data-driven components is particularly simple and robust, enabling automatic procedures. The few parameters to be set by the user are very transparent and do not require an experienced data scientist or process development engineer. The next section considers calibration of physical plant models and leads to propose a structure for model calibration. The calibrated model is suggested to be complemented by a data-driven model in parallel and a finite-state time-series model is suggested for this task. The approach is illustrated using the well known van der Vusse chemical reactor benchmark. Conclusions end the paper.

¹Enso Ikonen and István Selek are with Intelligent Machines and Systems, University of Oulu, FIN-90014 Oulun yliopisto, Finland
enso.ikonen@oulu.fi istvan.selek@oulu.fi

II. CALIBRATION OF PHYSICAL MODELS

Suppose that a given plant has a dynamic model

$$\mathbf{x}(k+1) = f_{PM}(\mathbf{x}(k), \mathbf{u}(k)) \quad (3)$$

$$\mathbf{y}_{PM}(k) = g_{PM}(\mathbf{x}(k)) \quad (4)$$

$f_{PM} : R^{n_x} \times R^{n_u} \rightarrow R^{n_x}$ and $g_{PM} : R^{n_x} \rightarrow R^{n_y}$. This model, called a ‘‘physical model’’ (PM) is likely to be derived based on 1st principles or detailed engineering knowledge. It may be complex. Nothing can be assumed about its internal structure or parameters. It is possible to simulate the propagation of \mathbf{y}_{PM} in real time in parallel with the plant, or using historical data. The plant model is assumed to have been calibrated as found appropriate, given the constraints on knowledge and resources. It is provided ‘‘as is’’. This kind of models emerge in industrial plant design, for example.

Corresponding I/O measurements from a real plant are available: $\mathbf{u}(i)$ and $\mathbf{y}(i)$, $i = 1, 2, \dots, k$. We want to use this data to calibrate the process model, so that the predictions would be consistent with the data. The purpose of calibration is to adjust the physical model to local conditions. The need for adjustment may be due to various reasons, such as correction of idealisations made during modeling or adaptation to systematic errors in plant measurements.

This wish to calibrate the physical model is constrained by a number of engineering requirements. Very often is not desired tune the parameters inside f_{PM} or g_{PM} , in order to preserve the plant design model. This is due to various reasons: Presumably these parameters appear in a complex manner. The plant model is trusted/accepted by the users, or the users are familiar with the model with its defects. The calibration must be automatic from data and should scale well. This requires that the procedure is simple and straightforward, requiring minimal insight to the physical model, and robust vs. measurement noise. The algorithm should be simple to code and run on-line in a distributed control system environment or alike. The approach should support the a priori assumption that physical models are often more accurate in their steady-state description, typically in the focus of plant design. The purpose of calibration is not to model phenomena/behaviour that is not described by the physical model. A more detailed plant modeling (of any unmodelled behaviour) is almost surely a project requiring an approach different from an automated procedure. Finally, the procedure should give some means to analyze the adjustments made to the physical model mapping due to calibration.

Joint physical/data-driven structures for modeling dynamic processes can be roughly categorized into the following

- A data-driven model (DM) is placed in parallel with the PM. The outcomes of the two independently generated models, DM and PM, can be compared by running both models in like situations, and this multimodel information then presented to the end-users. Typically data-driven approaches can provide an estimate of the prediction uncertainty. If such uncertainty estimate is

available for the PM also (e.g. via Monte Carlo), some sort of bayesian reasoning can be conducted to devise a more likely estimate.

- The DM is trained to minimize the residual between measurements and the PM output. The overall outcome is a sum of the PM and DM. This approach is appealing in that the DM is used to compensate for the defects in PM predictions, the behaviour modeled by PM need not be re-modelled by DM.
- The PM is combined with a DM by placing it in series. The DM can be placed either after or before the PM, so as to tune both static and dynamic behaviour of the PM. Optimization aims at minimizing the residual between the prediction and output data. A major obstacle in the latter structure is that in minimization of output residual using gradient based parameter estimation techniques requires computation of the derivatives through the PM.

A basic analysis of some of the differences of the structures is simple to derive using the linear case. Suppose that the physical model dynamics are given by $\frac{num_{PM}}{den_{PM}}$. The data from the true plant dynamics is obtained from $\frac{num_D}{den_D}$ (for simplicity the measurement dynamics are omitted). Suppose that the DM is trained to minimize the difference between DM output and measured plant output. We then obtain the following filters for DM:

- In the case of parallel DM: $\frac{num_{DM}}{den_{DM}} \approx \frac{num_D}{den_D}$
- In the case of parallel DM for PM residuals: $\frac{num_{DM}}{den_{DM}} \approx \frac{num_D}{den_D} + \frac{num_{PM}}{den_{PM}} - \frac{num_{DM}}{den_{DM}} \approx \frac{num_D}{den_D} + \frac{num_{PM}}{den_{PM}} - \frac{num_{DM}}{den_{DM}}$
- In the case of DM following PM: $\frac{num_{DM}}{den_{DM}} \approx \frac{num_{PM}}{den_{PM}} \frac{num_D}{den_D}$

From the analysis it is clear that the parallel DM model is by far the simplest to estimate. However, this approach provides no remedy for the PM calibration problem, it merely provides a data-driven alternative DM.

An alternative to $\frac{num_{DM}}{den_{DM}}$ (ARX) is to estimate only num_{DM} (finite impulse response, FIR):

$$y(k) = b_0 u(k) + b_1 u(k-1) + \dots + b_{n-1} u(k-n+1) \quad (5)$$

The FIR filters are known to be able to model complex dynamics, at the price of a large number of parameters to estimate, $n \gg \dim(num) + \dim(den)$. The FIR parameters can be estimated using least squares (LS), a number of extensions have been proposed to handle the nonparsimonious nature of FIR [3]. However, as illustrated in the above model structure analysis, the number of parameters in num_{DM} and den_{DM} can also be higher than expected at first glance. If, in addition, the models are to be used for predicting to the future, the ARX modeling is not feasible, but one needs to turn to output error (OE) modeling. Unfortunately, OE models can not be estimated with LS, but require iterative approaches. In conclusion, FIR modeling can be seen as an attractive structure for the problem at hand. This is further confirmed by the fact that a number of model-predictive control approaches use the FIR or finite step response descriptions for deriving the process controls [2].

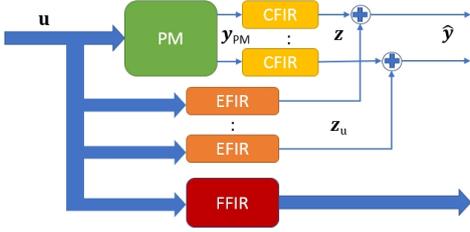


Fig. 1. Physical model calibration structure using FIR filters.

Many algorithms have been proposed for data-driven modeling of nonlinear dynamic processes [1][2]. Most -if not all- model structures can be interpreted as basis-function approaches: global (e.g. Volterra series), semi-global (e.g. sigmoid neural nets) or local (e.g. radial basis functions). Time series approach with external or internal dynamics can be applied. In practice, identification of nonlinear process models has turned out to be challenging task. This is mainly due to the lack of proper data. Processes in the heavy industry are often slow and experimentation is expensive if not impossible, and the data concentrates on the operation regions of plant production schedule. Therefore little information is available e.g. for extrapolation or optimization purposes. The iterative adjustment procedures may result in good reproduction of the data points, but much less can be said about the generalization properties of a particular model.

The PM provides one possible way to generate basis functions (PM followed by DM). In fact, since the PM states \mathbf{x} are often accessible as well, they could be used as a more extensive set of basis functions. However, this is likely to lead to collinearity problems and is not now pursued any further here. The PM can be expected to provide the essential nonlinear gains/dynamics of the plant. The steady state PM is a special case, leading to Wiener/Hammerstein structures, well-known and relatively commonly used in the industry due to their robustness.

III. PMFIR APPROACH

The above discussion leads to consider the following two-phase approach, Fig. 1. In the first step (CFIR, C for calibration) linear FIR filters are identified for each PM output to minimize the squared residual between FIR filter outputs and the corresponding measurements. This step provides a tuning for the physical model, both in terms of steady-state gain (affine correction) and dynamics (linear FIR). In the second step (EFIR) a FIR filter from the plant inputs to the residual error at the previous step is identified. This step provides a possibility to map unmodeled dynamics not covered by the physical model. The overall model (PMFIR) is provided by the sum of the two models.

Since the physical model (f, g) is to be evaluated “as is” with the known inputs $\mathbf{u}(k)$, the physical model is evaluated first. The physical model outputs y_{PM} are calibrated by SISO FIR filters at each output of the PM

$$z(k) = b_0 + b_1 y_{PM}(k) + \dots + b_n y_{PM}(k - n + 1) \quad (6)$$

where $y_{PM}(k)$ is one of the n_y elements of \mathbf{y}_{PM} (subscripting is omitted for simplicity). Note that the filter includes a bias term b_0 . This, together with the fact the FIR gain is not constrained, implies that the output can be scaled and shifted in an affine manner.

Each filter has $n+1$ coefficients to estimate, where n needs to be selected by the user. The estimation of the CFIR parameters from data is straightforward. If the physical model static performance is fully trusted (not to be calibrated), the steady state gain of the FIR can be constrained to 1. Since FIR parameter estimation is parsimonious [3], it can be justified to consider regularized approaches when n is considerably large. Then the goal is to minimize $(\mathbf{Y} - \Phi\theta)'(\mathbf{Y} - \Phi\theta) + \alpha\theta'\mathbf{Q}\theta$ where $\alpha \geq 0$ and \mathbf{Q} is a positive definite matrix. The solution is given by

$$\hat{\theta} = [\Phi_0'\Phi_0 + r\mathbf{Q}]^{-1} \Phi_0'\mathbf{Y}_0 \quad (7)$$

An increasing weighing for deviations from zero for increasingly delayed elements is obtained e.g. by choosing $\mathbf{Q} = \tilde{N}\text{diag}([1, 2, \dots, n])$ (in the SISO case with no b_0) and r is some small number (e.g., $r = 0.1$). A recursive parameter estimation algorithm can be devised e.g. from the Kalman filter.

The CFIR-calibrated PM can be further complemented with a linear FIR model of the remaining residuals. Consider the following MISO FIR structure with filter length n for each input:

$$z_u(k) = b_{u,0} + \dots + b_{u,n}u_1(k-n) + \dots + b_{u,1+np}u_p(k-n) \quad (8)$$

Note that the model causality is preserved by a one sample delay in inputs. The batch and recursive parameter estimation can be conducted as discussed above, with the exception that the targeted output is the residual error $r(k) = y(k) - z(k)$, instead of the measured output $y(k)$ (multiple-output indexes omitted for brevity).

The final calibrated PM outputs are then given by

$$\hat{y}_i(k) = z_i(k) + z_{u,i}(k) \quad (9)$$

The length of FIR filters is to be chosen by the modeller. This parameter can be set based on knowledge of plant settling time, or optimized using eg. the AIC. The two-phase estimation emphasizes the PM-calibration aspect of the approach. All nonlinearities in PMFIR originate from the physical model (simulator), the CFIR and EFIR corrections provide only linear tunings.

IV. FINITE STATE FIR

An additional parallel data-driven nonlinear dynamic model structure for catching possible nonlinear unmodelled dynamics in data complements the modeling tools.

In finite state modelling, the input, state and measurement spaces of f and g are discretized into a finite set of values. A state cell s is defined so that each point in the state space of interest is mapped exactly to one cell. Output cells m are defined in a similar manner. The process input actions a can

be determined in the same way, in control design the system inputs \mathbf{u} can be limited to a finite set. Mappings are built in the discretized spaces, leading to system state $(s, a) \rightarrow s'$ and measurement $s \rightarrow m$ equations:

$$\mathbf{p}(k+1) = \mathbf{P}^{a(k)} \mathbf{p}(k) \quad (10)$$

$$\mathbf{q}(k) = \mathbf{L} \mathbf{p}(k) \quad (11)$$

where $\mathbf{P}^{a(k)}$ is the $S \times S$ cell transition probability matrix given input $a(k)$, and \mathbf{L} is the $M \times S$ likelihood matrix. \mathbf{p} and \mathbf{q} represent the probability density of the state and measurement cells. The approach goes under various names, such as Markov Decision Processes (MDP), generalized cell mapping (GCM), and finite-state discrete-time Markov chains. The approach is particularly useful in modeling and analysis of nonlinear stochastic dynamic systems. Given the finite state description, very useful and exact results can be obtained on e.g. how to construct bayesian state estimates (grid-based filters) [4], devise optimal closed-loop controllers (using dynamic programming) [5] or analyze properties and behaviour of the complex dynamics [6][7][8].

The major problem is the curse of dimensionality, i.e. S will be large. In the field of approximate dynamic programming a number of techniques have been suggested to overcome the problem [5] [9] [10].

In what follows, a finite state FIR (FFIR) approach is suggested, based on maintaining a data base using a resolution set by a discretization of the input-output variables and constructing predictions using a local search. The system state is constructed from delayed inputs, in the spirit of FIR modeling. The corresponding output density is based on counts of observations, i.e. frequentist probabilistics, in the spirit of finite state systems.

The FFIR procedure is simple, straightforward, robust and easy to automate. It is a faithful representative of the family of data-driven modeling methods. The main drawback compared to alternative approaches is that the data compression can be small, which can make storage and searches in the data base resource consuming. The discretization also hides phenomena depending on signal amplitudes smaller than the discretization and brings quantization errors in the model.

Denote by Z_{u_i} ($i = 1, 2, \dots, n_u$) and Z_{y_i} ($i = 1, 2, \dots, n_y$) the number of discretized values considered for input $\mathbf{u} = [u_1, u_2, \dots, u_{n_u}]'$ and output measurement $\mathbf{y} = [y_1, y_2, \dots, y_{n_y}]'$. Discretization D_x maps each real-valued signal x into its finite-state counterpart, $a_i = D_{u_i}(a_i)$ and $m_i = D_{y_i}(y_i)$. The discretization is given by the model designer. A simple approach is to split each variable x into Z_x equidistant intervals. Each discretized cell is associated with a reference point, $y_i^{ref}(m_i)$, typically the center of the interval. In FIR modeling, the system state consists of delayed inputs $\varphi(k) = [u_1(k-1), u_1(k-2), \dots, u_1(k-n), u_2(k-1), u_2(k-2), \dots, u_{n_u}(k-n)]'$. Using the discretizations D_{u_i} it is straightforward to construct a discretization which maps each regressor $\varphi(k)$ into its cell counterpart, $s(k) = D_\varphi(\varphi(k))$.

Model building consists of counting the occurrences of

$(s(k), m(k))$ in plant measurement data. The information is stored in matrix \mathbf{C} , initially zeros, so that for cells $s(k) = i$

$$c_{j,i} = \lambda^{k-k^*} c_{j,i} + 1 \text{ if } m(k) = j \quad (12)$$

$$c_{j,i} = \lambda^{k-k^*} c_{j,i} \text{ otherwise} \quad (13)$$

λ is the forgetting factor, which implements an exponential decreasing weighing as a function of the distance between current sample k and the estimate at the last time of updating, k^* .

In this formulation the state transition probability matrices \mathbf{P}^a will be binary, consisting of the current inputs and delaying the cells associated with the past input vectors. They need not be constructed explicitly. The likelihood matrix \mathbf{L} is obtained by normalizing the column sums to one.

The matrix \mathbf{C} will be very sparse since it is very likely that most transitions will never occur. Also the size of the state space is huge and it is not possible to collect real measured data the fill the table for any table of reasonable size. Note that the state space consists of delayed inputs, so that the sparseness is not due to process characteristics. Unfortunately the simple counting formulated above is not possible to implement directly, even with sparse matrices, due to index overflow. This problem can be overcome by vectorized indexing.

In prediction, the first task is to construct the regression vector $\varphi(k)$ and find the corresponding state cell $s(k) = D_\varphi(\varphi(k))$. Two types of predictions can then be provided by a FFIR model, depending on the past data compressed in FFIR counters $c_{j,i}$.

If historical data on behaviour at state cell $s(k)$ exists, the information in the $s(k)$ 'th column of the counter \mathbf{C} can be used, and a desired prediction developed based on chosen statistics. Here, an approach is suggested where at least c_{min} samples are used to construct the estimate. If $\sum_j c_{j,i} \geq c_{min}$, the likelihood vector is given by the $s(k)$ 'th normalized column of \mathbf{C} . The mean output vector is then given by the weighted sum of reference points associated with the output cells, each cell is weighted by its likelihood vector. In a similar way estimates such as median or maximum likelihood can be devised, or the uncertainty can be expressed as quantiles, etc.

If $\sum_j c_{j,i} < c_{min}$, a local estimate is constructed instead in the neighborhood of $s(k)$. If individual signals take their values from a real interval (as they very commonly do when measuring physical quantities, such as temperatures and flows), the neighbourhood in the signal space comes out naturally. Here, an approach is suggested which finds the c_{min} closest cells to $s(k)$. The closeness is determined by the sum of absolute differences in each dimension of the discretized regression vector elements. The estimate can then be constructed as the mean, median or ML of the cell estimates.

V. NUMERICAL CSTR BENCHMARK EXAMPLE

The approach was tested using the well known nonlinear process control design benchmark of a highly nonlinear

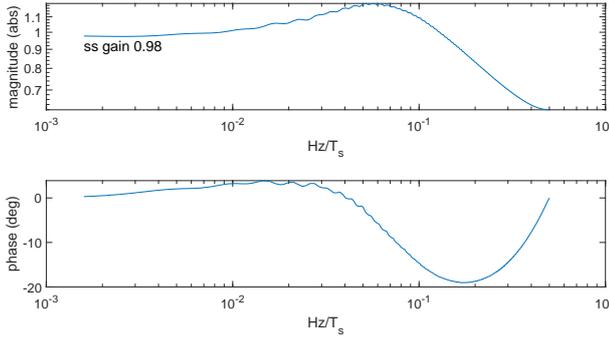


Fig. 2. PMFIR for concentration c_B : bode (CFIR) and PMFIR error distribution.

continuous stirred-tank reactor (CSTR) with a cooling jacket [11][10]. The system exhibits characteristics such as change of sign of gain in steady state behaviour and changes in zero dynamics. The control inputs are the normalized flow to the reactor \dot{V}/V_R and heat \dot{Q}_K withdrawn from the coolant by an external heat exchanger. System dynamics are given by nonlinear ode for component balances for the substances A and B, concentrations c_A and c_B ; and energy balances for the reactor and cooling jacket with temperatures ν and ν_K . The product B concentration c_B and reactor temperature ν are measured as outputs. A complete description of the equations and parameters can be found from [11].

The system was simulated for one year operation with a sampling interval of 20 seconds, resulting in 1578800 input-output data pairs. The inputs consisted of random ramps with set point changes at both inputs with a probability of once/5 hour and a random ramp length in the range from 20 seconds to 20 minutes. The amplitude of the ramps ranged between minimum and maximum values, $\dot{V}/V_R \in [3, 35]$ and $\dot{Q}_K \in [-9000, 0]$. Both output measurements were corrupted by additive zero mean noise with standard deviation of corresponding to 1% of the range of the signal

The PM model was taken to be the worst case scenario (see benchmark) while the data was generated with the nominal plant model. A FIR filter length of $n = 180$ samples was used for all filters, corresponding to a window of one hour. 10% of the data was left for test data, i.e. not used in parameter estimation. A total of $6n + 4 = 1084$ parameters were estimated from data. All estimations were conducted by first normalizing the data to zero mean unit variance and using a regularized LS with $r = 0.1$.

The estimated PMFIR filter for c_B is illustrated in Figs. 2–3, for ν similar figures can be drawn. Fig. 2. shows the CFIR bode diagram with absolute magnitude and phase in degrees. Fig 3. shows the cumulative sum of CFIR filter coefficients (step response), as well as the EFIR filters from the two inputs \dot{V}/V_R and \dot{Q}_K .

The impact of regularization was significant in reducing the ripple in FIR parameters. The PMFIR filter for the concentration of component B (Fig. 2) shows a slight lead of PM by the CFIR, the steady state gain is close to 1 (0.98). For the correction of the reactor temperature the

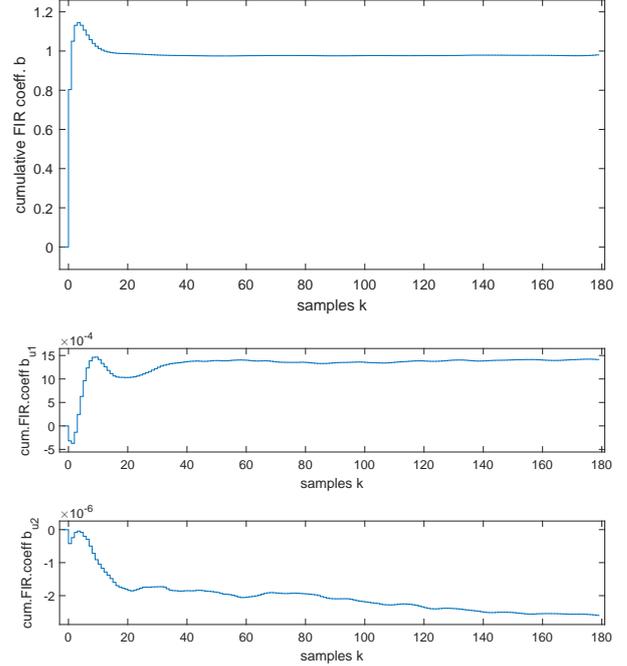


Fig. 3. PMFIR for concentration c_B : step responses CFIR (top) and EFIR (bottom).

CFIR resembled a lag filter. The static gain was 1.04, i.e. a 4% correction to PM is made. The interpretation of EFIR cumulated coefficients (Fig. 3) requires to take into account the scales of the inputs and outputs. In general, they seem to indicate that there is a relatively larger correction from input feed to c_B , similar observation was made for the adjustment from heat removal to temperature.

Root-mean-squared errors (RMSE) were computed on training and test data (see Table 1). There was a slight increase for c_B on test set, but an overall conclusion can be made that the impact of any noise captured to the PMFIR is insignificant. Some unexplained RMSE remained for c_B but the RMSE on ν can be explained by the measurement noises (standard deviations 0.013 and 1.09, respectively). It can be concluded that the calibrated PM now represents well the real plant.

The FFIR model was constructed using the same one year data as with PMFIR and the same filter lengths. The I/O variables were discretized into 100 equidistant intervals each, using the feasible data ranges from the benchmark description: $\dot{V}/V_R \in [3, 35]$, $\dot{Q}_K \in [-9000, 0]$, $c_B \in [0, 1.4]$ and $\nu \in [0, 130]$. The algorithm passed through the one year data, resulting in 512492 stored cells (33 %). A mean estimate for the output was maintained at each observed cell, as well as estimates on the quantiles, medians and ML values.

The model predictions were computed using four different values for the minimum sample size / neighborhood, $c_{min} \in \{1, 4, 25, 100\}$. A local model (constant) for c_B and ν was constructed in the neighborhood. In general, the procedure was time consuming, especially for deriving the predictions. The computing times were uncomfortable, but

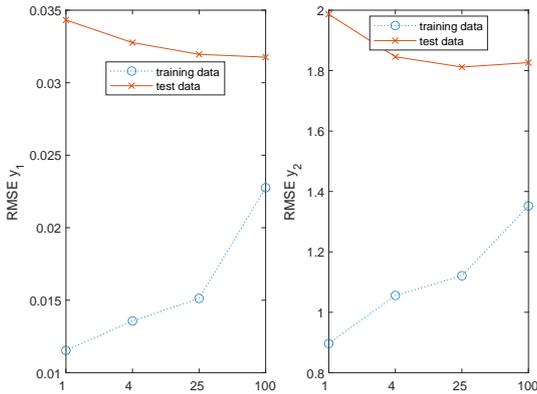


Fig. 4. FFIR RMSE for c_B (left) and ν (right) as a function of smoothing.

remained feasible even for many on-line purposes such as in monitoring and control.

Figure 4 illustrates the RMSE on training and test data predictions, as a function of the smoothing parameter c_{min} . The left plot shows the statistics for c_B , the right plot for ν . Both plots show the RMSE vs. c_{min} . The results on training data show that the RMSE are the smallest with $c_{min} = 1$, as expected. Since all data patterns are coded in the FFIR model, any smoothing can only worsen the prediction accuracy. The statistics on test set illustrate the generalization ability. The outcomes show surprisingly flat RMSE. This indicates that the balancing between measurement noise and mapping nonlinearity is not much affected by this parameter. The results suggest that even a simple nearest neighbor approach could be applied ($c_{min} = 1$), with the advantages of simplicity and speed.

It is interesting to compare the FFIR RMSEs to those obtained with the PMFIR. Table 1 indicates that the FFIR does a good job on training set, with $c_{min} = 1$ the RMSE are the smallest. The PMFIR corrects the mismatch between data and PM using a global linear calibration, while the local nonlinearities pose no problems to FFIR relying purely on observed data, given that sufficient data exists. The bottom rows in Table 1 give also the RMSE vs. uncalibrated (PM) and linear static output calibration. It is clear that dynamic calibration improves the prediction quality.

However, results on comparison of PMFIR and FFIR statistics must be interpreted recalling that a PM can be expected to extrapolate much better to operation regions not visited in real life, i.e. for which no data is available.

TABLE I
RMSE ON TEST AND TRAINING SETS

model/data	c_B training	c_B test	ν training	ν test
PMFIR	0.026	0.027	1.1	1.1
FFIR	0.012-0.023	0.031-0.034	0.090-1.35	1.8-2.0
PMFIR 2 mth	0.024	0.021	1.0	1.0
FFIR 2 mth	0.011-0.024	0.049-0.050	0.83-1.32	4.2
PM	0.039	0.042	2.8	2.7
PM with linear	0.030	0.031	1.2	1.3

On the other hand, the data-driven model will improve its performance as more data is collected, whereas the linear mapping has much less degrees of freedom to exploit the additional data. Both algorithms are simple to implement in a recursive form.

A second set of data was generated for one month data, to analyze the data quality requirements of the considered PMFIR and FFIR approaches. In both cases, all parameter values remained the same, the only difference being the data. For PMFIR Table 1 indicates that the decrease of the number of data points had no significant impact to RMSE. This was expected, as the main performance is given by the PM, and the PMFIR provides only a linear calibration. The FFIR is much more sensitive to the amount and quality of data. The RMSE errors on training/test data are clearly much larger than those based on one year data. While the measurement noise can be dealt with by increasing c_{min} , the fundamental problems on lack of data on the function mapping can not. This underlines the correctness of our hypothesis that even a miscalibrated physical model can provide a highly valuable component in process modelling.

VI. CONCLUSIONS

On-going work focuses at robust estimation of linear dynamics and ways of merging predictions from several plant models. In applied work the focus is in gaining experiences on the potential in applications in the heavy process industry. Two particularly interesting applications of plant models are in on-line state estimation/monitoring and control design/automatic feedback control.

ACKNOWLEDGMENT

The work in this paper was partly funded by the H2020 project COGNITWIN (grant number 870130).

REFERENCES

- [1] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Dylon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, *Automatica*, vol. 31, pp. 1691-1724, 1995.
- [2] E. Ikonen and K. Najim, *Advanced Process Identification and Control*, New York: Marcel Dekker, 2002.
- [3] M. Nikolauou and P. Vuthandam, FIR model identification: Parsimony through Kernel compression with Wavelets, *AICHE Journal*, vol. 44 pp. 141-150, 1998.
- [4] S. Ungarala, Z. Chen and K. Li, Bayesian state estimation of nonlinear systems using approximate aggregate Markov chains, *Ind. Eng. Chem. Res.*, vol. 45 pp. 4208-4221, 2006.
- [5] W. J. Powell, *Approximate Dynamic Programming - Solving the curses of dimensionality*. Wiley-Interscience, 2010.
- [6] J. Kemeny and J. Snell, *Finite Markov Chains*. New York: van Nostrand, 1960.
- [7] C. S. Hsu, *Cell-to-Cell Mapping - A method of global analysis for nonlinear systems*, Springer Verlag: 1987.
- [8] Z. Li, J. Jiang, L. Hong and J.-Q. Sun, On the data-driven generalized cell mapping method, *Int. J. Bifurcation and Chaos*, vol. 29 pp. 2019.
- [9] J.H. Lee, W. Wong, Approximate dynamic programming approach for process control, *Int. J. Process Control*, vol. 20 pp. 1038-1048, 2010.
- [10] E. Ikonen, I. Selek and K. Najim, Process control using finite Markov chains with iterative clustering, *Computers and Chemical Engineering*, vol. 93 pp. 293-308, 2016,
- [11] H. Chen, A. Gremling and F. Allgöwer, Nonlinear predictive control of a benchmark CSTR. In *European Control Conference 1995*, pp. 3247-3252.