



Empirical Modeling of Heating Element Power for the Czochralski Crystallization Process

M. Komperød¹ B. Lie²

¹*Faculty of Engineering, Østfold University College, N-1757 Halden, Norway. E-mail: magnus.komperod@hiof.no*

²*Faculty of Technology, Telemark University College, N-3901 Porsgrunn, Norway. E-mail: bernt.lie@hit.no*

Abstract

The Czochralski (CZ) crystallization process is used to produce monocrystalline silicon. Monocrystalline silicon is used in solar cell wafers and in computers and electronics. The CZ process is a batch process, where multicrystalline silicon is melted in a crucible and later solidifies on a monocrystalline seed crystal. The crucible is heated using a heating element where the power is manipulated using a triode for alternating current (TRIAC). As the electric resistance of the heating element increases by increased temperature, there are significant dynamics from the TRIAC input signal (control system output) to the actual (measured) heating element power. The present paper focuses on empirical modeling of these dynamics. The modeling is based on a dataset logged from a real-life CZ process. Initially the dataset is preprocessed by detrending and handling outliers. Next, linear ARX, ARMAX, and output error (OE) models are identified. As the linear models do not fully explain the process' behavior, nonlinear system identification is applied. The Hammerstein-Wiener (HW) model structure is chosen. The final model identified is a Hammerstein model, i.e. a HW model with nonlinearity at the input, but not at the output. This model has only one more identified parameter than the linear OE model, but still improves the optimization criterion (mean squared ballistic simulation errors) by a factor of six. As there is no nonlinearity at the output, the dynamics from the prediction error to the model output are linear, which allows a noise model to be added. Comparison of a Hammerstein model with noise model and the linear ARMAX model, both optimized for mean squared one-step-ahead prediction errors, shows that this optimization criterion is 42% lower for the Hammerstein model. Minimizing the number of parameters to be identified has been an important consideration throughout the modeling work.

Keywords: Czochralski Crystallization Process, Empirical Modeling, Hammerstein-Wiener Model, Heating Element, Nonlinear System Identification.

1 Introduction

The Czochralski (CZ) crystallization process was invented by the Polish scientist Jan Czochralski in 1916. The process is used to convert multicrystalline materials into monocrystalline materials, i.e. materials that have homogeneous crystal structures. Among the most important applications is production of monocrystalline silicon. Monocrystalline silicon is used in solar cell wafers and in computers and electronics. Solar cells

based on monocrystalline wafers have higher efficiency than those based on multicrystalline wafers. Hence, the CZ process is an important part of the solar cell industry.

The CZ process is a batch process. During the process multicrystalline silicon is melted in a crucible. The crucible is heated using a heating element. Tight control of the crucible temperature is most important for achieving high crystal quality. The heating element

power is manipulated using a triode for alternating current (TRIAC). As the electric resistance of the heating element increases with the temperature, there is a dynamic, nonlinear relationship between the TRIAC input signal and the heating element power.

System identification is the science of developing *dynamic* models based on observations of the process or system to be modeled. The identified models explain the process outputs as mathematical functions of the process inputs.

The contribution of this paper is to model the dynamics from the TRIAC input signal to the heating element power using system identification. The modeling work is based on a dataset logged at a real-life CZ process at SINTEF Materials and Chemistry in Trondheim, Norway. Initially, linear system identification is used. The identified linear models reveal that the process is nonlinear. Next the process is modeled using nonlinear system identification. Deciding a model structure that explains the process behavior using few parameters is emphasized. Also, providing the identification algorithm good initial values for the parameters to be identified is considered.

As the identified model provides a mathematical description of how the heating element power responds to the TRIAC input signal, the model serves several purposes. The most intuitive application is to simulate the power for specified sequences of the TRIAC input signal. The model is also most useful for analyzing the process' dynamic properties. Process models may be used to tune PID controllers. As the process is nonlinear, it may be desirable to use gain scheduling. The nonlinear process model contains information of which PID parameters to use for each interval in the gain scheduling scheme. If more advanced model-based control strategies are to be applied, such as model predictive control (MPC), process models are most important. In case of a power sensor failure, the model may be used to simulate the power for the purpose of process monitoring and control.

The literature of system identification is extensive. Among the most well-known books is Ljung (1999). This book gives a comprehensive introduction to system identification. Both theoretical and practical aspects are covered. Ljung (1999) also serves as an overview of the system identification literature.

Lee et al. (2005) presents an approach for batch-to-batch optimization of the CZ process, which includes model-based control. The paper includes two simple dynamic models empirically developed from step responses. However, these models cover different parts of the CZ process than the present paper. Except for Lee et al. (2005), the authors of the present paper have not been successful in finding any publications that present

results within empirical modeling of the CZ process. Several papers with focus towards mechanistic (first principle) modeling of the CZ process have been found. However, none of these papers seem to have validated the mechanistic models against datasets from real-life processes.

The authors have searched for literature covering modeling of heating element dynamics in general. Unfortunately, no interesting results were found.

2 Notations and Definitions

Table 1 presents the notations used in this paper. A variable with subscript k refers to the variable's value at timestep k . For example P_k refers to the heating element power at timestep k . Subscript "nom" refers to the variable's nominal value. The operating point $(s_{\text{nom}}, P_{\text{nom}})$ is defined to be a steady state operating point.

Polynomials to be used in linear polynomial models are defined as

$$A(q) \stackrel{\text{def}}{=} 1 + \sum_{i=1}^{n_a} a_i q^{-i}, \quad (1)$$

$$B(q) \stackrel{\text{def}}{=} \sum_{i=1}^{n_b} b_i q^{-i}, \quad (2)$$

$$C(q) \stackrel{\text{def}}{=} 1 + \sum_{i=1}^{n_c} c_i q^{-i}, \quad (3)$$

$$F(q) \stackrel{\text{def}}{=} 1 + \sum_{i=1}^{n_f} f_i q^{-i}. \quad (4)$$

The parameters a_i , b_i , c_i , and f_i are to be identified using system identification. The constants n_a , n_b , n_c , and n_f are to be specified to the system identification algorithm. There is a time delay of one sample in the dataset to be used in this paper. For processes without any time delay, the lower summation limit of $B(q)$ should be 0 instead of 1. The $A(q)$, $C(q)$, and $F(q)$ polynomials are defined as above in any case.

The linear polynomial models that will be considered in this paper are based on the ARX model structure

$$A(q)y_k = B(q)u_k + e_k, \quad (5)$$

the ARMAX model structure

$$A(q)y_k = B(q)u_k + C(q)e_k, \quad (6)$$

and the output error (OE) model structure

$$y_k = \frac{B(q)}{F(q)}u_k + \varepsilon_k. \quad (7)$$

Table 1: Notations used in this paper.

$e(\theta)$	One-step-ahead prediction error.
k	Index referring to sample number in the dataset.
N	The total number of samples in the dataset.
P	Actual (measured) power to the heating element [kW].
P_{nom}	Nominal power to heating element [kW] (see main text for explanation).
ΔP	Actual (measured) power to the heating element [kW] as deviation from the nominal power, i.e. $\Delta P = P - P_{\text{nom}}$.
q	The time-shift operator defined by $x_{k+1} = qx_k$ and $x_{k-1} = q^{-1}x_k$.
s	TRIAC input signal [%] (output from control system).
s_{nom}	Nominal TRIAC input signal [%] (see main text for explanation).
Δs	TRIAC input signal [%] as deviation from the nominal input signal, i.e. $\Delta s = s - s_{\text{nom}}$.
t	Time [s], relative to beginning of the CZ batch.
u	System input for a general system.
$V(\theta)$	Criterion to be optimized when using the system identification method PEM. This criterion is based on ballistic simulation errors. Please refer to Section 4 for further explanation.
$W(\theta)$	Criterion to be optimized when using the system identification method PEM. This criterion is based on one-step-ahead prediction errors. Please refer to Section 4 for further explanation.
y	System output for a general system.
$\varepsilon(\theta)$	Ballistic simulation error.
θ	A vector containing the parameters to be identified using PEM. Please refer to Section 4 for further explanation.

The term noise model refers to the dynamics from the one-step-ahead prediction error, e , to the system output, y . Solving (5) and (6) with respect to y_k shows that the noise models of ARX and ARMAX are $1/A(q)$ and $C(q)/A(q)$, respectively. The OE model has no noise model. Model structures having noise models, such as ARX and ARMAX, are suitable for n -step-ahead predictions, because the noise models correct the model states when there are errors between the measured process outputs and the simulated model outputs. Model structures having no noise model, like OE, are only suitable for ballistic simulations, not n -step-ahead predictions.

The term input-output model will be used to refer to the dynamics from the system input, u , to the system output, y . With respect to the input-output model, $A(q)$ of the ARX and ARMAX model structures is equivalent to $F(q)$ of the OE model structure.

Text written in `teletype` font refers to MATLAB commands, for example `pem`.

3 The Czochralski Crystallization Process

The Czochralski (CZ) crystallization process is used to convert multicrystalline materials into monocrystalline materials. The process considered in this paper converts multicrystalline silicon into monocrystalline silicon. Monocrystalline silicon is used in solar cell wafers and in computers and electronics. Monocrystalline silicon wafers give solar cells with higher efficiency than multicrystalline silicon wafers.

The CZ process is a batch process of which main steps are illustrated in Figure 1. (i) Initially multicrystalline silicon is melted in a crucible. (ii) When the silicon is molten, the tip of a seed crystal is dipped into the melt. The seed crystal is monocrystalline and has the crystal structure that is to be produced. (iii) When the tip of the seed crystal begins to melt, the crystal is slowly elevated. As the crystal is lifted, the molten silicon solidifies on the crystal. (iv) The crystal grows radially and axially. The produced crystal is referred to as an ingot. The crucible temperature and the vertical pulling speed are used to control the ingot diameter. Stable growing conditions are necessary to produce high crystal quality. (v) As the final ingot length is reached, the crystal growth is terminated by slowly decreasing the crystal diameter to zero. During the entire batch process, the crucible is rotated in one

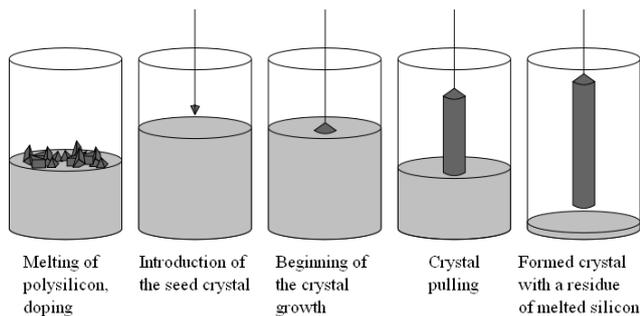


Figure 1: The main batch steps of the CZ process. Illustration from Wikipedia (the illustration is released to public domain by the copyright holder).

direction, and the seed crystal is rotated in the other direction.

SINTEF Materials and Chemistry in Trondheim, Norway, owns and operates a real-life CZ process. At this plant the crucible is heated using a cylinder-shaped heating element, which encircles the crucible. The heating element power is manipulated using a TRIAC. This paper considers empirical modeling of the dynamics from the TRIAC input signal, s , to the heating element power, P , based on a dataset logged from this plant.

The dataset that is used for empirical modeling is shown in Figure 2. The experiment was done without the crucible mounted in the process. The figure shows that increased s gives instantaneously increased P . However, as the temperature increases, the electric resistance of the heating element increases, causing decreased P . As shown later in this paper, these dynamics are nonlinear.

The authors have access to only one dataset that is considered to be suitable for the modeling work. This dataset will be used for model identification as well as model validation. The authors have chosen to use the entire dataset for both identification and validation. The reason for not splitting the dataset into an identification section and a validation section is that s tends to increase by time throughout the dataset. Hence, splitting the dataset in two halves will leave a small range of excitations in each half. As concluded in Section 6, the process is almost linear close to an operating point. Therefore, a large range of excitations is desirable to provide the identification algorithm sufficient information about the process' nonlinearity.

Figure 2 shows that P responds in a similar way for any step in s . This gives confidence in that the responses of P are caused by excitations of s , not by process disturbances nor measurement noise. However, as the models are not validated on independent data,

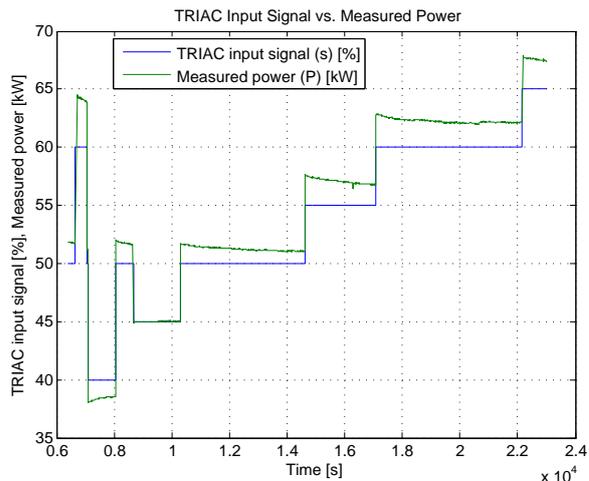


Figure 2: The dataset to be used for system identification.

it is most important to limit the number of parameters to be identified in order to avoid overfitting. The sampling time of the dataset is 2 seconds.

4 Prediction Error Method

System identification is the science of developing *dynamic* models based on observations of system inputs and system outputs. There exist many different system identification algorithms based on various mathematical approaches. This paper will be restricted to the prediction error method (PEM). PEM is one of the most used system identification approaches. When identifying a model using PEM, the identification algorithm is provided a model structure with one or more unknown parameters to be identified. The unknown parameters are stacked in a parameter vector θ . For example when identifying an ARMAX model with $n_a = 1$, $n_b = 2$, and $n_c = 1$ the parameter vector is

$$\theta = \begin{bmatrix} a_1 \\ b_1 \\ b_2 \\ c_1 \end{bmatrix}. \quad (8)$$

PEM computes the parameter vector that gives the least difference between the real process output and the simulated model output. What exactly is meant by “least difference” must be specified by a mathematical optimization criterion. This paper will consider two different criteria:

$$V(\theta) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=1}^N \varepsilon_k(\theta)^2, \quad (9)$$

$$W(\theta) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=1}^N e_k(\theta)^2. \quad (10)$$

When optimizing for ballistic simulations, $V(\theta)$ is used. When optimizing for one-step-ahead predictions, $W(\theta)$ is used. In this paper, unless otherwise specified, models with noise models are optimized with respect to $W(\theta)$, and models without noise models are optimized with respect to $V(\theta)$.

PEM can be used to estimate parameters in both linear and nonlinear model structures. If the model structure is developed based on mechanistic (first principle) modeling of the process, the model is referred to as a grey-box model, as it combines the principles of black-box modeling with process knowledge.

From a mathematical point of view, PEM is an optimization problem of which $V(\theta)$ or $W(\theta)$ is to be minimized with respect to θ . In most cases this optimization problem must be solved iteratively. A poor initial value of θ may cause the optimization algorithm to be trapped in a local minimum. It is therefore desirable for the model developer to find good initial values. ARX models can be identified using the ordinary least squares (OLS) method and are therefore not at the risk of being trapped in a local minimum.

Using system identification, models can in principle be developed without having any knowledge of the process at all, except for datasets of logged process inputs and outputs. However, in practice, basic knowledge of the process greatly helps the model developer during data preprocessing, model identification, and model validation.

When developing models using system identification, one should be aware of the limitation that the model may perform poorly outside the ranges of input and output values of the calibration and validation datasets. Even if the model structure has been developed using mechanistic (first principle) modeling, the model structure is often based on assumptions or simplifications that may not hold for larger ranges of input and output values. Also the estimation error of a parameter (the difference between the “true” parameter value and the estimated parameter value) may have larger impact for input and output values outside the calibration and validation ranges.

5 Data Preprocessing

Datasets logged from real-life processes often need preprocessing before they can be used for empirical mod-

eling and model validation. The dataset to be used in this paper has been preprocessed in two ways: (i) data detrending and (ii) outlier detection.

5.1 Data Detrending

The dataset to be used in this paper has been detrended by subtracting a steady state operating point, $(s_{\text{nom}}, P_{\text{nom}})$, from the raw data. The detrended dataset, $\Delta s = s - s_{\text{nom}}$ and $\Delta P = P - P_{\text{nom}}$, then refers to deviations from the steady state operating point. In particular when identifying linear models, subtracting a steady state operating point is desirable. The resulting linear model is then equivalent to a linearization (first order Taylor expansion) around the operating point.

If a steady state operating point is not known to the model developer, a commonly used approximation is the sample means, i.e. $s_{\text{nom}} = \frac{1}{N} \sum_{k=1}^N s_k$ and $P_{\text{nom}} = \frac{1}{N} \sum_{k=1}^N P_k$. For this particular dataset, Figure 2 shows that the process is very close to steady state in the time interval $t \in [8700, 10300]$. As the dataset happens to reveal a steady state operating point, this operating point is used for detrending, instead of the sample means. The steady state values found are $s_{\text{nom}} = 45\%$ and $P_{\text{nom}} = 45\text{kW}$. Please note that it is a coincidence that s_{nom} and P_{nom} happen to have the same value in the steady state operating point. This is not the case in general.

5.2 Outlier Detection

The term outlier refers to a sample, or a segment of samples, in the dataset that is not representative for the process’ behavior. It is therefore desirable to exclude the outliers from the dataset during model identification and model validation.

It is not trivial to decide which samples that are outliers. It may be intuitive to look for extreme values, i.e. samples of which values are significantly higher or lower than the other samples in the dataset. However, this approach is not reliable. A sample of extreme value may be real, while a non-extreme value may be an outlier. The latter case is shown for real-life industrial data in Komperød et al. (2008).

Deciding whether a sample is an outlier or not is to a large degree a subjective choice which requires practical knowledge of the process to be modeled. However, mathematical algorithms may be used to identify samples that are candidates for being outliers. An approach for outlier detection presented in Komperød et al. (2008) was used at the dataset shown in Figure 2: A high order ARX model with $n_a = 10$ and $n_b = 10$ was identified. Figure 3 shows the residual plot (one-step-ahead prediction errors), e , of this model. In general, samples having large residuals (absolute values)

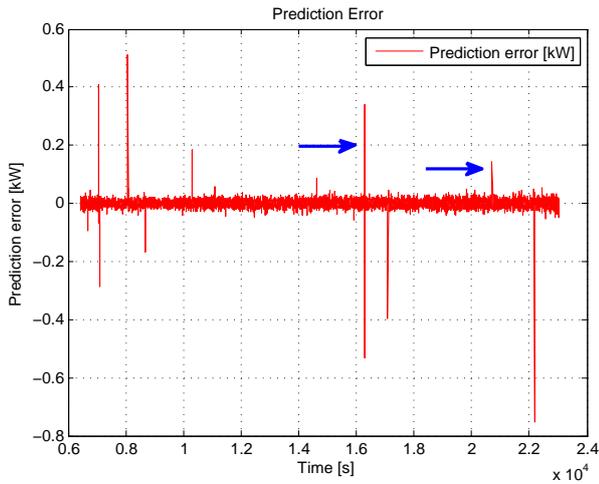


Figure 3: Residual plot of an ARX model with $n_a = 10$ and $n_b = 10$.

and their neighbor samples are candidates for being outliers. There are several large residuals in Figure 3. Plotting the residuals in the same plot as the heating element power, P , reveals that most of the large residuals correspond to steps in the heating element power (this plot is not shown). Hence, these residuals are most likely caused by model imperfection, rather than outliers. Only two large residuals do not correspond to power steps. These residuals are marked by blue arrows in Figure 3.

Figure 4 and Figure 5 show the residuals, e , and the measured power, ΔP , zoomed in close to the residuals marked by the left-most arrow and the right-most arrow in Figure 3, respectively. In Figure 4 there is one sample in ΔP (lower subplot), which value is significantly lower than the neighbor samples. This sample value can not be explained by the TRIAC input signal, s . The reason for this sample value is not known for sure. A reasonable explanation is voltage variations on the power grid. A voltage drop of short duration can occur for example when starting an electric motor. Please note from the lower subplot in Figure 4 that the power does not completely restore after the power drop. This observation strengthens the assumption that a larger load on the power grid was activated. The outlier is handled by replacing the extreme value sample by a linear interpolation between the two neighboring samples.

The residual in the upper subplot of Figure 5 is of a different nature than the one in Figure 4. In Figure 5 there are positive residuals for two to three samples, but no negative residual. The lower subplot shows that the power is lifted to a higher level and does *not* go back to the initial level. A reasonable explanation is that a

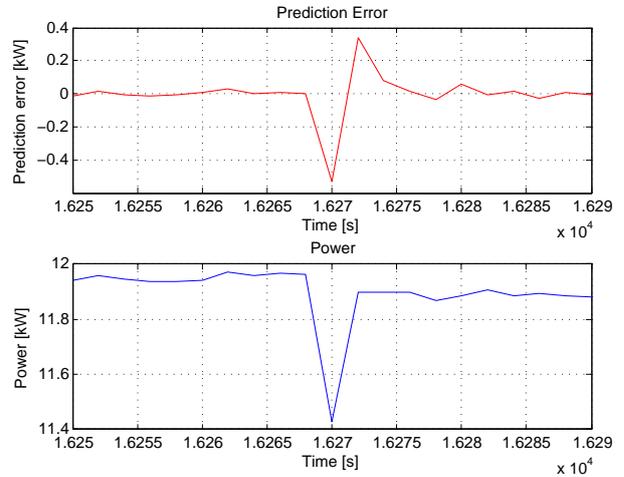


Figure 4: Residuals, e , (upper) and measured power, ΔP , (lower) zoomed in at the left-most arrow in Figure 3.

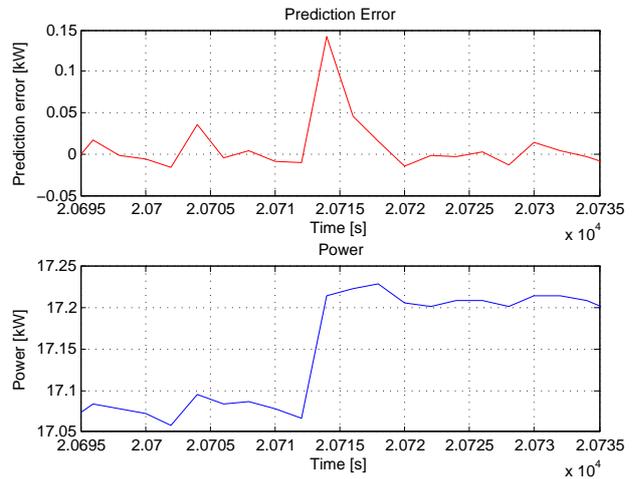


Figure 5: Residuals, e , (upper) and measured power, ΔP , (lower) zoomed in at the right-most arrow in Figure 3.

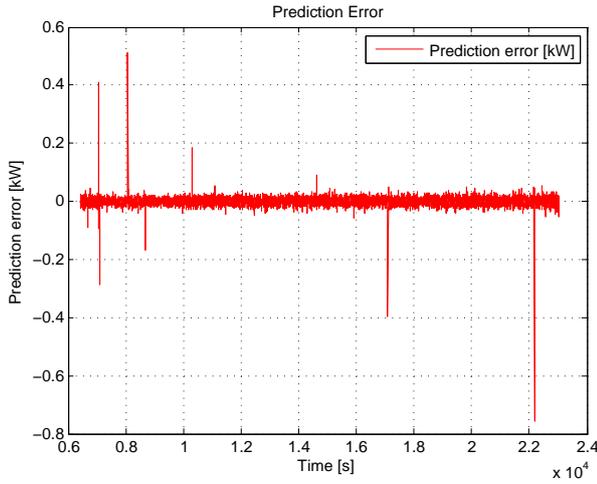


Figure 6: Residual plot after handling outliers.

large load at the power grid was shut down, increasing the grid voltage. This outlier is handled by subtracting 0.15kW from $\Delta P(t) \forall t \geq 20714s$.

After the outliers were handled, a new ARX model of the same polynomial orders was identified. The residual plot of this model is shown in Figure 6. The figure shows that the residuals marked by blue arrows in Figure 3 are no longer present.

Figure 7 shows the dataset after being preprocessed. This dataset will be used for linear and nonlinear system identification in Section 6 and Section 7, respectively.

6 Linear System Identification

For many processes it is not trivial to decide whether or not the process can be approximated by a linearization within the operating range of interest. It therefore makes sense to try linear models at first, as these models are simpler than nonlinear models. Even if the process is somewhat nonlinear, a linear model may be useful for understanding the process’ basic dynamics, such as model order and dominant time constant. Understanding these basic dynamics may be helpful if developing more complex, nonlinear models later.

In this section linear ARX, ARMAX, and OE models are identified. These three model structures have identical input-output model structures, but different noise models. The dataset to be used for system identification is the dataset shown in Figure 7. Before identification, the dataset has been preprocessed as explained in Section 5.

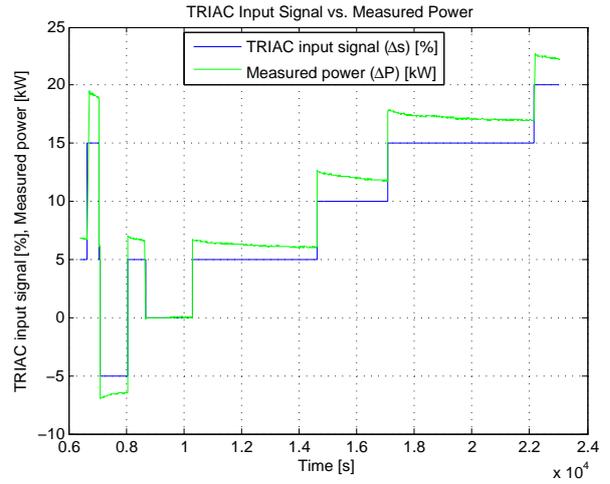


Figure 7: The dataset after being preprocessed. It is now ready to be used for linear and nonlinear system identification.

6.1 Deciding Polynomial Orders

A most important choice is how to select the polynomial orders n_a , n_b , n_c , and n_f in ARX, ARMAX, and OE models. In particular when there is no independent dataset available for model validation, the model developer should limit the number of parameters to identify in order to avoid overfitting. The polynomial orders to be used in this paper will now be derived based on the plot in Figure 7. For each positive step in the TRIAC input signal, Δs , the heating element power, ΔP , does an instantaneous positive step and then slowly decreases. The decrement will be approximated as a first order response with negative gain. There is a time delay of one sample from Δs to ΔP . The dynamics from Δs to ΔP are then given on the form

$$\Delta P_k = \left(\underbrace{\tilde{b}_2 q^{-1}}_{\text{step}} + \underbrace{\frac{-\tilde{b}_1 q^{-1}}{1 + a_1 q^{-1}}}_{\text{decrease}} \right) \Delta s_k, \quad (11)$$

where \tilde{b}_1 and \tilde{b}_2 are temporary parameters. On the right hand side of (11), the first term represents the instantaneous step and the second term represents the subsequent decrease. Equation (11) can be rewritten as

$$\Delta P_k = \frac{\tilde{b}_2 q^{-1} + \tilde{b}_2 a_1 q^{-2} - \tilde{b}_1 q^{-1}}{1 + a_1 q^{-1}} \Delta s_k. \quad (12)$$

Introducing $b_1 = \tilde{b}_2 - \tilde{b}_1$ and $b_2 = \tilde{b}_2 a_1$ gives

$$\Delta P_k = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1}} \Delta s_k. \quad (13)$$

This is the standard form for input-output models of polynomial models. Hence, the chosen polynomial orders for the ARX and ARMAX models are $n_a = 1$ and $n_b = 2$. As n_c of the ARMAX model is not given, this is chosen equal to n_a , i.e. $n_c = n_a = 1$. Equivalent, for the OE model the polynomial orders are $n_b = 2$ and $n_f = 1$. An ARX model, an ARMAX, and an OE model were identified using these polynomial orders. The identifications were performed using the commands `arx`, `armax`, and `oe` of the MATLAB System Identification Toolbox. A time-delay of one sample was specified to the identification algorithms, otherwise the default settings were used.

6.2 Model Validation and Discussions

Figure 8 shows ballistic simulations of the identified ARX, ARMAX, and OE models. In general, the models give small simulation errors close to the operating point ($s_{\text{nom}}, P_{\text{nom}}$), but give a somewhat poorer fit otherwise. Notice in particular the last step at $t = 22170$ s: Right before the step all the model simulations are significantly below the measured value. Right after the step all the simulations are significantly above the measured value. However, these simple linear models catch the basic dynamics of the process. The models may be usable for some applications, such as model-based PID controller tuning provided that the gain and phase margins are chosen sufficient large.

To examine whether the input-output model structure of (13) has sufficient polynomial degrees, n_a , n_b , n_c , and n_f were all increased by one to examine whether this improved the models' performance. Table 2 shows the values of the optimization criterion $V(\theta)$ of (9) for the models used in Figure 8 and for models with higher polynomial orders. The table reveals two interesting results: (i) According to $V(\theta)$, the OE models perform much better than the ARX and ARMAX models. This is to be expected as OE is optimized for $V(\theta)$ (ballistic simulation), while ARX and ARMAX are optimized for $W(\theta)$ (one-step-ahead prediction). (ii) Increasing the polynomial orders does not significantly improve the optimization criterion. Hence, it is concluded that the polynomial orders of (13) are sufficient. For the ARX and ARMAX models, increasing the polynomial orders actually gives slightly poorer performance. In the succeeding text, only the models of lower polynomial orders, i.e. rows 1, 3, and 5 of Table 2, will be considered.

Figure 9 shows the ballistic simulation errors, $\varepsilon(\theta)$, of the ARX, ARMAX, and OE models, i.e. the difference

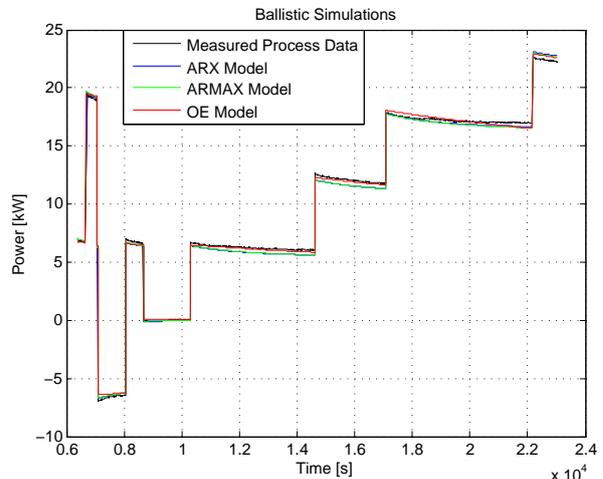


Figure 8: Ballistic simulations using the ARX model ($n_a = 1, n_b = 2$), ARMAX model ($n_a = 1, n_b = 2, n_c = 1$), and OE model ($n_b = 2, n_f = 1$).

Table 2: Optimization criterion for various model structures and polynomial orders.

Model	n_a	n_b	n_c	n_f	$V(\theta)$
ARX	1	2	-	-	0.106
ARX	2	3	-	-	0.109
ARMAX	1	2	1	-	0.108
ARMAX	2	3	2	-	0.110
OE	-	2	-	1	0.043
OE	-	3	-	2	0.042

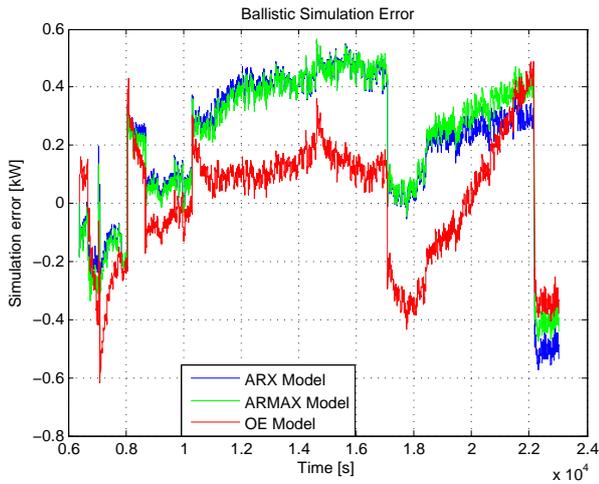


Figure 9: Ballistic simulations error using the ARX, ARMAX, and OE models.

between the measured data and the simulated data of Figure 8. A horizontal curve in Figure 9 indicates that the simulated data increase or decrease at the same rate as the measured data. An increasing curve shows that the simulated data increase too slowly or decrease too fast.

For all three models there are significant steps in Figure 9 at the steps in Δs . The most interesting result of Figure 9 is that the simulation error of the OE model is in general closer to zero than the other models, i.e. has better fit according to the criterion $V(\theta)$, but at the same time has a steeper curve in large parts of Figure 9. In particular in the time intervals $t \in [7116, 8076]$ and $t \in [17098, 22170]$ the OE model has a very steep curve compared to the other models. Hence, it seems that the ARX and ARMAX models explain the *dynamics* of the process more accurately than the OE model, but still give a poorer performance at ballistic simulation. During one-step-ahead predictions, which ARX and ARMAX are optimized for, the noise models handle the offsets that are caused by inaccurate explanation of the steps. However, as the noise models are to no avail during ballistic simulations, the good explanations of the dynamics do not fully compensate for the poorer explanations of the steps. The OE model on the other hand is optimized to balance these two considerations during ballistic simulations.

For the purpose of model-based tuning of a controller that should have a constant setpoint, i.e. constant operating point, it seems reasonable to recommend the ARX and ARMAX models over the OE model, as the ARX and ARMAX models seem to explain the dynamic more accurately.

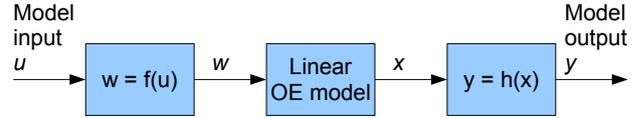


Figure 10: The Hammerstein-Wiener model structure as defined in the MATLAB System Identification Toolbox. The model structure has no noise model.

7 Nonlinear System Identification

Linear process models of the dynamics from Δs to ΔP were developed in Section 6. These models seem to explain the process dynamics well close to an operating point, but handle steps in Δs somewhat poorly.

This section will focus on nonlinear system identification of the dynamics from Δs to ΔP . The motivation for using nonlinear system identification is that the simulation errors of the linear models indicate that the process is nonlinear. The dataset to be used is shown in Figure 7. The dataset has been preprocessed as described in Section 5. As there is no independent dataset available for validation, an important consideration is to limit the number of parameters to be identified in order to avoid overfitting.

7.1 The Hammerstein-Wiener Model Structure

There are several modeling approaches and model structures that can be used for nonlinear system identification. In this paper a model structure referred to as Hammerstein-Wiener (HW) will be used. The motivations for choosing the HW model structure are: (i) The model structure is very simple and easy to understand. (ii) Linear system theory can be applied to analyze the models' stability properties.

The HW model structure is something between linear and nonlinear models. The core of the HW model is a linear model. As HW models are defined in the MATLAB System Identification Toolbox, the linear model is an OE model. The input to the OE model is processed by a static, nonlinear function $f(u)$. The output from the OE model is processed by another static, nonlinear function $h(x)$. The HW model structure is illustrated in Figure 10. A model having nonlinear processing of the input only, not the output, is referred to as a Hammerstein model. A model having nonlinear processing of the output only, not the input, is referred to as a Wiener model (Ljung, 1999, 2009).

Going from linear to nonlinear system identification gives new opportunities, but also introduces new challenges. General nonlinear model structures, such as

the HW structure, are more flexible than linear model structures. The flexibility comes with the price of several parameters to be identified and several choices where the model developer can go wrong.

For a linear OE model, the model developer's choices are the polynomial orders, i.e. n_b and n_f . When expanding the linear OE model to a HW model, the model developer must also choose which nonlinear functions $f(u)$ and $h(x)$ to use. Typically, $f(u)$ and $h(x)$ have parameters to be identified, for example saturation limits or polynomial coefficients. These parameters are added to the parameter vector θ , which was introduced in Section 4. When $f(u)$, $h(x)$, n_b , and n_f are chosen, θ is identified using PEM.

7.2 Default Settings Fail

In the MATLAB System Identification Toolbox, HW models can be identified either from the command line or by using the graphical user interface (GUI) of the toolbox. Using the command line, the model developer is forced to make explicit choices for $f(u)$, $h(x)$, n_b , and n_f . Using the GUI, default settings are provided. Deciding $f(u)$, $h(x)$, n_b , and n_f are most important, but difficult, choices. It therefore may be tempting for an inexperienced model developer to keep the default settings of the GUI.

Figure 11 shows ballistic simulation of a HW model, which was identified using the default settings for $f(u)$, $h(x)$, n_b , and n_f in the toolbox GUI. Also with respect to the optimization algorithm the default settings were used. Figure 11 speaks clearly for itself: The default settings give a very poor model. Even though this model structure has a large number of parameters to be identified and was validated at the same dataset that was used for identification, the model fit is very disappointing.

The example presented in Figure 11 illustrates what was explained above: The increased flexibility of the general nonlinear model structures also has significant disadvantages. In case of HW models, the model developer has to make clever choices for $f(u)$, $h(x)$, n_b , and n_f for the modeling work to be successful. Also, as the number of parameters to be identified increases, the risk of the optimization algorithm to be trapped in a local minimum increases significantly. Hence, the model developer may also spend some effort to provide the algorithm good initial values for the parameters to be identified.

7.3 Deciding Input and Output Nonlinearities

This subsection shows how examination of the dataset, together with the linear models developed in Section 6,

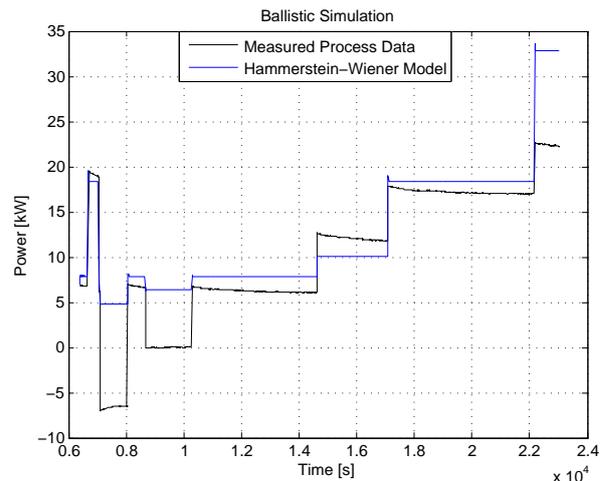


Figure 11: Ballistic simulation using the Hammerstein-Wiener model, which was identified using the default $f(u)$, $h(x)$, n_b , and n_f of the GUI in the MATLAB System Identification Toolbox.

can be used to make good choices for $f(u)$, $h(x)$, n_b , and n_f when developing a HW model for the dynamics from Δs to ΔP .

The preprocessed dataset used for system identification is shown in Figure 7. After $t = 1.0 \times 10^4$ s, four steps of Δs occur. For each step, Δs is increased by 5%. At first sight it seems that ΔP responds similarly to each step. However, more accurate examination of the dataset reveals that the magnitudes of the ΔP steps are different for each Δs step, even though all Δs steps are of the same magnitude.

To simplify notation, r [kW/%] is defined as the magnitude of a ΔP step divided by the magnitude of the corresponding Δs step. Hence, r can be considered the gain of which a step in Δs is amplified to the corresponding step in ΔP . Please note that the *instantaneous* responses in ΔP are considered, not the steady state values. Table 3 summarizes r for each step in Δs after $t = 1.0 \times 10^4$ s. The table shows that the higher values Δs and ΔP have prior to the step, the smaller is r .

In Section 6 it was concluded that the input-output models of the ARX and ARMAX models explain the *dynamics* of the process well, even though they handle the steps in Δs somewhat poorly. It is therefore reasonable to base the linear OE block of the HW model on the input-output model of the ARX model or the ARMAX model. Hence, the polynomial orders of the linear OE block is chosen $n_b = 2$ and $n_f = 1$. For the same reason, it is also reasonable to assume that the polynomial coefficients of $A(q)$ and $B(q)$ in the ARX

Table 3: Gain, r , for each step in Δs after $t = 1.0 \times 10^4$ s.

Step Δs [%]	Step ΔP [kW]	Gain, r , [kW/%]
0.00→ 5.00	0.04→ 6.78	1.35
5.00→10.00	6.09→ 12.64	1.31
10.00→15.00	11.72→ 17.87	1.23
15.00→20.00	16.99→ 22.68	1.14

and ARMAX models are suitable as initial values for identification of $F(q)$ and $B(q)$ in the linear OE block. The ARX model is chosen for initial values, because the ARX model has the simplest model structure and can be identified using the ordinary least squares (OLS) method.

The linear OE block of the HW model can not handle the variable gain shown in Table 3. The variable gain has to be handled by the input nonlinearity $f(u)$ and/or the output nonlinearity $h(x)$. A reasonable approach is to consider the gain, r , as a function of Δs and/or ΔP . It is here chosen to consider r as a function of Δs , i.e. $r = r(\Delta s)$. This choice will lead to a Hammerstein model (the reason for this choice is explained in Subsection 7.6). That is, $f(u)$ will be used to handle the nonlinearities, and $h(x)$ will simply be $h(x) = x$, or equivalent, the $h(x)$ block in Figure 10 will be absent.

A polynomial is a reasonable choice for explaining r as a function of Δs . As there are four $(\Delta s, r)$ pairs in Table 3, a third order polynomial will give no residual. However, as it is important to keep the number of parameters low, a first order polynomial is chosen. Hence, r is written as

$$r(\Delta s) = d_2 \Delta s + d_1, \quad (14)$$

where d_1 and d_2 are the polynomial coefficients. The process' behavior at each step, ignoring the general process dynamics, can then be approximated as

$$\begin{aligned} \Delta P_k &= \Delta P_{k-1} + r(\Delta s_{k-2}) \times (\Delta s_{k-1} - \Delta s_{k-2}) \quad (15) \\ &= \Delta P_{k-1} + (d_2 \Delta s_{k-2} + d_1) \times (\Delta s_{k-1} - \Delta s_{k-2}) \end{aligned}$$

for a step that occurs in Δs at timestep $k - 1$. The response in ΔP is delayed by one sample. The last term in (15) can be rewritten as

$$\begin{aligned} (d_2 \Delta s_{k-2} + d_1) \times (\Delta s_{k-1} - \Delta s_{k-2}) \quad (16) \\ = d_2 \Delta s_{k-1} \Delta s_{k-2} + d_1 \Delta s_{k-1} - d_2 \Delta s_{k-2}^2 - d_1 \Delta s_{k-2}. \end{aligned}$$

This equation gives a hint of how $f(u)$ could be chosen: The term $\Delta s_{k-1} \Delta s_{k-2}$ can not be included in $f(u)$, because $f(u)$, according to the definition of the HW model structure, should be a *static* function of u . Hence, one of these approximations must be chosen: $\Delta s_{k-1} \Delta s_{k-2} \approx \Delta s_{k-2}^2$ or $\Delta s_{k-1} \Delta s_{k-2} \approx \Delta s_{k-1}^2$. Using the first approximation, the term $d_2 \Delta s_{k-2}^2$ will cancel in (15). Hence, $r(\Delta s_{k-2})$ will be equal to the constant d_1 , which is in conflict with the intention of choosing $r(\Delta s)$ as a function of Δs . Using the other approximation, $\Delta s_{k-1} \Delta s_{k-2} \approx \Delta s_{k-1}^2$, makes sense. Define $f(\Delta s)$ as

$$f(\Delta s) \stackrel{\text{def}}{=} d_2 \Delta s^2 + d_1 \Delta s. \quad (17)$$

Using this definition and this approximation, (15) can be rewritten as

$$\Delta P_k = \Delta P_{k-1} + f(\Delta s_{k-1}) - f(\Delta s_{k-2}). \quad (18)$$

Although this derivation is based on some rough approximations, it will be shown shortly that $f(\Delta s)$ as defined in (17) is a good choice for the Hammerstein model to be identified.

For the chosen polynomial orders of the linear OE block, i.e. $n_b = 2$ and $n_f = 1$, the model structure to be used in this block is on the form

$$\Delta P_k + f_1 \Delta P_{k-1} = b_1 w_{k-1} + b_2 w_{k-2}, \quad (19)$$

where w is the output from the input nonlinearity block as illustrated in Figure 10. In order to handle the process' nonlinearity, the OE model of (19) is extended to a Hammerstein model by replacing w_{k-1} and w_{k-2} by $f(\Delta s_{k-1})$ and $f(\Delta s_{k-2})$, respectively. The Hammerstein model is then written as

$$\begin{aligned} \Delta P_k + f_1 \Delta P_{k-1} &= b_1 f(\Delta s_{k-1}) + b_2 f(\Delta s_{k-2}) \quad (20) \\ &= b_1 (d_2 \Delta s_{k-1}^2 + d_1 \Delta s_{k-1}) \\ &\quad + b_2 (d_2 \Delta s_{k-2}^2 + d_1 \Delta s_{k-2}). \end{aligned}$$

Now the desired model structure for the Hammerstein model has been developed. Subsections 7.4 and 7.5 consider how to identify the model parameters.

7.4 Computing Initial Values

For most model structures the PEM method uses an iterative optimization algorithm to compute the parameter vector θ . Such algorithms are generally at the risk of being trapped in a local minimum. The model developer may try to find good initial values for the

parameters to be identified, or he can try the iterative algorithm directly, hoping it will not be trapped in a local minimum. This subsection presents a suggestion for how to estimate good initial values for the parameters of the model (20).

The first issue is to estimate d_1 and d_2 . Equation (20) can be written on vector form as

$$\begin{aligned} \Delta P_k + f_1 \Delta P_{k-1} \\ = \begin{bmatrix} b_1 \Delta s_{k-1}^2 + b_2 \Delta s_{k-2}^2 \\ b_1 \Delta s_{k-1} + b_2 \Delta s_{k-2} \end{bmatrix}^T \begin{bmatrix} d_2 \\ d_1 \end{bmatrix}, \end{aligned} \quad (21)$$

where b_1 , b_2 , and f_1 are the parameters identified in the linear ARX model in Section 6 (a_1 of the ARX model corresponds to f_1 in (21)).

For each row in Table 3, a row in (21) is defined. This gives a linear, over-determined set of four equations with two unknown. Equation (21) is then solved for $[d_2 d_1]^T$ using the ordinary least squares (OLS) method. Figure 12 shows ballistic simulation of the Hammerstein model of (21) compared to the ARX model identified in Section 6. The Hammerstein model performs significantly better than the ARX model in terms of the optimization criterion $V(\theta)$. The Hammerstein model has $V(\theta) = 0.045$, while the ARX model has $V(\theta) = 0.106$. As the polynomial coefficients b_1 , b_2 , and f_1 (a_1 for the ARX model) are identical for the ARX model and the Hammerstein model, the improvement of the Hammerstein model can only be explained by the replacement of Δs in favor of $f(\Delta s)$ and the estimation of d_1 and d_2 . The Hammerstein model gives confidence in that the chosen model structure is well suited for explaining the process' behavior.

As good estimates of d_1 and d_2 are available, the parameters b_1 , b_2 , and f_1 will now be re-identified. That is, d_1 and d_2 in (20) are considered fixed, and b_1 , b_2 , and f_1 are identified using linear system identification. During this identification step, the system input is $f(\Delta s) = d_2 \Delta s^2 + d_1 \Delta s$ and the system output is ΔP . Please note that even if linear system identification is used, the identified model is a Hammerstein model, because the input to the linear model is $f(\Delta s)$, not Δs . Figure 13 shows Hammerstein models optimized this way using ARX and OE, respectively. These optimizations give significant improvements compared to the Hammerstein model of Figure 12. The optimization criterion is $V(\theta) = 0.024$ for the ARX-optimized model and $V(\theta) = 0.008$ for the OE-optimized model.

Although the OE-optimized model gives by far the best $V(\theta)$, the ARX-optimized model has a significant advantage: As ARX models can be identified using the ordinary least squares (OLS) method, all steps in identification of the ARX-optimized model can be solved

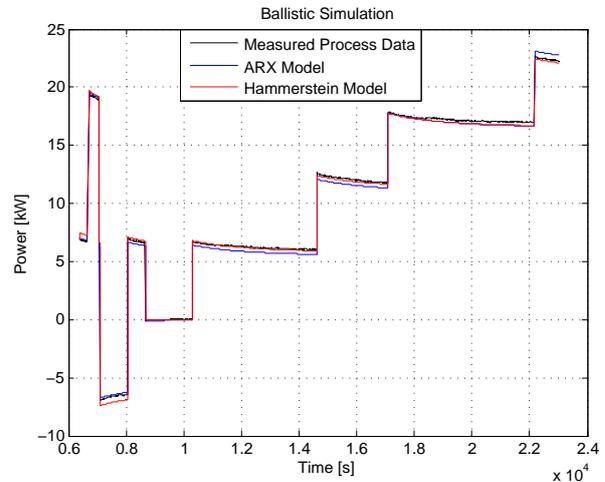


Figure 12: Ballistic simulation of the Hammerstein model of (21) after estimating d_1 and d_2 . The parameters b_1 , b_2 , and f_1 are identical to the ARX model identified in Section 6. The model is compared to measured process data and the ARX model.

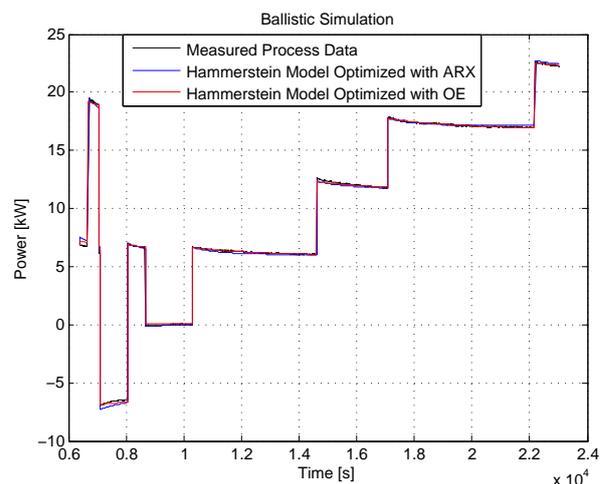


Figure 13: Ballistic simulations of Hammerstein models that are optimized with ARX and OE, respectively.

using OLS, involving no iterative optimization algorithms that are at the risk of being trapped in a local minimum. This also means that the ARX-optimized Hammerstein model can be identified by model developers not having the MATLAB System Identification Toolbox or other optimization software.

Figure 13 and the optimization criterion $V(\theta)$ show that the OE-optimized Hammerstein model fits the measured data very well. Hence, finding good initial values to be used in an iterative optimization algorithm has been successful.

7.5 Final Estimation

The final step of the nonlinear system identification work is to estimate all the parameters simultaneously using an iterative optimization algorithm. In Subsection 7.4, good initial values for the optimization algorithm were estimated.

A reasonable approach for identifying the final Hammerstein model is to use the function `nlhw` in the MATLAB System Identification Toolbox. Using this function, it can be specified that $f(\Delta s)$ should be a second order polynomial, and that there should be no output nonlinearity. Also n_b and n_f can be specified. However, to the authors’ knowledge, there is no way to force the constant term of the second order polynomial in $f(\Delta s)$ to be zero. In other words: $f(\Delta s)$ will be on the form $f(\Delta s) = d_2 \Delta s^2 + d_1 \Delta s + d_0$, where d_0 is a constant. Hence, an additional parameter, d_0 , has been introduced. This is unfortunate with respect to avoid overfitting.

A tailor-made workaround, which avoids the undesirable parameter d_0 , will be presented shortly. However, the first consideration is whether the model structure of (20) has the smallest possible number of parameters to be identified. The following parameters are defined

$$\tilde{b}_2 \stackrel{\text{def}}{=} \frac{b_2}{b_1}, \quad (22)$$

$$\tilde{d}_1 \stackrel{\text{def}}{=} b_1 d_1, \quad (23)$$

$$\tilde{d}_2 \stackrel{\text{def}}{=} b_1 d_2. \quad (24)$$

Using these definitions, (20) can be rewritten as

$$\Delta P_k + f_1 \Delta P_{k-1} = \tilde{d}_2 \Delta s_{k-1}^2 + \tilde{d}_1 \Delta s_{k-1} + \tilde{b}_2 (\tilde{d}_2 \Delta s_{k-2}^2 + \tilde{d}_1 \Delta s_{k-2}). \quad (25)$$

Hence, (20) can be rewritten with one less parameter. The parameters that are to be identified are \tilde{b}_2 , \tilde{d}_1 , \tilde{d}_2 , and f_1 .

The simplest solution the authors have found to identify the parameters in exactly the form of (25) is to

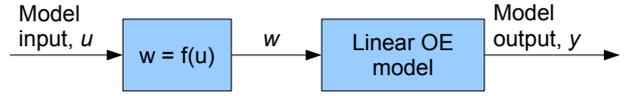


Figure 14: Block diagram of the final Hammerstein model.

use the functions `idgrey` and `pem` in the MATLAB System Identification Toolbox. These functions allow identification of arbitrary parameters in a linear state space model. Please refer to Ljung (2009) for a detailed explanation of these functions. Equation (25) can be rewritten to a discrete time state space model as

$$\begin{aligned} \begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} &= \begin{bmatrix} -f_1 & \tilde{b}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \\ &+ \begin{bmatrix} \tilde{d}_2 & \tilde{d}_1 \\ \tilde{d}_2 & \tilde{d}_1 \end{bmatrix} \begin{bmatrix} \Delta s_k^2 \\ \Delta s_k \end{bmatrix}, \\ \Delta P_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \varepsilon_k. \end{aligned} \quad (26)$$

In (26) the ballistic simulation error, ε , is added to emphasize that the model has no noise model. When the parameters \tilde{b}_2 , \tilde{d}_1 , \tilde{d}_2 , and f_1 have been identified, the model can be rewritten to a Hammerstein model with an OE model as its linear block

$$\Delta P_k + f_1 \Delta P_{k-1} = f(\Delta s_{k-1}) + \tilde{b}_2 f(\Delta s_{k-2}) + \varepsilon_k, \quad (27)$$

where

$$f(\Delta s) = \tilde{d}_2 \Delta s^2 + \tilde{d}_1 \Delta s. \quad (28)$$

Figure 14 shows a block diagram of this final Hammerstein model.

The ballistic simulation of the final model is shown in Figure 15, and the ballistic simulation error is shown in Figure 16. The optimization criterion is $V(\theta) = 0.007$. This is only a slight improvement over the OE-optimized model of Figure 13. The optimization algorithm used only three iterations to reach the final model. Hence the parameters of the OE-optimized model must have been very good initial values.

It was also tested whether poorer initial values led to the same parameters. When all initial values were set to zero, the optimization algorithm finds the same parameters as for the good initial values. Hence, in this case the work of estimating good initial values did not improve the final model. However, it is interesting

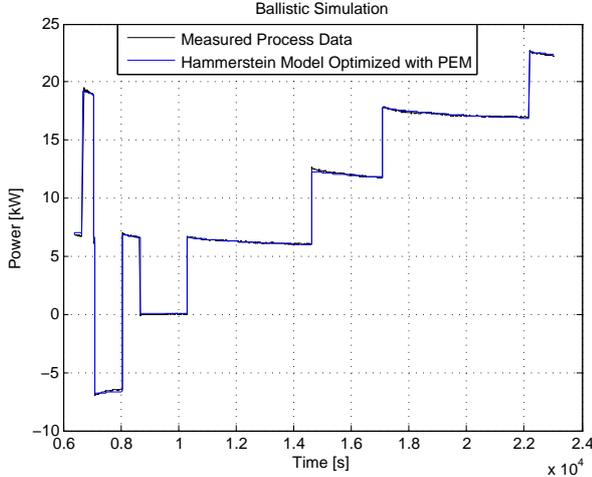


Figure 15: Ballistic simulation of the final Hammerstein model.

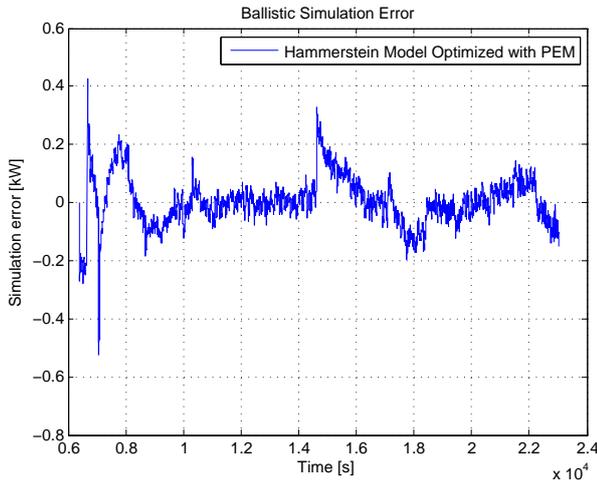


Figure 16: Ballistic simulation error of the final Hammerstein model.

to notice that the optimization algorithm used three iterations when having good initial values, and 17 iterations when having zeros as initial values.

Comparing the final Hammerstein model with the linear OE model identified in Section 6 shows that $V(\theta)$ was reduced by a factor of six when extending the linear OE model to a Hammerstein model. This extension includes only one additional parameter (the number of parameters to be identified is increased from three to four).

7.6 Noise Model

The HW model structure as defined in the MATLAB System Identification Toolbox has no noise model. This is reasonable as the output nonlinearity would significantly complicate the noise model, because there would be nonlinear dynamics from the one-step-ahead prediction error, e , to the model output, y .

The lack of noise model makes the HW model structure less suitable for some applications, such as (i) state estimation, (ii) measurement noise filtering using Kalman filter, and (iii) model predictive control (MPC).

In Subsection 7.3 the authors made the explicit choice of explaining the process nonlinearity using the input nonlinearity $f(u)$, instead of the output nonlinearity $h(x)$. This choice kept the door open to later replace the linear OE block of Figure 14 with a linear model structure having a noise model. By not applying a nonlinearity at the model output, the dynamics from the prediction error, e , to the model output, y , will be linear.

The state space model of (26) can easily be extended with a noise model

$$\begin{aligned} \begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} &= \begin{bmatrix} -a_1 & \tilde{b}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \\ &+ \begin{bmatrix} \tilde{d}_2 & \tilde{d}_1 \\ \tilde{d}_2 & \tilde{d}_1 \end{bmatrix} \begin{bmatrix} \Delta s_k^2 \\ \Delta s_k \end{bmatrix} \\ &+ \begin{bmatrix} k_1 \\ 0 \end{bmatrix} e_k, \\ \Delta P_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + e_k, \end{aligned} \quad (29)$$

where k_1 is the Kalman filter gain. The parameters a_1 , \tilde{b}_2 , \tilde{d}_1 , \tilde{d}_2 , and k_1 of (29) were identified using `idgrey` and `pem`. The parameter f_1 of (26) has been replaced by a_1 , because (29) can be rewritten to a Hammerstein model with an ARMAX model as its linear block

$$\begin{aligned} \Delta P_k + a_1 \Delta P_{k-1} &= f(\Delta s_{k-1}) + \tilde{b}_2 f(\Delta s_{k-2}) \\ &+ e_k + c_1 e_{k-1}, \end{aligned} \quad (30)$$

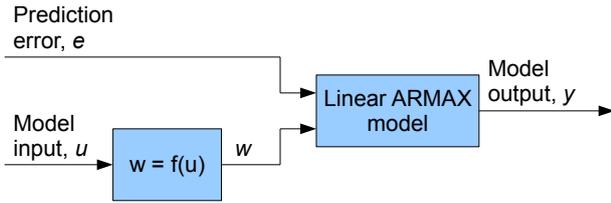


Figure 17: Block diagram of the final Hammerstein model extended with a noise model.

where

$$f(\Delta s) = \tilde{d}_2 \Delta s^2 + \tilde{d}_1 \Delta s \quad (31)$$

and

$$c_1 = a_1 + k_1. \quad (32)$$

Figure 17 shows a block diagram of the Hammerstein model of (30). This Hammerstein model has $V(\theta) = 0.032$, which is significantly poorer than the model of (27). This is because the model of (30) is optimized for $W(\theta)$ (one-step-ahead prediction), while the model of (27) is optimized for $V(\theta)$ (ballistic simulation). This result is similar to the result of Table 2, where the ARX and ARMAX models perform poorer than the linear OE model at ballistic simulation.

Figure 18 shows the one-step-ahead prediction error, e , for the linear ARMAX model identified in Section 6 with $n_a = 1$, $n_b = 2$, and $n_c = 1$ (upper subplot) and for the Hammerstein model of (30) (lower subplot). The figure shows that the largest peaks are considerably smaller for the Hammerstein model. The one-step-ahead prediction optimization criterion is $W(\theta) = 3.1 \times 10^{-4}$ for the ARMAX model and $W(\theta) = 1.8 \times 10^{-4}$ for the Hammerstein model, i.e. 42% lower for the Hammerstein model.

8 Model Weaknesses

The Hammerstein model of (27) gives very good model fit to the identification dataset, even with a small number of identified parameters. This strongly indicates that the identified model explains the process’ true input-output behavior. If the measured power, P , was significantly influenced by random process disturbances and measurement noise, it is very unlikely that a model with few parameters would give a good fit, even to the identification dataset.

Even though the model explains the process input-output behavior well for the dataset used in this paper, it is not known whether the model will perform

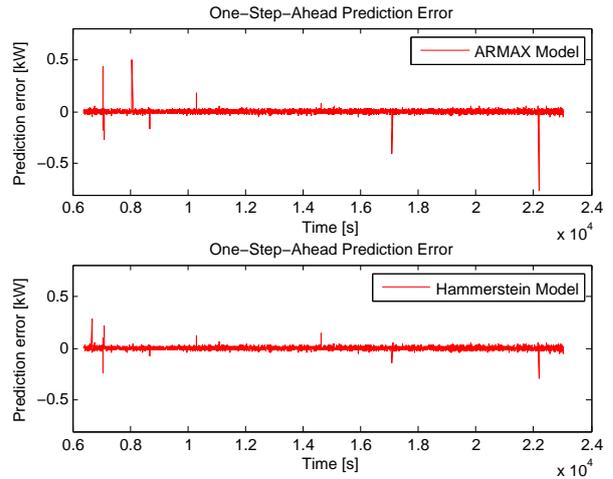


Figure 18: One-step-ahead prediction error, e , of the linear ARMAX model identified in Section 6 (upper) and the Hammerstein model of (30) (lower).

this good on datasets with a significantly different frequency content in the input signal. Further, it is not known whether the model will perform well on datasets of which Δs is significantly higher or lower than in the dataset used in this paper.

9 Further Research

The Hammerstein model identified in this paper is a black-box model in the sense that it explains *how* the measured power responds to the TRIAC input signal, but not *why* this happen. An interesting continuation of this work is to develop a grey-box model of the dynamics from the TRIAC input signal to the power. A grey-box model is a mechanistic model of which unknown parameters are estimated using PEM or other approaches for black-box modeling.

The modeling work presented in this paper is part of a larger modeling work, which aims to model the dynamics of the most important input / output pairs of the CZ process using black-box and grey-box modeling. Next these models are to be used for improving monitoring and control of the CZ process.

10 Conclusions

This paper considered empirical modeling of the dynamics from the TRIAC input signal to the actual (measured) heating element power at the Czochralski (CZ) crystallization process. The modeling was based

on a dataset logged from a real-life CZ process. Before modeling, the dataset was preprocessed by detrending the data and handling outliers. As no independent dataset was available for model validation, it was considered most important to limit the number of parameters to be identified.

Initially three linear polynomial models were identified; an ARX model, an ARMAX model, and an output error (OE) model. These models explain the process dynamics well close to the operating point, but perform somewhat poorer otherwise.

Because the linear models do not explain the process behavior very well over a larger range of the input signal, nonlinear system identification was used for modeling the process. The Hammerstein-Wiener (HW) model structure was chosen. At first, a model was identified using the default setting for HW models in the MATLAB System Identification Toolbox GUI. However, this model has a very poor performance, and was therefore rejected.

As the default settings of the toolbox failed, the model structure was developed by examination of the process' behavior at steps in the input signal. Also the results from linear system identification were used. The chosen model structure is a Hammerstein model, which has a second order polynomial as input nonlinearity and no nonlinearity at the output. The latter choice was done to allow a noise model to be added to the model.

The Hammerstein model has four parameters to be identified, which is only one more than the linear OE model. Still the Hammerstein model has improved the optimization criterion (mean squared ballistic simulation errors) by a factor of six compared to the linear OE model. As the Hammerstein model has good model fit despite few identified parameters, it is concluded that the modeling work has been successful.

This paper also extended the Hammerstein model with a noise model. For one-step-ahead predictions, the optimization criterion (mean squared one-step-ahead prediction errors) is reduced by 42% for the extended Hammerstein model compared to the linear ARMAX model. The extended Hammerstein model has five parameters to be identified, compared to four parameters in the ARMAX model.

Acknowledgements

The authors are most grateful to SINTEF Materials and Chemistry in Trondheim, Norway for giving access to data from the company's CZ process, and for allowing these data to be used in this paper and other publications. John Atle Bones at SINTEF is acknowledged for his decisive contributions in performing the exper-

iments on the CZ process. Eivind Johannes Øvrelid at SINTEF is acknowledged for sharing his knowledge and experience of the CZ process.

The first author is most grateful for the financial support of his PhD study from NorSun AS, Østfold Energi AS, and the Norwegian Research Council. The cooperation with NorSun AS, Prediktor AS, and SINTEF Materials and Chemistry is also very much appreciated.

References

- Komperød, M., Hauge, T., and Lie, B. Preprocessing of experimental data for use in model building and model validation. In *The 49th Scandinavian Conference on Simulation and Modeling (SIMS 2008)*. Oslo, Norway, 2008.
- Lee, K., Lee, D., Park, J., and Lee, M. MPC based feedforward trajectory for pulling speed tracking control in the commercial Czochralski crystallization process. *International Journal of Control, Automation, and Systems*, 2005. 3:252–257.
- Ljung, L. *System Identification - Theory for the User, 2nd Edition*. Prentice-Hall, Upper Saddle River, New Jersey, 1999.
- Ljung, L. *System Identification Toolbox 7 - User's Guide*. The MathWorks, Inc., 2009.