This document contains the **post-print pdf version** of the refereed paper:

“Hybrid derivative dynamic time warping for online industrial batch-end quality estimation”

by *Geert Gins, Pieter Van den Kerkhof, and Jan F.M. Van Impe*

which has been archived on the university repository Lirias (<https://lirias.kuleuven.be/>) of the KU Leuven.

The content is identical to the content of the published paper, but without the final typesetting by the publisher.

When referring to this work, please cite the full bibliographic info:

G. Gins, P. Van den Kerkhof and J.F.M. Van Impe (2012). Hybrid derivative dynamic time warping for online industrial batch-end quality estimation, Industrial & Engineering Chemistry Research, 51(17), 6071–6084.

<http://pubs.acs.org/journal/iecred>

<http://dx.doi.org/10.1021/ie2019068>

The corresponding author can be contacted for additional info.

Conditions for open access are available at:

<http://www.sherpa.ac.uk/romeo/>

Hybrid derivative dynamic time warping for online industrial batch-end quality estimation

G. Gins^a, P. Van den Kerkhof^a, J.F.M. Van Impe^{a,*}

^a*KU Leuven, Department of Chemical Engineering, Chemical & Biochemical Process Technology & Control (BioTeC), W. de Croylaan 46, B-3001 Leuven, Belgium*

Abstract

This paper discusses the design of an inferential sensor for the online prediction of the end-quality of an industrial batch polymerization process. Due to unequal batch speeds, measurement profiles must be synchronized before modeling. This makes profile alignment an integral part of any inferential sensor. In this work, a novel *online hybrid derivative dynamic time warping* data alignment technique is presented. The proposed technique allows for automatic adjustment of the warping resolution to achieve optimal alignment results for both slowly and rapidly varying parts of the measurement profiles. The proposed online data alignment technique is combined with a *multiway partial least squares* black box model to yield online predictions of the final quality of a running batch process. It is demonstrated that this inferential sensor is capable of accurately predicting the quality online for an industrial polymerization process, even when the production process is only halfway, that is, well before lab measurements become available. As a result of this early warning, batches violating the quality specifications can be corrected or even stopped. This leads to fewer off-spec batches, saves production time, lowers operational costs, and reduces waste material and energy.

1. Introduction

In the chemical process industry, batch processes are commonly used for the production of products with a high added value (e.g., high-performance polymers, pharmaceuticals, and biochemicals). While flexible production

*Corresponding author: jan.vanimpe@cit.kuleuven.be; tel. +32 16321466

is the main advantage of a chemical batch process, monitoring and control remain a major concern. Batch processes are characterized by a fixed production recipe that details a set of processing steps to be followed as closely as possible to achieve a satisfactory product quality. Owing to the lack of (reliable) online product quality measurements, quality measurements are most often only available after batch completion, that is, hours late. Hence, only minor, if any, a posteriori corrections can be made to bring the quality of faulty batches back within specifications. In the worst case, the production process must be run again, and the just-produced, off-spec material wasted.

Hence, there is a need for inferential sensors capable of making reliable estimations of the product's end quality well before lab measurements become available. Through this close monitoring of the production process, off-spec batches can be detected in an early stage, and actions can be taken to correct the product's end quality. If the batch is beyond recovery, the production run can be aborted early, and a new run started. Consequently, fewer off-spec batches are produced, saving valuable production time, lowering operational costs, and reducing waste material and energy.

Multivariate statistical methods, originally designed for monitoring continuous processes, have successfully been extended to batch processes[1–9]. In these works, online batch (phase) alignment is always assumed either (i) easy to obtain (e.g., by means of a monotonically increasing measurement variable), or even (ii) not needed (e.g., during offline analysis, or because all batches or batch phases already have identical lengths). In practice, however, these assumptions rarely hold.

For the alignment of monotonous profiles, using an indicator variable (IV)[10] is the easiest technique. IV uses a monotonously evolving measurement instead of time to express batch maturity. The major drawback of IV is that a monotonous variable is seldom present in a batch. Even if a different IV can be identified for different parts of a batch, online implementation requires that the IV always reaches the same value at the end of (each phase of) the batch.

Two major alignment techniques are available for alignment and synchronization of nonmonotonous variables. A first set of techniques are based on dynamic time warping (DTW)[11, 12]. DTW originated in speech technology but has been applied successfully in several fields, ranging from medicine[13], over bioinformatics[14], pattern recognition[15], and spectroscopic analysis[16], to batch processes[5, 12]. The DTW family of alignment techniques includes adaptations such as curve registration (CR)[17, 18] and dynamic locus anal-

ysis (DLA)[19–21], which first divide a batch in different phases via reference landmarks before synchronizing each phase, or derivative dynamic time warping (DDTW)[15], where the derivatives of the measurement profile are used to estimate the progress.

A second alignment technique is correlation optimized warping (COW)[22]. COW is applied mainly in spectroscopy[23, 24], but Fransson and Folestad[25] employed it for online batch process monitoring. COW was shown to be a heavily constrained version of DTW by Pravdova et al. [26]. Tomasi et al. [27] found that COW and DTW behave similarly for practical purposes.

DDTW has been shown to yield superior results compared to DTW when the evolution of a profile is more important than the actual measurement value[15]. For industrial processes, differences in sensor calibrations lead to differences in measured values between subsequent batches. The underlying process behavior, however, is captured by the evolution of a measurement (that is, its derivative). Therefore, DDTW is suited for aligning batch profiles. However, the computation of numerical derivatives of noisy data severely limits its online implementation for batch monitoring.

In this work, therefore, a novel online hybrid derivative dynamic time warping (HDDTW) data alignment technique is proposed. HDDTW solves the DDTW problem of taking numerical derivatives by using a piece-wise linear approximation of the measurement profile. Furthermore, it is shown to combine the best properties of COW with the flexibility of DTW via a dynamic warping resolution. HDDTW is combined with multivariate statistical methods to yield a true online inferential sensor for predicting batch-end quality. The approach is validated on a full scale industrial application.

Online data alignment using the proposed hybrid derivative dynamic time warping scheme is detailed in section 2. A short description of multiway partial least squares used to infer the final quality of a batch from the online measurement data is provided in section 3. Next, the actual inferential sensor is described in section 4. The industrial chemical batch process used as a case study in this work is described in section 5, and the practical implementation and inferential sensor performance are discussed in section 6 and section 7. Finally, section 8 discusses the application of the methodology presented in this work to other processes before final conclusions are drawn in section 9.

2. Data alignment

Before a multiway partial least squares mathematical model can be identified on the measurement data to relate the available measurements to the final batch quality (see section 3), all batches are required to be of identical length. Furthermore, similar events should occur at the same moment in all batches in order to improve model performance. In the absence of a monotonic indicator variables[28], a more advanced data alignment scheme is used. First, the offline data alignment technique is explained in section 2.1. Next, the online implementation is discussed in section 2.2.

2.1. Hybrid Derivative Dynamic Time Warping

In this work, the HDDTW approach is proposed for aligning measurement profiles. This novel technique combines the flexibility of DTW and DDTW, which has been adopted successfully in various fields of research[5, 12–16], with the robustness of COW, used mainly in chromatographic analysis[22, 25]. This is achieved by making piecewise linear approximations of the unaligned profiles and using these in DDTW. The resulting warping paths are then used to align the original measurement profiles.

HDDTW is based on DTW, which minimizes the difference between two profiles (that is, a test and a reference profile) by dynamically stretching and/or compressing the time axis of the test profile. A short description of DTW is provided here; the reader is referred to Sakoe and Chiba [11] or Kassidas et al. [12] for a full description.

The nonlinear time transformation (that is, warping) is obtained by first constructing the distance matrix $\mathbf{D}$ ($M \times N$) between the test profile ($x_1 \ x_2 \ \dots \ x_M$) and reference profile ($y_1 \ y_2 \ \dots \ y_N$). The element on the m^{th} row and n^{th} column of $\mathbf{D}$ is obtained as

$$\mathbf{D}(m, n) = (x_m - y_n)^2. \quad (1)$$

Next, the warping path $\mathcal{P}$, a set of P different elements of $\mathbf{D}$ (or (m, n) -pairs), is defined as the continuous path from $\mathbf{D}(1, 1)$ to $\mathbf{D}(M, N)$ which minimizes the total distance between both profiles.

$$\mathcal{P} = \arg \min_{\mathcal{P}} \left\{ \sum_{p=1}^P \mathbf{D}(m_p, n_p) \right\} \quad (2)$$

To compute the warping path $\mathcal{P}$, a dynamic programming problem is solved. Hereto, a cumulative distance matrix $\mathbf{D}_c$ ($M \times N$) is constructed recursively.

$$\mathbf{D}_c(m, n) = \mathbf{D}(m, n) + \min \left\{ \begin{array}{c} \mathbf{D}_c(m-1, n) \\ \mathbf{D}_c(m-1, n-1) \\ \mathbf{D}_c(m, n-1) \end{array} \right\} \quad (3)$$

By tracing the minimal total cost from $\mathbf{D}_c(M, N)$ back to $\mathbf{D}_c(1, 1)$, the warping path $\mathcal{P}$ is obtained.

To facilitate the backtracking, a backtracking matrix $\mathbf{B}$ ($M \times N$) is constructed in conjunction with the cumulative cost matrix $\mathbf{D}_c$. Each element $\mathbf{B}(m, n)$ contains a reference to the optimal predecessor of (m, n) , assuming the warping path would pass through this element. Hence, finding the warping path $\mathcal{P}$ is reduced to a simple table lookup procedure.

Several constraints can be imposed on the warping path, such as the commonly used Sakoe-Chiba adjustment window[11], the Itakura parallelogram [29], or warping path slope constraints to respectively limit the delay, the global speed difference, and the instantaneous speed difference between reference and test profiles. These constraints are required to retain physically correct solutions and are typically determined based on physical/chemical process insight or operator expertise. The second term of 3 needs to be adjusted to account for any active constraints, allowing only legal predecessors of the point (m, n) [11, 29].

In the derivative DTW approach, the distance between the derivatives of the profiles is minimized by redefining the distance measure used in the computation of the distance matrix $\mathbf{D}$ as

$$\mathbf{D}(m, n) = \left(\left. \frac{dx}{dt} \right|_m - \left. \frac{dy}{dt} \right|_n \right)^2. \quad (4)$$

This variant of the basic DTW algorithm has been demonstrated[15] to result in more intuitive alignment results and fewer alignment singularities. For most industrial processes, the evolution of a measurement (that is, its derivative) carries information concerning the speed of evolution.

In HDDTW, the measurement signals are first approximated using piecewise linear approximations because of the inherent numerical instability of computing derivatives of noisy data. The length of each linear part is determined dynamically based on the profile's properties by starting a new

linear part whenever the current approximation deviates too much from the true signal (e.g., when the difference between the linear approximation and the true measurement signal exceeds 3 times the standard deviation of the measurement error). (The standard deviation can be (i) computed from the known measurement error provided by the sensor manufacturer, or (ii) estimated by taking the standard deviations of the measurements in an approximately constant part of a measurement profile, where the only variation in the measurements is due to measurement noise.) In those areas where the profiles' main features are located, a single linear part will be suited for approximating the profile for only a few sample points. Hence, a very large number of linear parts is required to approximate the profile, and the resulting approximation will be (almost) identical to the original measurements. In contrast, the featureless zones of the profile, where a constant trend is observed, are approximated with a very small number of linear parts.

The approximated signal's derivative is constant in each linear part by construction. Hence, the distance matrix $\mathbf{D}$ contains rectangular zones where the distance between the derivatives of the test and reference profiles remains constant. The warping path follows the diagonal of any such zone through which it passes, resulting in a linear resampling in each zone.

The zones of equal cost will be very short in those areas where the profiles' main features are located and the high warping resolution of DDTW is achieved. In contrast, the approximation intervals are large in the relatively featureless parts of the profiles. This results in a low-resolution linear resampling of the featureless parts of the measurement profiles, similar to COW. This is an important advantage because measurement noise has a big impact on the warping path of DTW in featureless zones. In a featureless zone, the numerical derivative of the measurement profile is dominated by the noise. To compensate for these oscillations in the derivative, DDTW proposes a fast switching between local compression and expansion of time. The low-resolution linear resampling of the proposed HDDTW scheme negates this nervous behavior, and results in a smooth warping path without sudden speed changes.

The hybrid nature of HDDTW is illustrated in 1 for a hypothetical profile. The points where changes in compression/stretching are allowed are depicted in black for (a) DDTW, (b) COW, and (c) HDDTW. DDTW allows compression/stretching changes at every sample. With COW, compression/stretching changes are allowed less frequently, and larger intervals are each warped linearly. However, the feature-rich area near the end (indicated

by the dashed ellipse) poses a problem for COW: if the process speed changes at each peak, COW provides an incorrect alignment because it is incapable of compensating for multiple speed changes within each warping interval (that is, between two black points). In this case, the number of warping intervals must be increased, or the length of each interval must be adapted individually. HDDTW combines the properties of DDTW and COW: in the featureless zones, only a few changes in warping speed are allowed, as is the case with COW. Near the features, however, the speed of the process is monitored more closely, and the warping resolution of DDTW is recovered.

2.2. Online Implementation

When aligning two profiles, the optimal warping path $\mathcal{P}$ is determined via backtracking. Hereto, the point of the reference profile corresponding to the final point of the running (new) profile $\mathbf{x}$ must be known. When working online, it is unfortunately not known onto which point of the reference profile the last known point of $\mathbf{x}$ must be mapped, as illustrated in 2. The backtracking can, therefore, not be initiated, and correct online HDDTW is impossible.

Kassidas et al. [12] presented a method to obtain an approximation of the warping path. In this work, an improvement of the method of Kassidas et al. is proposed. This reduces the impact of measurement noise on the alignment and prediction results, as explained in 2.2.1. 2.2.2 discusses the computational complexity of the resulting algorithm for online implementation.

2.2.1. Methodology

When making an estimation of the warping path when only the first m_t samples of the new profile $\mathbf{x}$ are known, a partial cumulative cost matrix $\mathbf{D}_{c,t}$ of size $m_t \times N$ is constructed at each time instance. As the progress of the new measurement profile relative to the reference profile is unknown, the point of the reference profile n_t corresponding to m_t remains unknown, hence the warping path cannot be constructed as depicted in 2. Therefore, the point n_t of the reference profile corresponding to m_t has to be determined first. Next, the estimated warping path $\mathcal{P}_t$ is obtained by tracking back between (m_t, n_t) and $(1, 1)$, after which this result is used to warp the known part of the new profile $\mathbf{x}$.

In the original online DTW method for aligning batch trajectories proposed by Kassidas et al. [12], the final point of the new profile is mapped to

(a)

(b)

(c)

Figure 1: Points where changes in warping speed are allowed for (a) DDTW, (b) COW, and (c) HDDTW. HDDTW clearly combines the low warping resolution of COW in the constant or slowly varying parts (featureless) of the profile with the high warping resolution of DDTW in the rapidly changing parts (features) of the profile.

Figure 2: The cost matrix $\mathbf{D}$ with combined Itakura and Sakoe-Chiba constraints, which define a hexagonal zone in which the warping path $\mathcal{P}$ must be located. At time t , the length of the new measurement profile m_t is known, but the corresponding point of the reference profile n_t remains unknown, and the warping path $\mathcal{P}$ cannot be constructed.

the point of the reference profile with minimal current cumulative cost.

$$n_t = \operatorname{argmin}_n \left\{ \mathbf{D}_{c,t}(m_t, n) \mid m_t - \frac{w_c}{2} \leq n \leq m_t + \frac{w_c}{2} \right\} \quad (5)$$

Here, w_c is the width of the legal warping window, defined by any active Sakoe-Chiba constraints. Final Itakura parallelogram constraints on the end of the warping path (near (m_t, n_t)) cannot be imposed.

Speed differences between profiles can be observed by comparing the occurrence of some characteristic features (for example, a decreasing trend). For example, if the new measurement profile has a smaller slope, it proceeds at a slower rate than the reference profile. 5 gives a reasonable estimation of the matching point that would be obtained using offline alignment, provided profile features to indicate the relative speed difference are present.

This is illustrated in 3(a) for artificial test and reference profiles. Even in the presence of measurement noise, it is clear that the new (incomplete) profile proceeds at a slower pace than the reference. In this case, the procedure of Kassidas et al. yields reliable estimates of the reference point onto which the last point of the new profile must be mapped. 3(c) depicts the cumulative cost for mapping the second to last point of the new profile onto each point of the reference (that is, the estimated warping at time $t-1$). 3(e) displays the cumulative cost for the last point of the next profile (at time t). It is evident that the variation of the selected reference point is small during

Figure 3: Illustration of the performance of Kassidas' original method for selecting the reference point onto which the end point of a new profile is mapped. (a,b) Running and reference profiles for (a) feature-rich part of the running profile and (b) feature-poor part of the running profile. (c,f) The cumulative cost for mapping the final point of the running profile to each point of the reference for (c) feature-rich, time $t - 1$, (d) feature-poor, time $t - 1$, (e) feature-rich, time t , (f) feature-poor, time t .

feature-rich periods of the new profile. This results in stable warping results and stable quality predictions provided by the inferential sensor.

During feature-poor (for example, constant) episodes in the test profile, the mapping of the end point to the reference profile using Kassidas' method proves inaccurate, as illustrated in 3(b). Here, measurement noise masks information about the progress of the current measurement profile, and the minimum cumulative distance in 5 is heavily influenced by measurement noise. As a consequence, large changes in the selected reference point from the point obtained by offline alignment are observed at subsequent time points, and prediction accuracy deteriorates. This is visible in the large jump in the minimum of the cumulative cost between 3(d) at time $t - 1$ and 3(f) at time t . The large differences in the selected reference point at consecutive time points negatively impact the prediction quality and stability of the inferential sensor.

To eliminate the loss in prediction accuracy and stability during feature-poor zones, a modification of the original procedure of Kassidas et al. is proposed here. When a new measurement becomes available, the first step is to determine if a feature (for example, a peak, a clear increase or decrease) is occurring. If this is the case, the same technique as in Kassidas et al. for mapping the end point of the incomplete test profile to the reference is adopted. If no feature is detected (and the estimated mapping is inaccurate using the Kassidas et al. approach), the process is assumed to proceed at a constant pace until the current phase is completed.

First, information contained in the batch recipe is used to estimate the length of the current phase. Two types of phase changes exist: (iii) time-driven phase changes, and (ii) event-driven phase changes. Time-driven changes occur at a specific time indicated in the batch recipe, for example, adding extra initiator every 2 hours. Event-driven phase changes happen when some variables attain a value specified in the batch recipe, for example, adding initiator when the reactor temperature reaches a specified level.

If the phase change is event-driven, the phase duration is estimated by assuming the current batch will continue at its current speed. This speed is either inferred from the partial data, or the occurrence time of the phase change is estimated directly, for example, based on the initial conditions at the phase start.

When dealing with time-driven phase changes, the phase is estimated to end near the time stated in the batch recipe. The true transition time might deviate from the prescribed time due to operator variability. For a

well-chosen reference profile, the phase lengths are close to the mean length of each phase in the training set. Therefore, the estimated end point of the current measurement profile $m_{\text{end}}^{\text{est}}$ and the end point of the reference profile of the corresponding phase n_{end} are assumed equal.

When the estimated length of the current phase of the running batch $m_{\text{end}}^{\text{est}}$ is available, a path connecting $(m_t, n_{\text{prev}} + 1)$ and $(m_{\text{end}}^{\text{est}}, n_{\text{end}})$ is constructed. Here, n_{prev} is the previous matching point provided by the Kassidas et al. approach and $m_{\text{end}}^{\text{est}}$ is the estimated length of the current phase. The matching point is kept at n_{end} if the length of the monitored phase exceeds the number of samples of the reference phase.

As long as no feature is detected in the test profile at the following time samples, this linear path is followed. This corresponds to a simple linear resampling of the featureless zone, which matches the assumption that the process continues at a constant pace.

The industrial batch polymerization process studied in this work is a purely time-driven process: all transitions (for example, initiator shots, cooling) are performed by operators.

2.2.2. Computational Complexity

The cumulative cost matrix $\mathbf{D}_c$ and the backtracking matrix $\mathbf{B}$ are defined in a recursive way. Hence, when a new sample of the new profile $\mathbf{x}$ becomes available at time $t+1$, the results from time t can be reused by adding a single column to both $\mathbf{D}_c$ and $\mathbf{B}$. Only the backtracking between (m_{t+1}, n_{t+1}) and $(1, 1)$ must be performed again, requiring approximately $O(t)$ operations.

However, because of their definition, two possible warping paths cannot intersect; they must share the part from the intersection back to the origin $(1, 1)$. Using this property, it is easy to check during the backtracking process whether the current estimate of the warping path $\mathcal{P}_t$ intersects with a limited number of previous results. If this is the case, the further evolution of the warping path is immediately known, without the need for table lookups in the backtracking matrix $\mathbf{B}$. This drastically reduces the number of required operations.

In conclusion, the online estimation of the warping path requires $O(t)$ operations in a worst case scenario, or a nearly constant number of operations if the warping path does not change considerably between subsequent time steps.

3. Reactor Model

A black box model is used to describe the relation between the various measurements obtained from a batch reactor and the quality parameter. This model is described in section 3.1. The technique exploited to use a model trained on measurements of a complete batch run for the estimation of the final quality using only partially available measurements is detailed in section 3.2.

3.1. Model Structure

On a chemical (batch) reactor, most measurements are redundant and highly correlated. Therefore, a multiway partial least squares (MPLS) model, capable of dealing with this data structure, is used[1, 2, 30, 31].

The data matrix for a batch process is three-dimensional. For a known history consisting of I batches with J measurement variables per batch and K samples per measurement variable, this results in an input data matrix $\mathbf{X}$ ($I \times J \times K$). In MPLS, an extension of basic PLS[32], this three-dimensional data matrix is unfolded batch-wise. By placing all K slices of size $I \times J$ next to each other, an unfolded data matrix $\mathbf{X}$ ($I \times JK$) is obtained[1, 28]. This unfolded data matrix $\mathbf{X}$ is then used in a standard PLS model. When L quality variables are measured for each batch, the output matrix $\mathbf{Y}$ ($I \times L$) requires no unfolding.

Before the model is trained, both $\mathbf{X}$ and $\mathbf{Y}$ are mean centered and normalized to unit variance. This operation linearizes the problem around the average measurement profile behavior[1, 28]. By using the covariance matrix of the measurement variable trajectories, MPLS exploits not only the deviation of each process variable from its mean trajectory, but also the simultaneous and temporal correlation between each measurement variable[28].

In PLS, a latent variable modeling approach, the input and output spaces $\mathbf{X}$ and $\mathbf{Y}$ are projected into low dimensional spaces. This decomposition of $\mathbf{X}$ and $\mathbf{Y}$ leads to R latent variables.

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^T (+\mathbf{E}_\mathbf{X}) \\ \mathbf{Y} = \mathbf{U}\mathbf{Q}^T (+\mathbf{E}_\mathbf{Y}) \\ \mathbf{U} = \mathbf{T} (+\mathbf{E}_\mathbf{U}) \end{cases} \quad (6)$$

The scores matrices $\mathbf{T}$ ($I \times R$) and $\mathbf{U}$ ($I \times R$) are the low dimensional approximations of the input and output spaces $\mathbf{X}$ and $\mathbf{Y}$. The matrices $\mathbf{P}$ ($JK \times R$) and $\mathbf{Q}$ ($L \times R$) are the model weight matrices in input and output

space, respectively, and are identified on the available training data using, for example, the NIPALS algorithm[32]. The matrices $\mathbf{E}$ represent the modeling errors or residuals.

Because the matrix $\mathbf{P}$ is not invertible, a $JK \times R$ weight matrix $\mathbf{W}$ is introduced. This matrix has orthonormal columns such that $(\mathbf{P}^T \mathbf{W})^{-1}$ is upper triangular, with ones as diagonal elements. The projection of the input space $\mathbf{X}$ onto the scores space $\mathbf{T}$ is then obtained as

$$\mathbf{T} = \mathbf{XW} (\mathbf{P}^T \mathbf{W})^{-1}. \quad (7)$$

The regression matrix between the input space $\mathbf{X}$ and the scores space $\mathbf{T}$ can easily be identified from this relation.

$$\mathbf{B}_{\mathbf{XT}} \triangleq \mathbf{W} (\mathbf{P}^T \mathbf{W})^{-1}. \quad (8)$$

Using 7, the measurements of a (new) batch can easily be transformed into a prediction of the output variable $\mathbf{Y}$.

$$\hat{\mathbf{Y}} = \mathbf{XW} (\mathbf{P}^T \mathbf{W})^{-1} \mathbf{Q}^T \quad (9)$$

$$= \mathbf{XB}_{\mathbf{XT}} \mathbf{Q}^T \quad (10)$$

3.2. Online Implementation

When using an MPLS model in an online inferential sensor, the final columns of the input matrix $\mathbf{X}$, corresponding with the future measurement profiles, are unknown. Various solutions for this problem have been proposed[3, 7, 28, 33, 34], which can be classified into two general methodologies.

The first group uses a prediction of the future evolution of the measurement profiles to complete the input matrix $\mathbf{X}$; examples include the zero deviation (ZD) and constant deviation (CD) approaches. In contrast, the second group uses a part of the MPLS model corresponding with the currently known measurements to estimate the final quality parameter. Approaches such as known data regression (KDR), trimmed score regression (TSR), single component projection (SCP) and projection to the model plane (PMP) fall into this second group.

Arteaga et al. [34] demonstrated the superior performance of the KDR and TSR methods. Of these two methods, TSR is employed in this work because the estimation of the scores is required to compute the fault detection statistics as described in section 3.4.

At time t , only the first Jt columns of $\mathbf{X}$ are known (denoted by $\mathbf{X}_t$). With TSR, the final scores matrix $\mathbf{T}$ is estimated at time t using the first Jt columns of the regression matrix $\mathbf{B}_{\mathbf{XT}}$ and the training data matrix $\mathbf{X}_{\text{tr}}$ (denoted, respectively, by $\mathbf{B}_{\mathbf{XT},t}$ and $\mathbf{X}_{\text{tr},t}$).

$$\hat{\mathbf{T}}_t = \mathbf{X}_t \mathbf{B}_{\mathbf{XT},t} (\mathbf{B}_{\mathbf{XT},t}^T \mathbf{X}_{\text{tr},t}^T \mathbf{X}_{\text{tr},t} \mathbf{B}_{\mathbf{XT},t})^{-1} \mathbf{B}_{\mathbf{XT},t}^T \mathbf{X}_{\text{tr},t}^T \mathbf{X}_{\text{tr}} \mathbf{B}_{\mathbf{XT}} \quad (11)$$

3.3. Prediction Intervals

When making online predictions, it is important to quantify the uncertainty on the model predictions. Hereto, the prediction interval c_Y of the final quality estimate must be computed. Several methods are available to estimate the prediction interval for MPLS models, ranging from resampling methods via bootstrapping (for example, Aji et al.[35], Abdi[36]) to analytical expression based on error propagation (for example, Faber[37–39], Denham[40], Romera[41]). In this work, the approximation derived by Faber and Bro [39] is used for its mathematical simplicity, but the inferential sensor can also be developed using the resampling-based methods.

Faber and Bro [39] state that the standard deviation of the prediction interval for a new batch can be written as

$$\sigma_{\text{PI}} = \sqrt{\text{MSE} \left(1 + \hat{\mathbf{T}}_{\text{new}} (\mathbf{T}_{\text{tr}}^T \mathbf{T}_{\text{tr}})^{-1} \hat{\mathbf{T}}_{\text{new}}^T \right) - V_{\Delta y}}. \quad (12)$$

$$\text{MSE} = \frac{1}{I - R} \sum_{i=1}^I (\hat{y}_i - y_i)^2 \quad (13)$$

The scores for the training batches are $\mathbf{T}_{\text{tr}}$, and the estimated scores of the new batch are $\hat{\mathbf{T}}_{\text{new}}$. The *mean squared error* (MSE) is used to quantify the model accuracy numerically. $V_{\Delta y}$ is the variance of the quality measurement. Finally, the estimation and lab measurement of the polymer for each of the I batches, are $\hat{y}_i$ and y_i , respectively.

In their work, Faber and Bro mention that the subtraction of $V_{\Delta y}$ may lead to a negative σ_{PI} . Therefore, a more conservative estimate of the prediction interval is made by setting $V_{\Delta y}$ to 0. After this simplification, 12 simplifies to the formula used by Nomikos and MacGregor [2].

$$\sigma_{\text{PI}} = \text{RMSE} \sqrt{1 + \hat{\mathbf{T}}_{\text{new}} (\mathbf{T}_{\text{tr}}^T \mathbf{T}_{\text{tr}})^{-1} \hat{\mathbf{T}}_{\text{new}}^T} \quad (14)$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{I-R} \sum_{i=1}^I (\hat{y}_i - y_i)^2} \quad (15)$$

If the prediction error is Student's t-distributed (this assumption will be checked during implementation), the prediction interval with significance level α for the new batch is, therefore, obtained via

$$c_{\mathbf{Y},i} = \tau_{(I-1, \frac{\alpha}{2})} \text{RMSE} \sqrt{1 + \hat{\mathbf{T}}_{\text{new}} (\mathbf{T}_{\text{tr}}^T \mathbf{T}_{\text{tr}})^{-1} \hat{\mathbf{T}}_{\text{new}}^T}, \quad (16)$$

with $\tau_{(I-1, \alpha/2)}$ the width of the $\alpha/2$ percentile of Student's t-distribution with $I - 1$ degrees of freedom.

Nomikos and MacGregor use the training RMSE and $\mathbf{T}_{\text{tr}}$ in 16. This is under the assumption that training and validation cases come from identical distributions. In an industrial setting, however, this assumption does not necessarily hold. Therefore, cross-validation is used to determine the values of RMSE and $\mathbf{T}_{\text{tr}}$ in this paper.

To extend 16 to online estimates, a full online leave-one-out cross-validation is performed. This leads to time-evolving values for RMSE and $\mathbf{T}_{\text{tr}}$, which are then used to construct the prediction interval at each time point for new batches. This leads to the final, (warped) time-varying expression for the confidence bounds on the model predictions.

$$c_{\mathbf{Y},i}(t) = \tau_{(I-1, \frac{\alpha}{2})} \text{RMSE}(t) \sqrt{1 + \hat{\mathbf{T}}_{\text{new},t} (\mathbf{T}_{\text{tr},t}^T \mathbf{T}_{\text{tr},t})^{-1} \hat{\mathbf{T}}_{\text{new},t}^T} \quad (17)$$

The authors would like to reiterate that 17 is only a conservative approximation. In this work, 17 will be used to assess the convergence of the online quality estimate by evaluating the prediction error as a function of time. Here, the analytical expression for the prediction intervals is used, but can be easily interchanged with resampling-based methods (for example, Abdi[36], Aji et al.[35]).

3.4. Detection Of Abnormal Batches

Because the inferential sensor is a black box model, it is expected to perform poorly for new batches which are significantly different from the batches in the training set. Therefore, two abnormality statistics (T^2 and SPE) are used to identify abnormal batches[28].

3.4.1. T^2 Statistic

The columns of the scores matrix $\mathbf{T}$ are uncorrelated by definition and assumed multivariate normal-distributed. Hence, a simple confidence region for the scores can be constructed. To improve interpretability, however, the T^2 statistic is computed as the squared Mahalanobis distance from a batch's scores to the origin.

$$T_i^2(t) = \hat{\mathbf{T}}_{i,t} \mathbf{S}_{\mathbf{T},t}^{-1} \hat{\mathbf{T}}_{i,t}^T \quad (18)$$

The covariance matrix of the scores at time t , $\mathbf{S}_{\mathbf{T},t}$, is obtained from the training batches. At each time t , the training batches are considered partially unknown, and their scores $\hat{\mathbf{T}}_{\text{tr},t}$ are estimated using the TSR method. The covariance of these estimates is $\mathbf{S}_{\mathbf{T},t}$.

Because T^2 is $F_{(R, I_{\text{tr}} - R)}$ -distributed, an upper bound u_T for T^2 at a specified tolerance level α is easily obtained[2, 42, 43].

$$u_T = \frac{R(I_{\text{tr}}^2 - 1)}{I_{\text{tr}}(I_{\text{tr}} - R)} F_{(R, I_{\text{tr}} - R; \alpha)} \quad (19)$$

Here, R is the number of latent variables of the MPLS model, I_{tr} is the number of training batches, and $F_{(R, I_{\text{tr}} - R; \alpha)}$ is the upper critical value of the F -distribution with R numerator degrees of freedom and $I_{\text{tr}} - R$ denominator degrees of freedom, and tolerance α . As can be seen, all terms in 19 are time-independent.

Batches exceeding this limit value u_T are significantly different from the batches used during model construction, Quality predictions for these batches are unreliable and should be discarded.

3.4.2. SPE Statistic

The residuals of a new batch in the input space, $\mathbf{E}_{\mathbf{X}}$, express the similarity between the input $\mathbf{X}$ and its projection to the model's input space $\mathbf{T}\mathbf{P}^T$. Because high residual values indicate a large dissimilarity between $\mathbf{X}$ and $\mathbf{T}\mathbf{P}^T$, they are used to compute the *Squared Prediction Error* (SPE) statistic. The SPE is a measure for the reconstruction error at the most current time point t . This is performed for each time sample for a batch i .

$$\mathbf{E}_{\mathbf{X},i,t} = \mathbf{X}_{i,t} - \hat{\mathbf{T}}_{i,t} \mathbf{P}_t^T \quad (20)$$

$$\text{SPE}_i(t) = (\mathbf{E}_{\mathbf{X},i,t}^t) (\mathbf{E}_{\mathbf{X},i,t}^t)^T \quad (21)$$

The matrix $\mathbf{P}_t$ contains the columns of the input space weight matrix $\mathbf{P}$ corresponding to the known measurements at time t (that is, the first Jt

columns of $\mathbf{P}$). The $1 \times J$ matrix $\mathbf{E}_{\mathbf{x},i,t}^t$ contains only those entries of $\mathbf{E}_{\mathbf{x},i,t}$ corresponding with the most recent time point t (that is, the last J columns of $\mathbf{E}_{\mathbf{x},i,t}$).

The SPE is approximately $g\chi^2(k)$ -distributed with mean gk and variance g^2k [2, 28, 44]. The time-varying parameters g and k are estimated using the mean and standard deviation of the SPE for the training batches at each time t , respectively $\mu_{\text{SPE}}(t)$ and $\sigma_{\text{SPE}}(t)$, by matching the observed moments of the distribution.

$$g(t) = \frac{\sigma_{\text{SPE}}^2(t)}{2\mu_{\text{SPE}}(t)} \quad (22)$$

$$k(t) = \frac{2\mu_{\text{SPE}}^2(t)}{\sigma_{\text{SPE}}(t)} \quad (23)$$

Hence, time-varying upper limit $u_{\text{SPE}}(t)$ of the SPE is obtained.

$$u_{\text{SPE}}(t) = g(t) \chi^2(k(t); \alpha) \quad (24)$$

Here, $\chi^2(k(t); \alpha)$ is the upper critical value of the χ^2 -distribution with $k(t)$ degrees of freedom and tolerance level α .

4. Inferential Sensor

With the combination of the online data alignment and reactor model implementations, the inferential sensor is obtained. A schematic representation of the different steps in the identification and use of the sensor is provided in 4. For the running phase, the online HDDTW algorithm detailed in section 2.2 is used to provide an estimated warping path after each sample. Once a specific batch phase is completed, a full HDDTW alignment can be performed, and a final warping is obtained for the completed phase. Next, the aligned measurement profiles are obtained using the combined warping paths for each individual batch phase.

While the online HDDTW algorithm requires at most $O(t)$ operations, computations slow down as time progresses. However, because the online HDDTW estimation is only performed for the currently running batch phase, the time required to obtain a final product viscosity estimate is reset whenever a new batch phase is started.

Figure 4: Schematic illustration of the steps in the construction and use of the differential sensor.

5. Industrial Batch Polymerization Process

An industrial batch polymerization process is studied in this work. A schematic view of the industrial batch reactor is shown in 5. At the start of the production process, approximately half of the raw materials (mainly monomer, solvent, and initiator) stored in the premix vessel (P) are fed into the batch reactor (R) in a first loading step. Next, the reactor is heated by circulating hot water through the spiral (S). The polymerization starts when the initiator present in the reactor activates at a temperature T_{act} . Shortly after the start of the reaction, the second part of the premixer content is added to the reactor in the second loading step. During the production run, the reaction heat causes some of the solvent to evaporate. These gases are cooled and condensed in the top cooler (C). A negligible uncondensed vapor fraction is bled through the vent (V), while the liquid fraction is recirculated to the reactor. Approximately 2, 4, and 6 h after the start of the polymerization reaction, new initiator material is added manually to the reactor to compensate for initiator inactivation and to maintain a sufficiently high reaction rate. Finally, about 8 h after the start of the reaction, the polymerization reaction is terminated by cooling down the reactor, after which it is emptied.

During the batch process, 30 sensors record temperature, pressure, and flow rate at various positions, with samples taken in 15-second intervals. This results in approximately 2000 samples for every sensor during a complete batch run.

For each completed batch, offline lab analyses are used to verify the (final) polymer viscosity, which has upper and lower specification limits. While a large collection of historical measurements is available, the majority cannot be used for statistical analysis because of a too high compression ratio before measurement storage (a problem also reported by Kourti et al. [45], Thornhill et al. [46], and Imtiaz et al. [47]). For other batches, viscosity measurements are missing. As a result, 72 batches are available for model identification. At a later stage, 10 new batches were run for model validation.

From the measurement data, four phases can be clearly identified in each batch. A schematic overview of the batch recipe events and the final four identified batch phases is provided in 6. The start of the first phase coincides with the start of the polymerization reaction. Because the initiator is activated thermally, the reaction starts when the reactor temperature exceeds a specific threshold. The first batch phase lasts until the start of the sec-

Figure 5: Schematic view of the industrial batch reactor, consisting of a premixer (P), reactor (R), condenser (C), cooling or heating spiral (S) and gas vent (V).

ond loading phase, after which the second batch phase starts. The end of the second feeding step would be a logical choice for the end of the second batch phase. However, the feed rate signal differs significantly from its set point at the end of the second feeding step. This makes the unambiguous determination of the end point impossible. Therefore, this event cannot be used to mark the end of the second phase. The next event in the recipe is the second initiator shot, occurring approximately 2 h after the start of the polymerization reaction. However, the initiator shot is not observed in the temperature measurements of all batches, making this event also useless to mark the end of the second batch phase. The subsequent initiator shots can be clearly identified from the temperature measurements. Therefore, these events are used to mark the next batch phases.

The final result is that the following four phases are retained: (i) from the detection of the polymerization reaction until the start of the second feeding phase, (ii) from the start of the second feeding phase until the second initiator shot, 4 h after the reaction detection, (iii) between the second and third initiator shots, and (iv) from the third initiator shot until the start of the reactor cooling. The first phase lasts only a few minutes, while the second batch phase lasts approximately 4 h. Each of the following two batch phases take about 2 h to complete.

Figure 6: Schematic overview of the batch recipe events and the four identified batch phases.

6. Practical Implementation

Some features of the inferential sensor described in section 4 require specific tuning to the chemical batch process modelled in this work. In section 6.1 and section 6.2, some remarks concerning the online identification of the various batch phases and online HDDTW alignment are made. Section 6.3 details the specific parameters of the MPLS model. The determination of confidence intervals for the model prediction is discussed in section 3.3 and 3.4. Finally, the computational requirements of the inferential sensor are discussed in 6.4.

6.1. Online Batch Phase Transition

The identification of the transition points between the batch phases, required for the online data alignment, is also performed online. However, this identification is not instantaneous for the second and third initiator shot: once a candidate transition point for phase m , t_m^* , is detected, about extra 40 samples are required for the identification of the true transition point t_m , and the batch phase transition is identified with a small delay.

In all but a few batches, the true phase transition is located near the start of the interval $[t_m^*, t_m^* + 40]$. As can be seen in 7, the vapor temperature exhibits a sharp drop at the start of the third and fourth batch phase, differing significantly from the profiles at the end of the second and third batch phase, respectively. Including these large temperature drops in the online estimation of the, still running, previous batch phase would cause severe distortions of the model predictions. Hence, no model predictions are made during these small transition phases.

6.2. HDDTW Parameters

Because no suitable indicator variables are present in the measurements, HDDTW is used for data alignment. The vapor temperature at the entrance

Figure 7: Aligned vapor temperature profiles at the entrance of the top cooler for all training batches for the four batch phases.

of the top cooler is selected as the aligning variable in the HDDTW algorithm based on two criteria. First, the vapor temperature is not a controlled variable. Hence, its evolution is the result of the polymerization reaction and not -directly- of control actions. As a result, it contains information on the batch progress. The second criterion is that it features the temperature drops (that is, process features) prominently. The reference profile for this variable is obtained from a typical batch operation run. As can be seen in 7, the vapor temperature is nonmonotonous; a simple IV alignment cannot be used for aligning the measurement profiles. The width of the linear approximation windows and Sakoe-Chiba adjustment window for each batch phase are determined based on expert knowledge concerning typical measurement noise and batch speeds.

All phase transitions are effected by operators, therefore the methodology for a time-driven process explained in 2.2.1 is employed. The vapor temperature profiles shown in 7, consist of a sequence of temperature drops and zones of nearly constant temperature. If the vapor temperature is 10 °C or more below the activation temperature T_{act} as indicated by the horizontal solid line in 7, a feature is considered detected and the standard methodology of Kassidas et al. is applied. Otherwise, the current progress is estimated based on information contained in the batch recipe as detailed in 2.2.1.

6.3. MPLS Parameters

After offline alignment of the available measurement signals for each batch run, the optimal input variables for an order of the MPLS model are identified on the training data set. Hereto, a *forward branch and bound* approach is combined with *10-fold cross-validation*. First a single-input MPLS model is trained and crossvalidated on each of the 31 available input profiles. The input of the MPLS model with the best cross-validation performance is retained and combined with each other available input, resulting in 30 input pairs. Next, a two-input MPLS model is trained on each pair, and the inputs of the model with the best cross-validation is again retained. This is repeated until all 31 input signals are ranked from most to least important. The optimal input set is the set of inputs corresponding with the MPLS model with the lowest cross-validation error. For a more detailed description of the selection procedure, the reader is referred to Gins et al. [48].

The optimal input set for the industrial polymerization consists of (i) the reactor temperature, (ii) the vapor temperature in the reactor, and (iii) the vapor temperature at the entrance of the top cooler. For these inputs, a MPLS model with four latent variables yields the best cross-validation performance. The weights of the MPLS model are then identified on all 72 batch runs.

6.4. Computational Requirements

The inferential sensor is implemented in MATLAB[®] 7.8 (The MathWorks, Inc., Natick, MA), and tested on a computer equipped with an Intel[®] Core[™]2 Quad Q9550 2.83 GHz processor and 2×2 GB RAM. Even near the end of the second batch phase, the final quality estimate is updated (including graphical representation) in approximately 0.2 seconds, well before the next measurement sample becomes available.

7. Results

To assess the performance of the inferential sensor, the final product quality is estimated at each time step using the inferential sensor, both for the training and validation batches. First, the accuracy of the final estimate is investigated in section 7.1. Next, the convergence of the estimate towards the final model prediction is discussed in section 7.2 and section 7.3.

Table 1: Offline model performance during training and cross-validation.

	RMSE	R^2
Training	0.4192	90.92%
Cross-validation	0.5364	84.64%
Validation	0.5757	85.99%

7.1. Offline Quality Prediction

While the model prediction can converge to its final value very quickly, the performance of the inferential sensor is also determined by the quality of this final model prediction. Therefore, the accuracy of the offline quality estimates is investigated. The identified MPLS model is capable of accurately predicting the viscosity measurements when complete measurement profiles (and offline alignment results) are available. This is evidenced by 8(c), where a good similarity between model predictions and lab measurements is observed both during training (left) and validation (right). 1 lists the model performance (RMSE and R^2) for training, validation, and cross-validation. The small difference between validation and cross-validation again indicates a good model performance.

7.2. Pseudo-Online Quality Prediction With Offline HDDTW

The online inferential sensor combines TSR with online HDDTW. To correctly assess the performance of the proposed online HDDTW algorithm, it is first checked whether the identified MPLS model is capable of accurately predicting the final quality online if perfect alignment results are available. In this case, only TSR influences the pseudo-online quality estimates. The pseudo-online performance of the inferential sensor is expected to be an upper bound for the true online inferential sensor; a poor estimation performance in this best-case scenario would also be reflected in the stability and accuracy of online HDDTW.

Therefore, the offline alignment results are first used to obtain a pseudo-online estimate of the final quality. This corresponds to a best-case situation where the online and offline alignments are always in perfect agreement, that is, when the estimation of the final point of the warping path is always correct.

8 compares the offline lab measurements of the polymer viscosity with the available model predictions after the completion of each of the final three batch phases for both training and validation batches. As can be seen, a

Table 2: Comparison of training, validation and cross-validation RMSE after completion of each batch phase.

Batch progress	Training		Cross-validation		Validation	
	RMSE	R^2	RMSE	R^2	RMSE	R^2
50%	0.4770	88.07%	0.6460	76.77%	0.6214	81.30%
75%	0.4410	89.90%	0.5812	81.68%	0.5800	84.38%
100%	0.4192	90.92%	0.5364	84.64%	0.5757	85.99%

good approximation of the measured viscosities is already obtained after the second batch phase, approximately halfway through the batch. This indicates that the initial stages of the batch largely determine the final quality. The end phase of the batch has a smaller influence on the total quality. This corresponds with the chemical background of the process: maintaining a correct reaction rate at the start of the batch is important for the final viscosity. If the initial reaction rate is too high, too many short polymer chains are formed and the final viscosity is too low. If the reaction rate is too low, on the other hand, many long polymer chains will be formed. This will result in a very high final viscosity. Near the end of the batch, the reaction rate has a smaller influence on the final viscosity because the polymer already has a length in the correct range. This observation leads to the conclusion that the MPLS model correctly captures the underlying chemical phenomena of the polymerization.

The results are summarized in 2 for both training and validation sets. As expected, the results for the offline model and full online model are identical, as both take complete measurement profiles as inputs. The good agreement between cross-validation and validation RMSE values leads to the conclusion that the inferential sensor accurately predicts the final polymer viscosity for new batches.

The accuracy of the pseudo-online quality estimate is assessed via (conservative estimation of) the prediction intervals. The dominating factor in the width of the prediction interval is the RMSE (see section 3.3). 9 depicts the evolution over time of the RMSE for the training batches in cross-validation. As can be expected, the width of the prediction interval decreases as the batch nears completion. Starting from approximately 50% warped batch progress, the prediction interval is near its final width, indicating that reliable predictions are available after less than 4.5 h of batch operation, more than 4 h before lab measurements.

Figure 8: Comparison of the model prediction and the lab measurement of the polymer viscosity. For training (left) and validation sets (right) (a) after the second batch phase, (b) after the third batch phase, and (c) after batch completion.

Figure 9: Evolution of the RMSE in cross-validation.

These results lead to the conclusion that the developed MPLS model provides accurate estimates of the final batch quality if perfect alignment results were available.

7.3. Online Quality Prediction With Online HDDTW

When using the online inferential sensor to predict the final product quality, it is important that the model predictions converge to their final value as quickly as possible. The correct alignment of the partial profile is critical for a quick and smooth convergence. If the online alignment varies considerably between subsequent time points or deviates largely from the alignment that would be obtained in offline mode, the inferential sensor's stability and accuracy are seriously hampered. In this section, the alignment performance and its influence on the inferential sensor's online quality prediction is discussed.

7.3.1. Deviation From True Warping End Point

In 2.2.1, a new method for matching the end point of the current partial profile with the reference profile was proposed as an alternative to the approach of Kassidas et al.. 10 depicts the mean sample distance between the matching point of the partial profile and the matching point that would be obtained after alignment of the complete profile for Kassidas et al.'s approach (---) and the new approach detailed in 2.2.1 (—). It is clear from 10 that the new approach exhibits an increased stability and overall lower

Figure 10: Mean value of the absolute deviation from the offline matching point using the original approach of Kassidas et al. [12] (– –) and using the new method explained in 2.2.1 (—).

deviation from the offline alignment compared to Kassidas et al.’s approach. This especially rings true during the feature-poor zones at the end of each phase.

7.3.2. Deviation From Final Quality Estimate

To further study the performance difference between the method of Kassidas et al. and the method detailed in 2.2.1, for each batch of the training set the point in time is determined from which on the deviation from the final quality prediction does not exceed a specified tolerance level. Next, the number of batches having reached this accuracy level as a function of time is investigated. The results are depicted in 11(a) and (b) for a tolerated relative deviation of 5% and 10%, respectively.

Near the end of the second batch phase, which corresponds with 40-55% batch progress, the difference in prediction accuracy between the method of Kassidas et al. and the methodology described in 2.2.1 is significant. At 50% completion, approximately 96% of the training batches are within 10% of their final predicted polymer viscosity as compared to 86% when using Kassidas’ method. At that same time, 67% of the training batches show no deviation from their final prediction larger than 5% for the remainder of the batch operation. Using the Kassidas et al. approach, in only 48% of

Figure 11: Number of training batches having a relative deviation from the final quality prediction less than or equal to (a) 5% or (b) 10% beyond a specific batch progress. The dashed (—) and solid (—) curves represent the prediction accuracy performance using the approach of Kassidas et al. and the method described in 2.2.1, respectively.

the batches the final quality prediction exhibits a relative deviation smaller than 5%. The relative deviation is higher as a consequence of mismatch between the end point of the online profile and the progress in the reference profile during the feature-poor period at the end of the second phase. At the beginning of the third batch phase, the relative deviation of the final prediction using the method of Kassidas et al. drops sharply for all batches and approaches the performance of the new method in 2.2.1. After 60% batch completion, a maximum deviation of 10% is observed in 98% of the training batches, and 88% of the training batches deviate no more than 5% from their final prediction value for each method.

For the validation set, all batches are located in a 10% accuracy interval around their final prediction values after 50% of the batch process for both methods. In addition, 50% of the batches have already reached the 5% tolerance level. After 60% of the batch operation, 80% of the validation batches show a deviation from their final prediction no larger than 5%. Finally, in the final 20% of the batch production, no validation batch exhibits a deviation from its final prediction larger than 5%.

7.3.3. Reliability Of Online Estimates

The validation set might include batches which have a significant difference from the batches present in the training data set. Therefore, the two fault detection statistics (T^2 and SPE) for each validation batch are computed and compared with their respective control limits. 12 illustrates the evolution of the T^2 for each of the validation batches. The SPE has a similar

Figure 12: Input space residuals T^2 for the validation batches as a function of batch progress. The dashed (—) and dotted (···) lines represent the 99% and 99.9% confidence intervals, respectively.

evolution.

From these plots, it is clear that none of the batches exceeds the 99% confidence bounds for any statistic after an initial startup phenomenon. Therefore, the model predictions of all validation batches are considered reliable.

13 depicts the centered and rescaled predicted final product quality over the course of a complete production run for two training batches using the new approach of 2.2.1. 13(a) is representative for most of the training batches. For some batches, the behavior illustrated in 13(b) is observed. As can be seen, a very good prediction of the final polymer viscosity is obtained well before the batch's completion. The relative smoothness of the profiles indicates small variations in the warping path between subsequent time instances.

Similar results are observed for the validation batches, which indicates a good validation behavior of the inferential sensor, which provides stable, nonerratic predictions of the product quality.

The good agreement with the values observed on the training data set again suggests a very good validation of the inferential sensor. The inferential sensor can therefore be used to obtain estimates of the final product quality well before batch completion.

Figure 13: The evolution of the predicted final quality and its 95% confidence interval (—) during batch progression for two training batches, compared with the final lab quality measurement (---).

7.4. Validation Conclusions

Based on the good validation results obtained in section 7.1 through section 7.3, it is concluded that the developed inferential sensor is capable of quickly and accurately predicting the final polymer viscosity of the considered batch process. The proposed online HDDTW scheme yields more accurate predictions than the original scheme proposed by Kassidas et al..

8. Application To Other Processes

While the inferential sensor presented in this work is tuned to the specific batch process detailed in section 5, it is easily adapted to other (bio)chemical batch processes.

Expert knowledge is only used in the identification of relevant phases and corresponding HDDTW constraint parameters (such as, for example, the width of the Sakoe-Chiba adjustment window w_c). Even so, a detailed description of the production process often suffices to extract this information from the data.

The MPLS model weights, prediction intervals for the model predictions and fault detection statistics, detailed in section 6.3, section 3.3, and section 3.4, respectively, are obtained using standard model training and validation techniques. No expert knowledge is required, and these procedures can easily be applied to other (bio)chemical processes.

9. Conclusions

A novel *online hybrid derivative dynamic time warping* algorithm was proposed in this paper. The proposed technique combines the warping of flexibility of *derivative dynamic time warping* with the robustness against noise of *correlation optimized warping* by dynamically changing the warping resolution. Furthermore, a modification of the method of Kassidas et al. for online estimation of the data alignment was presented to reduce the impact of measurement noise and improve the stability of the results.

The presented *online hybrid derivative dynamic time warping* was used in the construction of an inferential sensor with the goal of quickly and accurately predicting the final product quality of a batch polymerization process. It was combined with a black box *multiway partial least squares* model of the polymerization reaction taking place inside the reactor. Using three on-line temperature measurements available on the chemical batch reactor (that is, (i) the reactor temperature, (ii) the vapor temperature inside the reactor, and (iii) the temperature of the vapor at the inlet of the top cooler) as inputs, the final polymer viscosity is estimated online. Concurrent to the model predictions, the reliability of the inferential sensor is provided using T^2 and SPE fault detection statistics.

Results indicate a significant improvement in stability of the final quality estimates with the proposed *online hybrid derivative dynamic time warping* in comparison with the original approach of Kassidas et al.. Next, it was shown that the inferential sensor is capable of predicting the final polymer viscosity quickly and accurately once a completion between 50% and 60% is reached, well before lab measurements become available. This early prediction of the final polymer viscosity will result in fewer off-spec batches being produced, saving valuable production time, lowering operational costs and reducing waste material and energy.

Acknowledgements

Work supported in part by Project PVF/10/002 (OPTEC Optimization in Engineering Center) of the Research Council of the KU Leuven, Project KP/09/005 (SCORES4CHEM) of the Industrial Research Council of the KU Leuven, and the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian Federal Science Policy Office. P. Van den Kerkhof has a Ph.D grant of the Agency for Innovation by Science and Technology in

Flanders (IWT-Vlaanderen). J. Van Impe holds the chair Safety Engineering sponsored by the Belgian chemistry and life sciences federation essenscia. The authors assume scientific responsibility.

The authors would like to thank J. De Wilde of CYTEC (Drogenbos, Belgium) for the data sets provided for this study.

- [1] Nomikos, P.; MacGregor, J.F. Monitoring batch processes using multiway principal component analysis. *AIChE J.* **1994**, *40*(8), 1361-1375.
- [2] Nomikos, P.; MacGregor, J.F. Multiway partial least squares in monitoring batch processes. *Chemom. Intell. Lab. Syst.* **1995**, *30*, 97-108.
- [3] Boqué, R.; Smilde, A.K. Monitoring and diagnosing batch processes with multiway covariates regression models. *AIChE J.*, **1999**, *45*(7), 1504-1520.
- [4] Dorsey, A.W.; Lee, J.H. Building inferential prediction models of batch processes using subspace identification. *J. Process Control* **2003**, *13*, 397-406.
- [5] García-Munoz, S.; Kourti, T.; MacGregor, J.F.; Mateos, A.G.; Murphy, G. Troubleshooting of an industrial batch process using multivariate methods. *Ind. Eng. Chem. Res.* **2003**, *42*, 3592-3601.
- [6] Kourti, T. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *J. Chemom.* **2003**, *17*, 93-109.
- [7] García-Munoz, S.; Kourti, T.; MacGregor, J.F. Model predictive monitoring for batch processes. *Ind. Eng. Chem. Res.* **2004**, *43*, 5929-5941.
- [8] Dorsey, A.W.; Lee, J.H. Monitoring of batch processes through state-space models. *AIChE J.* **2004**, *50*(6), 1198-1210.
- [9] Lu, N.; Yao, Y.; Gao, F. Two-dimensional dynamic PCA for batch process monitoring. *AIChE J.* **2005**, *51*(12), 3300-3304.
- [10] Kourti, T.; Lee, J.; MacGregor, J.F. Experiences with industrial applications of projection methods for multivariate statistical process control. *Comput. Chem. Eng.* **1996**, *20*(A), 745-750.

- [11] Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust., Speech, Signal Process.* **1978**, *ASSP-26*, 43-49.
- [12] Kassidas, A.; MacGregor, J.F.; Taylor, P.A. Synchronization of batch trajectories using dynamic time warping. *AIChE J.* **1998**, *44*(4), 864-875.
- [13] Caiani, E.G.; Porta, A.; Baselli, G.; Turiel, M.; Muzzupappa, S.; Pieruzzi, F.; Crema, C.; Malliani, A.; Cerutti, S. Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume. *Comput. Cardiol.* **1998**, *25*, 73-76.
- [14] Aach, J.; Church, G. Aligning gene expression time series with time warping algorithms. *Bioinformatics* **2001**, *17*, 495-508.
- [15] Keogh, E; Pazzani, M. Derivative dynamic time warping. In *First SIAM International Conference on Data Mining*, Chicago, IL, U.S.A., 2001.
- [16] Ramaker, H.-J.; Van Sprang, E.N.M.; Westerhuis, J.A.; Smilde, A.K. Dynamic time warping of spectroscopic BATCH data. *Anal. Chim. Acta* **2003**, *498*, 133-153.
- [17] Williams, B.; Ündey, C.; Çinar, A. Detection of process landmarks using registration for on-line monitoring. In *Proceedings of IFAC DYCOPS6*, Cheju Island (Korea), 2001, 257-263.
- [18] Ündey, C.; Çinar, A. Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Systems Magazine*, Oct 2002, 40-52.
- [19] Srinivasan, R.; Qian, M.S. Offline temporal signal comparison using singular points augmented time warping. *Ind. Eng. Chem. Res.* **2005**, *44*, 4697-4716.
- [20] Srinivasan, R.; Qian, M.S. Online fault diagnosis and state identification during process transitions using dynamic locus analysis. *Chem. Eng. Sci.* **2006**, *61*, 6109-6132.
- [21] Srinivasan, R.; Qian, M.S. Online temporal signal comparison using singular points augmented time warping. *Ind. Eng. Chem. Res.* **2007**, *46*, 4531-4548.

- [22] Nielsen, N.-P.V.; Carstensen, J.M.; Smedsgaard, J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr., A* **1998**, *805*, 17-35.
- [23] Bylund, D.; Danielsson, R.; Malmquist, G.; Markides, K.E. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography-mass spectrometry data. *J. Chromatogr., A* **2002**, *961*, 237-244.
- [24] Skov, T.; Van den Berg, F.; Tomasi, G.; Bro, R. Automated alignment of chromatographic data. *J. Chemom.* **2006**, *20*, 484-497.
- [25] Fransson, M.; Folestad, S. Real-time alignment of batch process data using COW for on-line process monitoring. *Chemom. Intell. Lab. Syst.* **2006**, *84*, 56-61.
- [26] Pravdova, V.; Walczak, B.; Massart, D.L. A comparison of two algorithms for warping of analytical signals. *Anal. Chim. Acta* **2002**, *456*, 7792.
- [27] Tomasi, G.; Van den Berg, F.; Andersson, C. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *J. Chemom.* **2004**, *18*, 231-241.
- [28] Nomikos, P.; MacGregor, J.F. Multivariate SPC charts for monitoring batch processes. *Technometrics* **1995**, *37*(1), 41-59.
- [29] Itakura, F. Minimum prediction residual principle applied to speech recognition. *IEEE Trans. Acoust., Speech, Signal Process.* **1975**, *ASSP-23*, 52-57.
- [30] Kourti, T.; Nomikos, P.; MacGregor, J.F. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *J. Process Control* **1995**, *5*, 277-284.
- [31] Wold, S.; Geladi, P.; Ebersen, K.; Öhman, J. Multi-way principal components- and PLS-analysis. *J. Chemom.* **1987**, *1*(1), 41-56.
- [32] Geladi, P.; Kowalski, B.R. Partial least-squares regression: a tutorial. *Anal. Chim. Acta* **1986**, *185*, 1-17.

- [33] Nelson, P.R.C.; Taylor, P.A.; MacGregor, J.F. Missing data methods in PCA and PLS: score calculations with incomplete observations. *Chemom. Intell. Lab. Syst.* **1996**, *35*, 45-65.
- [34] Arteaga, F.; Ferrer, A. Dealing with missing data in MSPC: several methods, different interpretations, some examples. *J. Chemom.* **2002**, *16*, 408-418.
- [35] Aji, S.; Tavoraro, S.; Lantz F.; Faraj, A. Apport du bootstrap à la régression PLS: application à la prédiction de la qualité des gazoles. *Oil Gas Sci. Technol.* **2003**, *58*(5), 599-608.
- [36] Abdi, H. Partial least squares regression and projection on latent structure regression (PLS Regression). *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 97-106.
- [37] Faber, K.; Kowalski, B.R. Propagation of measurement errors for the validation of predictions obtained by principal component regression and partial least squares. *J. Chemom.* **1997**, *11*, 181-238.
- [38] Faber, K.M. Comparison of two recently proposed expressions for partial least squares prediction error. *Chemom. Intell. Lab. Syst.* **2000**, *52*, 123-134.
- [39] Faber, K.M.; Bro, R. Standard error of prediction for multiway PLS 1. Background and a simulation study. *Chemom. Intell. Lab. Syst.* **2002**, *61*, 133-149.
- [40] Denham, M.C. Prediction intervals in partial least squares. *J. Chemom.* **1997**, *11*, 39-52.
- [41] Romera, R. Prediction intervals in Partial Least Squares regression via a new local linearization approach. *Chemom. Intell. Lab. Syst.* **2010**, *103*, 122-128.
- [42] Choi, S.W.; Martin, E.B.; Morris, A.J.; Lee, I.-B. Fault detection based on a maximum likelihood PCA mixture. *Ind. Eng. Chem. Res.* **2005**, *44*(7), 2316-2327.
- [43] Tracy, N.D.; Young, J.C.; Mason, R.L. Multivariate control charts for individual observations. *J. Quality Technol.* **1992**, *24*(2), 88-95.

- [44] Box, G.E.P. Some theorems on quadratic forms applied in the study of analysis of variance problems: Effect of inequality of variance in one-way classification. *Ann. Math. Stat.* **1954**, 25(2), 484-498.
- [45] Kourti, T. Abnormal situation detection, three-way data and projection methods; robust data archiving and modeling for industrial applications. *Ann. Rev. Control* **2003**, 27, 131-139.
- [46] Thornhill, N.F.; Choudhury, M.A.A.S.; Shah, S.L. The impact of compression on data-driven process analyses. *J. Process Control* **2004**, 14, 389-398.
- [47] Imtiaz, S.A.; Choudhury, M.A.A.S.; Shah, S.L. Building multivariate models from compressed data. *Ind. Eng. Chem. Res.* **2007**, 46, 481-491.
- [48] Gins, G.; Vanlaer, J.; Van Impe, J.F.M. Combined online quality prediction and fault detection in (bio-)chemical batch processes: discriminating between critical and non-critical process disturbances. *Submitted to Ind. Eng. Chem. Res.*, **2011**.