Chemometrics and Intelligent Laboratory Systems 167 (2017) 161–170

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Selecting local constraint for alignment of batch process data with dynamic time warping

Max Spooner [a,*], David Kold [b], Murat Kulahci [a,c]

[a] DTU Compute, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Chr. Hansen A/S, Hvidovre, Denmark
[c] Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

ARTICLE INFO

Keywords:
Batch process
Dynamic time warping
Local constraint
Global constraint
Step pattern
Trajectory synchronisation

ABSTRACT

There are two key reasons for aligning batch process data. The first is to obtain same-length batches so that standard methods of analysis may be applied, whilst the second reason is to synchronise events that take place during each batch so that the same event is associated with the same observation number for every batch. Dynamic time warping has been shown to be an effective method for meeting these objectives. This is based on a dynamic programming algorithm that aligns a batch to a reference batch, by stretching and compressing its local time dimension. The resulting "warping function" may be interpreted as a progress signature of the batch which may be appended to the aligned data for further analysis. For the warping function to be a realistic reflection of the progress of a batch, it is necessary to impose some constraints on the dynamic time warping algorithm, to avoid an alignment which is too aggressive and which contains pathological warping. Previous work has focused on addressing this issue using global constraints. In this work, we investigate the use of local constraints in dynamic time warping and define criteria for evaluating the degree of time distortion and variable synchronisation obtained. A local constraint scheme is extended to include constraints not previously considered, and a novel method for selecting the optimal local constraint with respect to the two criteria is proposed. For illustration, the method is applied to real data from an industrial bacteria fermentation process.

1. Introduction

In industrial batch processes data is often collected for I batches and J variables measured over the duration of the batch at K_i observations. It is often the case that the duration (number of observations, K_i) varies from batch to batch. However, most statistical methods for analysis of such three-way data require data in the format of an $I \times J \times K$ cube such that each batch has the same number of observations, K. This is the case in many methods of statistical batch process monitoring such as multi-way principal component analysis [13], multi-way partial least squares [14], PARAFAC/Tucker3 models [7] as well as numerous variations on these approaches found in the literature.

Furthermore, it is usually reasonable to suppose that the variation in overall batch duration is just one aspect of a more prevalent time-variation throughout the entirety of each batch. Industrial batch processes often consist of a complex sequence of stages. For example in the case of bacteria fermentation, the living cells multiply, consume nutrients, change food source and so on. The different stages may be reflected in the features (peaks, valleys, slopes etc.) of the variable trajectories. Each stage may occur faster or slower from batch to batch. Even if overall duration is standardised across batches, at similar time-points, different events may be occurring in each batch. Early approaches to processing of batch data did not address this aspect when dealing with uneven batch lengths. For example, all batches were simply cut to the same length, or linearly interpolated to obtain the same number of observations and form the $I \times J \times K$ data cube for further analysis. The first method ignores variation in local batch time, whilst the second method assumes that local batch time is expanded or contracted uniformly throughout the batch, which is unlikely in most processes and so the data will not be synchronised.

Reasons for synchronising batch data are well expressed in [3]. Firstly, if the aim is to apply a latent structure model, then such a model will encounter difficulty in identifying meaningful structure in the time dimension when the data is unsynchronised. This is because data for the same observation number, say k, will be contrasted for different batches and this does not make sense if observation number k corresponds to

* Corresponding author. DTU Compute, Asmussens Alle 322, 2800, Kgs. Lyngby, Denmark.
E-mail address: mpsp@dtu.dk (M. Spooner).

http://dx.doi.org/10.1016/j.chemolab.2017.05.019
Received 3 February 2017; Received in revised form 19 May 2017; Accepted 23 May 2017
Available online 25 May 2017

different events in different batches (e.g. in one batch observation 10 could occur during a lag phase, whilst in another batch observation 10 could occur during a growth phase). Synchronising the data will allow the model to better explain the variation in data for the same event across batches. A second reason for synchronising batch data is to gain a more detailed insight into the dynamics of the process.

A method which takes into account time variation throughout the batch as well as overall duration, has become known as the indicator variable method [2,11]. This method is useful if there exists a suitable indicator variable, which must be monotonically increasing or decreasing throughout each batch. Then the data can be interpolated against even increments of this variable which is used to indicate the progress of the batch or local batch time. However, for many batch processes there may be no suitable variable, or at best one that can only be used for one stage of the process such as in [2], in which case the advantage of this method, i.e. its simplicity, is lost.

Dynamic time warping is a synchronisation method which can align features at every point of each trajectory and does not require an indicator variable. This method originates from the field of speech recognition [17] and was later applied to batch process data by Kassidas et al. [6]. DTW aligns a query trajectory to a reference trajectory by finding the optimum matching of indices of each trajectory, subject to certain constraints, such that the overall distance between the trajectories is minimised. This optimum matching is found through a dynamic programming algorithm. The resulting time warping function contains information on how the local batch times of the trajectories progress. The application of DTW to batch process data has been investigated further by, for example, Ramaker et al. [16] and González-Martínez et al. [5].

An alternative alignment algorithm, Correlation Optimized Warping (COW), was presented by Nielsen et al. [12] for aligning chromatographic data. Tomasi et al. [18] evaluate both COW and DTW performance in the application of chromatographic data alignment and conclude that COW is preferable. With regards to alignment of batch process data (understood as engineering variables measured over time), COW has been further explored by [1], but DTW is the more widely used method in the literature. In this paper the practical issues of the application of DTW to industrial batch data will be discussed.

There are an array of options for tuning the DTW algorithm to a particular alignment problem and two of the key aspects are choice of global constraint and choice of local constraint. Briefly, the global constraint limits the maximum timing difference in the matching between query and reference trajectories found by DTW. There is freedom in how the trajectories are warped as long as the maximum timing difference is not exceeded. In contrast, the local constraint limits the amount of expansion/contraction that DTW applies throughout every point of the trajectories. If DTW is applied without either type of constraint, then the alignment may result in unrealistic, extreme warping. Researchers who have considered the problem of avoiding extreme distortions when aligning batch process data with DTW have focused on the global constraint, with little or no discussion of the local constraint [5]. Recently, Lu et al. [8] presented a method for specifying global constraints so that warping only takes place in selected regions of the trajectories such that key features are not distorted unduly. Again, a local constraint is not used. The primary goal of the present work is to provide a method for avoiding unrealistic warping with DTW whilst achieving a reasonable synchronisation of the key events in the trajectories.

In this paper, besides reviewing the many relevant aspects of DTW as applied to batch process data, we focus especially on the effects of local constraints. We argue that local constraints are preferable to global constraints for limiting unrealistic warping with DTW. The local constraint scheme of Sakoe and Chiba [17] is extended and stronger local constraints than previously seen in the literature are applied. A novel method for selecting the most appropriate local constraint is proposed which is readily generalisable to any batch process. A case study of real data from an industrial bacteria fermentation process is presented and the proposed methods are demonstrated on this data. Our local constraint method is shown to be superior to a global constraint in this case and the results are discussed.

2. Methods

2.1. Basic theory of DTW

In this section the various elements of DTW are introduced. Let $\mathbf{X} \in \mathbb{R}^{K_1 \times J}$ be a query batch of J variables collected at K_1 time points, and let $\mathbf{R} \in \mathbb{R}^{K_2 \times J}$ be a reference batch of J variables collected at K_2 time points. In order to align $\mathbf{X}$ to $\mathbf{R}$ with DTW, the first step is to construct a local distance matrix, $\mathbf{C} \in \mathbb{R}^{K_1 \times K_2}$ where $\mathbf{C}_{k_1,k_2}$ is the distance between observation k_1 in $\mathbf{X}$ and observation k_2 in $\mathbf{R}$ (denoted by the row vectors $\mathbf{X}_{k_1\bullet}$ and $\mathbf{R}_{k_2\bullet}$ respectively). Usually, the squared Euclidean distance is used

$$C_{k_1,k_2} = [\mathbf{X}_{k_1\bullet} - \mathbf{R}_{k_2\bullet}]\mathbf{W}[\mathbf{X}_{k_1\bullet} - \mathbf{R}_{k_2\bullet}]^T \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{J \times J}$ is a diagonal matrix with $W_{j,j}$ being the weight for the j^{th} variable. Variables with a large weight will contribute more to the distance and will therefore have greater influence on the alignment. The choice of these weights will be revisited in section 2.4. An advantage of using the squared Euclidean distance, rather than the Euclidean distance, is that the weights of each variable are preserved in the computed local distance.

The DTW algorithm considers a warping function $f(t)$, $t = 1, \ldots, T$

$$f(t) = (f_X(t), f_R(t)) \tag{2}$$

$$f_X(t) \in \{1, \ldots, K_1\} \tag{3}$$

$$f_R(t) \in \{1, \ldots, K_2\} \tag{4}$$

Thus, f maps the indices of $\mathbf{X}$ and $\mathbf{R}$ to a common time axis and defines a path through the cost-distance matrix consisting of T steps. After applying a given warping function, f, to $\mathbf{X}$ and $\mathbf{R}$ then the accumulated distance between the resulting warped trajectories is given by

$$D_f(\mathbf{X}, \mathbf{R}) = \frac{1}{M_f} \sum_{t=1}^{T} m_f(t) \mathbf{C}_{f(t)} \tag{5}$$

The term $m_f(t)$ denotes the weight assigned to the t^{th} step in the path, with M_f being the normalisation constant $\sum_{t=1}^{T} m_f(t)$.

Several basic constraints are usually imposed on the warping function f. In the case of global DTW alignment (synchronising all of $\mathbf{X}$ to all of $\mathbf{R}$), the following boundary constraints must apply:

$$f(1) = (1, 1) \text{ and } f(T) = (K_1, K_2) \tag{6}$$

The warping function must preserve the monotonicity of the original time dimension:

$$f_X(t) \leq f_X(t+1) \text{ and } f_R(t) \leq f_R(t+1) \tag{7}$$

Finally, the time warping function should be continuous:

$$f_X(t+1) - f_X(t) \leq 1 \text{ and } f_R(t+1) - f_R(t) \leq 1 \tag{8}$$

The goal of DTW is to identify warping function f satisfying these constraints, which minimises the accumulated distance $D_f(\mathbf{X}, \mathbf{R})$. The DTW algorithm achieves this through dynamic programming in $O(K_1 \cdot K_2)$ time. The details of this solution may be found in [10]. DTW thus produces a warping function $f = \text{argmin}_f(D_f(\mathbf{X}, \mathbf{R}))$ denoting how $\mathbf{X}$ and $\mathbf{R}$ can be warped to make them most similar, as well as the accumulated distance $D = \min_f(D_f(\mathbf{X}, \mathbf{R}))$ which quantifies the similarity between $\mathbf{X}$ and $\mathbf{R}$ after warping.

In order for the minimisation of the distance in Eq. (5) to be solved through dynamic programming it is necessary that the normalization constant (M_f) be independent of the path, f. This limits the possibilities

for the choice of step weights. The most well known step weights for symmetric DTW are those devised in [17] where diagonal steps are given a weight of 2, and horizontal and vertical steps are each given weights of 1. In this case, the normalization factor will be $M_f = K_1 + K_2$, independently of the warping path. An implication of this weighting scheme is that the minimisation of Eq. (5) does not favour warping paths of shorter length (small T). For the computed DTW distances to be comparable for different batches having different durations we suggest using this approach for the step weights.

Under the above constraints DTW operates symmetrically, warping both the reference and the query to obtain the closest alignment. The usual approach with batch process data is to apply a symmetric form of DTW between each batch $\mathbf{X}^{(i)}$ and the reference batch $\mathbf{R}$. However, the resulting warping functions will still have varying lengths. A second step is necessary to obtain warped batches of equal length to the reference batch. This step entails aggregating observations of the query batch whenever the warping function aligns several observations of the query to a single observation of the reference. This aggregation may be performed by taking the mean [6] or the median [8]. For example, in Fig. 1, the 3^{rd} observation of R is aligned to both observations 1 and 2 of X. Therefore, the 3^{rd} value of the warped X is taken to be the mean of X_1 and X_2. We choose to take the mean so that no information from the query trajectory is removed completely.

The basic form of dynamic time warping outlined in this section may be used to process batch data so each batch has the same length as the reference batch. However, the method will often result in extreme warpings and it is therefore preferable to impose additional constraints on the algorithm. In the following section we propose a novel method for selecting a local constraint so that unrealistic warping is avoided.

2.2. Local constraint

The basic DTW method allows a lot of flexibility for the warping path through the local distance matrix. For example, unlimited vertical and horizontal paths are permissible. These entail that either a single observation in the query is expanded to match a section of the reference or that a section of the query is compressed to match a single observation of the reference. In many contexts, extreme warpings may be unrealistic. To limit such warping various constraints have been proposed on the slope of the path through the cost-distance matrix. This family of constraints, known as local constraints or slope constraints are expressed as an allowable step pattern showing which predecessor points are allowable paths for a given point in the local distance matrix. Sakoe and Chiba [17] devised step patterns based on the maximum allowable number of horizontal or vertical steps that may be taken before a diagonal step must be taken and classified them according to the ratio of these two numbers, the parameter P. They proposed 4 step patterns for $P = 0, 0.5, 1$ and 2, as shown in Fig. 2. We have found it necessary to extend this scheme to obtain arbitrarily strong constraints of $P = 3, 4, 5, \ldots,$ etc., as shown in Fig. 2. The original four step patterns of [17], as well as our extension of this scheme are described in more detail as follows:

- $P = 0$: Infinitely many steps in either the horizontal or vertical directions are permitted regardless of previous steps, i.e. there is no constraint on the slope (equivalent to the basic DTW method).
- $P = \frac{1}{2}$: up to 2 steps in either the horizontal or vertical directions are permitted provided they are preceded by 1 diagonal step.
- $P = 1$: each horizontal or vertical step must be preceded by at least 1 diagonal step
- $P = 2$: each horizontal or vertical step must be preceded by at least 2 diagonal steps
- $P = p$: each horizontal or vertical step must be preceded by at least p diagonal steps

For the mathematical expressions and how these step patterns are incorporated into the dynamic programming algorithm see the original work [17]. The choice of local constraint is in effect a trade-off between trajectory synchronisation, and time distortion. If the constraint is too strong, then the trajectories may not be adequately synchronised, whilst if the constraint is too weak, the trajectories will be closely synchronised but at the cost of unrealistic warpings in the time dimension (extreme contractions or expansions). A method for evaluating this trade-off is the primary goal of this work.

It is not realistic that a choice of local constraint could be made based on process knowledge. Kassidas et al. [6] use DTW without a local constraint, and other authors appear to take the same approach, in some cases using instead a global constraint to limit extreme warpings (to be discussed in section 2.3). To our knowledge, the effects of different local constraints on DTW alignment of batch process data have not been previously investigated. We will demonstrate the advantages of exploiting the local constraint possibilities in DTW to obtain realistic data alignment.

In order to select the best local constraint, we first consider the two extremes: the most lenient constraint, $P = 0$, and the most restrictive constraint which we call $P = p_{max}$. These define the feasible region for choosing the local constraint. The value of p_{max} depends on the length of the query batch K_i relative to the length of the reference batch K_{Ref}. Consider the $K_i \times K_{Ref}$ local distance matrix $\mathbf{C}$. As P increases, the warping function is constrained to a path closer to the diagonal. If $K_i \neq K_{Ref}$ then $\mathbf{C}$ is non-square and a path from $(1,1)$ may consist of at most $\min(K_i, K_{Ref})$ diagonal steps before reaching the edge of the matrix (the first step "onto" (1,1) is conventionally considered a diagonal step). For the path to reach the corner at (K_i, K_{Ref}), requires $|K_i - K_{Ref}|$ additional horizontal or vertical steps. This is illustrated in Fig. 3. In the plot of the 5×8 matrix, there may be at most 5 diagonal steps requiring 3 horizontal steps to form a path from $(1,1)$ to $(5,8)$. This is the case regardless of how the 5 diagonal and 3 vertical steps are distributed. The ratio of the maximum number of diagonal steps possible to the associated number of

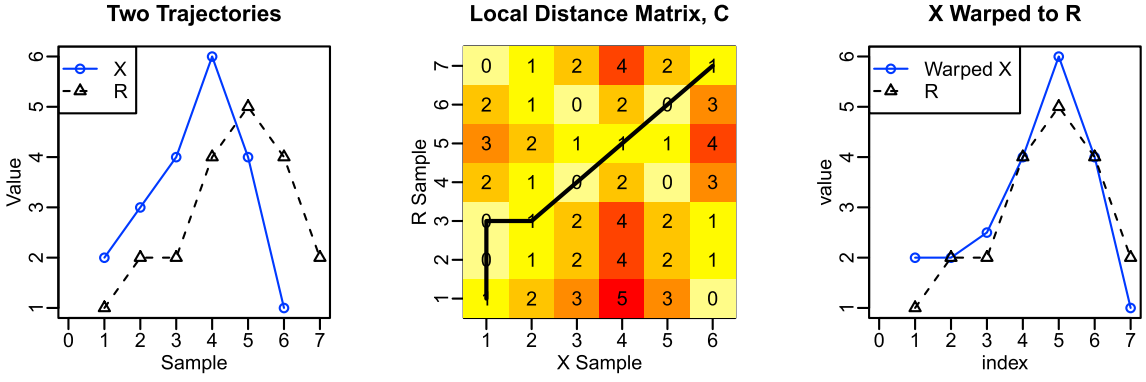

Fig. 1. Two simple trajectories (left), the local distance matrix and the optimum warping path (black line) found by DTW (center), and the resulting warped trajectory X.

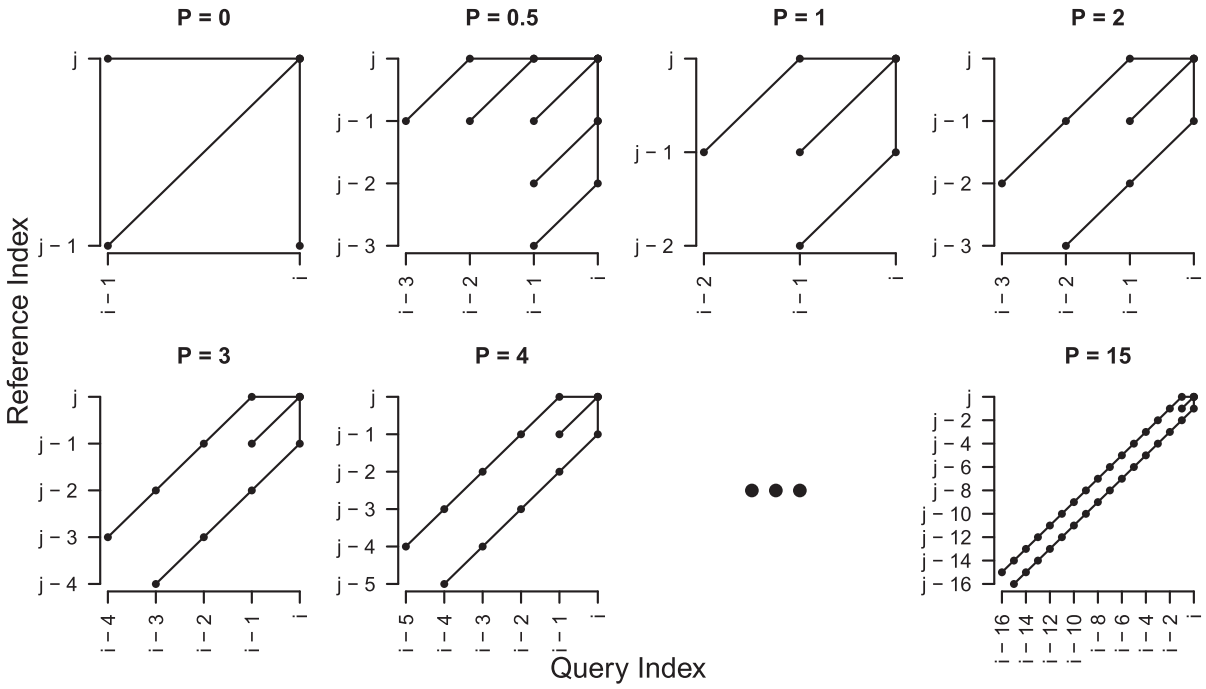

Fig. 2. The four symmetric step patterns proposed by [17] (top row) and our extension of this scheme for arbitrarily large integer P (bottom).

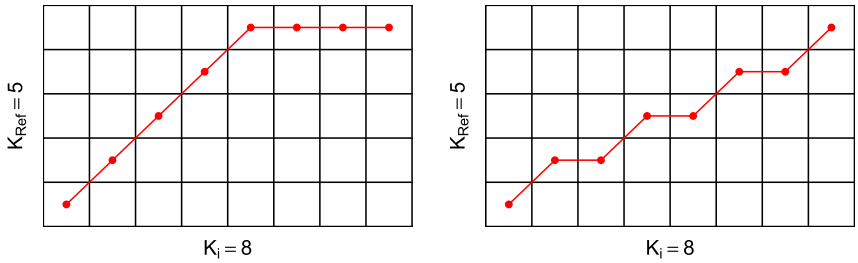

Fig. 3. Two possible warping paths for $K_i = 8$ and $K_{ref} = 5$, showing that the greatest number of diagonal steps possible is 5 and this requires 3 horizontal steps to complete a valid path.

horizontal or vertical steps required provides the greatest P value which still allows a possible warping path to be found. As we cannot allow fractions of a diagonal step, the floor of this ratio is taken to obtain the strictest local constraint for a particular query batch. Once this is done for every query batch in the dataset, p_{max} is taken to be the smallest of the values, so that the same $P = p_{max}$ local constraint can be used on all the batches.

$$p_{max} = \min_{1<i<I} \left\lfloor \frac{\min(K_i, K_{ref})}{|K_i - K_{ref}|} \right\rfloor \tag{9}$$

The $P = p_{max}$ local constraint is the strictest constraint under which every query batch in the dataset can be aligned to the reference batch using DTW, and corresponds to the DTW alignment that is most comparable to linear interpolation of each query batch at K_{ref} sampling points, i.e., minimal warping takes place so events within the batch are not well synchronised. Conversely, the $P = 0$ constraint allows unlimited warping, and the batches are aligned as closely as possible. The problem is to select a local constraint between these extremes that can produce the closest alignment without extreme warpings. To evaluate this trade-off, the following measures are first defined to quantify the two effects.

With regards to degree of variable synchronisation, recall that the DTW distance, Eq. (5), is precisely a measure of the similarity between the query trajectories and the reference trajectories, after DTW synchronisation by the warping function f. In addition, this quantity is returned by the DTW algorithm and so requires no additional computation. To quantify the degree of synchronisation for a particular local constraint $P = p$, align each batch under this constraint. Let D_i be the DTW distance calculated for the i^{th} batch. Define the synchronisation value, $\overline{D}$, as the mean of the DTW distances across the I batches

$$\overline{D} = \sum_{i=1}^{I} D_i / I \tag{10}$$

A small synchronisation value indicates a close alignment of the batches.

To quantify the degree of time distortion, we note that warping only occurs at horizontal and vertical transitions in the warping path whilst diagonal transitions correspond to no change. Therefore, we define the time distortion measure $\overline{N}_{HV}$ as the mean number of horizontal or vertical steps per batch alignment. Denote the warping function for the i^{th} batch by $f^{(i)}(t) = (f_X^{(i)}(t), f_R^{(i)}(t))$, $t = 1, \ldots, T^{(i)}$. A horizontal or vertical step entails that either $f_X^{(i)}(t)$ remains the same for a step, or that $f_R^{(i)}(t)$ remains the same, and the distortion measure is defined as

$$\overline{N}_{HV} = (1/I) \sum_{i=1}^{I} \sum_{t=1}^{T^{(i)}-1} \mathbb{1}_0\left(\left(f_X^{(i)}(t+1) - f_X^{(i)}(t)\right) \cdot \left(f_R^{(i)}(t+1) - f_R^{(i)}(t)\right)\right) \tag{11}$$

where $1_0(x) = \mathbb{1}$ for $x = 0$ and 0 otherwise.

In order to combine $\overline{D}$ and $\overline{N}_{HV}$ into a single "alignment score" for

each local constraint they should be scaled to make them comparable (the range spanned by $\overline{N}_{HV}$ for different local constraints will differ in magnitude to the range spanned by $\overline{D}$). This is done by subtracting the minimum and dividing by the range so each measure varies from 0 to 1. The combined alignment score may then be calculated as the euclidean distance of $(\overline{N}_{HV(\text{scaled})}, \overline{D}_{(\text{scaled})})$ to the origin $(0, 0)$:

$$\text{Alignment Score} = \sqrt{\overline{N}^2_{HV(\text{scaled})} + \overline{D}^2_{(\text{scaled})}} \tag{12}$$

The local constraint which has the smallest alignment score is chosen as the most suitable and the aligned data under this constraint may be used for further analysis. This constraint will synchronise the key events in the process without unrealistic time distortions.

With regards to the variable weights **W**, the above procedure requires that the same variable weights are used for each alignment so that $\overline{D}$ is comparable for different local constraints. We suggest that **W** is calculated prior to local constraint selection, using the method in [16] with the basic DTW algorithm ($P = 0$ local constraint). These variable weights are then used thereafter. In this way, the variable weights are based on the potential warping information, under the most flexible local constraint.

Our proposed method for selecting the local constraint is summarised as follows:

- Determine variable weights **W** by applying the method from [16] with $P = 0$ local constraint.
- Select the reference batch (see section 2.4)
- Calculate the value of p_{max} (Eq. (9))
- For p in $0, \frac{1}{2}, 1, 2, \ldots, p_{max}$, align the batches to the reference batch using DTW with $P = p$ local constraint
- For each different local constraint alignment calculate the synchronisation value $\overline{D}$ and the distortion measure $\overline{N}_{HV}$
- For each local constraint calculate the alignment score by adding the scaled synchronisation value and scaled distortion measure.
- Select that local constraint which results in the smallest synchronisation score for the final alignment of the data

2.3. Global constraint

A global constraint may be specified such that the warping function may not enter certain regions of the local distance matrix. A simple global constraint is the band constraint [17], where the time difference between the warped query and reference series may not exceed some value b: $|f_X(t) - f_R(t)| \leq b$. This means that the warping function is confined to a band of width b along the main diagonal of the distance matrix. The band constraint is problematic when the query and reference are of different lengths, as the main diagonal of a non square matrix does not go from corner to corner.

Alternatively, arbitrary regions of the distance matrix may be selected as off-limits to the warping function. González-Martínez et al. [5] define global constraints for use in their on-line DTW implementation as an empirical envelope derived from historical warping functions in an off-line analysis. Lu et al. [8] use global constraints to avoid warping regions in the data that contain important features. This procedure requires identifying features and selecting which features should not be warped, requiring much analysis and interpretation. It does not seem physically realistic that local batch time would follow the reference time exactly in certain predefined regions, but warp freely in other regions.

Global constraints entail that there is a maximum possible time difference between the warped series. As long as this maximum is not exceeded, extreme compressions and expansions may still occur. This abrupt cut-off in maximum allowable time difference does not seem realistic if the warping function is interpreted physically as the local batch time. It seems unlikely that local batch time could naturally adhere to such a cut-off. In contrast, the local constraint may be physically interpreted as a limit to the rate of change of local batch time relative to the reference, which is more realistic.

Finally, it has been shown that the local constraints for $P > 0$ implicitly limit the region of possible warping paths to the so-called Itakura parallelogram [9], the sides of which have slopes $1/S$ and S where $S = (p+1)/p$ for local constraint $P = p$. For example, local constraint $P = 1$ limits the warping path to the Itakura parallelogram with sides of slope $1/2$ and 2 (Fig. 4). Therefore, we do not see an advantage to introducing additional global constraints if a local constraint is used.

2.4. Other DTW considerations

Our main interest in this work is the use of local and global constraints to control the performance of DTW as outlined in sections 2.2 and 2.3. In this section we consider several issues which are relevant for applying DTW to alignment of batch process data but which are not related to the local or global constraints.

During DTW, each query batch is aligned to the same reference batch in order to obtain aligned batches of the same length as the reference. The choice of reference batch is of importance as the data will be re-expressed in terms of its relation to the reference batch. To aid interpretation of the DTW results, it is preferable that the reference batch be a typical batch found under normal operating conditions. Following [5,6,16], we select the batch with a duration closest to the median duration of the dataset as the reference. It should also be noted that it is important to carry out additional checks that the chosen batch represents normal operating conditions.

Prior to DTW alignment, the data should be scaled so that the local distance measure in Eq. (1) is not influenced by the different engineering units of the variables. Following [6] this is done by dividing each variable by its average range across batches. Ramaker et al. [16] hold that scaling is not necessary for spectroscopic data where each variable has the same unit and similar range. Mean standard deviation or interquartile range could also be used as a scaling factor.

After the data has been scaled, the appropriate weight of each variable, **W** in Eq. (1) must be determined. Variables with a large relative weight will contribute more to the distance and will thereby have greater influence on the warping function and the final alignment. Therefore appropriate weighting will lead to better synchronisation. There have been two proposed methods for defining variable weights. Both are iterative methods. The first method, from Kassidas et al. [6], aims to give large weights to variables that are consistent from batch to batch. These are the variables which can be most closely synchronised by DTW. However, as noted by [16], the method assigns large weights to flat, featureless variables which are actually poor indicators for alignment. For this reason, we choose to use the second method, proposed by Ramaker et al. [16], which assigns large weights to variables which produce a steep valley in the surface plot of the local distance matrix. In section 3.3 we apply both the method from [6] and from [16] and find that the latter results in more intuitive weights based on visual inspection of the data.

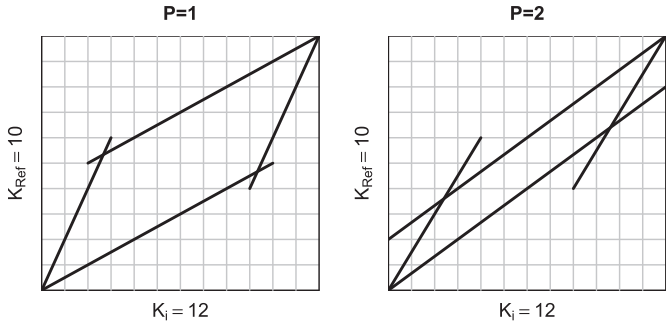

Fig. 4. Itakura parallelograms for the $P = 1$ local constraint, having sides that slope $1/2$ and 2, and the $P = 2$ local constraint, having slides that slope $2/3$ and $3/2$

Finally, we consider the case where batches are stopped at varying degrees of completeness. This may occur in processes where there is no reliable way to measure the completeness of a batch exactly. In this case, some batches will contain process events which are not present in other batches. One approach to aligning such data is to use partial DTW alignment in which the endpoint constraints of Eq. (6) are relaxed to allow for all of $\mathbf{X}$ to be aligned to a section of $\mathbf{R}$. The DTW solution may be interpreted as performing K_{ref} DTW alignments, using $\mathbf{R}(1:k,:)$ as the reference for $k = 1, \ldots, K_{ref}$. Then the alignment which gives the minimum value of $D_f(\mathbf{X}, \mathbf{R}(1:k,:))$ is adopted and indicates that the batch completion of $\mathbf{X}$ corresponds to the first k observations of $\mathbf{R}$. Similarly, a partial alignment can account for uncertainty concerning the start time of the batches, or of both start time and end time. Of course, it is necessary that a reference batch which is "more complete" than all the query batches is chosen.

A disadvantage of conducting a partial alignment is that after synchronisation, the batches will still have varying lengths. Secondly, in omitting the start point or end point constraint, the algorithm is given a great deal more flexibility which may result in inappropriate synchronisation. The constraint of fixed endpoints is a great advantage, (if the start and end endpoints do indeed represent the same events in all batches) as they ensure that at least these points are appropriately aligned, leading to a greater confidence in the alignment of the intervening points. With an open ended alignment, there is uncertainty whether the reference section selected by DTW truly represents the events in the query, or whether a chance match was found to a section containing different events, because of atypical behaviour in the batch.

We suggest dealing with varying completion amounts as follows:

1. Using process knowledge, identify a feature in one or more of the variable trajectories, near the end of the batch, which is present in all batches and which may be assumed to represent the same event across batches. Truncate each batch at the time of this end-point event.
2. Select local constraint using global DTW and synchronise data using the optimum local constraint.
3. For subsequent analysis, each synchronised batch matrix may be appended with the very last observation from the unsynchronised batch matrix in order to represent the information on what took place after the endpoint event for each batch. As a single observation is used for this, all the synchronised batches will still have the same length.

A similar method could be used if the start-point of the batches varies. If an end or start event cannot be identified, then partial DTW may be of value.

3. Results and discussion

The methods described in section 2 were applied to real data using the statistical software R [15]. To implement DTW with various local constraints, global constraints, variable weights etc. the DTW R package was used [4]. The Sakoe and Chiba [17] local constraints for $P = 0, 0.5, 1$ and 2 are built in to this package, and the additional constraints described in section 2.2 for integer $P > 2$ may be readily defined by the user.

3.1. The data

Data was provided by the company Chr. Hansen for an industrial bacteria fermentation process used to produce a multi-strain bacteria culture for use in the dairy industry. Process data was obtained for $I = 23$ batches consisting of $J = 6$ variables measured throughout the batch duration. The raw data was processed to obtain measurements at evenly spaced time intervals resulting in $403 \leq K_i \leq 532$ observations. This processed data is shown in Fig. 5. A batch begins with the introduction of bacteria cells to a fermentation vessel which has been pre-filled with growth media. As the bacteria cells grow and multiply they produce acid which lowers the pH inside the fermenter. Once the pH reaches a predefined set point roughly half-way through the process, a controller is

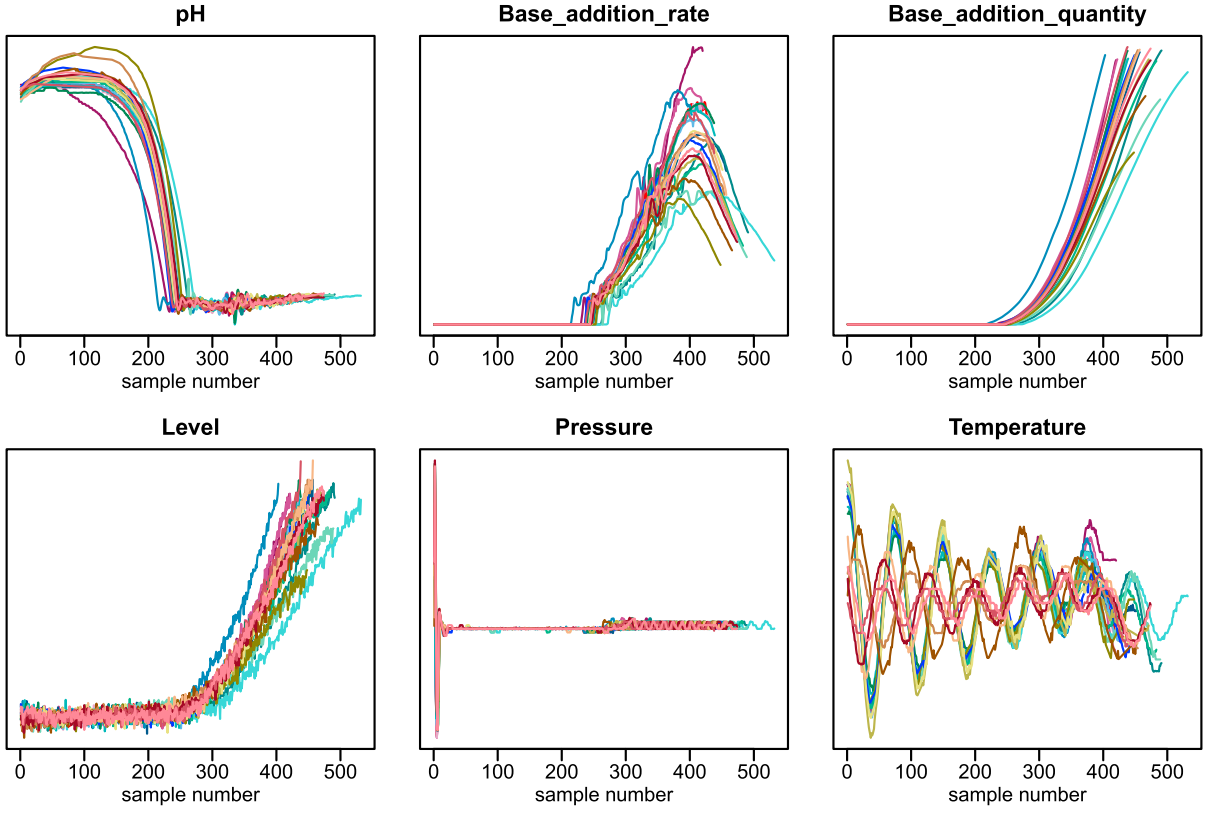

Fig. 5. The 23 batches in the dataset obtained from the batch fermentation process. The scale of the ordinate axis is omitted for confidentiality reasons.

activated which is designed to maintain the pH at this set point for the remainder of the batch. The controller does this by manipulating the amount of base (ammonia) added to the fermenter in response to changes in measured pH. The rate of base addition is measured, as well as the total quantity of base added to the fermenter. The level inside the tank rises as base is added. Temperature is kept within a narrow range. The point at which to stop the batch is decided by the operator based on process knowledge (judgement on the degree of bacteria growth based on amount or current rate of base addition) and logistics (e.g. availability of equipment downstream). In this process, the pH and base addition rate/quantity can be assumed to be direct indicators of the biological state of the process. Level is closely correlated to base addition. Pressure is kept constant. However, it was determined that the temperature variable is not a reflection of the progress of bacteria growth, but rather depends on the physics of maintaining the temperature of a large body of fluid, and the mixing of this fluid in relation to the sensor placement. Therefore, the temperature variable is not a direct indicator of the biological progress of the bacteria fermentation. Based on this prior knowledge, the temperature variable should not influence the alignment of the data.

Time variation between batches is clearly visible in the process. As well as the overall batch duration which varies between 403 and 523 observations, there is also variation in the timing of the events that take place during the process. The lag phase (the time before the bacteria multiply enough to decrease the pH) varies from around 50, to around 200 observations. The time at which the pH reaches the set-point varies from 200 to 270 observations. The time at which base addition rate peaks varies between 350 and 450. In addition to these more visible events in the biological process, we may suppose that there are many other features in the variable trajectories that reflect other events in the process, and these also vary in time from batch to batch.

3.2. Processing data and selecting reference batch

The batches have been stopped at varying degrees of completion, as illustrated by how complete a curve is traced by the base addition rate trajectory after it has peaked. The last observation in each batch does not correspond to the same event, and so should not be synchronised to each other. The methods proposed in this paper determine the most appropriate constraints to use for DTW, by applying the global alignment version of DTW. Therefore, in cases like this, some endpoint that is common to all batches must be inferred in order to truncate the data so that all batches stop at the same event. The time of reaching peak base addition rate was selected as the imposed endpoint. This time was determined by fitting a cubic polynomial to the last section of the base addition rate trajectory, and the batches were truncated at this point.

Next, the reference batch was selected to be that batch with duration closest to the median duration, having confirmed with operators that the selected batch had desirable properties. The data was scaled by dividing each variable by its mean range across all batches and centred by subtracting the minimum (across all batches).

The truncated data and reference batch are shown in Fig. 6.

3.3. Determining variable weights

The variable weights are determined by applying the method from [16] with unconstrained DTW ($P = 0$) and initial weights of 1 for all variables (Temperature excluded as previously discussed). After seven iterations the weights converged ($\varepsilon < 0.002$) to the values shown in Fig. 7. It is reasonable that pH is given the greatest weight as it is the main indicator of the state of the process. Base addition quantity and rate are also important indicators, but are only relevant for the second half of the process so receive less weight. Pressure and Level contain very little information on the state of the process and so it is appropriate that they are given small weights.

The weights that result from the method in [6] are also shown in Fig. 7. This method required 19 iterations before $\varepsilon < 0.002$ and it weighted pressure and level greater than base addition rate and base addition quantity. This contradicts our prior interpretation that level and

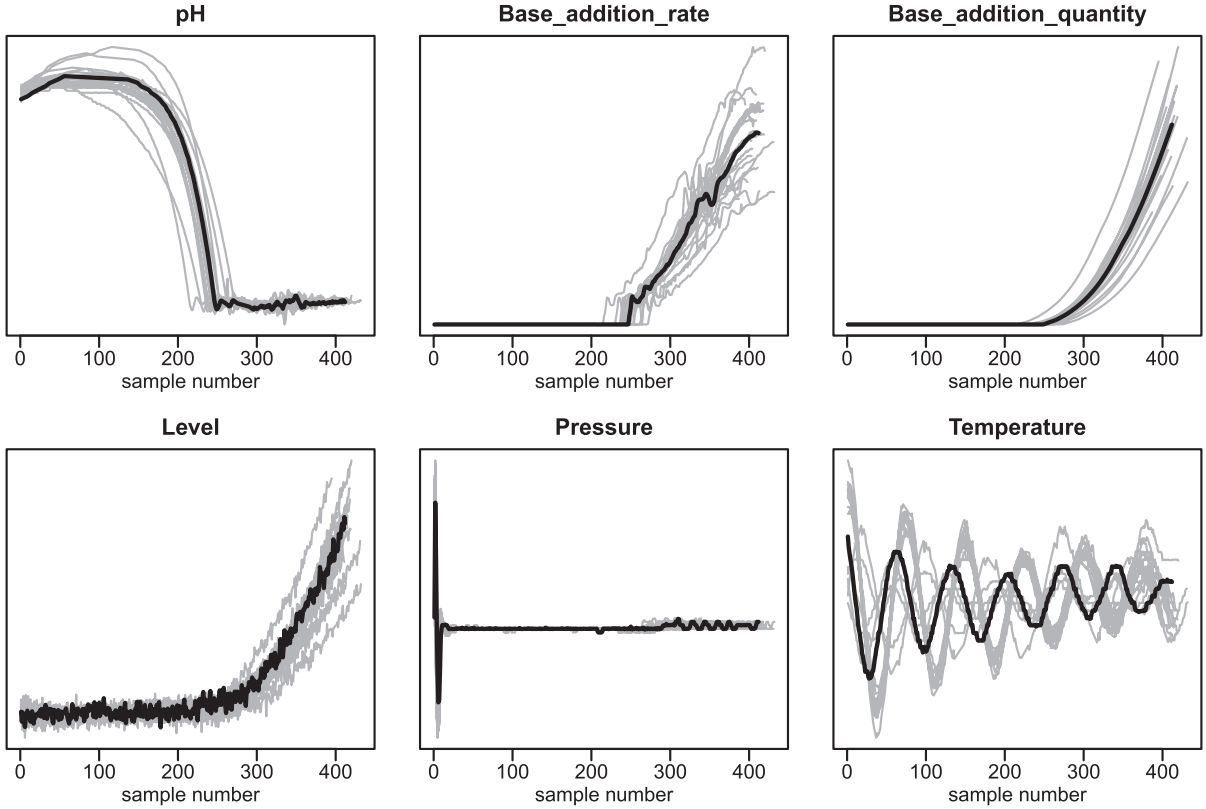

Fig. 6. The truncated data with reference batch in black and other batches in grey.

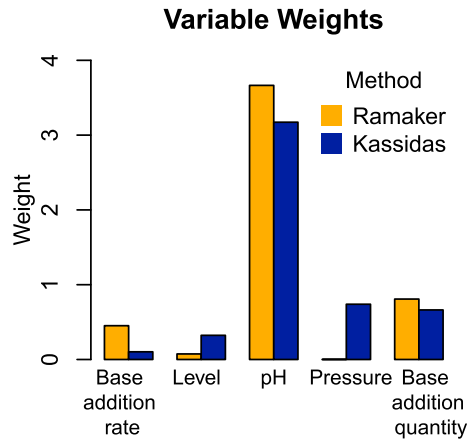

Fig. 7. Variable weights resulting from the methods of Ramaker et al. [16] and Kassidas et al. [6].

pressure are the least informative variables for alignment and supports our choice of the Ramaker method for determining variable weights.

3.4. Selecting local constraint

After truncating the batches to a common end point, they range in length from 387 to 432 observations. The reference batch is of length 412 observations. From Eq. (9), the strictest constraint which can be applied to all batches is $P_{max} = 15$. Therefore a choice must be made amongst the constraints $P \in [0,\ 0.5,\ 1,\ \ldots,\ 15]$. DTW alignments were applied for each of these 17 local constraint choices. The resulting time warping functions and aligned trajectories of the pH variable are shown for $P = 0, 3$ and 15 in Fig. 8.

Fig. 8 clearly shows that in the traditional DTW method where $P = 0$ and there is no constraint on the slope of the warping function, there are many extreme warpings in the time warping functions: long sections of the query batch are compressed to an instant or a single observation in the query batch may be expanded to cover long sections of the reference batch. The variable trajectories are aligned too aggressively by traditional DTW. The strongest constraint, $P = 15$ approaches similar results to a linear interpolation. The aligned batches have the same number of observations as the reference, but events during each batch are not synchronised. The intermediate constraint shown in Fig. 8 ($P = 3$), results in much less extreme warping than for $P = 0$, whilst appearing to succeed in synchronising the key events in the pH variable.

For each local constraint alignment, $\overline{N}_{HV}$ (degree of time distortion) and $\overline{D}$ (synchronisation value) was calculated. The statistics are scaled and combined to form the alignment score. The constraint resulting in the smallest alignment score was the $P = 3$ constraint, although $P = 2$ has a score very close to it. This procedure is shown in Fig. 9.

With no local constraint ($P = 0$, i.e. traditional DTW), the closest variable synchronisation is obtained, but also the greatest time distortion, $\overline{N}_{HV} \approx 500$. As the local constraint is strengthened up to $P = 3$, time distortion is reduced significantly to $N_{HV} < 100$, without much deterioration in variable synchronisation (small increase in $\overline{D}$). Further strengthening of the local constraint up to $P = 15$ does not lead to substantial reduction in time distortion but results in rapid deterioration in variable synchronisation. This relationship results in an "elbow" in the plotted values which could form the basis for choosing P without the need for scaling and score calculation steps. The "elbow" indicates the choice of P at which strengthening the constraint further will result in substantially poorer alignment with little reduction in time distortion. This heuristic may be summarised as follows:

- Plot $\overline{D}$ against $\overline{N}_{HV}$
- Locate the bend in the plot and select the smallest local constraint closest to the bend

This approach would lead us to select $P = 3$ or $P = 2$. Although the

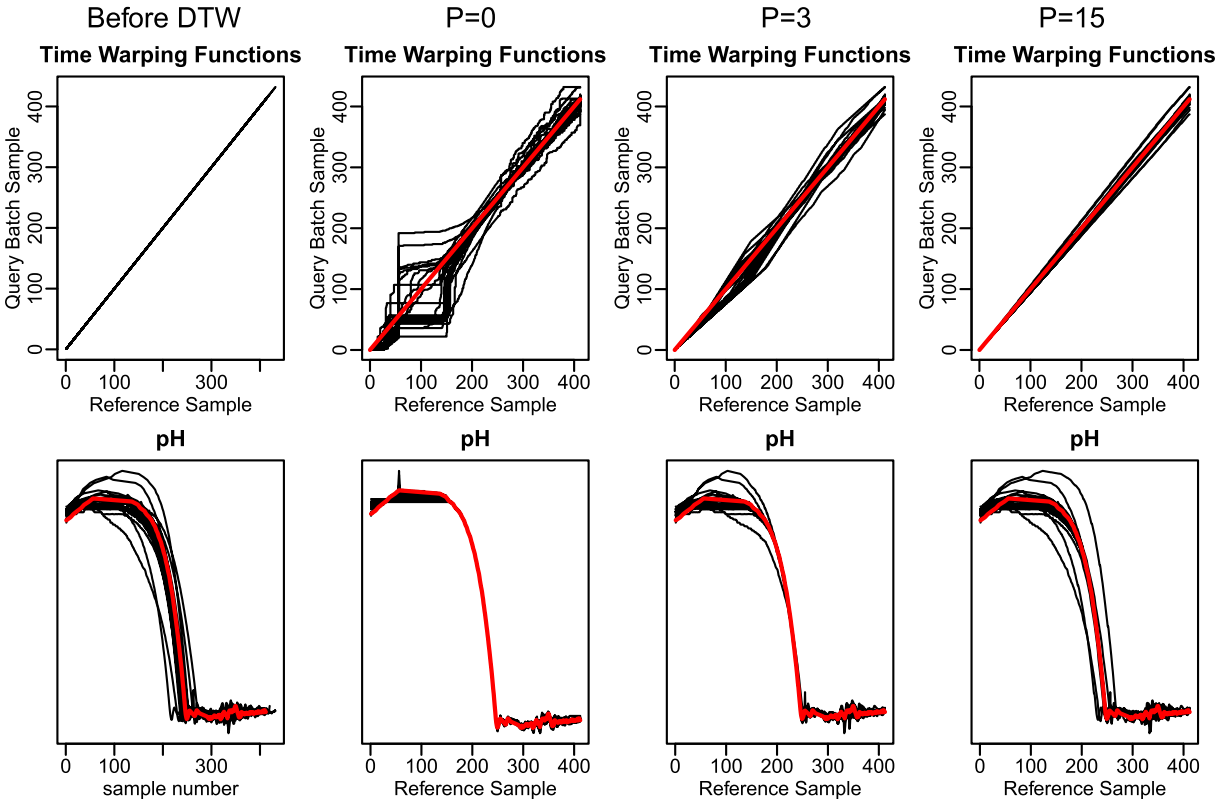

Fig. 8. The warping functions (top) and the aligned trajectories for the pH variable (bottom) are shown from DTW alignment with local constraints of $P = 0, 3$ and 15 (left to right).

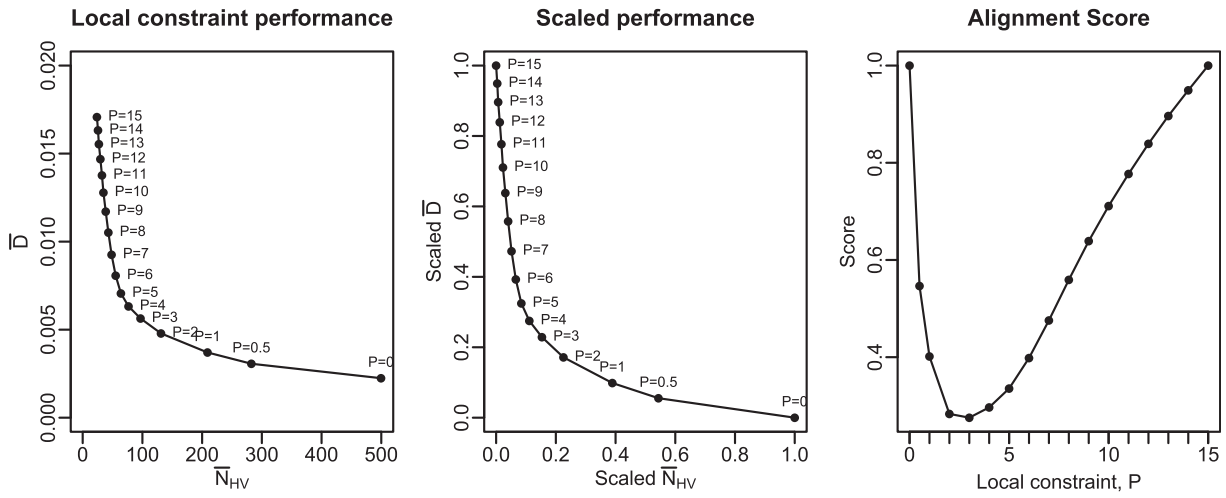

Fig. 9. Choosing the local constraint parameter, P, based on the alignment performance according to $\overline{N}_{HV}$ and $\overline{D}$ (left), their scaled counterparts (center) and the alignment score (right).

method is more subjective, in this instance it leads to the same conclusion as the score method.

In summary, the above off-line analysis has determined the variable weights and the appropriate local constraint for DTW alignment of this data. Using these parameters, the data may then be aligned for further analysis by multi-way methods. In addition, these parameter settings can be used on new batch data from the process, or even in applying DTW in an on-line application.

3.5. Comparison to DTW with global constraint

For comparison, the effect of using a global constraint was investigated by applying the "slanted band" global constraint from [4]. This global constraint limits the warping function to a band that is b columns wide and is centred on the line joining start-corner and end-corner of the local distance matrix. The data was aligned with the slanted band global constraint for bandwidths of $b = 5, 10, 20, 40$ and 80 without local constraint ($P = 0$ for all b). For each bandwidth, time distortion value $\overline{N}_{HV}$, as well as variable synchronisation value $\overline{D}$ was calculated. The resulting values are compared to the previous results from varying the local constraint in Fig. 10. The global constraint is not as effective as the local constraint in reducing time distortion, and strengthening the global constraint (decreasing b) leads to rapid deterioration in variable synchronisation (increase in $\overline{D}$) without much reduction in time distortion. We conclude that the local constraint approach is superior to the slanted band global constraint at limiting unrealistic warping without serious loss in variable synchronisation quality.

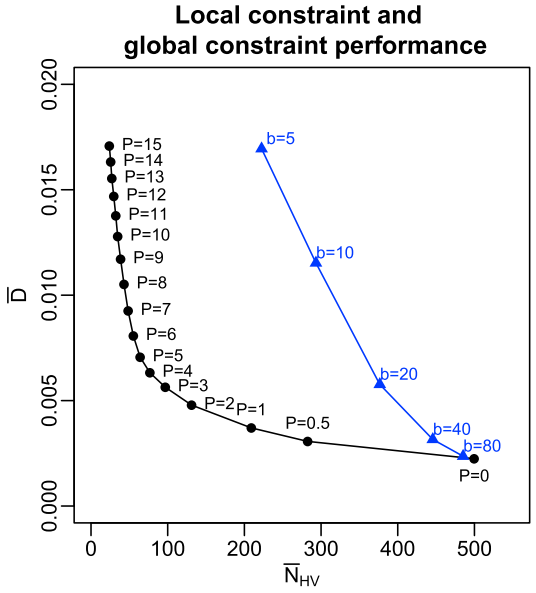

Fig. 10. $\overline{N}_{HV}$ and $\overline{D}$ are shown for varying local constraint (black dots) and varying width of the slanted band global constraint (blue triangles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Conclusions

DTW has been widely used for the alignment of batch process data, though there is not a single accepted approach to how the algorithm should be applied. We have investigated the use of local constraints, an area which has received little attention previously in this application. We have shown how the local constraint may be used to avoid unrealistic warping in the aligned data and presented a novel method for selecting the most appropriate local constraint for an alignment problem. This method optimises the trade-off between variable synchronisation and time distortion. We maintain that the problem of unrealistic warping in DTW should be addressed using the proposed local constraint method, rather than a global constraint approach for two main reasons. Firstly, the global constraint does not avoid extreme contractions or expansions less than some upper limit, whilst the local constraint method limits extreme warping more pervasively. Secondly, the local constraint method we propose is very generalizable to other datasets, whilst many global constraint methods require much tailoring to the dataset at hand. By adopting the proposed method to select an appropriate local constraint, batch process data may be aligned realistically, allowing the warping functions of the aligned batches to be interpreted as progress signatures of the batches.

Acknowledgements

The project received financial support from Innovation Fund Denmark through the BIOPRO2 strategic research center (Grant number 4105-00020B). We thank Chr. Hansen A/S for providing access to data and expertise on the production process.

References

[1] M. Fransson, S. Folestad, Real-time alignment of batch process data using cow for on-line process monitoring, Chemom. Intell. Lab. Syst. 84 (1) (2006) 56–61.
[2] S. García-Muñoz, T. Kourti, J.F. MacGregor, A.G. Mateos, G. Murphy, Troubleshooting of an industrial batch process using multivariate methods, Ind. Eng. Chem. Res. 42 (15) (2003) 3592–3601.

[3] S. García-Muñoz, M. Polizzi, A. Prpich, C. Strain, A. Lalonde, V. Negron, Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling, J. Process Control 21 (10) (2011) 1370–1377.
[4] T. Giorgino, et al., Computing and visualizing dynamic time warping alignments in r: the dtw package, J. Stat. Softw. 31 (7) (2009) 1–24.
[5] J.M. González-Martínez, A. Ferrer, J.A. Westerhuis, Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping, Chemom. Intell. Lab. Syst. 105 (2) (2011) 195–206.
[6] A. Kassidas, J.F. MacGregor, P.A. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE J. 44 (4) (1998) 864–875.
[7] D. Louwerse, A. Smilde, Multivariate statistical process control of batch processes based on three-way models, Chem. Eng. Sci. 55 (7) (2000) 1225–1235.
[8] B. Lu, S. Xu, J. Stuber, T.F. Edgar, Constrained selective dynamic time warping of trajectories in three dimensional batch data, Chemom. Intell. Lab. Syst. 159 (2016) 138–150.
[9] M. Müller, Information Retrieval for Music and Motion, vol. 2, Springer, 2007.
[10] C. Myers, L. Rabiner, A. Rosenberg, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, IEEE Trans. Acoust. Speech, Signal Process. 28 (6) (1980) 623–635.
[11] D. Neogi, C.E. Schlags, Multivariate statistical analysis of an emulsion batch process, Ind. Eng. Chem. Res. 37 (10) (1998) 3971–3979.
[12] N.-P.V. Nielsen, J.M. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, J. Chromatogr. A 805 (1) (1998) 17–35.
[13] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (8) (1994) 1361–1375.
[14] P. Nomikos, J.F. MacGregor, Multi-way partial least squares in monitoring batch processes, Chemom. Intell. Lab. Syst. 30 (1) (1995) 97–108.
[15] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2014. http://www.R-project.org/.
[16] H.-J. Ramaker, E.N. van Sprang, J.A. Westerhuis, A.K. Smilde, Dynamic time warping of spectroscopic batch data, Anal. Chim. Acta 498 (1) (2003) 133–153.
[17] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, Acoust. Speech Signal Process. IEEE Trans. 26 (1) (1978) 43–49.
[18] G. Tomasi, F. Van Den Berg, C. Andersson, Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data, J. Chemom. 18 (5) (2004) 231–241.