An internal dynamics approach to predicting batch-end product quality in plastic injection molding using Recurrent Neural Networks

Alexander Rehmer[1], Marco Klute[2], Andreas Kroll[1] and Hans-Peter Heim[2]

***Abstract*— Estimating a dynamic model for the task of batch end quality prediction from process measurements without continuous measurements of the quality variable but only a single measurement after the batch is finished is a difficult modeling task. It is common practice to circumvent building a dynamic model altogether by instead using a static model approach of some kind. These static approaches, i.e. Multivariate Statistical Process Control (MSPC) or mapping features extracted from process measurements to the quality measurement, work in general very well. Nevertheless they neglect some of the information present in the data and come with some drawbacks, i.e. the necessity for all batches having equal length in the case of MSPC methods. The purpose of this paper is therefore to demonstrate how an internal dynamics model approach, more specifically a Recurrent Neural Network, can be used to estimate a true dynamic model for batch end quality prediction from process measurements. The resulting model may be used for in-batch or batch-to-batch optimization depending on the real-time requirements of the respective process. The proposed model approach is applied to a plastic injection molding (PIM) process, which is a switched system, and compared to static model approaches that are commonly employed in the PIM community.**

I. Introduction

In order to obtain a consistent and desirable batch end-product quality, precise quality prediction models are necessary that can be employed for in-batch or batch-to-batch optimization. Many batch processes, such as the plastic injection molding (PIM) process, do not allow for continuous in-process quality measurements. Only after the part is ejected can the quality variables of interest, e.g. weight or geometrical features, be quantified. Estimating a dynamical model with trajectories for the inputs, but only a single datum for the output, i.e. the quality measurement, is an unusual and difficult identification task. Probably for this reason, the dynamic modeling task of predicting part quality in batch processes has almost exclusively been treated as a static one. The most straightforward approach is to predict batch end-product directly from process setpoints. By doing so, process dynamics are implicitly assumed to have a negligible effect on product quality. Another approach is to extract features from process measurements based on expertise and map those to batch-end product quality. Depending on how meaningful the selected features are, this approach can be very successful, but still uses not all the information available in the data. Finally and probably most widespread, Multivariate Statistical Process Control (MSPC) methods, such as multi-way principal component analysis (PCA) and multi-way projection to latent structures (PLS), are employed to correlate process variable trajectories with final product quality [1]. Although these methods exploit all the information in the data, as static models they are subject to certain restrictions, e.g. that all batches need to have the exactly same length [2]. Hence, methods must be employed that somehow scale all trajectories to the same length, e.g. Dynamic Time Warping. By doing so, the original information contained in the measured data is affected to an unknown extent. Estimating a dynamic nonlinear model that maps process measurements to part quality has not yet been attempted.

The purpose of this paper is to present and examine a modeling approach for estimating a dynamic model that maps continuous process measurements to a single quality datum. This will be achieved via using an internal dynamics model structure. The benefits of the internal dynamics approach for final part quality prediction over existing approaches are, that batches of varying length are handled naturally by the model. Also, internal dynamics models tend to have significantly fewer parameters in the Multiple Input Multiple Output (MIMO) case, which is beneficial if little data is available. The resulting model can be employed in model-based numerical optimal control for in-batch or batch-to-batch part quality optimization.

II. Related Work

Final part quality prediction in batch processes in general has been the subject in a vast body of research. Usually, MSPC methods, such as multi-way PCA and multi-way PLS, are employed to correlate the process variable trajectories with the final product quality [1]. Regarding the PIM process specifically, there is no known application of MSPC methods for final part quality prediction, merely an application of multi-way PCA for fault detection purposes [3]. Instead, final part quality is usually directly predicted from process setpoints [4] or from features extracted from process measurements [5, 6, 7]. An exception is the work [8], who estimated a linear state space model for a rotational molding process. The final part quality was then predicted solely from the final state of the process model. This implies, that only the final process state (and not the evolution that led to that state) is relevant for part quality.

*This work was not supported by any organization

[1]University of Kassel, Department of Measurement and Control, Mönchebergstr. 7, 34125 Kassel, Germany alexander.rehmer@mrt.uni-kassel.de

[2]University of Kassel, Institute of Material Engineering - Polymer Engineering, Mönchebergstr. 3, 34125 Kassel, Germany marco.klute@uni-kassel.de

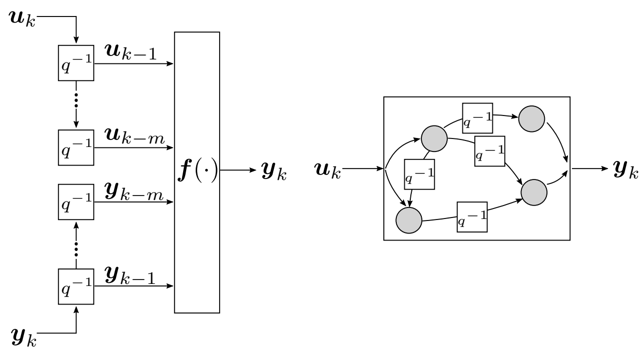

Fig. 1. External and internal dynamics approach

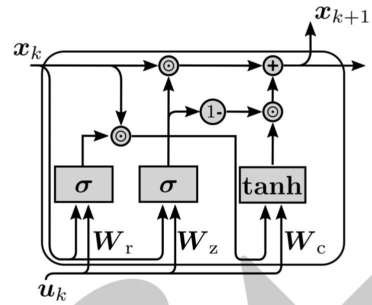

Fig. 2. Gated Recurrent Unit architecture

To the best of the authors knowledge, an internal dynamics approach has never been applied to the problem of final part quality prediction, neither in batch processes in general nor in PIM specifically.

III. Preliminaries

A. External vs. Internal Dynamics Approach

Dynamic models can amongst other criteria be differentiated into external and internal dynamics approaches, see Fig. 1. In the far more widespread external dynamics approach a static model $\boldsymbol{f}(\cdot)$ is provided with past inputs $\boldsymbol{y}_{k-m}$ and outputs $\boldsymbol{u}_{k-m}$ to predict the current output $\boldsymbol{y}_k$, i.e. the dynamics are realized via external filters that in their most simple form correspond to time-delays q^{-1}.

$$\boldsymbol{y}_k = \boldsymbol{f}\left(\boldsymbol{y}_{k-1}, \ldots, \boldsymbol{y}_{k-m}, \boldsymbol{u}_{k-1}, \ldots, \boldsymbol{k}_{k-m}\right) \tag{1}$$

The internal dynamics approach is basically a state-space (SS) model, without any information about the true process states. The internal model states are merely used to approximate the dynamic behavior of the true process and do not have any physical interpretation. Like any SS model, an internal dynamics model is only provided with the current input to predict the next output:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\ \boldsymbol{y}_k &= \boldsymbol{g}\left(\boldsymbol{x}_k\right) \end{aligned} \tag{2}$$

Although the internal dynamics approach does have its advantages, like a lower dimensional input space especially for higher order MIMO systems, its major shortcoming is that the training of internal dynamics models poses a very difficult optimization problem. Most significantly, an internal dynamics model can be unstable from the outset, which will terminate the optimization procedure immediately. Also the evolution of the internal states must reflect the true process dynamics, but can only be deduced from input-output data while simultaneously the model parameters have to be estimated. In contrast to that, in an external approach the dynamics are fixed by the filters specified by the user such that only the model parameters have to be estimated from the data. For the problem at hand, i.e. the prediction of final product quality from process measurements, the application of the internal dynamics approach is virtually inevitable, if a dynamic model is to be estimated. Since only the output $\boldsymbol{y}_T$ at the very last time instance T of each batch is available, delayed outputs cannot be used as model inputs. Hence all external dynamics model structures with output feedback, e.g. Autoregressive with inputs (ARX), Autoregressive moving average with inputs (ARMAX), etc., and their nonlinear counterparts are excluded from consideration. Only external dynamics model structures without output feedback, e.g. Finite Impulse Response (FIR), would be viable. However, such a model would treat every time instance as a distinct input which would result in thousands of inputs (and model parameters) depending on the length and number of measured process variables. The amount of experimental data needed to estimate these parameters makes this approach unfeasible. An internal dynamics model on the other hand, would only have as many inputs as there are process measurements, which significantly reduces the input space and number of model parameters compared to an external dynamics model. It's internal state, that evolves from time instance to time instance under the influence of the process variables, can be viewed as an abstract representation of the produced part or its properties. As mentioned above, the task of parameter optimization for an internal dynamics model is already a difficult problem, if trajectories for input and output are available. The task at hand further amplifies the severity of these problems, since only a single measurement for the output at the end of each batch is available for estimating the entire dynamical process.

B. Recurrent Neural Networks (RNN)

Although any recurrent model structure can be employed to realize a nonlinear internal dynamics model, RNN are the most common. In recent years, great progress has been made in the design of RNN-architectures, resulting in easier to optimize RNNs, most notably the long short-term memory (LSTM) [9] and Gated Recurrent Unit (GRU) [10]. Since the latter has significantly fewer parameters and has been shown to perform as good or even better on various applications [11] it is considered for the task of final part quality prediction. The GRU is depicted in Fig. 2, its state equation is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{\mathrm{z}} \odot \boldsymbol{x}_k + (\boldsymbol{1} - \boldsymbol{f}_{\mathrm{z}}) \odot \boldsymbol{f}_{\mathrm{c}}. \tag{3}$$

The operator $\odot$ denotes the Hadamard product. The activations of the so-called reset gate $\boldsymbol{f}_{\mathrm{r}}$, update gate $\boldsymbol{f}_{\mathrm{z}}$ and the output gate $\boldsymbol{f}_{\mathrm{c}}$ are given by

$$\begin{aligned} \boldsymbol{f}_{\mathrm{r}} &= \boldsymbol{\sigma}\left(\boldsymbol{W}_{\mathrm{r}} \cdot [\boldsymbol{x}_k, \boldsymbol{u}_k]^T + \boldsymbol{b}_{\mathrm{r}}\right), \\ \boldsymbol{f}_{\mathrm{z}} &= \boldsymbol{\sigma}\left(\boldsymbol{W}_{\mathrm{z}} \cdot [\boldsymbol{x}_k, \boldsymbol{u}_k]^T + \boldsymbol{b}_{\mathrm{z}}\right), \\ \boldsymbol{f}_{\mathrm{c}} &= \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}} \cdot [\tilde{\boldsymbol{x}}_k, \boldsymbol{u}_k]^T + \boldsymbol{b}_{\mathrm{c}}\right), \end{aligned} \tag{4}$$

with $\tilde{\boldsymbol{x}}_k = \boldsymbol{f}_{\mathrm{r}} \odot \boldsymbol{x}_k$ and $\boldsymbol{W}_{\mathrm{r}}, \boldsymbol{W}_{\mathrm{z}}, \boldsymbol{W}_{\mathrm{c}} \in \mathbb{R}^{n_x \times n_x + n_u}$, $\boldsymbol{b}_{\mathrm{r}}, \boldsymbol{b}_{\mathrm{z}}, \boldsymbol{b}_{\mathrm{c}} \in \mathbb{R}^{n_x}$ and $\boldsymbol{f}_{\mathrm{r}}, \boldsymbol{f}_{\mathrm{z}}, \boldsymbol{f}_{\mathrm{c}} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$. $\boldsymbol{\sigma}(\cdot)$ denotes the logistic function. The internal state $\boldsymbol{x}_k$ is usually mapped to the output $\boldsymbol{y}_k$ via a feedforward Neural Network with a nonlinear hidden layer and a linear output layer:

$$\begin{aligned} \boldsymbol{h}_k &= \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{h}} \cdot \boldsymbol{x}_k + \boldsymbol{b}_{\mathrm{h}}\right) \\ \boldsymbol{y}_k &= \boldsymbol{W}_{\mathrm{o}} \cdot \boldsymbol{h}_k + \boldsymbol{b}_{\mathrm{o}} \end{aligned} \tag{5}$$

C. Backpropagation Through Time

In the Machine Learning community the process of calculating the gradient of the loss function $\mathcal{L}(\boldsymbol{e})$ and therefore the prediction error $\boldsymbol{e} = \hat{\boldsymbol{y}} - \boldsymbol{y}$ with respect to the model parameters $\boldsymbol{\theta}$ is denoted backpropagation (BP). If the model is recurrent, the derivative $\frac{\partial \boldsymbol{e}_k}{\partial \theta}$ depends on the state of the previous time step $\boldsymbol{x}_{k-1}$, which again depends on the model parameters and so on. Hence, the application of the chain rule requires to propagate the error back to the very first time instance, therefore the name Backpropagation Through Time (BPTT). If the output is a trajectory $\boldsymbol{y}_{1,\ldots,T}$, a prediction error can be calculated at each time step and the loss becomes

$$\sum_{k=1}^{T} \mathcal{L}(\boldsymbol{e}_k) \tag{6}$$

Since only the final product quality, i.e. $\boldsymbol{y}_T$, is known, the sum in (6) reduces to a single term, showing how little information is available for parameter optimization.

D. Plastic Injection Molding Process

For the production of plastic parts by the PIM process, the base material is fed in the form of granules into the hopper of the injection molding machine. Within the injection unit of the machine, the material is molten and transported by a rotary movement of the screw. A defined melt volume, which is collected in front of the screw, is then injected into the mold cavity under high pressure at a defined velocity by a translatory movement of the screw. To prevent shrinkage during the cooling of the plastic part within the mold, a defined packing pressure is applied. Depending on the process variables, the PIM process can be devided into three phases:

1) Injection: As mentioned ealier, a defined dosing volume of melt is collected infront of the screw, which is then injected with a defined injection velocity until a switching point is reached, that is defined by the screw position. In order to prevent damage to the machine, a maximal allowed injection pressure is defined.

2) Packing: In contrast to injection, the packing phase is pressure controlled, to reach a defined pressure curve over a specified time. The packing pressure is composed of multiple pressure ramps. In this study, three packing pressure levels were set, with the first and third each remaining constant and only the second being varied in the experimental design.

3) Cooling: Although cooling of the plastic part begins directly upon entry of the melt into the cavity, the defined cooling phase does not start until after the packing phase. During this pressureless cooling phase, the melt volume for the next cycle is metered against a given back pressure. At the end of the cooling time, the mold opens and the plastic part is ejected.

IV. Data Driven Modeling

A. Data Acquisition

Since the goal is the estimation of a model that maps process measurements to part quality, the experiments for data acquisition should ideally be designed in the process measurement space. However, due to the constraints imposed by the process a direct manipulation of process measurements is not possible. Therefore the experiment was instead designed in the process setpoint space. A process setpoint $\boldsymbol{s} \in \mathbb{R}^8$ is defined by the eight setpoint parameters defined in Table I. A comparison of different DoE methods, that are applicable for PIM processes, has shown that central composite designs (CCD) are most effective [12]. They represent an extension of a full factorial design (FFD) with star points (SP), enabling them to register non-linear effects. In this study a face centered CCD ($\boldsymbol{\alpha} = 1$) was chosen and extended by a central point (CP). For a total of eight setpoint parameters with two factor levels each (Table I) and the central point (two repetitions), the resulting experimental design contains a total of 274 experiments (FFD: 256, SP: 16, CP: 2).

The experiments were carried out using an ALLROUNDER 470S injection molding machine (ARBURG GmbH + Co KG, Loßburg, Germany) with a screw diameter of 35 mm. For each of the 274 setpoints ten parts were produced using an Ultramid® B3S polyamide (BASF SE, Ludwigshafen, Germany), that was dryed for several hours at $80°C$ prior to processing. As part quality feature, the inner diameter of the produced parts was measured using a digital measuring projector IM-7020 (Keyence Corporation, Osaka, Japan).

In addition to the machine setting parameters (Table I), decisive process variables were also taken into account for learning the models. For this purpose, trajectories were retrieved via the Open Platform Communications - Unified Architecture (OPC-UA) interface of the injection molding machine. The injection mold was equipped with an additional sensor that recorded both pressure and temperature inside the mold cavity.

The following process variables were retrieved as trajectories with a sampling rate of 50 Hz: cavity pressure, cavity temperature, injection pressure, melt volume (infront of the screw), melt volume flow rate.

TABLE I
FACTORS VARIED FOR THE EXPERIMENTAL DESIGN

Setpoint Parameter	Unit	Lower limit	Upper limit
Nozzle temperature	°C	250	260
Mold temperature	°C	40	50
Injection velocity	cm^3/s	16	48
Switching point	cm^3	13	14
Packing pressure	MPa	50	60
Packing time	s	3	5
Back pressure	MPa	2.5	7.5
Cooling time	s	15	20

TABLE II
DEFINITION OF FEATURES f_i CALCULATED FROM p_{cav} AND T_{cav}

Feature	Definition	Feature	Definition
$p_{\text{cav}}^{\max}$	$\max(p_{\text{cav}})$	$T_{\text{cav}}^{\text{int}}$	$\sum_{k=0}^{t_{\text{cyc}}} p_{\text{cav}}$
$t_p^{\max}$	$\arg\max\limits_k (p_{\text{cav}})$	$t_T^{\max}$	$\arg\max\limits_k (T_{\text{cav}})$
$p_{\text{cav}}^{\text{int}}$	$\sum_{k=0}^{t_{\text{cyc}}} p_{\text{cav}}$	$T_{\text{cav}}^{\max}$	$\max(T_{\text{cav}})$
$p_{\text{cav}}^{\text{drop}}$	$\frac{1}{t_{\text{cyc}}-t_{\text{hold}}} \sum_{k=t_{\text{hold}}}^{t_{\text{cyc}}} p_{\text{cav}}$		

B. Model structures

Although the goal is to identify an internal dynamics quality model for quality prediction, static modeling approaches that are common in practice and science were estimated for comparison. The different modeling approaches are introduced in the following.

1) Setpoints model: The most straightforward approach to predict part quality is to map the process setpoints listed in Table I directly to the resulting part quality. A linear regression model, a polynomial regression model and a Multilayer Perceptron (MLP) were employed as model structures.

2) Measurement-features model: The most widespread approach is to predict final part quality from features extracted from process measurements based on expert knowledge. The features extracted from the cavity pressure signal were chosen to be the same as in [6] with the addition of the time instance $t_p^{\max}$, when the maximal cavity pressure $p_{\text{cav}}^{\max}$ occurs. The features extracted from the cavity temperature signal were chosen to be: The temperature at the beginning of a cycle T_{cav}^0, since it indicates the mold temperature before the injection of the melt. The maximal occurring temperature during the cycle $T_{\text{cav}}^{\max}$, since it correlates with the melt temperature and the integral $T_{\text{cav}}^{\text{int}}$ over the temperature signal, which mainly reflects the cooling process. The features are summarized in Table II and visualized in Fig. 3

3) Internal dynamics model: The GRU described in subsection III-B was chosen as the internal dynamics model structure to predict batch-end product quality from process measurements. The PIM process is known to be time-varying due to switches between different machines internal controllers, see subsection III-D. It is to be expected that the formation of the parts quality characteristics is also a time-varying process: The part undergoes major temperature changes as well as a change of aggregate state from fluid to solid. The pressure in the cavity for one should have different

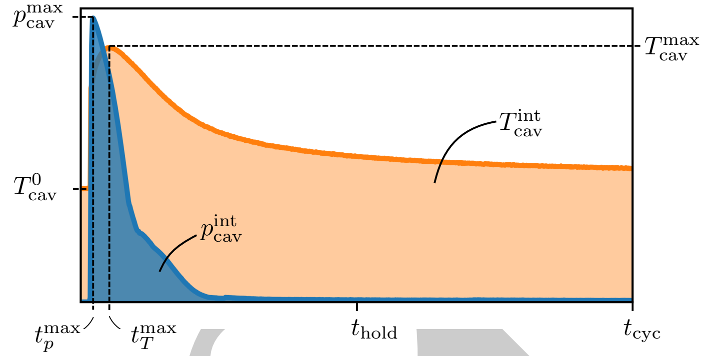

Fig. 3. Features extracted from p_{cav} and T_{cav}

effects depending on the state the part is currently in. In order to incorporate this time-varying behavior in the model approach, the quality process was divided into the same three phases as the PIM-process itself. Hence, the part quality process model is a time-varying switched system comprised of three subsystems $i = 1, 2, 3$, respectively representing the injection, packing and cooling phase. The parameter vectors of each subsystem are denoted $\boldsymbol{\theta}^i$.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{h}^i\left(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta}^i\right), \quad \boldsymbol{x}_0 = \boldsymbol{0} \\ \boldsymbol{y}_k &= \boldsymbol{g}^i\left(\boldsymbol{x}_k; \boldsymbol{\theta}^i\right) \end{aligned} \tag{7}$$

The recurrent part $\boldsymbol{h}^i(\cdot)$ of each subsystem is modeled via a GRU, see (3). Since the subsystem representing the cooling phase must also map the internal state to the output, it is additionally equipped with an MLP with 10 $\tanh$-neurons in the hidden layer as output function $\boldsymbol{g}^3(\cdot)$. The complete model architecture is depicted in Fig. 4. Parameter initialization was found to be crucial in order to obtain any useful models. As in any (stable) dynamical system the contribution of the GRU's state $\boldsymbol{x}_k$ at a given time instance k to the output $\boldsymbol{y}_{k+T}$ at a later time instance decreases exponentially with T. Since the model to be optimized is recurrent in the states, the same is true for its parameters. This phenomenon is known as the *vanishing gradient* in the ML community. If care is not taken during initialization, especially the contribution of the first two subsystems ($i = 1, 2$) will be negligible small. The solution to this dilemma is to initialize the bias of the update gate $\boldsymbol{b}_{\text{z}}^{1,2}$ with large positive values. By doing so the state equation (3) becomes $\boldsymbol{x}_{k+1} \approx \boldsymbol{x}_k$, i.e. the GRU just passes on the state. This ensures the maximal possible gradient, at least at the very beginning of the optimization procedure. If need be, the optimizer will then reduce the bias to an appropriate value, such that the GRUs dynamics represents the dynamics of the true process. For this reason the bias must also not be chosen too large, otherwise it would take an excessive amount of optimization steps to reduce the bias to its appropriate value. For this case study, the best results were obtained by drawing $\boldsymbol{b}_{\text{z}}$ from a random uniform distribution $\mathcal{U}_{[4,10]}$. Without this initialization the estimated models were merely able to reproduce the mean of the training data.

It should be noted, that this is not necessarily the best way to account for the time variance of the quality process. The

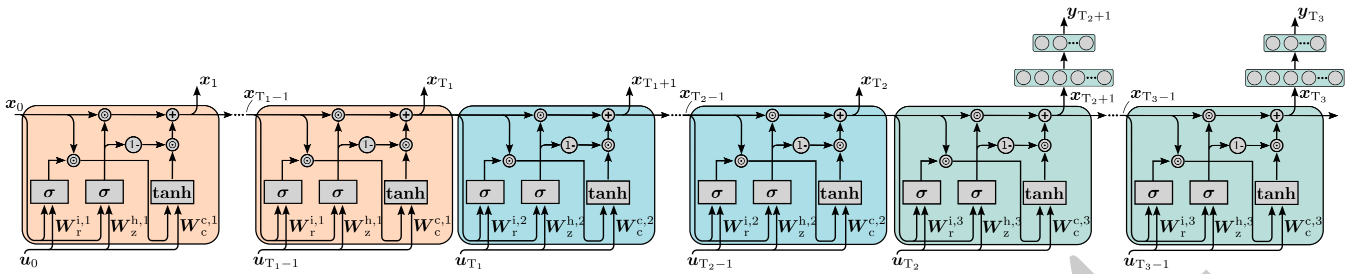

Fig. 4. Internal dynamics modeling approach with three subsystems

TABLE III

CONSIDERED CANDIDATE MODEL STRUCTURES FOR QUALITY PREDICTION.

Label	Definition
$\mathrm{PR}^n_{\mathrm{s,f}}$	n-th degree polynomial regression model mapping either process setpoints $\boldsymbol{s}$ or process measurement features $\boldsymbol{f}$ to product quality D_{i}.
$\mathrm{MLP}^n_{\mathrm{s,f}}$	MLP with n neurons in single hidden layer mapping either process setpoints $\boldsymbol{s}$ or process measurement features $\boldsymbol{f}$ to product quality D_{i}.
ID^n_i	Internal dynamics model consisting of i subsystems mapping process measurements $\boldsymbol{p}$ to product quality D_{i}. The recurrent model part is realized via GRUs with n internal states. The mapping from internal state to product quality D_{i} is realized via an MLP with 10 tanh neurons in the hidden layer.

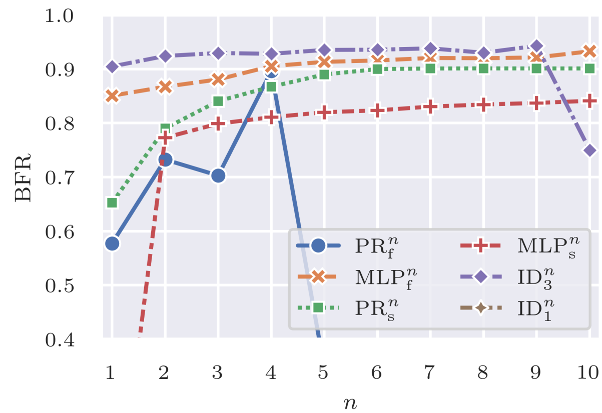

Fig. 5. BFR of all candidate models from Table III on $\mathcal{D}_{\mathrm{val}}$ depending on the model complexity n.

time variance of the emerging part does not necessarily need to coincide with the time variance of the PIM process itself. The time instance when the part changes its aggregate state would be an ideal candidate for the switching instance. Since the solidification is a spatially distributed process, this time instance is not well defined. In order to evaluate, whether the chosen approach for modeling the time variance was appropriate, a quality model with only a single subsystem, i.e. without time variance, was estimated for comparison.

C. Model training and validation

All models were optimized on the same training dataset $\mathcal{D}_{\mathrm{train}}$ to minimize the mean-squared-error (MSE) of the predicted quality variable y. $\mathcal{D}_{\mathrm{train}}$ comprises the process setpoints $\boldsymbol{s}^c$, process measurements $\boldsymbol{p}^c = \left\{p^c_{\mathrm{cav},k}, T^c_{\mathrm{cav},k}\right\}$, $k = 0, \ldots, t^c_{\mathrm{cyc}}$ and resulting inner diameter D_{i} of the produced part D^c_{i} of a production cycle c. The optimized models were then evaluated on a validation dataset $\mathcal{D}_{\mathrm{val}}$ to determine their generalization capabilities. In total $\mathcal{D}_{\mathrm{train}}$ comprises 2105 and $\mathcal{D}_{\mathrm{val}}$ 548 production cycles. All considered candidate models for quality prediction are summarized in Table III. For model comparison, the Best Fit Rate (BFR) (8) was calculated on the validation data:

$$\mathrm{BFR} = \max\left(0, 1 - \frac{\sum\limits_c y_{\mathrm{c}} - \hat{y}_{\mathrm{c}}}{\sum\limits_c y_{\mathrm{c}} - \bar{y}_{\mathrm{c}}}\right) \cdot 100\ \% \tag{8}$$

For parameter estimation of the nonlinear models a gradient based optimization method (IPOPT [13]) with quasi-Newton approximation of the hessian was employed.

V. RESULTS & DISCUSSION

The performance of each candidate model structure in terms of the BFR on $\mathcal{D}_{\mathrm{val}}$ is depicted in Fig. 5. First of all, the static setpoint models, i.e. $\mathrm{PR}^n_{\mathrm{s}}$ and $\mathrm{MLP}^n_{\mathrm{s}}$ perform already quite well. The best fit (BFR = 90 %) is achieved with a 10th degree polynomial model $\mathrm{PR}^{10}_{\mathrm{s}}$. This is probably due to the fact, that external disturbances on the process, such as the ambient temperature or fluctuations of material properties, were eliminated as far as possible. This allowed for a prediction of part quality with high accuracy directly from machine setpoints. Nevertheless, the feature models, especially $\mathrm{MLP}^n_{\mathrm{f}}$, outperform the setpoint models, albeit by a small margin. This indicates, that the additional dynamic information encoded in the features is in fact relevant for predicting part quality. The best feature model was an MLP with 10 Neurons in the hidden layer, i.e. $\mathrm{MLP}^{10}_{\mathrm{f}}$, with BFR = 93 %. The polynomial feature models $\mathrm{PR}^n_{\mathrm{f}}$ show underfitting for $n < 4$, indicating a significant nonlinear relation between features $\boldsymbol{f}$ and part quality. For $n > 4$ on the other hand, the polynomial models exhibit overfitting. This is most likely due to the tendency for oscillatory interpolation and extrapolation behavior of higher degree polynomials. The polynomial setpoint models do not exhibit this behavior, because the experiment was designed to ensure an equidistant distribution of data samples in the setpoint space.

Comparing the two internal dynamics approaches with each other, i.e. the switched ID^n_3 and the single subsystem approach ID_1^n clearly validates the switched system approach. The performance of the single subsystem approach ID_1^n is not able to explain additional variance in the data compared to its mean and therefore has a BFR of 0. The switched system approach with three internal states, i.e. ID_3^3, performs best among the internal dynamics models with a BFR of approximately 93 % and matches the performance of the best static feature model $\mathrm{MLP}_\mathrm{f}^{10}$. It should also be mentioned, that both ID_3^3 and $\mathrm{MLP}_\mathrm{f}^{10}$ possess roughly 100 free model parameters. The reason for the decreasing performance of ID_3^n with increasing model complexity n is either due to premature termination of the optimization procedure or due to convergence to poor local optima, which becomes increasingly likely the larger the parameter space is. In this case study 10 multi-starts with random initializations had been conducted for each nonlinear in the parameters model structure and the optimization procedure was terminated after a maximum of 1000 iterations was reached. It is likely, that better local optima can be found, if more multi-starts and/or optimization steps are performed.

Comparing the internal dynamics approach to the static feature model approach, it should be kept in mind, that the internal dynamics models estimate part quality from raw sensor data. The feature models on the other hand estimate part quality from pre-processed features that are chosen based on expert knowledge, which poses a simpler task. Nevertheless, the internal dynamics approach manages to find an internal representation of the data, that is as informative as the tailored features. Considering the circumstances, i.e. that no pre-processing was done and that during learning a single error datum per batch had to be backpropagated over 1000 timesteps, this result is quite impressive. It should also be noted, that the feature model can only be used to predict part quality after the fact, i.e. once the batch is completed. The internal dynamics model on the other hand can be used for in-batch optimization, provided the real-time requirements of the process at hand allow for it.

Considering the fact that the static setpoint models already perform that well shows that there is little dynamic information available in the data that is relevant for quality prediction in this case study. From a purely practical viewpoint the application of feature models let alone internal dynamics models, which are very time consuming to estimate, is not required for predicting part quality in this specific scenario. However, under realistic operating conditions the PIM process is subject to external disturbances due to changing ambient temperatures, fluctuations in raw material properties, machine wear, etc. These disturbances affect part quality and are reflected in the process measurements. E.g. changes of the ambient temperature affect the cavity temperature and through the temperature dependence of the melt viscosity also the cavity pressure. Therefore, under realistic operating conditions the static setpoint models would probably be rendered useless.

VI. Conclusions and Outlook

An internal dynamics approach for batch-end quality prediction from in-process measurements has been proposed and applied to a switched process, the PIM process. The proposed approach performed as well as static model approaches which rely on feature extraction based on expert knowledge and surpassed static models that map process setpoints to part quality on the chosen case study. However, a case study is as usual only a snap-shot. While in this specific application dynamics had an almost negligible effect on part quality, this is of course not true for all batch processes. The proposed approach is therefore a viable option, if static setpoint models and static feature models fail to predict part quality accurately and/or if the goal is to estimate a dynamic model for in-batch process optimization.

Acknowledgment

Funded by the federal state of Hesse and the European Regional Development Fund (ERDF 2014-2020), Project: Digital Twin of Injection Molding (DIM) FKZ: 0107/20007409

References

[1] Yuan Yao and Furong Gao. "A survey on multistage/multiphase statistical modeling methods for batch processes". In: *Annual Reviews in Control* 33.2 (2009), pp. 172–183.

[2] J. Wan, O. Marjanovic, and B. Lennox. "Uneven batch data alignment with application to the control of batch end-product quality". In: *ISA transactions* 53.2 (2014), pp. 584–590.

[3] Yun Zhang et al. "A statistical quality monitoring method for plastic injection molding using machine built-in sensors". In: *The International Journal of Advanced Manufacturing Technology* 85.9 (2016), pp. 2483–2494.

[4] Wen-Chin Chen and Denni Kurniawan. "Process parameters optimization for multiple quality characteristics in plastic injection molding using Taguchi method, BPNN, GA, and hybrid PSO-GA". In: *International journal of precision engineering and manufacturing* 15.8 (2014), pp. 1583–1593.

[5] W.-C. Chen et al. "A neural network-based approach for dynamic quality prediction in a plastic injection molding process". In: *Expert systems with Applications* 35.3 (2008), pp. 843–849.

[6] Kun-Cheng Ke and Ming-Shyan Huang. "Quality Prediction for Injection Molding by Using a Multilayer Perceptron Neural Network". In: *Polymers* 12.8 (2020), p. 1812.

[7] H. Jung et al. "Application of Machine Learning Techniques in Injection Molding Quality Prediction: Implications on Sustainable Manufacturing Industry". In: *Sustainability* 13.8 (2021), p. 4120.

[8] A. Garg et al. "Model predictive control of uniaxial rotational molding process". In: *Computers & Chemical Engineering* 121 (2019), pp. 306–316.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[10] K. Cho, B. v. Merrienboer, and C. Gulcehre. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, 1724–1734.

[11] J. v. d. Westhuizen and J. Lasenby. "The unreasonable effectiveness of the forget gate". In: *CoRR* abs/1804.04849 (2018).

[12] Julian Heinisch, Yannik Lockner, and Christian Hopmann. "Comparison of design of experiment methods for modeling injection molding experiments using artificial neural networks". In: *Journal of Manufacturing Processes* 61 (2021), pp. 357–368.

[13] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.