Data-Driven Fault Detection in Industrial Batch Processes Based on a Stochastic Hybrid Process Model

Stefan Windmann

***Abstract*— This paper presents a novel fault detection approach for industrial batch processes. The batch processes under consideration are characterized by the interaction between discrete system modes and non-stationary continuous dynamics. Therefore, a stochastic hybrid process model (SHPM) is introduced, where process variables are modeled as time-variant Gaussian distributions, which depend on hidden system modes. Transitions between the system modes are assumed to be either autonomous or to be triggered by observable events such as on/off signals. The model parameters are determined from training data using expectation-maximization techniques. A new fault detection algorithm is proposed, which assesses the likelihoods of sensor signals on the basis of the stochastic hybrid process model. Evaluation of the proposed fault detection system has been conducted for a penicillin production process, with the results showing a significant improvement over the existing baseline methods.**

***Note to Practitioners*—Automatic fault detection makes it possible to limit the effects of faults by taking countermeasures at an early stage. In this work, a data-driven fault detection method for industrial batch processes is proposed, in which the underlying process model is learned from training data. The proposed fault detection system can be used for various industrial batch processes without the need for complex and error-prone manual configuration. In contrast to many other data-driven approaches such as neural networks, only a few process cycles are required to create a robust process model. It should be noted that in data-driven fault detection methods, the training data should cover a large part of the process states that occur during error-free process cycles. The developed method is therefore particularly suitable for cyclical processes, which, however, can have alternative process paths and variability between the process cycles.**

***Index Terms*— Fault diagnosis, process monitoring, Hidden Markov Models.**

I. Introduction

THE main part of industrial production systems are safety-critical systems, which require continuous and reliable operation. Potential faults have to be detected as early as possible to prevent system performance degradations and in the worst case the damage or collapse of the whole plant (see e.g. [1], [2]). Thereby, both abrupt changes in the process behavior, for example blockages and switching errors, and continuous performance degradations have to be taken into account.

The scope of this paper is the fault detection in hybrid batch processes like the penicillin production process described in section VII. Monitoring and fault detection for hybrid systems entail challenges due to the fact that continuous dynamics and discrete events are mutually dependent and interact. Hybrid automata are the most common approach to represent hybrid systems within the scope of fault detection and isolation (see section II). In this approach it is assumed that the process can be divided into distinct system modes, i.e. phases of continuous process behavior. Transitions between system modes are usually modeled to be deterministically driven by discrete events (see e.g. [3]). Hidden Markov Models (HMMs) and Segmental HMMs (SHMMs) are generalizations of this approach, which also take into account stochastic transitions between the system modes. Stochastic transitions can be attributed to unobservable discrete events or autonomous changes in continuous process behavior. Existing HMM- or SHMM-based fault detection methods are predominantly focused on modeling of progressive degradations of the plant condition, which are adequately modeled as hidden state sequence with stochastic transitions between failure modes (see e.g. [4], [5]). Initial approaches to model the normal behavior of hybrid industrial systems using HMMs have been introduced in [6] and [7].

The method proposed in this paper is based on the work of [6], where an HMM-based fault detection method for piecewise stationary processes with autonomous transitions between system modes has been developed. Piecewise stationary processes can be considered as sequences of distinct system modes, in which the expectation values and variances of the process variables are constant. In the work of [6], the system modes are modeled as hidden state variables of a HMM. Continuous process variables are assumed to be Gaussian distributed with constant second order statistics in each system mode.

However, for many industrial production processes such as the penicillin production process described in section VII, the assumption of piecewise stationary processes does not hold. The signal progression of penicillin concentration, which

Manuscript received 28 August 2021; revised 6 December 2021; accepted 18 December 2021. Date of publication 5 January 2022; date of current version 13 October 2022. This article was recommended for publication by Associate Editor C.-Y. Lee and Editor F.-T. Cheng upon evaluation of the reviewers' comments.

The author is with the Fraunhofer IOSB-INA, Fraunhofer Center for Machine Learning, 32657 Lemgo, Germany (e-mail: stefan.windmann@iosb-ina.fraunhofer.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2021.3138925.

Digital Object Identifier 10.1109/TASE.2021.3138925

TABLE I
NOMENCLATURE

$\alpha_{i,j}$, $A_{i,j}$	transition probability between system mode i and j
$A_{i,j}$	$\log(\alpha_{i,j})$
$b_{i,l}(\mathbf{o}_{k_0}^{k_0+l-1})$	output density of the hybrid process model
$B_{i,l}(\mathbf{o}_{k_0}^{k_0+l-1})$	$\log(b_{i,l}(\mathbf{o}_{k_0}^{k_0+l-1}))$
$\beta_{i,\nu}$	probability that d adopts the value ν in system mode i
$\mathbf{c}_i$	offset of the trajectory $\mathbf{f}_i(k)$
d	discrete measurement
δ_i	forward variable in the extend Viterbi algorithm
Δ_i	$log(\delta_i)$
$\mathbf{D}$	discrete measurements $(d_0 \dots d_{n-1})$
$\mathbf{\Delta}_{x,i}$	prediction error ($\mathbf{\Delta}_{x,i} = \mathbf{x}(k) - \mathbf{f}_i(k)$)
ϵ, ϵ_{min}	merge error, minimum merge error
$\mathbf{f}_i(k)$	trajectory model in segment i
f_d	detected fault in the discrete event system
f_x	detected fault in the continuous process
$f_s^{(opt)}$, $f_s^{(max)}$	splitting factor, threshold for the splitting factor
h	threshold on the log likelihood ratios in the extended CUSUM algorithm
i, j	system modes
γ	segment identifier
Γ	sequence of segments
k	discrete time index
$\bar{k}_i$	mean value of k in system mode i
$k_{0,\gamma}$	start time of segment γ
l_γ	length of segment γ
l_{min}, l_{max}	lower/upper bound on the segment length
λ	process model
$\lambda^{(0)}$	initial estimate for λ
m	length of a sequence of segments
$\mathbf{m}_i$	slope of the trajectory $\mathbf{f}_i(k)$
μ_{l_i}	expectation value of the segment length in segment i
n	number of observations
n_c	number of components of the measurement vector
N	number of process cycles
$\mathbf{o}$	measurement vector $\mathbf{o} = (d, \mathbf{x})$
$\mathbf{O}$	measurements $(\mathbf{o}_0 \dots \mathbf{o}_{n-1})$
π_i	prior probability distribution of a system mode at $k = 0$
Π_i	$\log(\pi_i)$
s	hidden system mode
s_γ	hidden system mode in segment γ
s_d	initial segment labels
$\mathcal{S}$	value set for the hidden system modes
$\mathbf{S}_q$	segmentation $(q_0, \dots, q_{m-1})$ of the process data
σ_{l_i}	standard deviation of the segment length in segment i
Σ_i	covariance matrix of the trajectory model in segment i
t	iteration index in the estimation maximization algorithm
T	number of iterations of the EM algorithm
T_d	threshold on the probability of discrete events in the extended CUSUM algorithm
$\mathcal{T}$	transition matrix with transition probabilities $\alpha_{i,j}$
$\mathbf{\Theta}$	set of segment models Θ_i
q_γ	segment state
$\mathcal{V}$	value set for the discrete measurements d
$\mathbf{x}$	continuous measurement vector with components x_c
$\bar{\mathbf{x}}_i$	mean value of $\mathbf{x}$ in system mode i
$\mathbf{X}$	continuous measurements $(\mathbf{x}_0 \dots \mathbf{x}_{n-1})$

is shown in Fig. 4, can for example be divided in several segments, in which the expectation value of the observed signal is not a constant but rather a linear or quadratic function of the batch progress.

A substantial novelty of the proposed method is the introduction of a stochastic hybrid process model (SHPM), which allows to model gradually changing system behavior in hidden system states, and, therefore, to reduce the number of hidden states compared to HMMs with piecewise stationary emission distributions. In contrast to existing fault detection systems based on SHMMs (see e.g. [5]), the hidden states of the proposed SHPM do not correspond to different failure modes, but to different process phases that are traversed during a fault-free process. Thus, it is possible to learn the structure and the parameters of the SHPM from observations of the normal process behavior. Model learning is accomplished in two phases: In the first phase, an initial process model is learned using a novel heuristic, which includes both discrete events and continuous signals. Top-down segmentation and bottom-up merging methods are introduced for this task. In the second phase, the model is refined with an expectation-maximization algorithm, which is commonly used in a similar form to learn SHMMs in the context of automatic speech recognition (see e.g. [8]). Finally, a novel fault detection methodology is derived in this paper, in which the SHPM is used to assess the likelihood of faults in hybrid batch processes (see section VI).

The remaining part of the paper is structured as follows: In section II, a summary of the related work is given. Subsequently, the underlying probabilistic framework for fault detection in industrial production processes is outlined in section III. The proposed process model and model learning algorithms are introduced in section IV and section V, respectively. The novel fault detection method is described in section VI. In section VII, the overall fault detection approach is evaluated in two application scenarios. Section VIII gives a conclusion. The most important notations used in this paper are shown in Table I.

II. Related Work

Generally speaking, three classes of algorithmic approaches exist for the detection of faults and anomalous situations in industrial production processes: signal-based, knowledge-based and model-based methods [9].

1) In signal-based methods, features such as mean values, trends or peaks are extracted in time domain or frequency domain from the measured signals. Expert knowledge is then used to assess whether the feature values indicate faults [9]. The advantage of this approach is that it does not require a complex model describing the system behavior, since the analysis relies on features that are derived from the signal waveforms. On the other hand, signal-based fault detection methods often need a high number of classification rules to discriminate between all possible combinations of symptoms. Hence, this approach is used for small modules rather than for complex manufacturing systems.

2) Knowledge-based fault detection methods employ large volumes of historic data to check the consistency between the observed process behavior and the knowledge base, leading to a decision about a possible fault with the aid of a classifier. Knowledge-based approaches can be further classified into qualitative methods and quantitative methods [9], where in particular quantitative methods are of great practical importance. Qualitative methods employ symbolic representations of the input data such as rules or sequences of symbols. The large part of the existing quantitative knowledge-based methods is based on transformations of the multivariate input signals into a low-dimensional subspace. In particular, artificial neural network (ANN) methods (see e.g. [10], [11]) and multivariate

analysis (MVA) methods such as principal component analysis (PCA), independent component analysis (ICA), partial least squares (PLS), and canonical correlation analysis (CCA) are used to transform the input signals [9]. For fault detection, statistics as in particular the squared prediction error (SPE) resulting from the reconstruction of the input data and the T^2 statistics of the dimension-reduced data are evaluated.

The advantage of knowledge-based fault detection methods compared to signal-based approaches is that they can be used in different application fields without much prior knowledge and configuration effort. On the other hand, the automatic generation of ANNs requires large historical data sets, while the basic forms of the MVA methods are limited to linear dimensional reductions. To close the gap, non-linear MVA methods have been developed such as kernel PCA-based methods [12], [13], kernel PLS-based methods [14], kernel ICA-based methods [15] or combinations of deep belief networks and CCA-based methods [16]. The core of the knowledge-based fault detection methods is further limited to processes where the dependencies between the measured signal values and the dimension-reduced features do not change. Some recent approaches overcome this limitation by carrying out the dimensional reductions depending on the current process phase [17], [18]. However, processes, in which the distributions of the process variables change gradually, can only be modeled inadequately with the existing MVA methods.

3) Model-based fault detection methods employ a model to simulate the normal process behavior [9]. If the actual measurements vary significantly from the simulation results, the behavior is classified as anomalous (see e.g. [2], [3], [19]). Model-based approaches allow to monitor complex processes even in the case of gradually changing process behavior. Furthermore, no complicated decision rules are required as far as sufficiently accurate process models are available. However, in practice, model-based methods require a high level of manual model creation effort or a sufficient amount of data for automatic identification of the model parameters. Therefore, model-based methods are particularly suitable for processes with a limited number of system modes in which sufficient data for automatic model generation is available after a few process cycles. As this is the case for the batch processes under consideration, a model-based approach is used in this work.

The following sections outline the related work in model-based fault detection and model learning.

A. Model-Based Fault Detection

Model-based fault detection methods can be divided into approaches for discrete events systems and approaches for continuous systems. Fault detection methods for continuous systems can further be divided into deterministic and stochastic methods. The batch processes investigated in this paper are hybrid processes, which are characterized by the interaction of discrete event systems and continuous processes.

1) Faults in discrete event systems such as incorrect transitions between system modes and timing errors are straightforwardly detected by comparing the observed discrete events with events that are predicted using a model of the discrete event system (see e.g. [20], [21]). In the large part of the existing application cases, this approach is based on Petri nets or automata.

2) Deterministic fault detection methods for continuous systems are based on observers that use deterministic models of the monitored processes to simulate the system behavior and to predict the system output. The observers compute residuals, i.e. the differences between the actual and the expected outputs of the monitored processes, which are analyzed for fault detection (see e.g. [22]). However, many processes behave non-deterministically due to process and measurement noise, so that stochastic methods are often required in in industrial applications.

3) Stochastic fault detection methods for continuous systems employ stochastic models to generate the residuals [23]. In many of these approaches, a stochastic state space model is used to model the temporal transitions of continuous state variables, which are related to the observations via a measurement model. Common approaches are Kalman filters (e.g. [24]–[26]) and particle filters [27], in which the probability distributions of the state variables are approximated by second order statistics or particles, respectively. Considering the measurement and state space models, the filters determine estimates of the state variables at each time instance, which are used to compute the residuals. An alternative approach to state space models, which allows to model more complex nonlinear dependencies in signals, is the use of recurrent neural networks like Long Short Term Memories (LSTMs) for process modeling [28]. However, neural networks usually require a larger amount of training data compared to state space models. Potential faults are determined in the stochastic fault detection methods using statistical analysis of the residuals. For this, likelihoods of the residuals are computed or statistical tests e.g. $\mathcal{X}^2$ or multiple hypothesis tests are carried out on the residuals [29]. The tests can be applied either instantaneously or cumulatively, whereby calculating cumulative sums generally allows to find smaller errors while accepting delayed detection.

4) Fault detection in hybrid systems has been investigated e.g. in [3], [6], [7], [21], [30]. The approaches are based on the aforementioned methods for continuous systems and discrete event systems and can be divided accordingly into deterministic and stochastic approaches. Most of the deterministic approaches for hybrid systems use automata to describe the underlying discrete event system. Different formalisms have been investigated to represent the continuous dynamics within discrete system modes, e.g. differential equations [3], [21] or regression models [30]. In all of these approaches, the parameters of the continuous models depend on automaton states related to the system modes. However, the differential equations proposed in [3], [21] are more general compared to the regression models proposed in [30], as they consider continuous state variables in addition to the automaton states.

Stochastic hybrid process models such as Hidden Markov Models (HMMs) and switching Kalman filters have been proposed to deal with hidden system modes and process and modeling uncertainties that are not covered by deterministic hybrid automata. The authors in [31] and [32] employ HMMs to capture the behavior of the underlying discrete event system,

while Kalman Filters represent the continuous dynamics in the individual system modes. In [6] and [7], the process behavior in the individual system modes is modeled using stationary Gaussian distributions. Compared to the switching Kalman filters introduced in [32], HMMs with stationary Gaussian distributions are more suitable to model long-term dependencies in the data, leading to higher accuracy in many application cases [6]. However, HMMs with stationary Gaussian distributions capture non-stationary process behavior in a system mode only in the form of an increased variance, which reflects the increased modeling error. As endorsed in the introduction, the proposed methods aim to close this gap using a hybrid process model, which is based on a segmental HMM.

B. Model Learning

A large part of the existing model-based fault detection approaches employ manually created process models. However, manual creation of process models is error-prone and time-consuming due to the high complexity and flexibility of industrial production systems [33]. Hence, model learning approaches have been developed, which allow to create process models automatically from historical process data. Model learning methods for continuous process models have been the subject of extensive research. A large number of learning algorithms has been developed for neural networks, which are often based on gradient descent methods (see e.g. [28]). The identification of stochastic process models has also been extensively studied and is usually based on maximizing the log-likelihood of the measured output signals (e.g. [24]). In the case of hidden state variables, which are used for example in Kalman filters or HMMs, maximization of the log-likelihood leads to Expectation Maximization (EM) approaches, where the hidden state variable estimation and parameter maximization are performed iteratively. Learning discrete event system models and in particular automata from historical process data is not as well studied. Learning algorithms for automata can be divided into offline algorithms (e.g. ALERGIA [34], MDI [35] and BUTLA [36]) and online algorithms (e.g. OTALA [37]). In the main part of the offline algorithms, prefix trees are computed by combining common beginnings of discrete event sequences that are used as training data. Subsequently, similar states in the prefix trees are iteratively merged until the final automata are obtained. The online learning algorithm OTALA creates a specific state in the automaton for each signal vector of the inputs/outputs [37]. OTALA needs least computing time of the aforementioned model learning algorithms and achieves the same model learning accuracy as BUTLA [37], which in turn was shown to be superior to ALERGIA and MDI [38].

An initial approach to learn a deterministic hybrid process model from data is the HyBUTLA algorithm introduced in [20]. In this approach, an automaton is generated using the BUTLA algorithm and a regression model is learned for each state of the automaton to model the continuous process behavior. In the present work, a model learning approach for the proposed hybrid process model is developed, which is based on learning procedures for SHMMs introduced in section V, which have been originally developed for acoustic models in the field of automatic speech recognition [8]. In contrast to the HyBUTLA algorithm, this approach particularly takes account of hidden system modes, temporal correlations in the continuous signals, and process and modeling uncertainties.

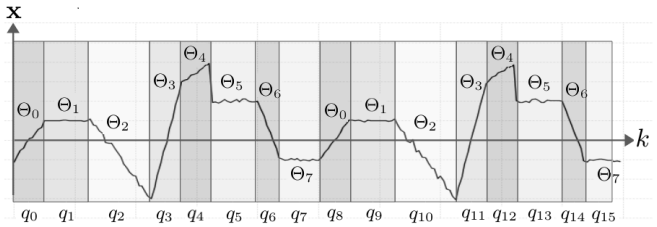

Fig. 1. Segmentation $\mathbf{S}_q = (q_0, \ldots, q_{15})$ of two example process cycles, in which the 16 segments are represented by the 8 models $\Theta_i, i = 0, \ldots, 7$.

III. Overall Fault Detection Approach

Fault detection is carried out using a process model which allows to detect deviations between the observed and predicted signal courses. In this section, this model is briefly described before model learning and model-based fault detection are outlined.

The proposed stochastic hybrid process model (SHPM) is based on the assumption that the monitored processes consist of several process modes $i \in \mathcal{S}$ with qualitatively different process behavior. The observed signals are segmented according to the process modes, and segment models Θ_i are used to model the process behavior in the individual segments $q_\gamma, \gamma = 0, \ldots, m-1$ (see Fig. 1).

To model the dynamics of the continuous observations $\mathbf{x}(k)$, time-dependent linear models with modeling errors captured in the standard deviations of the segment models are used. It is worth noting that HMMs with stationary Gaussian-distributed emission distributions, which are e.g. applied in [6], require in general a larger number of states within each segment to achieve the same modeling accuracy as the proposed SHPM due to the assumption of constant expected values $E[\mathbf{x}]$ for each state. Furthermore, in contrast to HMM-based approaches an explicit probabilistic model for the length of the individual segments is considered in the SHPM.

Fault detection is conducted in two phases, which are shown in Fig. 2:

1) Training phase: In this phase, the SHPM is learned from historic training data.
2) Operating phase: Fault detection is conducted during the operation of the industrial process.

In the training phase, an Expectation-Maximization (EM) algorithm is used to learn the SHPM λ from historical training data

$$\mathbf{O} = \{o(0), \ldots, o(n-1)\}. \tag{1}$$

The EM algorithm consists of two steps, which are repeated in the iterations $t = 0 \ldots T$ (see section V-B for details):

- Maximization step: Computation of the model parameters $\lambda^{(t)}$) for a given segmentation $\mathbf{S}_q^{(t)}$
- Expectation step (extended Viterbi algorithm): Estimation of the optimal segmentation $\mathbf{S}_q^{(t+1)}$ for a given model $\lambda^{(t)}$.

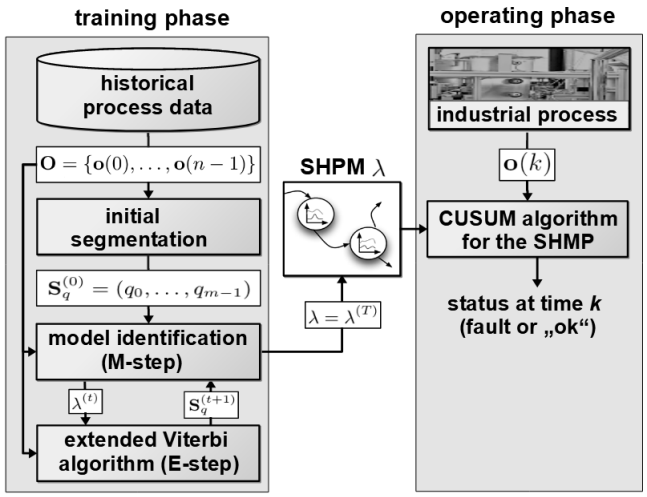

Fig. 2. Model-based fault detection with the stochastic process model λ.

An initial segmentation $\mathbf{S}_q^{(0)}$ of the training data is required to initialize the EM algorithm, which is determined using a heuristic introduced in section V-A.

The model $\lambda = \lambda^{(T)}$ that results from T iterations of the EM algorithm is used for fault detection during the operating phase of the industrial process. Fault detection at time k is based on predictions $\hat{\mathbf{o}}(\kappa)$ of the measurements $\mathbf{o}(\kappa)$, which are determined with the SHPM for time $\kappa = 0 \ldots k$. Based on the residuals $r(\kappa) = o(\kappa) - \hat{o}(\kappa), \kappa = 0 \ldots k$, a cumulative fault measure is computed in the CUSUM (cumulative sum) algorithm introduced in section VI. A fault is reported as soon as the cumulative fault measure exceeds a given threshold.

IV. Stochastic Hybrid Process Model

The fault detection approach employs a stochastic hybrid process model (SHPM), which allows to capture both discrete and non-stationary continuous behavior of industrial production processes. The proposed SHPM is based on a Segmental Hidden Markov Model (SHMM), which has been originally introduced for acoustic modeling in automatic speech recognition ([8], [39]), i.e. for the modeling of continuous speech features $\mathbf{x}(k)$.

In this work, standardization of the continuous measurements $\mathbf{x}(k)$ is conducted to compensate for different scales in the components $x_c(k)$:

$$x_c(k) \mapsto \frac{x_c(k) - \min\limits_{\kappa=0\ldots n-1} x_c(\kappa)}{\max\limits_{\kappa=0\ldots n-1} x_c(\kappa) - \min\limits_{\kappa=0\ldots n-1} x_c(\kappa)} \tag{2}$$

Discrete observations $d(k)$ are integrated into this approach to exploit measurable events, such as the switching of valves.

The proposed process model, which is depicted in Fig. 3, divides the sequence of observations into a sequence of segments with segment indices $\gamma \in \Gamma$, starting times $k_{0,\gamma}$ and segment lengths l_γ. The length of the sequence is denoted by $m = |\Gamma|$ in the following.

Formally, the proposed model is a 3-tuple $\lambda = (\mathcal{S}, \mathcal{T}, \Theta)$ with:

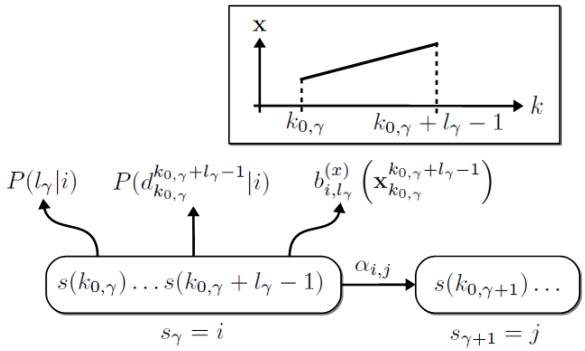

Fig. 3. Stochastic hybrid process model.

- A finite set of system modes $\mathcal{S}$: At each time instance k the system is assumed to be in the system mode

$$s(k) = i, \quad i \in \mathcal{S} \tag{3}$$

with the probability $P(s(k) = i)$. The system modes are assumed to be constant within a segment γ, i.e.

$$s(k_{0,\gamma}) = \ldots = s(k_{0,\gamma} + l_\gamma - 1) = s_\gamma . \tag{4}$$

To combine the variables s_γ, $k_{0,\gamma}$ and l_γ, the segment state

$$q_\gamma = \left(k_{0,\gamma}, l_\gamma, s_\gamma\right). \tag{5}$$

is introduced.

- Transition matrix $\mathcal{T} = \{\alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{|\mathcal{S}|,|\mathcal{S}|}\}$: $\mathcal{T}$ comprises the transition probabilities between two segments $i \in \mathcal{S}$ and $j \in \mathcal{S}$.
- Segment models $\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_{|\mathcal{S}|}\}$: A segment model provides a joint model for a random-length sequence of the discrete observations

$$d_{k_0}^{k_0+l-1} = [d(k_0), \ldots, d(k_0 + l - 1)] \tag{6}$$

and the continuous observations

$$\mathbf{x}_{k_0}^{k_0+l-1} = [\mathbf{x}(k_0), \ldots, \mathbf{x}(k_0 + l - 1)], \tag{7}$$

which is assumed to be generated for each segment $i \in \mathcal{S}$ with segment start time k_0 and segment length l according to the density

$$\begin{aligned} &p(d_{k_0}^{k_0+l-1}, \mathbf{x}_{k_0}^{k_0+l-1}, l|i) \\ &\quad = p(\mathbf{x}_{k_0}^{k_0+l-1}|l, i) P(d_{k_0}^{k_0+l-1}|i) p(l|i). \end{aligned} \tag{8}$$

Each segment model Θ_i consists of the following components:

- A prior probability distribution for the segment model at time instance $k = 0$: $\pi_i = p(l = 1, i)$.
- A duration distribution $p(l|i)$ that gives the probability density of segment length l and, thereby, the likelihood of a particular segmentation of a process cycle. In this work, a Gaussian duration distribution

$$p(l|i) = \mathcal{N}(l; \mu_{l,i}, \sigma_{l,i}) \tag{9}$$

with mean $\mu_{l,i}$ and standard deviation $\sigma_{l,i}$ is assumed. Additionally, boundaries l_{min} and l_{max}, $l_{min} \leq l \leq l_{max}$,

for the segment length l are defined to restrict the length of the individual segments.

- A family of output densities $\{b_{i,l}; l_{min} < l \leq l_{max}\}$ that describes observation sequences of different lengths l. In this work, the output densities are composed of conditional probabilities $b_{i,l}^{(d)}(d_{k_0}^{k_0+l-1}) = P(d_{k_0}^{k_0+l-1}|i)$ for the discrete events and linear trajectory segment models $b_{i,l}^{(x)}(\mathbf{x}_{k_0}^{k_0+l-1})$ for the continuous process behavior:

$$b_{i,l}\left(d_{k_0}^{k_0+l-1}, \mathbf{x}_{k_0}^{k_0+l-1}\right) = b_{i,l}^{(d)}(d_{k_0}^{k_0+l-1}) b_{i,l}^{(x)}\left(\mathbf{x}_{k_0}^{k_0+l-1}\right) \quad (10)$$

In the following, the discrete observations in a segment are assumed to be conditionally independent, i.e. the approximation

$$b_{i,l}^{(d)}(d_{k_0}^{k_0+l-1}) = P(d_{k_0}^{k_0+l-1}|i) \approx \prod_{k=k_0}^{k_0+l-1} P(d(k)|i) \quad (11)$$

is made. The probability that $d(k)$ adopts the value $v \in \mathcal{V}$ in system mode i is denoted as

$$\beta_{i,v} = P(d(k) = v|i). \quad (12)$$

The continuous output densities are modeled using the Gaussian probability density

$$b_{i,l}^{(x)}\left(\mathbf{x}_{k_0}^{k_0+l-1}\right) = \prod_{k=k_0}^{k_0+l-1} \mathcal{N}(\mathbf{x}(k); \mathbf{c}_i + \mathbf{m}_i(k-k_0)), \boldsymbol{\Sigma}_i) \quad (13)$$

where $\mathbf{m}_i$ and $\mathbf{c}_i$ denote the slope and the offset of the trajectory

$$\mathbf{f}_i(k-k_0) = \mathbf{c}_i + \mathbf{m}_i(k-k_0) \quad (14)$$

in segment i. The co-variance matrix of the trajectory $\mathbf{f}_i(k)$ is referred to as $\boldsymbol{\Sigma}_i$.

Altogether, the individual segment models can be written as 7-tuples

$$\Theta_i = \left(\pi_i, \mu_{l,i}, \sigma_{l,i}, \beta_{i,v}, \mathbf{c}_i, \mathbf{m}_i, \boldsymbol{\Sigma}_i\right). \quad (15)$$

V. Model Learning

This section describes the learning of the SHPM λ from historical process data $\mathbf{O}$. The basic concept behind the proposed model learning method consists in (see Fig. 2 in section III) (A) creating an initial segmentation of the historical process data, and (B) iteratively improving the model parameters and the initial segmentation using maximum likelihood (ML) parameter estimation.

A. Initial Segmentation

To obtain an initial segmentation in which both the discrete events and the continuous dynamics are considered, the following steps are performed:

- Step 1 Segmentation of the process data with respect to the observed discrete signals
- Step 2 Top-down segmentation: Segments with discontinuities in the continuous signals are split.
- Step 3 Bottom-up merging: Similar segments are assigned to a single system mode for which a common segment model is learned further on.

The three steps are realized in the following way (Alg. 1):

Algorithm 1 Segmentation Heuristic

Step 1: Segmentation with respect to the discrete signals
Given: Discrete observations $\mathbf{D} = \{d(0), \ldots, d(n-1)\}$
Result: segment states $q_\gamma, \gamma \in \Gamma'$
(01) $\forall d(k) \in \mathbf{D}$:
(02) $\quad s_d(k) = \mathrm{f}_d(d(k))$
(03) $\Gamma' = \emptyset, \quad k_{0,0} = 0, \quad \gamma = 0$
(04) **for** $k = 1 \ldots n-1$:
(05) $\quad$ **if** $s_d(k) \neq s_d(k-1)$:
(06) $\quad\quad k_{0,\gamma+1} = k, \quad l_\gamma = k - k_{0,\gamma}, \quad s_{d,\gamma} = s_d(k-1)$
(07) $\quad\quad q_\gamma = (k_{0,\gamma}, l_\gamma, s_{d,\gamma}), \quad \Gamma' = \Gamma' \cup \{\gamma\}$
(08) $\quad\quad \gamma = \gamma + 1$
(09) $q_\gamma = (k_{0,\gamma}, n - k_{0,\gamma} - 1, s_d(n-1)), \quad \Gamma' = \Gamma' \cup \{\gamma\}$

Step 2: Top-down segmentation
Given: Observations $\mathbf{O} = \{\mathbf{o}(0), \ldots, \mathbf{o}(n-1)\}$,
segment states $q_\gamma, \gamma \in \Gamma'$
Result: refined segment states $q_\gamma, \gamma \in \Gamma$
(01) **while** $|\Gamma'| > 0$:
(02) $\quad$ Select $\gamma \in \Gamma'$ with segment state $q_\gamma = \left(k_{0,\gamma}, l_\gamma, s_\gamma\right)$
(03) $\quad$ **for** $c = 1 \ldots n_c$:
(04) $\quad\quad e_c^{(\gamma)} = \sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1} \left(x_c(k) - f_{s_\gamma,c}(k-k_{0,\gamma})\right)^2$
(05) $\quad f_s^{(opt)} = 1.0$
(06) $\quad$ **for** $k = k_{0,\gamma} + \frac{l_{min}}{2} - 1 \ldots k_{0,\gamma} + l_\gamma - \frac{l_{min}}{2}$:
(07) $\quad\quad q_{\gamma^{(l)}} = (k_{0,\gamma}, k - k_{0,\gamma} + 1, s_{\gamma^{(l)}})$
(08) $\quad\quad q_{\gamma^{(r)}} = (k+1, k_{0,\gamma} + l_\gamma - 1 - k, s_{\gamma^{(r)}})$
(09) $\quad\quad$ Update $\mathbf{m}_{s_{\gamma^{(l)}}}$ and $\mathbf{m}_{s_{\gamma^{(r)}}}$ according to (17)
(10) $\quad\quad$ Update $\mathbf{c}_{s_{\gamma^{(l)}}}$ and $\mathbf{c}_{s_{\gamma^{(r)}}}$ according to (18)
(11) $\quad\quad$ **for** $c = 1 \ldots n_c$:
(12) $\quad\quad\quad$ **if** $e_c^{(opt)}/l_\gamma > e_{min}$:
(13) $\quad\quad\quad\quad \tilde{e}_c^{(l)} = \sum_{\kappa=k_{0,\gamma}}^{k} \left(x_c(\kappa) - f_{s_{\gamma^{(l)}},c}(\kappa - k_{0,\gamma})\right)^2$
(14) $\quad\quad\quad\quad \tilde{e}_c^{(r)} = \sum_{\kappa=k+1}^{k_{0,\gamma}+l_\gamma-1} \left(x_c(\kappa) - f_{s_{\gamma^{(r)}},c}(\kappa - k_{0,\gamma})\right)^2$
(15) $\quad\quad\quad\quad \tilde{e}_c = \tilde{e}_c^{(l)} + \tilde{e}_c^{(r)}$
(16) $\quad\quad\quad\quad$ **if** $\tilde{e}_c/e_c^{(\gamma)} < f_s^{(opt)}$:
(17) $\quad\quad\quad\quad\quad \gamma^{(l,opt)} = \gamma^{(l)}, \gamma^{(r,opt)} = \gamma^{(r)}, f_s^{(opt)} = \tilde{e}_c/e_c^{(\gamma)}$
(18) $\quad\quad$ **if** $f_s^{(opt)} < f_s^{(max)}$:
(19) $\quad\quad\quad \Gamma' = \Gamma' \cup \{\gamma^{(l,opt)}, \gamma^{(r,opt)}\} \setminus \{\gamma\}$
(20) $\quad$ **else**: $\Gamma' = \Gamma' \setminus \{\gamma\}, \Gamma = \Gamma \cup \{\gamma\}$

Step 3: Bottom-up merging
Given: segment states $q_\gamma = (k_{0,\gamma}, l_\gamma, s_\gamma), \gamma \in \Gamma$
Result: updated segment labels s_γ for the segments $\gamma \in \Gamma$
(01) $e_{min} = \infty$
(02) **do**
(03) $\quad \forall(\gamma_1, \gamma_2)((\gamma_1, \gamma_2) \in \Gamma \times \Gamma \wedge \gamma_1 \neq \gamma_2)$:
(04) $\quad\quad$ **if** $s_{d,\gamma_1} = s_{d,\gamma_2}$:
(05) $\quad\quad\quad$ Compute merge error e according to (19)
(06) $\quad\quad\quad$ **if** $e < e_{min}$: $e_{min} = e$, $\gamma_1^{(opt)} = \gamma_1$, $\gamma_2^{(opt)} = \gamma_2$
(07) $\quad$ **if** $e_{min} < 1$: $s_{\gamma_1^{(opt)}} = s_{\gamma_2^{(opt)}}$
(08) **while** $e_{min} < 1$: $\forall \gamma \in \Gamma$: $s_\gamma = \mathrm{rank}(s_\gamma)$

- *Step 1: Segmentation With Respect to the Discrete Signals* In the first step, a segmentation with respect to the discrete events $d(k)$ is computed, which is refined in step 2 based on the continuous observations $\mathbf{x}(k)$. For this, initial approximations $s_d(k)$ of the system modes $s(k)$ are computed from the discrete input signals $d(k)$ using an automata learning approach. In this work, the OTALA algorithm [37] is used for this purpose

(lines (1) and (2) of step 1). In this approach, a bijective mapping $s_d(k) = \mathrm{f}_d(d(k))$ from the set of discrete events to unique state IDs is applied to determine the initial system modes $s_d(k)$. The segment states $q_\gamma, \gamma \in \Gamma'$ are straightforwardly obtained from the initial system modes $s_d(k)$ (lines (03) - (09)) by starting a new segment at splitting points k, where the condition $s_d(k) \neq s_d(k-1)$ holds. In this process, the segment states q'_γ are updated using the splitting times k and the segment labels at time instances $k-1$. It is worth noting that a single segment will result after step 1, if the discrete inputs are not available. In this case, segmentation is performed only in step 2 based on the continuous signals.

- *Step 2: Top-Down Segmentation of the Observed Process Data*
  In the second step, a top-down segmentation is accomplished to refine the initial segmentation Γ'. The fundamental principle of the proposed method consists in reducing the overall model error

$$e_c = \sum_{\gamma \in \Gamma'} \sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1} \left(x_c(k) - f_{s_\gamma,c}(k-k_{0,\gamma})\right)^2 \quad (16)$$

between the components $x_c(k)$, $c \in \{1 \dots n_c\}$, of the training data and the components $f_{s_\gamma,c}(k - k_{0,\gamma})$ of the modeled trajectory within the segments $\gamma \in \Gamma'$ with segment states $q_\gamma = \left(k_{0,\gamma}, l_\gamma, s_\gamma\right)$. For this issue, each segment is iteratively divided into smaller segments until the desired accuracy is achieved or no further reduction of the overall model error is possible (see step 2 in Alg. 1). In line (02) of step 2 in Alg. 1, an arbitrary segment $\gamma \in \Gamma'$ with start time $k_{0,\gamma}$, length l_γ and segment model s_γ is selected as a candidate for splitting. In line (04), the model error $e_c^{(\gamma)}$ within this segment is computed for each component $x_c(k)$ of the training data. Splitting is accomplished in lines (06)-(10) for all potential time instances $k = k_{0,\gamma} + \frac{l_{min}}{2} - 1 \dots k_{0,\gamma} + l_\gamma - \frac{l_{min}}{2}$ within the respective segment γ. In doing so, segment γ is divided into two smaller segments γ', $\gamma' \in \{\gamma^{(l)}, \gamma^{(r)}\}$, with parameters $k_{0,\gamma'}$, $l_{\gamma'}$ and $s_{\gamma'}$ (lines (07) and (08)). For both segments, slope

$$\mathbf{m}_{s_{\gamma'}} = \frac{\sum_{k=k_{0,\gamma'}}^{k_{0,\gamma'}+l_{\gamma'}-1} \left(k - \frac{l_{\gamma'}}{2} - k_{0,\gamma'}\right)\left(\mathbf{x}(k) - \bar{\mathbf{x}}_{s_{\gamma'}}\right)}{\sum_{k=k_{0,\gamma'}}^{k_{0,\gamma'}+l_{\gamma'}-1} \left(k - \frac{l_{\gamma'}}{2} - k_{0,\gamma'}\right)^2} \quad (17)$$

and offset

$$\mathbf{c}_{s_{\gamma'}} = \bar{\mathbf{x}}_{s_{\gamma'}} - \mathbf{m}_{s_{\gamma'}} \frac{l_{\gamma'}}{2} \quad \text{with} \quad \bar{\mathbf{x}}_{s_{\gamma'}} = \frac{1}{l_{\gamma'}} \sum_{k=k_{0,\gamma'}}^{k_{0,\gamma'}+l'_\gamma-1} \mathbf{x}(k) \quad (18)$$

are updated (lines (09) and (10)). The splitting factor, i.e. the quotient $e_c/e_c^{(\gamma)}$ of the model errors $\tilde{e}_c$ in the divided segments and the model error $e_c^{(\gamma)}$ in the overall segment γ, is used to select the optimal splitting point for segment γ. In doing so, the model errors $\tilde{e}_c$ are computed in lines (11)-(15), while the splitting factor $f_s^{(opt)}$ is updated in lines (05), (16) and (17). Condition (12) is used to restrict the number of segments by introducing a threshold e_{min} on the desired accuracy. In line (19), segment γ is divided, if the splitting factor $f_s^{(opt)}$ is lower than a given threshold $f_s^{(max)}$. Otherwise, γ is included in the set Γ of refined segments (line (20)).

- *Step 3: Bottom-Up Merging of Similar Segments*
  To reduce the number of segments, pair-wise bottom-up merging of similar segments is accomplished. Pairs of segments γ_1 and γ_2 are merged, where the discrete state labels s_{d,γ_1} and s_{d,γ_2} are identical and where the merge error

$$e = \max_{c=1\dots n_x} \frac{1}{e_{c,\max}} \frac{e_{c,\gamma_1} + e_{c,\gamma_2}}{l_{\gamma_1} + l_{\gamma_2}} \quad (19)$$

with

$$e_{c,\gamma} = \sum_{k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma+l_\gamma-1} \left(x_c(k) - f_{i,c}(k-k_{0,\gamma})\right)^2$$

$$e_{c,max} = \max_{\gamma \in \Gamma} \frac{1}{l_\gamma} \sum_{k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma+l_{\gamma_1}-1} \left(x_c(k) - f_{i,c}(k-k_{0,\gamma})\right)^2$$

  is below a given threshold (lines (01)-(07) in step 3 of Alg. 1). In this process, the two segments with the lowest merge error e are successively merged by assigning identical segment labels s_γ to the respective segments. In line (08), each ID is replaced by the respective rank to avoid gaps in the set of IDs. It should be noted that in the initial segmentation heuristic, only segments with identical discrete state labels are merged. Segments with different discrete state labels are potentially merged in the course of the EM algorithm introduced in section V-B.

The initial segmentation $\mathbf{S}_q^{(0)} = (q_0, \dots, q_{m-1}), m = |\Gamma|$, is obtained by arranging the segment states $q_\gamma, \gamma \in \Gamma$, determined in steps 2 and 3 of Alg. 1 in ascending order by the start times $k_{0,\gamma}$.

B. Maximum Likelihood Estimation

The initial parameters $\lambda = \lambda^{(0)}$ of the hybrid process model are computed from the initial segmentation $\mathbf{S}_q^{(0)}$ as described in section V-B1. Subsequently, the parameters λ are improved using maximum likelihood (ML) parameter estimation, i.e. $p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1})$ is optimized with respect to λ. Since $p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1})$ comprises the hidden segment states q_γ, some form of iteration is required. Similar as in [40], the expected value

$$\begin{aligned} Q(\lambda|\lambda^{(t)}) &= E_{q_0^{m-1}|\mathbf{o}_0^{n-1},\lambda^{(t)}}[\log\ p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1})] \\ &= \sum_{q_0^{m-1}} P(q_0^{m-1}|\mathbf{o}_0^{n-1}, \lambda^{(t)}) \log\ p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1}) \end{aligned} \quad (20)$$

of $\log p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1})$ is maximized by iteratively applying an expectation-maximization algorithm with the following steps:

1) Estimation step: Compute the expected value $Q(\lambda|\lambda^{(t-1)})$ for the current estimates of the parameters $\lambda^{(t-1)}$ according to (20). This step is implicitly achieved by calculating the probabilities $P(q_0^{m-1}|\mathbf{o}_0^{n-1}, \lambda^{(t-1)})$ in (20), since $\lambda^{(t-1)}$ and $\mathbf{o}_0^{n-1}$ are given.
2) Maximization step: Find the parameters $\lambda^{(t+1)}$ that maximize $Q(\lambda|\lambda^{(t)})$:

$$\lambda^{(t)} = \underset{\lambda}{\operatorname{argmax}}\ Q(\lambda|\lambda^{(t-1)}) \tag{21}$$

with $t = 1 \ldots T$ denoting the iteration index. Two common realizations of the expectation-maximization algorithm are the generalized forward-backward algorithm [41] and methods, which find the most likely hidden state sequence at each iteration [42]. Due to the shorter computation time, the second approach is chosen in this paper. In this approach, the Viterbi approximation [40] is applied, i.e. the sum over all possible state sequences in the computation of $Q(\lambda|\lambda^{(t)})$ is approximated by the summand $\tilde{Q}(\lambda|\lambda^{(t)})$ corresponding to the most probable sequence of states in (20):

$$Q(\lambda|\lambda^{(t)}) \approx \tilde{Q}(\lambda|\lambda^{(t)}) = \log\ p_\lambda(\hat{q}_0^{\hat{m}-1}, \mathbf{o}_0^{n-1}) \tag{22}$$

with

$$(\hat{m}, \hat{q}_0^{\hat{m}-1}) = \underset{q_0^{m-1}}{\operatorname{argmax}}\ p_\lambda(q_0^{m-1}, \mathbf{o}_0^{n-1}) \tag{23}$$

As described in detail in section V-B1, the maximization step (21) is conducted by computing the model parameters $\lambda^{(t)}$ for the most probable sequence of states $\mathbf{S}_q^{(t)} = \hat{q}_0^{\hat{m}-1}$. The estimation step, where the sequence $\mathbf{S}_q^{(t)} = \hat{q}_0^{\hat{m}-1}$ is computed as approximation of $P(q_0^{m-1}|\mathbf{o}_0^{n-1}, \gamma^{(t)})$, is detailed in section V-B2.

1) Maximization Step (Model Identification): The transition probabilities $\alpha_{i,j}$ between the system modes and the parameters of the linear trajectory segment models

$$\Theta_i = (\pi_i, \mu_{l,i}, \sigma_{l,i}, \beta_{i,\nu}, \mathbf{m}_i, \mathbf{c}_i, \mathbf{\Sigma}_i) \tag{24}$$

for the individual system modes i are estimated as follows for a given sequence of states $\mathbf{S}_q = q_0^{m-1}$ (see appendix I for the derivations):

$$\pi_i = \frac{1}{m}\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i) \tag{25}$$

$$\alpha_{i,j} = \frac{1}{m-1}\sum_{\gamma=0}^{m-2}\delta(s_\gamma - i, s_{\gamma+1} - j) \tag{26}$$

$$\mu_{l,i} = \frac{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)l_\gamma}{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)}, \quad \sigma_{l,i}^2 = \frac{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\left(l_\gamma - \mu_{l,i}\right)^2}{\left(\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\right) - 1} \tag{27}$$

$$\beta_{i,\nu} = \frac{\sum_{\gamma=0}^{m-1}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - i, d(k) - \nu)}{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)l_\gamma} \tag{28}$$

$$\mathbf{m}_i = \frac{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\left(k - \bar{k}_i\right)(\mathbf{x}(k) - \bar{\mathbf{x}}_i)}{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\left(k - \bar{k}_i\right)^2} \tag{29}$$

$$\mathbf{c}_i = \bar{\mathbf{x}}_i - \mathbf{m}_i\bar{k}_i \tag{30}$$

$$\mathbf{\Sigma}_i = \frac{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\mathbf{\Delta}_{x,i}(k_{0,\gamma}, k)\mathbf{\Delta}_{x,i}(k_{0,\gamma}, k)^T}{\sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)l_\gamma} \tag{31}$$

with

$$\bar{\mathbf{x}}_i = \sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\frac{\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\mathbf{x}(k)}{l_\gamma},$$

$$\bar{k}_i = \sum_{\gamma=0}^{m-1}\delta(s_\gamma - i)\frac{l_\gamma}{2}$$

$$\mathbf{\Delta}_{x,i}(k_{0,\gamma}, k) = \mathbf{x}(k) - \mathbf{f}_i(k - k_{0,\gamma}) \tag{32}$$

In (25)-(31), the Dirac function is denoted as $\delta(\ldots)$.

2) Expectation Step (Extended Viterbi Algorithm): To find both the most likely number of segments $\hat{m}$ and the most likely sequence of segments $\hat{q}_0^{\hat{m}-1}$, the optimization problem (22) is solved for given model paramters $\lambda^{(t)}$. Using the definition $q_\gamma = (k_{0,\gamma}, l_\gamma, s_\gamma)$, eq. (23) can be written as

$$(\hat{m}, \hat{k}_{0,0}^{\hat{m}-1}, \hat{l}_0^{\hat{m}-1}, \hat{s}_0^{\hat{m}-1}) = \underset{m, k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-1}}{\operatorname{argmax}}\ p_\lambda(k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-1}, \mathbf{o}_0^{n-1}) \tag{33}$$

For efficient solution, the variable $\delta_i(k)$ is defined to be the probability of the most likely segmentation sequence ending with segment label i for observations $\mathbf{o}_0^{n-1} = \{o_0, \ldots, o_{n-1}\}$ [40]:

$$\delta_i(k) = \max_{m, q_0^{m-1}}\ p(k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-2}, s_{m-1} = i, \mathbf{o}_0^{n-1}). \tag{34}$$

It is worth noting that the relationship

$$\tilde{Q}(\lambda|\lambda^{(t)}) = \max_{i\in\mathcal{S}}\ \log\delta_i(n-1) \tag{35}$$

holds, so that the final segment s_m of the most likely segmentation sequence can be calculated directly from $\delta_i(n-1)$:

$$\hat{s}_m = \underset{i\in\mathcal{S}}{\operatorname{argmax}}\ \log\delta_i(n-1) = \underset{i\in\mathcal{S}}{\operatorname{argmax}}\ \delta_i(n-1) \tag{36}$$

The recursion

$$\delta_i(k) = \begin{cases} \pi_i b_{i,0}^{(d)}(d(0))b_{i,0}^{(x)}(\mathbf{x}(0)), & \text{if } k = 0 \\ \max\limits_{\substack{l=1\ldots k \\ j\in\mathcal{S}}} \left\{ \begin{array}{l} \delta_j(k-l+1)\alpha_{ij}\,p(l|i) \\ \quad \cdot P(d(k-l+1)\ldots d(k)|i) \\ \quad \cdot b_{i,l}^{(x)}(\mathbf{x}(k-l+1)\ldots\mathbf{x}(k)) \end{array} \right\}, & \text{else} \end{cases} \tag{37}$$

which results directly from (34), allows for an efficient computation of the forward variable $\delta_i(k)$. In this process, the

traceback information

$$(\hat{l}_i,\,\hat{j}_i) = \underset{\substack{l=0,\dots,k-1\\ j\in\mathcal{S}}}{\operatorname{argmax}} \left\{ \begin{array}{l} \delta_j(k-l+1)\alpha_{ij}\,p(l|i) \\ \quad \cdot P(d(k-l+1)\dots d(k)|i) \\ \quad \cdot b_{i,l}^{(x)}(\mathbf{x}(k-l+1)\dots\mathbf{x}(k)) \end{array} \right\} \quad (38)$$

and $\hat{k}_{0,i} = k - \hat{l}_i + 1$ is stored in $\Psi_i(k) = (\hat{k}_{0,i}, \hat{l}_i, \hat{j}_i)$, which contains the optimal starting time $\hat{k}_{0,i}$ and length $\hat{l}_i$ of the current segment i as well as the label $\hat{j}_i$ of the best previous segment that led to $\delta_i(k)$.

The extended Viterbi algorithm, which is used to compute the most probable sequence of states $\hat{q}_0^{\hat{m}-1}$ from periodically obtained observations $\mathbf{o}(k) = (d(k), \mathbf{x}(k))$, is shown in Alg. 2.

Algorithm 2 Extended Viterbi Algorithm

Inputs:
(1) Hybrid process model $\lambda = (\mathcal{S}, \mathcal{T}, \Theta)$
(2) Observations $\mathbf{o}(k) = (d(k), \mathbf{x}(k))$
Output: State sequence $\hat{q}_0 \dots \hat{q}_{\hat{m}-1}$ with $\hat{q}_\gamma = (\hat{k}_{0,\gamma}, \hat{l}_\gamma, \hat{s}_\gamma)$

(01) **for** $i \in \mathcal{S}$: $\Delta_i(0) = B_{i,0}(\mathbf{x}(0)) + \Pi_i$
(02) **for** $k = 1 \dots n-1$:
(03) **for** $i \in \mathcal{S}$:
(04) $\Delta_i(k) = \max\limits_{\substack{l=l_{min}\dots l'_{max,k}\\ j\in\mathcal{S}}} \{\Delta_j(k-l+1) + A_{ij} + L_{i,l} + B_{i,l}^{(d)}(d_{k-l+1}^k) + B_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k)\}$
(05) $(\hat{l}_i, \hat{j}_i) = \operatorname{argmax}\limits_{\substack{l=l_{min}\dots l'_{max,k}\\ j\in\mathcal{S}}} \{\Delta_j(k-l+1) + A_{ij} + L_{i,l} + B_{i,l}^{(d)}(d_{k-l+1}^k) + B_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k)\}$
(06) $\Psi_i(k) = (k - \hat{l}_i + 1, \hat{l}_i, \hat{j}_i)$
(07) $\hat{s}'_1 = \operatorname{argmax}\limits_{i\in\mathcal{S}} \Delta_i(n-1)$
(08) $k = n-1,\ \gamma = 1$
(09) **while** $k > 0$:
(10) $(\hat{k}'_{0,\gamma}, \hat{l}'_\gamma, \hat{s}'_{\gamma+1}) = \Psi_{s_\gamma}(k)$
(11) $k \leftarrow \hat{k}'_{0,\gamma},\ \gamma \leftarrow \gamma + 1$
(12) $\hat{m} = \gamma,\ \hat{k}'_{0,\hat{m}-1} = 0,\ \hat{l}'_{\hat{m}-1} = \hat{k}'_{0,\hat{m}-2} + 1$
(13) **for** $\gamma \in 0 \dots \hat{m}-1$:
(14) $(\hat{k}_{0,\gamma}, \hat{l}_\gamma, \hat{s}_\gamma) = (\hat{k}'_{0,\hat{m}-\gamma}, \hat{l}'_{\hat{m}-\gamma}, \hat{s}'_{\hat{m}-\gamma})$

For numerical reasons, the following variables are used in the implementation of the algorithm:

$$\begin{aligned} \Delta_i(k) &= \log\,\delta_i(k), \quad \Pi_i = \log\,\pi_i, \quad A_{ij} = \log\,\alpha_{ij} \\ L_{i,l} &= \log\,p(l|i) = -\frac{1}{2}\log(2\pi\sigma_{l,i}) - \frac{1}{2}\left(\frac{l-\mu_{l,i}}{\sigma_{l,i}}\right)^2 \\ B_{i,l}^{(d)}(d_{k-l+1}^k) &= \log\,b_{i,l}^{(d)}(d_{k-l+1}^k) = \log\,P(d_{k-l+1}^k|i) \\ B_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k) &= \log\,b_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k) \\ &= -\frac{1}{2}\sum_{\kappa=k-l+1}^{k} \big(\log\big((2\pi)^{n_x}\det(\boldsymbol{\Sigma}_i)\big) \\ &\quad + \boldsymbol{\Delta}_{x,i}^T(k-l+1,\kappa)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_{x,i}(k-l+1,\kappa)\big) \end{aligned} \quad (39)$$

with n_x denoting the dimensionality of $\mathbf{x}(k)$. $\boldsymbol{\Delta}_{x,i}(k)$ has been introduced in (32). The forward variable $\Delta_i(k)$ is initialized in line (01) according to eq. (37). In lines (02)-(06), the forward variable $\Delta_i(k)$ and the trace-back information $\Psi_i(k)$ are computed for all values of $k = 1, \dots, n-1$ and $i = 0, \dots, |\mathcal{S}| - 1$ using eqs. (37) and (38). It is worth noting that the values of $B_{i,l-\delta}(\mathbf{x}_{k_1-l+\delta+1}^{k_1})$ are a subset of the values of $B_{i,l}(\mathbf{x}_{k_2-l+1}^{k_2})$, for two time instances k_1 and $k_2 = k_1 + \delta$ with an arbitrary delay of δ. For this reason, the computation time can be significantly reduced by storing the values of $B_{i,l}(x_{k-l+1})$ in a lookup-table. Further speedup is achieved by limiting the segment length to a range of $[l_{min}, l_{max}]$. In lines (04) and (05), the upper bound $l'_{max,k} = \min(l_{max}, k+1)$ is used to avoid negative indexes. In lines (07)-(14), eq. (36) and the trace-back information computed according to (38) are used to determine the most probable state sequence $\hat{q}_0^{\hat{m}-1}$. In doing so, the auxiliary states $q'_{0,\gamma} = q_{0,m-\gamma}$ with reversed indexes are computed first, because $\hat{m}$ is not known before the trace-back is finished.

VI. Fault Detection

The proposed fault detection method is based on evaluating the conditional probability density function

$$p_\lambda(\mathbf{o}_0^{n-1}|q_0^{n-1}) = P_\lambda(\mathbf{d}_0^{n-1}|q_0^{n-1})\,p_\lambda(\mathbf{x}_0^{n-1}|q_0^{n-1}) \quad (40)$$

of the measurements $\mathbf{o}_0^{n-1}$ regarding the optimal state sequence q_0^{n-1}, which is determined using the extended Viterbi algorithm introduced in section V-B. To detect unlikely discrete events, the conditional probability

$$P_\lambda(d(k)|q(k)) = P_\lambda(d(k)|s(k)) < T_d \quad (41)$$

of the measured discrete events $d(k)$ is compared to a given threshold T_d. Fault detection in the continuous measurements $\mathbf{x}(k)$ is more complex. For this issue, the CUSUM algorithm [43] for stationary Gaussian processes, which is commonly used in statistical quality control, has been extended for hybrid process models.

The CUSUM algorithm is based on the distinction between the hypotheses $\mathcal{H}$ of error-free process behavior and the hypothesis $\tilde{\mathcal{H}}$ of faulty process behavior. In hypothesis $\tilde{\mathcal{H}}$, the faulty process behavior is assumed to start at time instance k_f. To identify the correct hypothesis, the ratio

$$\Delta_{\mathcal{L}}(k, k_f) = \mathcal{L}_{\tilde{\mathcal{H}}}(k, k_f) - \mathcal{L}_{\mathcal{H}}(k) \quad (42)$$

of the log likelihoods

$$\mathcal{L}_{\mathcal{H}}(k) = \log\,p_\lambda(\mathbf{x}_0^k|q_0^k) \quad (43)$$

and

$$\mathcal{L}_{\tilde{\mathcal{H}}}(k) = \log\,p_\lambda(\mathbf{x}_0^{k_f-1}|q_0^{k_f-1}) + \log\,p_{\tilde{\lambda}}(\mathbf{x}_{k_f}^k|q_{k_f}^k) \quad (44)$$

is evaluated.

In the extended CUSUM algorithm, $\mathcal{L}_{\mathcal{H}}(k)$ is computed for the hybrid process model λ. The process behavior according to hypothesis $\tilde{\mathcal{H}}$ is described by means of a modified model $\tilde{\lambda}$, which simulates the faulty process behavior from time k_f. A fault is detected for

$$\Delta_{\mathcal{L}}(k, k_f) > h \quad (45)$$

where h is a threshold set by the user. Inserting (43) and (44) into (42) leads to the log likelihood ratio

$$\begin{aligned} \Delta_{\mathcal{L}}(k, k_f) &= \log\,p_{\tilde{\lambda}}(\mathbf{x}_{k_f}^k|q_{k_f}^k) - \log\,p_\lambda(\mathbf{x}_{k_f}^k|q_{k_f}^k) \\ &= \sum_{\kappa=k_f}^{k} \delta_{\mathcal{L}}(k|q(k)) \end{aligned} \quad (46)$$

with the instantaneous log likelihood ratio

$$\delta_{\mathcal{L}}(k|q(k)) = \log\ p_{\tilde{\lambda}}(\mathbf{x}(k)|q(k)) - \log\ p_{\lambda}(\mathbf{x}(k)|q(k)). \quad (47)$$

Under assumption of a systematic fault of the form

$$\tilde{\mathbf{f}}_i(k-k_0) = \mathbf{f}_i(k-k_0) + \boldsymbol{\delta}_x, \quad (48)$$

which affects the continuous process behavior $\mathbf{f}_i(k-k_0)$ in each state $q(k) = (k_0(k), l(k), s(k) = i)$ in the same way, the following relationship results (see appendix II):

$$\delta_{\mathcal{L}}(k|q(k)) = \boldsymbol{\delta}_x^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\right) \quad (49)$$

It is worth noting that positive and negative values of different components of $\boldsymbol{\delta}_x$ cancel each other out with respect to the computation of $\delta_{\mathcal{L}}(k|q(k))$. For this reason, analogous to the two-sided version of the standard CUSUM algorithm a separate detection is performed for each of the fault vectors

$$\begin{aligned}\boldsymbol{\delta}_{x,c}^{+} &= (0\ldots0, |\delta_{x,c}|, 0\ldots0)\\ \boldsymbol{\delta}_{x,c}^{-} &= (0\ldots0, -|\delta_{x,c}|, 0\ldots0)\end{aligned} \quad (50)$$

where $|\delta_{x,c}|$ denotes the absolute value of the fault magnitude for component x_c, $c \in \{1\ldots n_c\}$. Since in (46) both q_0^k and k_f are unknown, they have to be replaced by their maximum likelihood estimates. Maximizing (46) with respect to k_f leads to the following recursion for $\Delta_{\mathcal{L}_c}^{+}(k)$ and $\Delta_{\mathcal{L}_c}^{-}(k)$:

$$\begin{aligned}\Delta_{\mathcal{L}_c}^{\pm}(k) &= \max_{1\le k_f\le k} \Delta_{\mathcal{L}_c}^{\pm}(k,k_f)\\ &= \max\left\{\max_{1\le k_f\le k-1} \Delta_{\mathcal{L}_c}^{\pm}(k,k_f),\ \Delta_{\mathcal{L}_c}^{\pm}(k,k)\right\}\\ &= \max\left\{\max_{1\le k_f\le k-1} \sum_{\kappa=k_f}^{k} \delta_{\mathcal{L}_c}^{\pm}(\kappa|q(\kappa)),\ \delta_{\mathcal{L}_c}^{\pm}(k|q(k))\right\}\\ &= \max\left\{\Delta_{\mathcal{L}_c}^{\pm}(k-1) + \delta_{\mathcal{L}_c}^{\pm}(k|q(k)),\ \delta_{\mathcal{L}_c}^{\pm}(k|q(k))\right\}\\ &= \max\left\{\Delta_{\mathcal{L}_c}^{\pm}(k-1),\ 0\right\} + \delta_{\mathcal{L}_c}^{\pm}(k|q(k))\\ &= \left\{\Delta_{\mathcal{L}_c}^{\pm}(k-1)\right\}^{+} + \delta_{\mathcal{L}_c}^{\pm}(k|q(k))\end{aligned} \quad (51)$$

where $\delta_{\mathcal{L}_c}^{\pm}$ is computed for each fault vector $\boldsymbol{\delta}_{x,c}^{\pm}$ defined in (50) using (49). The maximum likelihood estimate $\hat{q}_0^k$ of the state sequence q_0^k is computed with the extended Viterbi algorithm introduced in section V-B. The incremental update of $\Delta_{\mathcal{L}_c}^{\pm}(k)$ according to (51) is integrated in the forward iteration of the Viterbi algorithm. In this process, it has to be considered that the estimate $\hat{q}_0^k$ is not yet final at time instance k, $k < n$, of the forward iteration due to future observations. For this reason, $\Delta_{\mathcal{L}_{i,c}^{\pm}}(k)$ is updated for each possible system mode i rather than recomputing the recursion (51) at each time instance k for a complete state sequence $\hat{q}_0^k$.

This procedure is detailed in the overall fault detection algorithm for hybrid processes (Alg. 3). The extended CUSUM algorithm processes continuous observations $\mathbf{x}(k)$ and discrete observations $d(k)$ at time instances k, $k = 0\ \ldots\ n-1$. Initialization of the forward variables and the CUSUM variables is conducted in line (01). In lines (02)-(15), these variables are updated for the time instances $k = 1\ \ldots\ n-1$ simultaneously with the actual fault detection. The forward variable $\Delta_i(k)$ is updated in line (04) using eq. (37).

Algorithm 3 CUSUM Algorithm for the SHPM

Inputs:
(1) Hybrid process model $\lambda = (S, T, \Theta)$
(obtained with Algorithm 2 from training data)
(2) Observations $\mathbf{o}(k) = (d(k), \mathbf{x}(k))$, $k = 0, \ldots, n-1$
Output: Detected faults f_d, f_x (if existing, else "OK")
(01) **for** $i \in \mathcal{S}$: $\Delta_{\mathcal{L}_{i,c}}^{+} = 0$, $\Delta_{\mathcal{L}_{i,c}}^{-} = 0$, $\Delta_i(0) = B_{i,0}(\mathbf{o}(0)) + \Pi_i$
(02) **for** $k = 1, \ldots n-1$:
(03) **for** $i \in \mathcal{S}$:
(04) $\Delta_i(k) = \max_{l=l_{min}\ldots l'_{max,k},\ j\in\mathcal{S}} \{\Delta_j(k-l+1) + A_{ij} + L_{i,l} + B_{i,l}^{(d)}(d_{k-l+1}^k) + B_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k)\}$
(05) $(\hat{l}_i, \hat{j}_i) = \operatorname{argmax}_{l=l_{min}\ldots l'_{max,k},\ j\in\mathcal{S}} \{\Delta_j(k-l+1) + A_{ij} + L_{i,l} + B_{i,l}^{(d)}(d_{k-l+1}^k) + B_{i,l}^{(x)}(\mathbf{x}_{k-l+1}^k)\}$
(06) $\hat{k}_{0,i} = k - \hat{l}_i + 1$, $\tilde{\Delta}_{\mathcal{L}_{i,c}}^{\pm}(\hat{k}_{0,i}) = \Delta_{\mathcal{L}_{\hat{j}_i,c}}^{\pm}(\hat{k}_{0,i} - 1)$
(07) **for** $\kappa = \hat{k}_{0,i} \ldots k$:
(08) $\tilde{\Delta}_{\mathcal{L}_{i,c}}^{\pm}(\kappa) = \left\{\tilde{\Delta}_{\mathcal{L}_{i,c}^{\pm}}(\kappa-1)\right\}^{+} + \boldsymbol{\delta}_{x,c}^{\pm\,T} \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}(\kappa) - \boldsymbol{f}_i(\kappa - \hat{k}_{0,i}) - \frac{\boldsymbol{\delta}_{x,c}^{\pm}}{2}\right)$
(09) $\Delta_{\mathcal{L}_{i,c}}^{\pm}(k) = \tilde{\Delta}_{\mathcal{L}_{i,c}}^{\pm}(k)$
(10) $\hat{s}(k) = \operatorname{argmax}_{i\in\mathcal{S}}\ \Delta_i(k)$
(11) **if** $P_\lambda(d(k)|\hat{s}(k)) < T_d$:
(12) Report fault f_d.
(13) **for** $c = 1 \ldots n_x$:
(14) **if** $\Delta_{\mathcal{L}_{\hat{s}(k),c}}^{+}(k) > h$ **or** $\Delta_{\mathcal{L}_{\hat{s}(k),c}}^{-}(k) > h$:
(15) Report fault f_x.

In line (05), the length $\hat{l}_i$ and the predecessor $\hat{j}_i$ of the current segment on the optimal path are estimated using eq. (38). Based on the estimates $\hat{l}_i$ and $\hat{j}_i$, the CUSUM variables $\Delta_{\mathcal{L}_{i,c}^{\pm}}(k)$ are computed from $\Delta_{\mathcal{L}_{\hat{j}_i,c}^{\pm}}(k - \hat{l}_i)$ by repeated application of recursion (51) using the auxiliary variables $\tilde{\Delta}_{\mathcal{L}_{i,c}^{\pm}}(\kappa)$ (lines (06)-(09)). In line (10), the most likely system mode at time instance k is determined using eq. (36). In lines (11)-(12), eq. (41) is evaluated to detect anomalies f_d in the discrete measurements. Finally, the log likelihood ratios $\Delta_{\mathcal{L}_{\hat{s}(k),c}^{\pm}}$ of the continuous measurements are assessed in lines (13)-(15).

VII. Evaluation

The proposed fault detection method has been evaluated on the test database introduced in [44], which is based on the Pensim benchmark model of [45]. The Pensim benchmark model describes the growth of biomass and the production of penicillin in a fed-batch reactor. This is an industrial batch process with hidden system modes.

Initially, the fermentation is operated in batch mode at high substrate concentrations to stimulate biomass growth. Once the initial substrate is nearly exhausted, the process switches to fed-batch mode with a nominal feed rate of 0.06 L/h to maintain a low but non-zero substrate concentration. Under these stressful conditions, penicillin is produced by the biomass. The reactor is stirred and aerated to provide the biomass with oxygen during the entire operation. Temperature and pH are controlled via PID loops. Feed rate, feed temperature, aeration rate, agitator power, and cold and hot water temperatures are controlled in open loop. A batch is terminated after a total of 25 L of substrate has been added, for a total batch

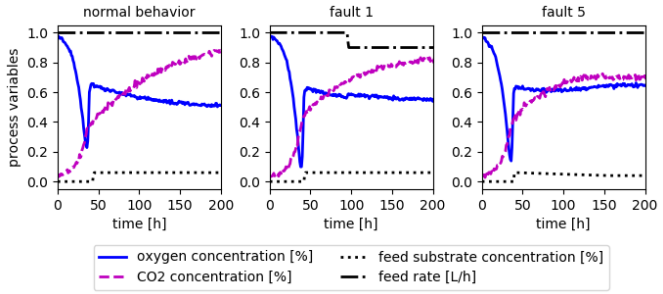

Fig. 4. Process variables for the cases of normal behavior, an abrupt signal change (fault 1) and a gradual drift (fault 5).

duration of approximately 460 h. Detailed descriptions of the mathematical model, its parameter values, and the process installation are presented in [45]. During creation of the test data base, Gaussian measurement noise has been added to the measured variables with standard deviations as listed in [44]. Initial conditions have been randomly determined to introduce additional batch-to-batch variability. All sensors have been sampled every 0.2 h. Subsequently, data reduction has been accomplished by replacing the data within non-overlapping windows of length 5 by the respective mean values, leading to an effective sampling interval of 1 h. The following set of process variables is used for model learning and fault detection: time, dissolved oxygen concentration, dissolved CO_2 concentration, reactor temperature, pH, feed rate, feed temperature, agitator power, cooling water flow rate, cumulative base flow rate, cumulative acid flow rate. By changing the properties of the initial conditions and/or model parameters, four subsets have been generated, each containing 400 batches with normal operation and several batches with faults. The faulty batches each contain one of 15 fault types with the same frequency. Nine fault types are abrupt changes of individual process variables due to sensor failures or changes in the operation, while the other six error types are gradual drifts. A detailed description of the faults is given in [44].

Fig. 4 shows the example course of four process variables for the cases of normal process behavior, an abrupt signal change (fault 1) and a gradual drift (fault 5). The *feed rate* and the *feed substrate concentration* are not directly measurable process variables, which cause the deviation from normal behavior in the two fault cases. As can be seen from the figure, fault 1 is caused by an abrupt change in the *feed substrate concentration*, fault 5 by a gradual deviation of the *feed rate* from the normal behavior. The *dissolved CO2 concentration* and the *oxygen concentration* (solid line) are examples of the measured process signals, which are the basis for model learning and fault detection.

A hybrid process model is learnt from the initial N batches of normal operation using the approach described in section V with T iterations of the extended Viterbi algorithm. 10 additional error-free and 10 faulty batches of each subset are used to optimize the fault detection parameters, i.e. the parameters h and $\delta_{x,c}$ of the CUSUM algorithm. Fault detection was carried out on another set of 150 error-free batches and 150 faulty batches.

TABLE II
EVALUATION RESULTS FOR THE PENICILLIN PRODUCTION

	Δ_{acc}	t_{train}	t_{eval}
HMM	0.777	435s	0.005s
SHPM	0.823	1740s	0.026s

The results of a systematic evaluation and a sensitivity analysis of the developed fault detection method are presented in the next two subsections.

A. Systematic Evaluation

A systematic evaluation of the developed fault detection approach was conducted with parameters $h = 300$, $\delta_{x,c} = 3.5$, $N = 20$ and $T = 3$. Fault detection was carried out on 150 batches of normal operation and 150 batches with faults. For comparison, the HMM-based fault detection were evaluated in addition to the developed SHPM.

The averaged fault detection results and runtimes on the test data sets are shown in table II.

The table shows that the proposed fault detection system (SHPM) achieves a significantly higher balanced accuracy than the baseline method. For a more detailed analysis of the results, the sensitivity $\Delta_{sens} = \frac{TP}{TP+FN}$ of the SHPM for each of the 15 fault types has been determined (Fig. 5 d). As can be seen from the figure, in particular fault types 9, 10, 13 and 14 impair the fault detection results, while the sensitivity for the other fault types is very high. The small sensitivity for the critical fault types appears to be due to the fact that the four fault types did not directly affect the process variables but the control of the pH value and the reactor temperature. This evidently led to relatively small deviations of the measured signals from the normal behavior, which are not obvious even on visual inspection. To improve the detection of the four critical fault types, further research could aim at exploiting additional correlations between the different sensor signals. Excluding the four critical faults types from the test data set results in a balanced accuracy of $\Delta_{acc} = 0.93$ for $N = 20$ training batches and a balanced accuracy of $\Delta_{acc} = 0.96$ for $N = 100$ training batches.

Furthermore, cross validation was carried out to analyze the impact of different data splits on the detection results. In each split, the 170 data sets, on which the results in table II are based, were randomly partitioned into 20 training sets and 150 test sets. The evaluation of the SHMP for 20 randomly chosen splits with the parameters given at the beginning of section VII-A resulted in the mean balanced accuracy $\bar{\Delta}_{acc} = 0.82$ with a standard deviation of 0.015. All in all, in the investigated use case, the impact of the data partitioning on the balanced accuracy is relatively small compared to the impact of the parameters analyzed in section VII-B.

The high runtimes t_{train} for the training of the SHPM are acceptable in the investigated application scenario, since the training only has to be carried out once during the commissioning of the fault detection system. With respect to the processing time t_{eval}, which is required to evaluate a new

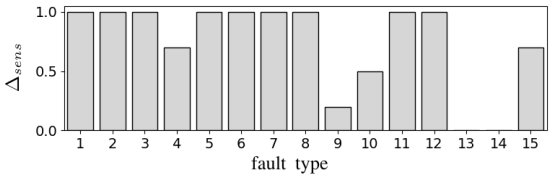

Fig. 5. Sensitivity Δ_{sens} of the SHPM for the different fault types.

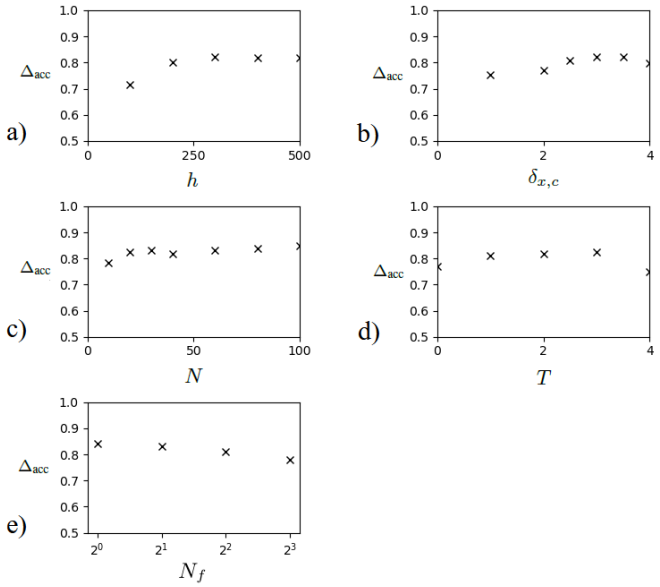

Fig. 6. Sensitivity analysis for the parameters h, $\delta_{x,y}$, N, T and N_f.

observation during the operating phase of the fault detection system, there are no practical disadvantages of the SHPM, since penicilin production is very slow with a cycle time of about 460 h, and for many errors a time delay in correction is therefore tolerable. But even if rapid troubleshooting is required, the time for this is generally not significantly increased by the algorithm's processing time, which is in the order of $t_{eval} = 0.026s$ per observation.

B. Sensitivity Analysis

The sensitivity analysis was conducted using the default values for h, $\delta_{x,c}$, N and T, which are specified in section VII-A. Furthermore, the influence of outliers in the training data on the fault detection results was investigated. The analysis results are shown in Fig. 6. As in section VII-A, fault detection was carried out on 150 faulty and 150 error-free process cycles.

The balanced accuracy is stable over a wide range of the parameter h for $h > 200$ as shown in Fig. 6 a). Fig. 6 b) shows that the maximum of the balanced accuracy under variation of the $\delta_{x,c}$ parameter is achieved at $\delta_{x,c} \approx 3.5$. Furthermore, as expected, a tendential increase of the balanced accuracy with the number of training cycles N can be observed (Fig. 6). For $N = 20$ the balanced accuracy is already relatively high ($\Delta_{acc} = 0.823$), but still below the value $\Delta_{acc} = 0.847$, which is reached for $N = 100$. Furthermore, a minor increase of the balanced accuracy with the number of training data can be observed as shown in Fig. 6 d). For $T > 3$, however, it decreases rapidly again, which could be due to an over-adaptation of the model to the training data, a known problem of expectation-maximization approaches.

The impact of outliers in the training data on the balanced accuracy is shown in Fig. 6 e) for $N = 50$ training batches. In each of the experimental runs, N_f faulty training batches were randomly selected and added to the error-free training data. This procedure was repeated 10 times for each of the values $N_f = 1, 2, 4, 8$. Subsequently, the false positives, false negatives, true positives and true negatives were added up for each value of N_f and the resulting sums were used to calculate the balanced accuracies shown in Fig. 6 e). The balanced accuracy decreases with the number of faulty training batches N_f. For $N_f = 8$ faulty training batches, the balanced accuracy is reduced to a value of 0.783 compared to 0.843 for $N_f = 0$. However, it can be seen from the figure that the fault detection results are not significantly affected for an outlier fraction less than 5% in the training batches.

VIII. Conclusion

This study investigated a novel fault detection approach for industrial batch processes, which consist of several process phases in which the process variables change over time. Fault detection is based on a stochastic hybrid process model (SHPM), which describes the process behavior in the different process phases with time-dependent linear models. In contrast to Hidden Markov Models (HMMs) with stationary, Gaussian-distributed emission distributions, which have been extended in the SHPM, this approach allows to model time-dependent changes of process variables within a process phase, leading to a significant reduction in the number of hidden states. Furthermore, discrete observations as well as an explicit probabilistic model for the length of process phases are considered in the SHPM.

An expectation-maximization (EM) algorithm is used to learn the SHPM from training data. Iteration is carried out between the segmentation of the input data and the estimation of the model parameters. To determine an initial segmentation as a starting point for the EM algorithm, a new heuristic was developed that takes into account both discrete system modes and continuous process phases. To analyze the impact of the EM algorithm on the fault detection results, fault detection was evaluated for models that resulted from a different number of iterations of the EM algorithm. With model parameters calculated directly from the initial segmentation, a balanced accuracy of 0.77 was achieved for $N = 20$ training batches. Three iterations of the EM algorithm resulted in an improved balanced accuracy of 0.82.

To assess the likelihood of faults during the operating phase of the fault detection system, a new CUSUM algorithm for the SHPM was introduced. In this approach, the SHPM is employed to simultaneously determine the process phases and a cumulative log-likelihood ratio of the observations, on the basis of which fault detection is performed. Experimental results showed significant improvements for the proposed fault detection method compared to an existing method where the continuous process behavior is modeled with Gaussian HMMs. On average, the balanced accuracy in the investigated

application scenario with 15 fault types was increased from 0.78 to 0.82, whereby four fault types related to manipulations of the pH and temperature control were only detected in rare cases. Excluding these fault types from the evaluations led to a balanced accuracy of 0.93. The evaluations further showed that the results depend on the number of training data. For $N = 100$, fault detection with the SHPM resulted in increased balanced accuracies of 0.85 and 0.96 with and without the four critical fault types, respectively.

Furthermore, the influence of the two parameters h and $\delta_{x,c}$ as well as of outliers in the training data was investigated. The evaluations have shown that fault detection results are stable over a relatively wide range of the parameters h and δ. The balanced accuracy was observed to decrease with the number of outliers in the training data. However, the experimental results show that the method is able to cope with a small number of faulty training batches.

Appendix I
Details on the Parameter Estimation

To determine the parameters $\lambda^{(t)}$, $Q(\lambda|\lambda^{(t)})$ has to be maximized according to (21). Using the Viterbi approximation (22) in (21) leads to the optimization problem

$$\lambda^{(t+1)} = \underset{\lambda}{\operatorname{argmax}} \log\ p_\lambda(\mathbf{o}_0^{n-1}, \hat{q}_0^{m-1}) \tag{52}$$

Thus, the log likelihood

$$\begin{aligned}\mathcal{L}(\lambda) &= \log\ p_\lambda(\mathbf{o}_0^{n-1}, q_0^{m-1})\\ &= \log\ p_\lambda(\mathbf{o}_0^{n-1}, k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-1})\\ &= \log\ p_\lambda(\mathbf{o}_0^{n-1}|k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-1})\\ &\quad + \log\ p_\lambda(l_0^{m-1}|s_0^{m-1}) + \log\ p_\lambda(s_0^{m-1})\\ &= \sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1} \log\ p_\lambda(\mathbf{o}(k)|k_{0,\gamma}, l_\gamma, s_\gamma)\\ &\quad + \sum_{\gamma\in\Gamma}\log\ p_\lambda(l_\gamma|s_\gamma) + \sum_{\gamma\in\Gamma}\log\ p_\lambda(s_\gamma|s_{\gamma-1})\end{aligned} \tag{53}$$

has to be maximized for the most probable state sequence $\hat{q}_0^{m-1}$. The maximization is achieved by computing the partial derivations of $\mathcal{L}(\lambda)$ with respect to the components of λ. It is worth noting that the components π_i, $\alpha_{i,j}$, $\mu_{l,i}$ and $\sigma_{l,i}$ dependent only on the log likelihoods $\log\ p_\lambda(l_0^{m-1}|s_0^{m-1})$ and $\log\ p_\lambda(s_0^{m-1})$ but not on $\log\ p_\lambda(\mathbf{o}_0^{n-1}|k_{0,0}^{m-1}, l_0^{m-1}, s_0^{m-1})$:

$$\frac{\partial}{\partial\pi_i}\mathcal{L}(\lambda) = \frac{\partial}{\partial\pi_i}\log\ p_\lambda(s_0^{m-1}) \tag{54}$$

$$\frac{\partial}{\partial\alpha_{i,j}}\mathcal{L}(\lambda) = \frac{\partial}{\partial\alpha_{i,j}}\log\ p_\lambda(s_0^{m-1}) \tag{55}$$

$$\frac{\partial}{\partial\mu_{l,i}}\mathcal{L}(\lambda) = \frac{\partial}{\partial\mu_{l,i}}\log\ p_\lambda(l_0^{m-1}|s_0^{m-1}) \tag{56}$$

$$\frac{\partial}{\partial\sigma_{l,i}}\mathcal{L}(\lambda) = \frac{\partial}{\partial\sigma_{l,i}}\log\ p_\lambda(l_0^{m-1}|s_0^{m-1}) \tag{57}$$

The partial model $\log\ p_\lambda(l_0^{m-1}|s_0^{m-1}) + \log\ p_\lambda(s_0^{m-1})$ formally corresponds to a HMM with a scalar Gaussian output density, which is e.g. used in [6]. For this reason, the determination equations (25)-(27) for these parameters can be adopted from [6].

To determine $\beta_{i,\nu} = P(d(k)|i)$, the equation

$$\begin{aligned}&\frac{\partial}{\partial\beta_{i,\nu}}\mathcal{L}(\lambda)\\ &= \frac{\partial}{\partial\beta_{i,\nu}}\log\ p_\lambda(\mathbf{o}_0^{n-1}, q_0^{m-1})\\ &= \frac{\partial}{\partial\beta_{i,\nu}}\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\log\ p_\lambda(d(k)|s_\gamma)\\ &= \frac{\partial}{\partial\beta_{i,\nu}}\left(\sum_{j\in\mathcal{S}}\sum_{\nu\in\mathcal{V}}\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - j, d(k)-\nu)\log\ \beta_{j,\nu}\right)\end{aligned} \tag{58}$$

has to be evaluated with respect to the constraints

$$\sum_{\tilde{\nu}\in\mathcal{V}}\beta_{i,\tilde{\nu}} = 1 \quad\text{and}\quad \beta_{i,\nu}\geq 0 \quad \text{for } i\in\mathcal{S},\quad \nu\in\mathcal{V} \tag{59}$$

Lagrange multipliers $\boldsymbol{\zeta} = \{\zeta_i\}$ are used to assert the equality constraints in (59), while the inequality constraints can be safely ignored, since the optimal solution implicitly produces positive values for $\beta_{i,\nu}$:

$$\begin{aligned}&\frac{\partial}{\partial\beta_{i,\nu}}\mathcal{L}(\lambda,\boldsymbol{\zeta})\\ &= \frac{\partial}{\partial\beta_{i,\nu}}\sum_{j\in\mathcal{S}}\sum_{\nu\in\mathcal{V}}\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - j, d(k)-\nu)\log\beta_{j,\nu}\\ &\quad + \frac{\partial}{\partial\beta_{i,\nu}}\sum_{i\in\mathcal{S}}\zeta_i\left(1-\sum_{\nu\in\mathcal{S}}\beta_{i,\nu}\right)\\ &= \frac{1}{\beta_{i,\nu}}\left(\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - i, d(k)-\nu)\right) - \zeta_i \equiv 0\\ &\Rightarrow \beta_{i,\nu} = \frac{1}{\zeta_i}\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - i, d(k)-\nu)\end{aligned} \tag{60}$$

Substituting back in and setting the partial with respect to $\boldsymbol{\zeta}$ equal to zero:

$$\begin{aligned}\frac{\partial}{\partial\zeta_i}\mathcal{L}(\lambda,\boldsymbol{\zeta}) &= 1 - \sum_{\nu\in\mathcal{V}}\frac{1}{\zeta_i}\sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - i, d(k)-\nu)\\ \Rightarrow \zeta_i &= \sum_{\gamma\in\Gamma}\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\delta(s_\gamma - i) = \sum_{\gamma\in\Gamma}\delta(s_\gamma - i)l_\gamma\end{aligned} \tag{61}$$

Inserting from (61) to (60) results in (28).

The parameters $\mathbf{c}_i$ and $\mathbf{m}_i$ result from the partial derivatives $\frac{\partial}{\partial\mathbf{c}_i}\mathcal{L}(\lambda)$ and $\frac{\partial}{\partial\mathbf{m}_i}\mathcal{L}(\lambda)$:

$$\begin{aligned}\frac{\partial}{\partial\mathbf{c}_i}\mathcal{L}(\lambda) &= \frac{\partial}{\partial\mathbf{c}_i}\sum_{\gamma\in\Gamma}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\Big(\log((2\pi)^{n_x}\det(\boldsymbol{\Sigma}_i))\\ &\quad + \big(\mathbf{x}(k) - \mathbf{c}_i - \mathbf{m}_i\big(k-k_{0,\gamma}\big)\big)^T\\ &\quad \times \boldsymbol{\Sigma}_i^{-1}\big(\mathbf{x}(k) - \mathbf{c}_i - \mathbf{m}_i\big(k-k_{0,\gamma}\big)\big)\Big)\\ &= \sum_{\gamma\in\Gamma}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\boldsymbol{\Sigma}_i^{-1}\Delta_i(k_{0,\gamma}, k) \equiv 0\end{aligned} \tag{62}$$

and

$$\frac{\partial}{\partial \mathbf{m}_i}\mathcal{L}(\lambda) = \sum_{\gamma\in\Gamma}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}(k-k_{0,\gamma})\boldsymbol{\Sigma}_i^{-1}\Delta_i(k_{0,\gamma},k) \equiv 0 \tag{63}$$

with $\Delta_i(k_{0,\gamma},k) = \mathbf{x}(k) - \mathbf{c}_i - \mathbf{m}_i\big(k-k_{0,\gamma}\big)$. Solving the equations (62) and (63) with respect to $\mathbf{c}_i$ and $\mathbf{m}_i$ leads to the equations (29) and (30).

For the estimation of $\boldsymbol{\Sigma}_i$, a similar procedure as for a multivariate Gaussian distribution with a constant expected value is possible (see e.g. [46]). To simplify the derivation, $\mathcal{L}(\lambda)$ is not minimized with respect to $\boldsymbol{\Sigma}_i$ but maximized with respect to $\boldsymbol{\Sigma}_i^{-1}$:

$$\begin{aligned}&\frac{\partial}{\partial \boldsymbol{\Sigma}_i^{-1}}\mathcal{L}(\lambda)\\ &= \frac{\partial}{\partial \boldsymbol{\Sigma}_i^{-1}}\sum_{\gamma\in\Gamma}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\Bigg(-\frac{1}{2}\log(\det(\boldsymbol{\Sigma}_i))\\ &\quad -\frac{1}{2}\boldsymbol{\Delta}_{x,i}(k_{0,\gamma},k)^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_{x,i}(k_{0,\gamma},k)\Bigg)\\ &= \sum_{\gamma\in\Gamma}\delta(s_\gamma - i)\sum_{k=k_{0,\gamma}}^{k_{0,\gamma}+l_\gamma-1}\left(\frac{1}{2}\boldsymbol{\Sigma}_i - \frac{1}{2}\boldsymbol{\Delta}_{x,i}(k_{0,\gamma},k)\boldsymbol{\Delta}_{x,i}(k_{0,\gamma},k)^T\right)\\ &\equiv 0\end{aligned} \tag{64}$$

In (64), the relationships [46]

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_i^{-1}}\det(\boldsymbol{\Sigma}_i) = -\boldsymbol{\Sigma}_i \tag{65}$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_i^{-1}}\boldsymbol{\Delta}_{x,i}^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_{x,i} = \boldsymbol{\Delta}_{x,i})\boldsymbol{\Delta}_{x,i}^T \tag{66}$$

are exploited. Solving (64) with respect to $\boldsymbol{\Sigma}_i$ leads to (31).

Appendix II
Details on the Computation of the Instantaneous Log Likelihood Ratio

By exploiting the invariance of the trace tr of a matrix under cyclic permutations, the instantaneous ratio $\delta_{\mathcal{L}}(k|q(k))$ of the log likelihoods

$$\begin{aligned}\log\ p_{\tilde{\lambda}}(\mathbf{x}(k)|i) &= -\frac{1}{2}\log\big((2\pi)^{n_x}\det(\boldsymbol{\Sigma}_i)\big)\\ &\quad -\frac{1}{2}\tilde{\boldsymbol{\Delta}}_{x,i}^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\tilde{\boldsymbol{\Delta}}_{x,i}(k_0,k)\end{aligned} \tag{67}$$

$$\begin{aligned}\text{and}\quad \log\ p_{\lambda}(\mathbf{x}(k)|i) &= -\frac{1}{2}\log\big((2\pi)^{n_x}\det(\boldsymbol{\Sigma}_i)\big)\\ &\quad -\frac{1}{2}\boldsymbol{\Delta}_{x,i}^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_{x,i}(k_0,k)\end{aligned} \tag{68}$$

in system mode $s(k) = i$ can be written as

$$\begin{aligned}&\delta_{\mathcal{L}}(k|q(k))\\ &= \delta_{\mathcal{L}}(k|k_0,l,s(k)=i)\\ &= \log\ p_{\tilde{\lambda}}(\mathbf{x}(k)|i) - \log\ p_{\lambda}(\mathbf{x}(k)|i)\\ &= -\frac{1}{2}\Big(\tilde{\boldsymbol{\Delta}}_i^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\tilde{\boldsymbol{\Delta}}_i(k_0,k) - \boldsymbol{\Delta}_i^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_i(k_0,k)\Big)\\ &= -\frac{1}{2}\mathrm{tr}\Big(\tilde{\boldsymbol{\Delta}}_i^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\tilde{\boldsymbol{\Delta}}_i(k_0,k) - \boldsymbol{\Delta}_i^T(k_0,k)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Delta}_i(k_0,k)\Big)\\ &= -\frac{1}{2}\mathrm{tr}\Big(\Big(\tilde{\boldsymbol{\Delta}}_i(k_0,k)\tilde{\boldsymbol{\Delta}}_i^T(k_0,k) - \boldsymbol{\Delta}_i(k_0,k)\boldsymbol{\Delta}_i^T(k_0,k)\Big)\boldsymbol{\Sigma}_i^{-1}\Big)\end{aligned} \tag{69}$$

with

$$\boldsymbol{\Delta}_i(k_0,k) = \mathbf{x}(k) - \mathbf{f}_i(k-k_0), \tag{70}$$

$$\tilde{\boldsymbol{\Delta}}_i(k_0,k) = \mathbf{x}(k) - \tilde{\mathbf{f}}_i(k-k_0). \tag{71}$$

Under the assumption $\tilde{\mathbf{f}}_i(k-k_0) = \mathbf{f}_i(k-k_0) + \boldsymbol{\delta}_x$, $\delta_{\mathcal{L}}(k|q(k))$ can be further transformed as follows:

$$\begin{aligned}&\delta_{\mathcal{L}}(k|q(k))\\ &= -\frac{1}{2}\mathrm{tr}\Big(\Big(\big(\mathbf{x}(k) - \boldsymbol{f}_i(k-k_0) - \boldsymbol{\delta}_x\big)\big(\mathbf{x}(k) - \boldsymbol{f}_i(k-k_0) - \boldsymbol{\delta}_x\big)^T\\ &\quad -\big(\mathbf{x}(k) - \boldsymbol{f}_i(k-k_0)\big)\big(\mathbf{x}(k) - \boldsymbol{f}_i(k-k_0)\big)^T\Big)\boldsymbol{\Sigma}_i^{-1}\Big)\end{aligned} \tag{72}$$

$$\begin{aligned}&= \frac{1}{2}\mathrm{tr}\Bigg(\Bigg(\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg)\boldsymbol{\delta}_x^T\\ &\quad +\boldsymbol{\delta}_x\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg)^T\Bigg)\boldsymbol{\Sigma}_i^{-1}\Bigg)\end{aligned} \tag{73}$$

$$\begin{aligned}&= \frac{1}{2}\mathrm{tr}\Bigg(\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg)\boldsymbol{\delta}_x^T\boldsymbol{\Sigma}_i^{-1}\\ &\quad +\boldsymbol{\Sigma}_i^{-1}\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg)\boldsymbol{\delta}_x^T\Bigg)\end{aligned} \tag{74}$$

$$= \mathrm{tr}\Bigg(\boldsymbol{\delta}_x^T\boldsymbol{\Sigma}_i^{-1}\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg)\Bigg) \tag{75}$$

$$= \boldsymbol{\delta}_x^T\boldsymbol{\Sigma}_i^{-1}\Bigg(\mathbf{x}(k) - \mathbf{f}_i(k-k_0) - \frac{\boldsymbol{\delta}_x}{2}\Bigg). \tag{76}$$

In (74) and (75), the invariance of the trace under transposition and cyclic permutations are exploited, respectively.

References

[1] L. Christiansen, A. Fay, B. Opgenoorth, and J. Neidig, "Improved diagnosis by combining structural and process knowledge," in *Proc. IEEE 16th Conf. Emerg. Technol. Factory Automat. (ETFA)*, Toulouse, France, Oct. 2011, pp. 1–8.

[2] O. Niggemann and V. Lohweg, "On the diagnosis of cyber-physical production systems: State-of-the-art and research agenda," in *Proc. 29th Conf. Artif. Intell. (AAAI)*, Austin, TX, USA, Jan. 2015, pp. 1–8.

[3] M. A. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Context-sensitive modeling and analysis of cyber-physical manufacturing systems for anomaly detection and diagnosis," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 29–40, Jan. 2020.

[4] C. Yuan, "Unsupervised machine condition monitoring using segmental hidden Markov models," in *Proc. 24th Int. Joint Conf. Conf. Artif. Intell. (IJCAI)*, Buenos Aires, Argentina, Jul. 2015, pp. 1–8.

[5] H. Yu, "A novel semiparametric hidden Markov model for process failure mode identification," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 506–518, Apr. 2018.

[6] S. Windmann, F. Jungbluth, and O. Niggemann, "A HMM-based fault detection method for piecewise stationary industrial processes," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Automat. (ETFA)*, Luxembourg City, Luxembourg, Sep. 2015, pp. 1–6.

[7] R. D. Telford, S. Galloway, B. Stephen, and I. Elders, "Diagnosis of series DC arc faults—A machine learning approach," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1598–1609, Aug. 2017.

[8] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 5, pp. 1060–1089, May 2013.

[9] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault tolerant techniques," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3774, Jun. 2015.

[10] Z. Ren, W. Zhang, and Z. Zhang, "A deep nonnegative matrix factorization approach via autoencoder for nonlinear fault detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5042–5052, Aug. 2020.

[11] Q. Jiang, S. Yan, X. Yan, H. Yi, and F. Gao, "Data-driven two-dimensional deep correlated representation learning for nonlinear batch process monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2839–2848, Apr. 2020.
[12] R. T. Samuel and Y. Cao, "Nonlinear process fault detection and identification using kernel PCA and kernel density estimation," *Syst. Sci. Control Eng.*, vol. 4, no. 1, pp. 165–174, Jun. 2016.
[13] M. Mansouri, M. Nounou, H. Nounou, and N. Karim, "Kernel PCA-based GLRT for nonlinear fault detection of chemical processes," *J. Loss Prevention Process Industries*, vol. 40, pp. 334–347, Mar. 2016.
[14] C. Botre, M. Mansouri, M. Nounou, H. Nounou, and M. N. Karim, "Kernel PLS-based GLRT method for fault detection of chemical processes," *J. Loss Prevention Process Industries*, vol. 43, pp. 212–224, Sep. 2016.
[15] J. Fan and Y. Wang, "Fault detection and diagnosis of non-linear non-Gaussian dynamic processes using kernel dynamic independent component analysis," *Inf. Sci.*, vol. 259, pp. 369–379, Sep. 2014.
[16] Q. Jiang and X. Yan, "Learning deep correlated representations for nonlinear process monitoring," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6200–6209, Dec. 2019.
[17] Y. Wang, F. Sun, and B. Li, "Multiscale neighborhood normalization-based multiple dynamic PCA monitoring method for batch processes with frequent operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1053–1064, Jul. 2018.
[18] Q. Jiang and X. Yan, "Multimode process monitoring using variational Bayesian inference and canonical correlation analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1814–1824, Oct. 2019.
[19] R. Isermann, "Model-based fault detection and diagnosis—Status and applications," *Annu. Rev. Control*, vol. 29, pp. 71–85, 2005, doi: 10.1016/j.arcontrol.2004.12.002.
[20] O. Niggemann, B. Stein, A. Vodenčarević, A. Maier, and H. K. Buning, "Learning behavior models for hybrid timed systems," in *Proc. 26th Conf. Artif. Intell. AAAI*, Jul. 2012.
[21] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-time manufacturing machine and system performance monitoring using Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1735–1747, Feb. 2018.
[22] X. Dai, Z. Gao, T. Breikin, and H. Wang, "Disturbance attenuation in fault detection of gas turbine engines: A discrete robust observer design," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 2, pp. 234–239, Mar. 2009.
[23] A. Shui, W. Chen, P. Zhang, S. Hu, and X. Huang, "Review of fault diagnosis in control systems," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2009, pp. 5324–5329.
[24] S. Helm, M. Kozek, and S. Jakubek, "Combustion torque estimation and misfire detection for calibration of combustion engines by parametric Kalman filtering," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4326–4337, Nov. 2012.
[25] P. Lall, R. Lowe, and K. Goebel, "Prognostics health management of electronic systems under mechanical shock and vibration using Kalman filter models and metrics," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4301–4314, Nov. 2012.
[26] G. H. B. Foo, X. Zhang, and D. M. Vilathgamuwa, "A sensor fault detection and isolation method in interior permanent-magnet synchronous motor drives based on an extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3485–3495, Aug. 2013.
[27] M. Wang and R. Dearden, "Detecting and learning unknown fault states in hybrid diagnosis," in *Proc. 20th Int. Workshop Princ. Diagnosis, DX*, Stockholm, Sweden, 2009, pp. 19–26.
[28] B. Eiteneuer, N. Hranisavljevic, and O. Niggemann, "Dimensionality reduction and anomaly detection for CPPS data using autoencoder," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 1286–1292.
[29] W. Liu and I. Hwang, "Robust estimation and fault detection and isolation algorithms for stochastic linear hybrid systems with unknown fault input," *IET Control Theory Appl.*, vol. 5, no. 12, pp. 1353–1368, Aug. 2011.
[30] A. Vodenčarević, H. K. Büning, O. Niggemann, and A. Maier, "Using behavior models for anomaly detection in hybrid systems," in *Proc. 23rd Int. Symp. Inf., Commun. Automat. Technol.*, Oct. 2011, pp. 1–8.
[31] M. W. Hofbaur and B. C. Williams, "Mode estimation of probabilistic hybrid systems," in *Proc. Int. Conf. Hybrid Syst., Comput. Control*, 2002, pp. 253–266.
[32] S. Windmann, S. Jiao, O. Niggemann, and H. Borcherding, "A stochastic method for the detection of anomalous energy consumption in hybrid industrial systems," in *Proc. 11th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2013, pp. 194–199.
[33] S. Gilani, S. Windmann, F. Pethig, B. Kroll, and O. Niggemann, "The importance of model-learning for the analysis of the energy consumption of production plants," in *Proc. IEEE 18th Conf. Emerg. Technol. Factory Automat. (ETFA)*, Sep. 2013, pp. 1–8.
[34] R. C. Carrasco and J. Oncina, "Learning stochastic regular grammars by means of a state merging method," in *Grammatical Inference and Applications*. Berlin, Germany: Springer-Verlag, 1994, pp. 139–152.
[35] F. Thollard, P. Dupont, and C. de la Higuera, "Probabilistic DFA inference using Kullback–Leibler divergence and minimality," in *Proc. 17th Int. Conf. Mach. Learn.* San Mateo, CA, USA: Morgan Kaufmann, 2000, pp. 975–982.
[36] A. Maier, A. Vodenčarević, O. Niggemann, R. Just, and M. Jäger, "Anomaly detection in production plants using timed automata," in *Proc. 8th Int. Conf. Informat. Control, Automat. Robot. (ICINCO)*, Noordwijkerhout, The Netherlands, Jul. 2011, pp. 363–369.
[37] A. Maier, "Online passive learning of timed automata for cyber-physical production systems," in *Proc. 12th IEEE Int. Conf. Ind. Informat. (INDIN)*, Porto Alegre, Brazil, Jul. 2014, pp. 60–66.
[38] O. Niggemann, B. Stein, A. Vodenčarević, A. Maier, and H. K. Büning, "Learning behavior models for hybrid timed systems," in *Proc. 26th Conf. Artif. Intell. AAAI*, Toronto, ON, Canada, 2012, pp. 1083–1090.
[39] M. Ostendorf, V. V. Digalakis, and O. A. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 360–378, Sep. 1996.
[40] M. Ostendorf, V. V. Digalakis, and O. A. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 360–378, Sep. 1997.
[41] A. P. Dempster, N. M. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
[42] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang, "A segmental *k*-means training procedure for connected word recognition," *AT&T Tech. J.*, vol. 65, no. 3, pp. 21–40, 1986.
[43] P. Granjon, "The CuSum algorithm—A small review," Jun. 2013.
[44] J. Van Impe and G. Gins, "An extensive reference dataset for fault detection and identification in batch processes," *Chemometric Intell. Lab. Syst.*, vol. 148, pp. 20–31, Nov. 2015.
[45] G. Birol, C. Ündey, and A. Cinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Chemometrics Intell. Lab. Syst.*, vol. 26, pp. 1552–1565, Nov. 2002.
[46] J. Wu, "Some properties of the Gaussian distribution," 2004.

Stefan Windmann received the Dipl.-Ing. and Dipl.-Inf. degrees in electrical engineering and technical computer sciences and the Ph.D. degree in electrical engineering from Paderborn University, Germany, in 2004 and 2008, respectively. He is currently employed as a Senior Scientist at the Fraunhofer IOSB-INA, Lemgo, Germany. His current research interests include machine learning algorithms and methods for diagnosis and optimization of automated production systems.