Piecewise Linear Representation Segmentation in Noisy Domains with a Large Number of Measurements: the Air Traffic Control Domain

José Luis Guerrero

Group of Applied Artificial Intelligence, Computer Science Department
University Carlos III of Madrid, Avda. Universidad Carlos III, 22
28270. Colmenarejo, Madrid. España
jguerrer@inf.uc3m.es

Jesús García

Group of Applied Artificial Intelligence, Computer Science Department
University Carlos III of Madrid, Avda. Universidad Carlos III, 22
28270. Colmenarejo, Madrid. España
jgherrer@inf.uc3m.es

José Manuel Molina

Group of Applied Artificial Intelligence, Computer Science Department
University Carlos III of Madrid, Avda. Universidad Carlos III, 22
28270. Colmenarejo, Madrid. España
molina@ia.uc3m.es

The importance of time series segmentation techniques is rapidly expanding, due to the growth in collection and storage technologies. Among them, one of the most used ones is Piecewise Linear Representation, probably due to its ease of use. This work tries to determine the difficulties faced by this technique when the analyzed time series shows noisy data and a large number of measurements and how to introduce the information about the present noise in the segmentation process. Both difficulties are met in the Air Traffic Control domain, which exhibits position measurements of aircraft's trajectories coming from sensor devices (basically surveillance radar and aircraft-derived data), being used as the motivating domain. Results from the three main traditional techniques are presented (sliding window, top down and bottom up approaches) and compared with a new introduced approach, the Hybrid Local Residue Analysis technique.

Keywords: Time series segmentation; piecewise linear representation; piecewise linear approximation; BLUE residue; movement models; air traffic control.

1. Introduction

Time series domain involves a set of different concepts and procedures to understand its importance and the available techniques. The objectives of this introductory section are to present those concepts in a simple way, leading the reader to easily understand the growing importance of time series, the different representation issues and how they have been faced by the available approaches, introduce the air traffic control domain in the context of time series, and finally outline the main aspects of the technique developed in this work.

Time series are sequences of data vectors, containing each of these vectors a timestamp as one of its values. Typical examples can be found in genetic research,[1] financial,[2] medicine[3] manufacturing[4] or tourism[5] applications. Also research fields require the analysis of time series, such as Several processes can be defined regarding time series, such as their analysis (in order to extract different meaningful characteristics or patterns from them, which can be used by additional processes) or forecasting (the development of models in order to predict future values).

The importance of time series has grown exponentially in recent years, due to the explosion in the application of collection and storage technologies, generating huge amounts of data to be processed. In the financial domain, a clear example is the tracking of stock prices, being constantly updated in different markets all over the world.[6] The processing of these massive amounts of data requires an approximate representation of the information that can be more efficiently and effectively handled (as the analysis of every time point is usually not necessary nor practical, and can even be unaffordable). Time series segmentation is a tool presented in order to resolve this issue, by means of reducing the data dimension with appropriate models for representation and approximation.

To achieve that dimensionality reduction, segmentation processes may use different high level representations, such as Fourier Transforms,[7] Wavelets,[8] Symbolic Mappings[9] or the approach, recently explored by the data mining community,[6,10,11] which will be covered in this paper: Piecewise Linear Representation (PLR, also named Piecewise Linear Approximation, PLA). We have centered the scope of our proposal in this representation due to its wide application to different domains. This representation can be treated both as a final result itself and/or as the basis for different additional processes, such as fast similarity search.[12] The extended use of this particular technique may be caused by its simplicity and ease of use: PLR segmentation is based on the approximation of a Time Series T with length n by means of a set of K segments (where $K \ll n$), approximating each of these segments by a linear model.

A segmentation technique, in general, is responsible of the division of a time series into a certain number of segments (ideally, as few as possible) and the approximation of the data in each segment by a certain simple function. This introduces several interesting issues, such as the measurement of the quality of a certain segmentation result and the consideration of the implied cost to obtain that quality (remembering that the purpose of the whole schema is to perform a dimensionality reduction over the original data). Different classifications can be performed over segmentation techniques along with the high level representation used, being a capital one, regarding their applicability to different processes, their online or offline nature. Offline segmentation algorithms may use the whole data from the time series to obtain their segment approximation,[10] whereas online algorithms perform their segmentation based only on the available data of an incomplete data series.[11] This nature usually has an impact on the complexity and accuracy of the resulting algorithm (offline algorithms benefit from their complete knowledge of the time series to obtain a more accurate segmentation, while their computational complexity is more impacted by the size of the considered time series).

Regardless of the concrete technique applied (or according to it, considering that this fact may determine the concrete approximation used), representation of the available information is the key to obtain effective and efficient segmentation results.[13] Time series may be affected by a series of factors, such as large quantity of measurements and the presence of severe noise in them which may prevent those achievements. Thus, dealing with those handicaps is an extremely important issue for segmentation processes. These factors are especially relevant in time series exhibiting sensor data or video tracking information.[14] A paradigmatic time series domain related with the previous difficulties is Air Traffic Control (ATC), which analyzes the data coming from sensors measuring the position of aircraft, which is recorded for offline validation, resulting in time series usually named *opportunity traffic*. This *opportunity traffic* information is the only available experimental data in this domain.

As exposed, we will use ATC as a source for opportunity traffic time series in order to perform PLR segmentation over them. The particular importance of these time series is related to the domain activities: ATC is a critical area related with safety, requiring strict validation in real conditions,[15] being this one of the previously mentioned domains where the amount of data has gone under an exponential growth (in this case due to the increase in the number of passengers and flights). This has led to the need of automation processes in order to help the work of human operators.[16] These automation procedures can be basically divided into two different basic processes: the required online tracking of the aircraft (along with the decisions required according to this information) and the offline validation of sensor data processors. The evaluation task is usually separated into two sub-processes, segmentation,[17,18] showing a slightly different meaning to the one introduced in this section, as it only covers the division of the time series into a series of segments, and reconstruction,[15] which covers the approximation of the segments which the trajectory was divided into. Artificial intelligence techniques have been applied for different purposes, such as establishing a formalization of the domain theory and its associated validation process.[19] Considering it from the segmentation point of view, opportunity traffic provides very interesting time series due to the difficulties which segmentation processes have to face in them. These difficulties, along with the characteristics of data measurements, may include reconsiderations of the quality functions used to measure the accuracy of a segmentation result (due to the high noise in the measurement values and the knowledge of the motion models which the aircraft may perform).

The purpose of this paper is to review the performance of traditional segmentation techniques and propose a new approach for these particularly difficult domains to deal with for PLR segmentation: noisy domains with a large number of measurements. The technique presented is built according to established segmentation design characteristics, also discussing the treatment which these design characteristics have received in the available algorithms. The presented approach is based on the technique and results presented in Refs. 17 and 18, generalizing the proposal in those works to the general PLR domain, and leading to the introduction of the Hybrid Local Residue Analysis (HLRA) segmentation technique, including the mechanisms to handle these situations, and particularizing the results for the ATC domain. The introduction of quality measures is also required in order to cope with the noisy data and the multi-objective nature of the problem solutions, which, along with the proper statistical tests, will be used to test the relevance of the obtained results over a dataset containing opportunity traffic trajectories from the ATC domain.

The introduction to the domain and its difficulties are explained in the second section. The third section presents the segmentation issue as a whole, along with three traditional techniques whose results will be compared with the algorithm proposal. The HLRA algorithm is presented in the fourth section, while the fifth compares the computational complexity of the four options. The sixth section presents the individual results and their comparison, leading to the final conclusions section.

2. Segmentation Issues in the ATC Domain

The traditional PLR segmentation techniques exhibit a series of problems and issues in domains with high noise and very long series. These domains are particularly interesting when there is available information about the introduced noise (or accurate estimations of its value), being a clear example of this fact the measurements obtained by sensor devices (having an individual model for their measuring errors), which have in the ATC domain one of its most representative examples. Thus, the introduction to these domains will be performed by the description of the particular ATC domain, in order to be able to build the argumentation leading to the proposed technique.

The basic data in the ATC domain are the trajectories recorded from flying commercial aircraft, containing sensor plots with the following components: stereographic projections of their x and y components (which are a representation in a common reference frame of the different radar measurements), covariance matrix (representation of the noise introduced by the positioning system: radar, GPS, multilateration, etc) and detection time. This domain also allows us to exploit some related knowledge due to the fact, as has already been pointed out in several references[17,15] that the movement models (MM's) of commercial flights have a certain uniformity in their values (meaning that they tend to follow certain MM's smoothly, without abrupt changes in the position values). This prevents the application of approaches to detect abrupt changes, such as the one exposed in Ref. 13, based on the identification of those changes (named feature points).

In this domain, the models followed by an aircraft can be usually simplified into three different possibilities achieving remarkable results[15]: uniform, turn and acceleration MM's (which might be reduced even to only two, considering that a turn is only a transversal acceleration MM). An important consideration is the length of the maneuvers when we compare them to the uniform segments of the trajectory. If we consider that in a whole time series q measurements were recorded while the aircraft was performing some kind of maneuver and p measurements were recorded while the aircraft was performing a uniform MM, for the vast majority of trajectories, $p >>> q$. These trajectories are performed in airways areas, the most common situation in the available airspace. When the plane approaches a terminal in the surroundings of airports, it gets into terminal maneuvering areas (TMA), where this rule does not apply. To illustrate these differences we have included into the considered dataset racetracks examples, the trajectories performed by aircraft during the landing procedures, which we will analyze later.

Therefore, the right identification of the uniform segments becomes the key factor in this domain (which involves the difficulty of being able to differentiate the effects of the noise from those due to the start or end of a maneuver). An effective PLR segmentation technique should be able to adequately identify those long uniform segments accurately. There is very valuable information which algorithms must seek to introduce. This information includes noise and maneuvers data. The noise introduced in the time series' values is caused, as explained, by different measuring devices, usually external, such as radars[20] or automatic dependent surveillance (ADS) systems based on GPS.[21] Usually the segmenting algorithm is provided this information by a covariance matrix under Gaussian assumptions, not requiring it to know or apply special noise considerations depending on the measuring device. The additional important source of information involves the minimum and maximum length of the maneuvers the aircraft may take (which is specially delimited when handling commercial air traffic). This data can provide us with configuration parameters for our algorithms, in order to adjust them to the kind of traffic they will be dealing with.

According to the analysis presented in this section, PLR segmentation techniques will have to deal in the ATC domain time series with three difficulties: the noise introduced by the measurement device, the large number of measurements which compose each trajectory performed by an aircraft and the proper segmentation of the long uniform segments which these aircraft exhibit.

3. Time Series Segmentation Techniques

The objective of a segmentation process is to divide a data sequence into a series of segments and approximate these segments with a simple function. In the case of study in this work, PLR, those segments are approximated with piecewise linear models.

The segmentation process can be seen as a search over the time series measurements trying to obtain the structure of segments that minimize (or maximize) a certain quality function. Considering each measurement represented as $\vec{x}_k$ for a time series T, the segmentation process is formalized in (1)

$$T = \{\vec{x}_k\},\ S(T) = \{B_m\},\ \ B_m = \vec{x}_j,\ j \in [k_{min}, \dots, k_{max}] \\ \to \begin{matrix} min \\ max \end{matrix}\ f_{quality}(\{B_m\}) \tag{1}$$

where $S(T)$ is the result of segmentation according to the criteria in the given function $f_{quality}$, which is minimized or maximized according to the given requirements, and B_m is a concrete segment from the solution (which covers the points between k_{min} and k_{max} boundaries). The best possible solution for this process could be obtained by considering every possible segment obtained from the different $\vec{x}_k$ measurements of the time series and deciding the output value according to the summation of the function error values for those segments. Equivalently, this can be seen as a search over the different possible measurements which divide the trajectory into different segments. Unfortunately, these search processes are computationally unaffordable, leading to different segmentation techniques which apply different heuristics.

The traditional criteria to determine the quality of a segmentation process[10,11] are the following:

(1) Minimizing the overall representation error (*total_error*)
(2) Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)
(3) Minimizing the number of segments so that the total representation error does not exceed *total_error*

where *total_error* and *max_segment_error* are user defined parameters for the algorithm.

The previous criteria highlight the fact that, instead of a single quality function, these processes usually have to minimize (or maximize) a set of different error functions jointly (typically an error function measuring the distance to the original time series, for example an Euclidean distance, and a different one measuring the cost of that error, for example the number of segments used for the segmentation), changing this approach into a multi-objective optimization problem (MOOP),[22] formally represented by (2). Additionally, sets of restrictions over the quality functions may be set.

$$T = \{\vec{x}_k\},\ S(T) = \{B_m\},\ \ B_m = \vec{x}_j,\ j \in [k_{min}, \dots, k_{max}] \\ \to \begin{matrix} min \\ max \end{matrix}\ \{f_{q1}(\{B_m\}), \dots, f_{q1}(\{B_m\})\}^{\cdot} \tag{2}$$

Following the formalization in (2), we may introduce the three quality criteria presented for the PLR problem obtaining Eq. (3).

$$B_m = \vec{x}_j,\, j \in [k_{min}, \ldots, k_{max}],\, m \in [1, \ldots, seg_{num}] \rightarrow min\{d(S(T)), T), seg_{num}\}$$

$$\text{such that} \begin{cases} d(S(T), T) \leq total\_error \\ \forall m, d(f_{ap}(B_m), B_m) \leq max\_segment\_error \end{cases} \tag{3}$$

where $d(P, Q)$ is a distance error function between series P and Q, $f_{ap}(B)$ is the approximation function result over series B (in PLR the resulting line which approximates the data in series B), and seg_{num} is the number of segments obtained by the applied segmentation algorithm.

The reader may notice that minimizing the number of segments seems to be a key factor in the quality of the segmentation process (as it appears in two out of the three criteria). Even so, most capital references on this topic[10] (even though they state the three previous criteria) base their quality comparisons on only one factor: $total\_error$. Only recently[11] the number of segments is beginning to be compared as a performance metric over the quality of a segmentation process.

The lack of attention to a performance metric which is, at the same time, stated to be a very important one, can be explained by looking at the design of traditional algorithms (which will be covered in the next subsections) and the absence in them of mechanisms to actually control that the number of segments is kept to an allowable minimum. Their approach is based on Eq. (1), using a leading quality function based only on the approach error compared to the original time series. The segmentation approach proposed in this work will take into account that value not only as a quality comparison value, but also as a design consideration. This is especially important for noisy domains, since considering only the representation error leads to oversegmentation in the trajectories, due to the algorithm's excessive focus on the position changes caused by the noise.

An obvious determinant factor not yet commented is the computational complexity of the segmentation process. In general, segmentation processes are required to have low computational complexity (or at least a scalable one), either in online (due to the real-time requirements) or offline (due to the huge amounts of data required) processing.

Along with the differentiation between online and offline algorithms, the linear segmentation process can be divided, as well, into two different approaches: linear interpolation and linear regression. The former uses the equation of a straight line given two points which belong to it (using the initial and end points of the segments) to obtain the approximation segment. This produces a segmentation of the time series with continuous piecewise lines. On the other hand, linear regression, as its own name indicates, uses a regression line to approximate all the points belonging to the segment with a criterion of minimum residual error, producing a set of disjointed lines. The overall error obtained by a linear regression is always less than or equal to the one obtained with a linear approximation, which, along with the low computational complexity it involves, are some of the reasons for its usual choice.[10,11] There are additional difficulties faced by linear interpolation approaches in noisy domains, due to their high sensitiveness to the position of the initial and final measurements of the segment, which can redefine the interpolated segment completely. According to this analysis, we will introduce linear regression into the applied techniques presented in this work.

A final decision regarding a general segmentation algorithm is whether the final segments will be continuous (meaning that the end measurement of segment i is the beginning measurement of segment $i + 1$) or discontinuous (each measurement belongs only to one segment). Some algorithms are more sensitive to this choice than others, being bottom up approaches the most affected by it (since a discontinuous approach introduces limitations in the possible sizes of the output segments). The algorithms presented in this work will use a continuous approach, in order to prevent possible limitations introduced by discontinuous segments.

Three different traditional approaches will be used as comparison tools for the proposed technique, each of them using different heuristics to solve the segmentation issue. Their implementation will be based in the pseudocodes provided in Ref. 10, considering the choices explained in this section. Sliding window[23] is an online algorithm based on building growing windows from the beginning of the time series until a certain user boundary is exceeded by the result of an error function, leading to the creation of a new segment at that measurement and the restart of the process. Several improvements have been performed over this basic version, such as the Incremental Sliding Window,[11] or the different complementary approaches.[24] It is also important to notice that the sliding window algorithm is reported to give pathological results under certain circumstances[25] and also to obtain a best relative performance on noisy data,[10] being this last statement specially important, since the ATC domain presents such noisy data.

Top Down algorithm[10] is an offline process based on finding the best splitting point (understanding by this that measurement which divides the trajectory into the two segments with the lowest added errors) recursively, until all the resulting segments have an error value bellow a user defined boundary. The Top Down algorithm is applied in a wide variety of domains and fields, being also known by different names.[26,27] As in the case of the sliding window, there are numerous improvements to the basic top down algorithm. Alternative approaches[28] perform different initializations based on valleys and peaks, which is reported to perform poorly on noisy datasets, and thus would be inapplicable in our domain.

Bottom up algorithm[10] is an offline process complementary to Top Down, where the time series is initially divided into every possible segment (composed of two measurements) and finds the best possible segment fusion afterwards (understanding by this the fusion which obtains the segment with the lowest error) until any possible fusion obtains a segment having an error above a user defined boundary. The bottom up algorithm, as well, has spread to different fields and research areas using different names, such as the computer graphics domain and decimation methods.[29]

4. The Hybrid Local Residue Analysis Technique

In the presentation of traditional segmentation techniques, the multi-objective nature of segmentation algorithms and the importance of the number of segments was introduced. Even so, these techniques do not provide any explicit mechanism to deal with this performance metric, which explains the lack of coverage this parameter has in most of the available literature. This fact, which is important in any general domain, is even more so in the ATC domain. The reason for this importance is that, as was explained in the second section of this work, typical ATC time series consist of very long uniform segments causing that, if our segmentation technique outputs a large number of segments, it probably means that we are oversegmenting, using that information to cover the position changes introduced by the noise in the measurements (which, in terms of storage, is a waste of capacity, and difficulties the processing of the output for a different range of processes, such as reconstruction[15]). This introduces a different quality factor to our domain: we do not only want a number of segments as small as possible, but we also want to concentrate those segments on the maneuver sections of our time series.

The general idea of our proposed approach, the Hybrid Local Residue Analysis (HLRA) technique, is to analyze each measurement of the trajectory according to a surrounding window and assign a classification value to it (local classification according to a residue value). This classification determines if the measurement is considered to belong to a uniform MM or non-uniform MM. Adjacent measurements sharing the same classification are considered to belong to the same segment. Once the whole time series has been classified following this approach, those segments which were classified as belonging to a non-uniform MM are segmentated according to the bottom up algorithm (hybrid segmentation schema). The segmentation positions obtained this way are relative to the beginning of the non-uniform segments, which require to be corrected to their respective positions in the complete time series in order to be added to the final solution. Figures 1 and 2 present an overview of the two phases of this process, while figure 3 shows an example over a turn trajectory. This example shows that the first phase of the algorithm, by the use of the local classification information, is able to accurately segmentate the time series data where the aircraft was performing a uniform MM, while those sections where a non-uniform MM was being performed are handled afterwards and segmentated by the general bottom-up algorithm.

This idea of using individual classification of the measurements in order to perform the segmentation process has been previously explored in the works,[17,15] where each point is classified according to a set of possible MM's in order to launch a reconstruction process over that classification. Also, in Ref. 17, a comparison showed that classification results were substantially better if measurements were classified individually instead of as part of whole segments (which would be an approach comparable to a sliding window algorithm without its usual left anchor where, instead of growing to the right, it would move the whole window). These

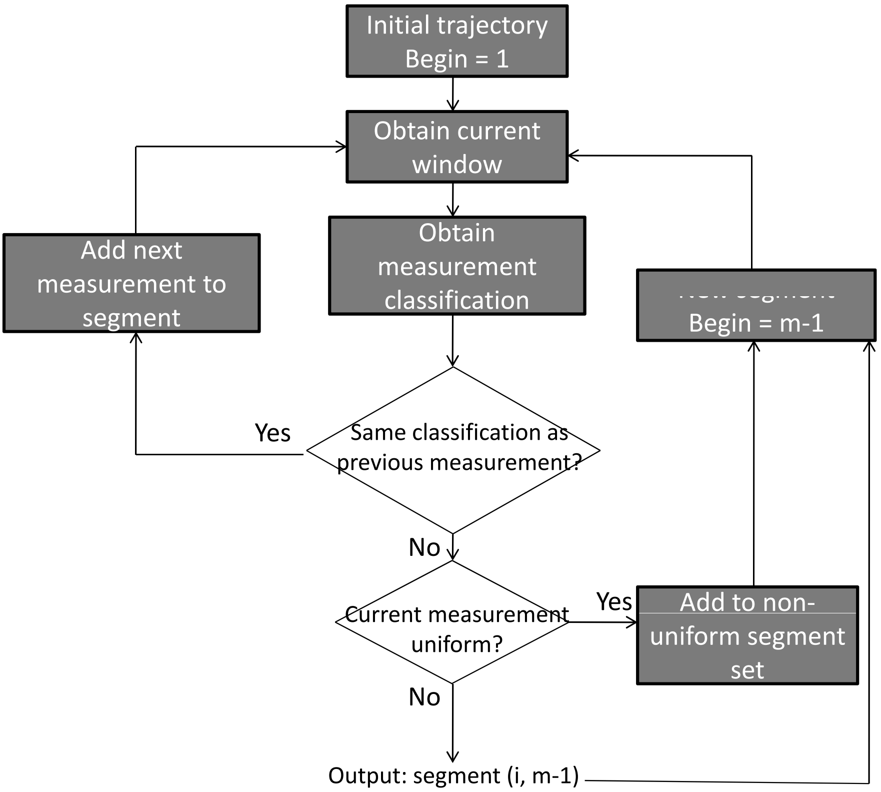

Initial trajectory
Begin = 1
Obtain current window
Add next measurement to segment
Obtain measurement classification
Begin = m-1
Yes
Same classification as previous measurement?
No
Current measurement uniform?
Yes
Add to non-uniform segment set
No
Output: segment (i, m-1)

Fig. 1. Hybrid local residue analysis first phase overview.

ideas will be adapted to our current PLR context. This means that, according to the previously explained needs, our algorithm will be centered in finding the uniform segments of our time series.

The offline processing allows us to use information both from our past and our future. Introducing this fact into a local representation, we will restrict that information to a certain local segment around the measurement which we would like to classify. These intervals are centered on that measurement, but the boundaries for them can be expressed either in number of measurements (4) or timestamp values (5).

$$S_j^i = \{\vec{x}_k^i\},\ k \in [j-p, j+p],\ p \in [j-1, N-j]\,, \tag{4}$$

$$S_j^i = \{\vec{x}_k^i\},\ t_k^i \in [t_j^i - m, t_j^i + m],\ m \in [t_j^i - t_1^i, t_N^i - t_j^i] \tag{5}$$

where S_j^i is a given segment from the trajectory centered on measurement j, N is the number of measurements contained in the time series, p is the sample window size and determines the possible boundaries for a given segment according to its number of measurements (from measurement 1 to measurement N) and m is the time

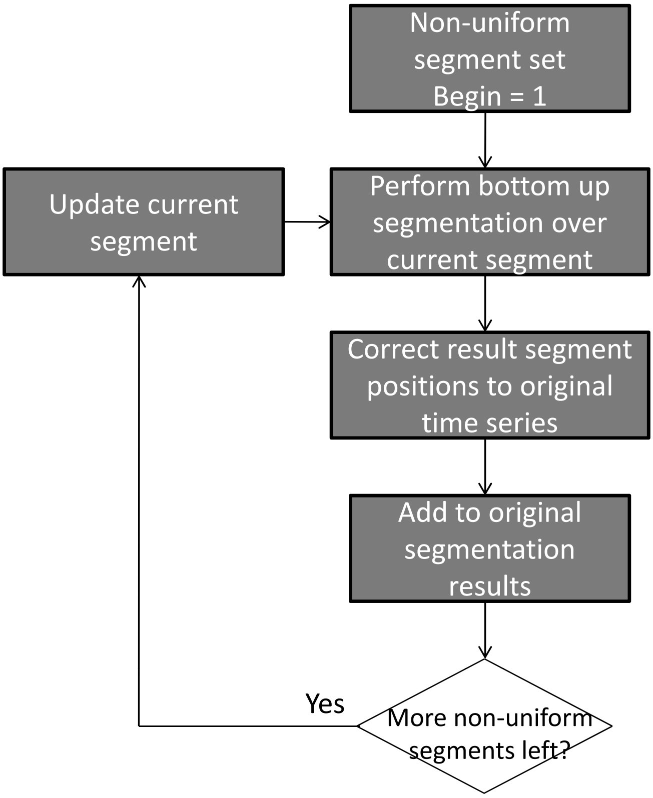

Non-uniform segment set Begin = 1
Perform bottom up segmentation over current segment
Correct result segment positions to original time series
Add to original segmentation results
More non-uniform segments left?
Yes
Update current segment

Fig. 2. Hybrid local residue analysis second phase overview.

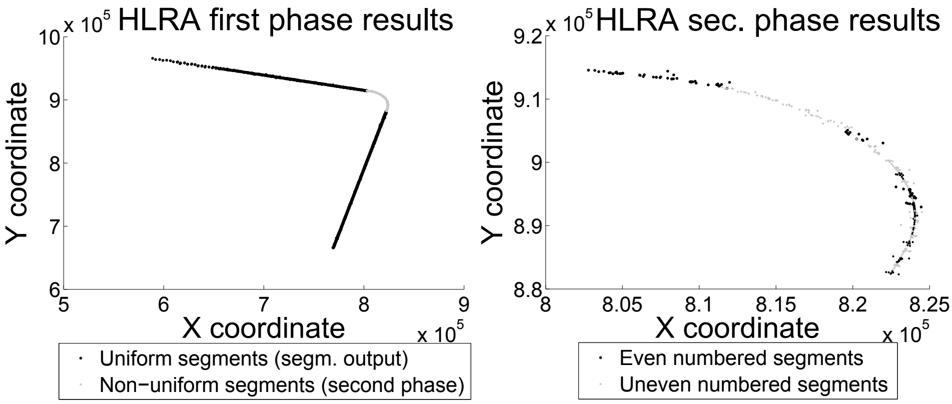


Fig. 3. Example of the HLRA's results over a sample turn trajectory.

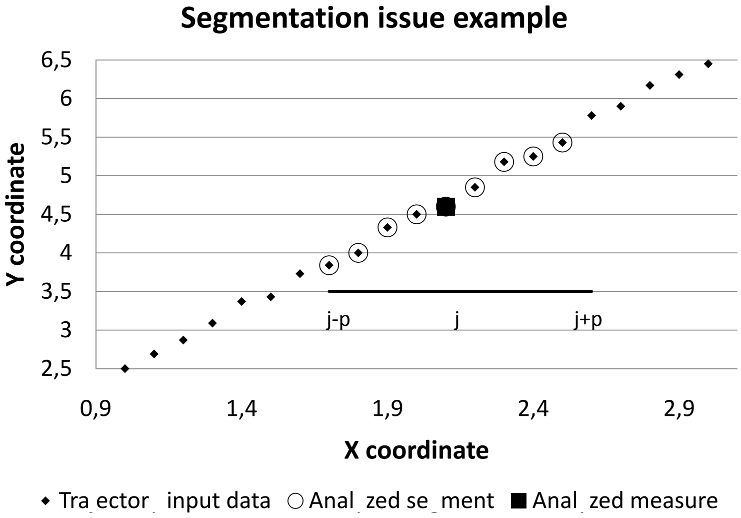


Fig. 4. Local approach segmentation overview.

window size, which determines those same possible boundaries according to their timestamp. Once we have chosen a window around our current measurement, we will have to apply a function to that segment in order to obtain its classification. This general classification function $F(\vec{x}_j^i)$, using measurement boundaries, is represented in (6)

$$F(\vec{x}_j^i) = F(\vec{x}_j^i|T^i) \to F(\vec{x}_j^i|S_j^i) = F_p(\vec{x}_{j-p}^i, \dots, \vec{x}_j^i, \dots, \vec{x}_{j+p}^i) \,. \tag{6}$$

From this formulation of the problem we can already see some of the choices available: how to choose the segments (according to (4) or (5)), which classification function to apply in (6) and how to perform the final segment synthesis. An example of the segmentation issue according to the local classification formulation is presented in figure 4.

The segment boundaries are defined by the domain knowledge. As exposed in the domain section, this knowledge is usually in the form of average duration (in time) of typical maneuvers, so we will use (5), setting a value for m adjusted to 1/2 of the longest possible maneuver. Once the analysis window has been fixed, the classification function will be based on a Best Linear Unbiased Estimator (BLUE) residue value (domain transformation), in order to introduce noise information in the uniform segment detection, and an automatic threshold choosing technique to determine the final classification over that value.

4.1. *Introducing noise information: The BLUE residue*

The first phase of our algorithm covers the process where we must synthesize an attribute from our input data to represent each of the trajectory's measurements in a transformed domain and choose the appropriate thresholds in that domain to effectively differentiate those which belong to our model from those which do not do so.

The transformation function decision is crucial. In Ref. 17 the discussion of whether introducing noise information in the domain transformation function allows us to improve our results was presented. The results proved that, as expected, that noise information improves the overall results. Therefore we will use a BLUE residue value, where we will be able to introduce the noise information by means of a covariance matrix R_k. The assumed linear model is presented in (7)

$$\vec{x}_m(k) = \begin{bmatrix} x_m(k) \\ y_m(k) \end{bmatrix} = \begin{bmatrix} 1 & t_k & 0 & 0 \\ 0 & 0 & 1 & t_k \end{bmatrix} \begin{bmatrix} x_0 \\ vx_0 \\ y_0 \\ vy_0 \end{bmatrix} + \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} = H(t_k)\vec{\theta} + \vec{n}(k)\,. \tag{7}$$

The first component $H(t_k)\vec{\theta}$ represents the ideal estimated parameters for a uniform segment (initial position and velocity). The best estimator of these parameters with minimum squared weighted residual is introduced in Eq. (8). The noise information is introduced in (8) in the form of its covariance matrix, R_k. Then, with estimator $\vec{\theta}$ the interpolated positions for the x and y components of the points can be calculated with (9). Finally, with the previous values, the normalized BLUE residue can be obtained with (10).

$$\left\langle \vec{\theta} \right\rangle = \begin{bmatrix} \langle x_0 \rangle \\ \langle vx_0 \rangle \\ \langle y_0 \rangle \\ \langle vy_0 \rangle \end{bmatrix} = \left(\sum_k H(t_k)^T R_k^{-1} H(t_k) \right)^{-1} \sum_k H(t_k)^T R_k^{-1} \vec{x}_m(k)\,, \tag{8}$$

$$x_{int}(k) = \langle x_0 \rangle + \langle vx_0 \rangle\, k\,, \qquad y_{int}(k) = \langle y_0 \rangle + \langle vu_0 \rangle\, k\,, \tag{9}$$

$$res = \frac{1}{kmax - kmin + 1} \sum_{k=kmin}^{kmax} \begin{pmatrix} x(k) - x_{int}(k) & y(k) - y_{int}(k) \end{pmatrix} \times R_k^{-1} \begin{pmatrix} x(k) - x_{int}(k) \\ y(k) - y_{int}(k) \end{pmatrix} \tag{10}$$

where $x(k)$, $y(k)$ are the sensor measurements values, R_k is the covariance matrix (associated to the sensor) and $x_{int}(k), y_{int}(k)$ are interpolated values using (9).

The threshold choosing technique is closely related to the domain transformation, involving how we determine if a measurement belongs to the model or not. The choice for this parameter will be detailed in the next section.

4.2. *Threshold choosing technique*

The threshold choice[18] involves determining the boundary above which transformed measurements will be considered as unknown. Figure 5 shows an example of a possible choice over the presented transformed domain.

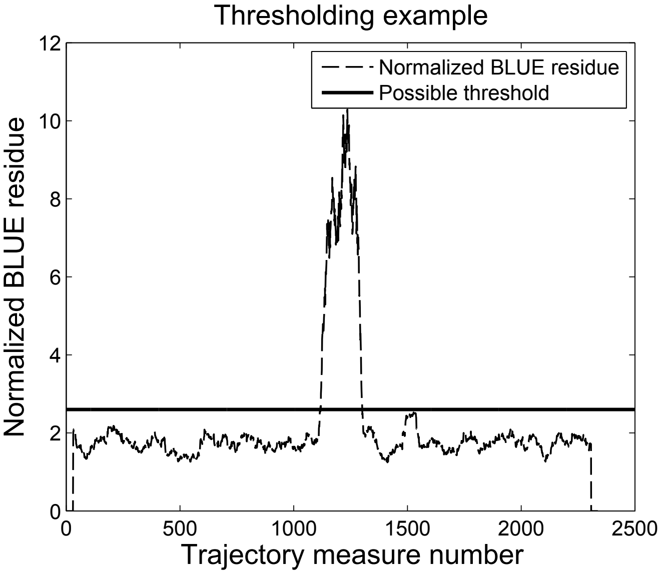


Fig. 5. Threshold choosing example.

According to previous considerations, the objective is to classify the measurements belonging to a uniform MM correctly, with a special attention regarding the limits where the aircraft's MM changes to a different one. Graphically over figure 5, that implies getting the red line as low as possible, leaving only the central section over it (where the maneuver takes place, making its residue value high enough to get over our threshold).

The presented residue value in (10) follows a Chi-squared probability distribution function (pdf) normalized by its degrees of freedom, n. n is given by twice the number of 2D measurements contained in the interval minus the dimension of P (P=4 in our uniform segments, as we are imposing four linear restrictions). For a valid segment residual, res behaves with distribution $\frac{1}{kmax-kmin+1}\chi^2_{2(kmax-kmin+1)-P}$, which has the mean and variance detailed in (11).

$$\begin{aligned} \mu &= 2 - \frac{P}{kmax - kmin + 1}, \\ \sigma^2 &= \frac{4}{kmax - kmin + 1} - \frac{2P}{(kmax - kmin + 1)^2}. \end{aligned} \tag{11}$$

The residue distribution allows us to establish our criterion based on the percentage of measurements belonging to uniform MM. We may use the Chevychev's inequality[30] to determine a threshold which should leave the 90% of the measurements belonging to our linear model above it, with $\mu + 3\sigma$ value. Using the values from (11), Eq. (12) presents the obtained threshold value.

$$thres = 2 - \frac{4}{N} + 3\sqrt{\frac{4}{N} - \frac{8}{N^2}}\,. \tag{12}$$

This threshold depends on the resolution of the segment, which also influences the residue value in (10). It is interesting to notice that the highest threshold value is reached with the lowest resolution. This is a logical result, since to be able to keep our percentage of uniform measurements correctly classified (usually called True Positives Rate or TPR), which has been fixed with the inequality at 90%, with short segments, we need to have a high threshold, in order to counteract the noise effects (while longer segments are more resistant to that noise and thus the threshold value may be lower).

We would like to determine how precisely the chosen χ^2 distribution represents the normalized BLUE residue in non-uniform trajectories with estimated covariance matrix. In the following figures we compare the results obtained with Eq. (12) with the optimal result of the threshold choice (dotted lines), manually chosen to obtain the highest possible TPR while FPR (False Positives Rate, measurements not belonging to the uniform model missclassified) remains in a zero value. Figure 6 shows the used trajectories for this comparison, whereas figure 7 shows the actual comparison for the proposed trajectories between the optimal TPR and the one obtained with (12) for increasing threshold values.

In the two trajectories in figure 7 we may appreciate two distortion effects introduced by the presented approximation. The turn trajectory shows an underestimation of the TPR value due to the inexactitude in the covariance matrix R_k.

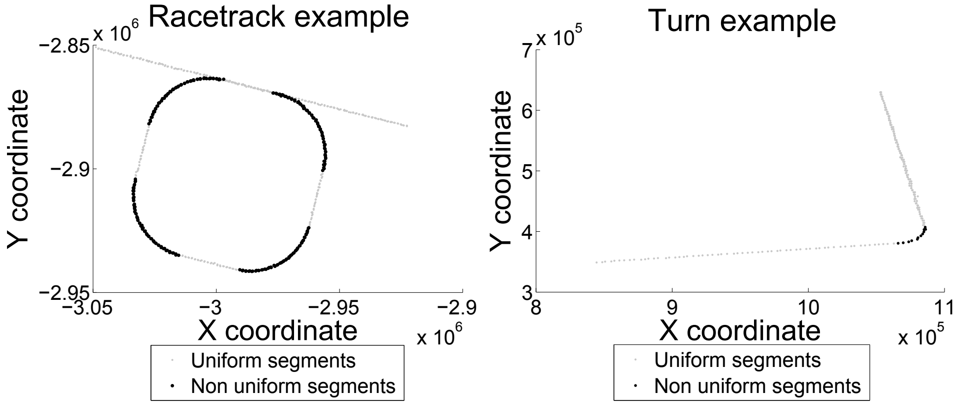


Fig. 6. Considered trajectories for the threshold choice effects analysis.

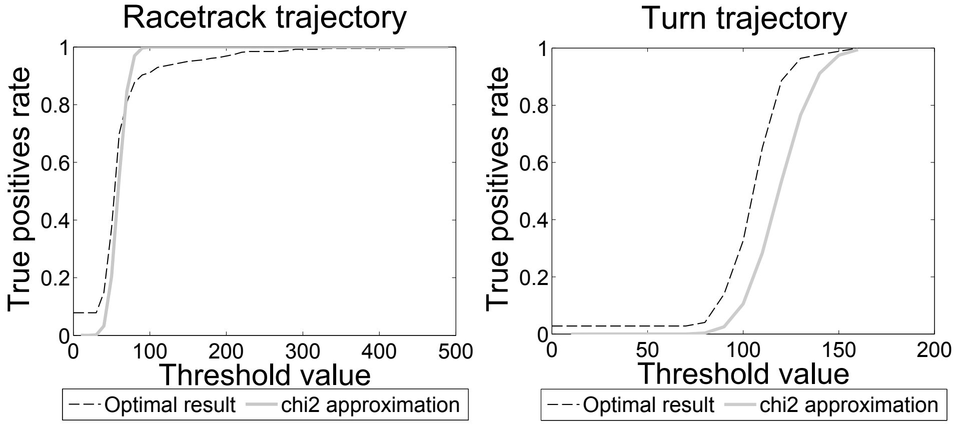


Fig. 7. χ^2 approximation comparison.

This inexactitude assumes a higher noise than the one which is present in the trajectory, and thus will require the choice of a higher threshold than necessary in order to obtain the desired TPR margin.

In the racetrack trajectory we perceive the same underestimation at the lower values of the threshold, but then the approximation result crosses the optimal results and reaches a value over it. This is caused by the second distortion effect, the maneuver's edge measurements. The measurements close to a maneuver beginning or end tend to have a higher residue value than the theoretical one for a uniform trajectory (due to their proximity to the non-uniform segments), making us increase the threshold value to classify them correctly (which causes the optimal result to show a lower TPR in the figure).

These two effects show that we may need a heuristic tuning in our χ^2 distribution in order to adapt it to these distortion effects. For our PLR approach, it is enough to higher the threshold value to $\mu + 5\sigma$, knowing that we may misclassify some non-uniform measurements close to the points where the MM changes (considering that this does not have an important penalization impact).

4.3. *Algorithm definition*

After the theoretical considerations behind our proposed segmentation technique have been covered, we would like to specify the pseudocode for it. The functions required for this pseudocode are programmed with the proposed equations in the two previous sections. Algorithm 1 presents the first phase of our technique.

The second phase of the algorithm applies the bottom-up technique (as described in its section) and corrects the segmenting points obtained to their positions in the original trajectory, providing the final output of the algorithm as a series of segmenting points.

Algorithm 1 Hybrid Local Residue Analysis Algorithm, first phase

```
Input: time sequence $(a_1, \ldots, a_k)$, $time\_length\_window$
Output: uniform segments $(s_1, \ldots, s_k)$, non uniform segments $(sn_1, \ldots, sn_m)$
classifications = uniform_segments = non_uniform_segments = empty set
initial_point = 1
current_point = 1
sequence_length = length (time sequence)
while current_point <= sequence_length do
  current_segment = obtain_segment (time_sequence, current_point, time_length_window)
  current_length = length(current_segment)
  current_threshold = obtain_threshold (current_length)
  current_residue = obtain_residue(current_segment)
  if current_residue > current_threshold then
    add(classifications, non_uniform_class)
  else
    add(classifications, uniform_class)
  end if
  if current_point > 1 && (classifications (current_point) != classifications (current_point-1) || current_point == sequence_length) then
    if current_point == sequence_length then
      if classifications(current_point) == uniform_class then
        add (uniform_segments, current_point)
      else
        add (non_uniform_segments, current_point)
      end if
    else
      if classifications(current_point) == uniform_class then
        add (non_uniform_segments, current_point)
      else
        add (uniform_segments, current_point)
      end if
      initial_point = current_point-1
    end if
  end if
  current_point++
end while
```

5. Computational Complexity Analysis

The complete complexity analysis for the three different traditional techniques can be found in Ref. 10. The results presented are the following:

- Sliding-Window = $O(Ln)$
- Top-Down = $O(n^2K)$
- Bottom-Up = $O(Ln)$

where n is the number of measurements in the time series, K is the number of segments and L is the mean length of the obtained segments. It is important to notice that, for each complexity order, there is at least one parameter not known *a priori*, (either K or L) which makes this complexity orders harder to be accurately established.

5.1. *Hybrid local residue analysis algorithm*

The proposed algorithm involves two different phases, each of them based on different approaches. We will present them separately in order to obtain the complexity order. There are, as well, two different main steps involved in the first phase of the algorithm:

- *Threshold value*: for a fixed window size, this parameter can be computed only once, with a constant order complexity. If the window is not fixed (or established with time boundaries, which result in windows with different sizes at every measurement) this involves computing the threshold n times, where n is the number of measurements, thus adding an $O(n)$ term.
- *Residue value obtaining*: The residue has to be calculated, with a certain window size, at every value of the time series. Calculating each residue involves a cost of $O(w_l)$, where w_l represents the *window_length* involved, and applied to each value of the time series, involves a cost of $O(w_l n)$.

The second phase shows the computational complexity of the bottom up algorithm, which is, as presented in the previous section, $O(Pq^2)$, applied t times, where t is the number of non-uniform segments in the trajectory, P is the mean length of the sub-segments in those segments and q their number of measurements, giving us a complexity order for this second phase of $O(tPq^2)$.

The final complexity order of the trajectory is, adding the terms from the previous two phases, $O(w_l n)+O(tPq^2)$. Considering that P is a small value (as the secondary segmentation is applied to non-uniform sections of the time series, which cannot be well approximated by long uniform segments), $q <<< n$ and $t <<< w_l$ (with the possible exception of extremely long time series, where the value of n compensates for the possible increase in the value of t), we can determine that the dominating order term is $O(w_l n)$, being this the complexity of our proposed approach.

Compared to the presented complexities of the traditional techniques, the proposed solution shows the advantage of having parameters that can be either fixed by the user or accurately approximated, opposed to some of the terms presented by traditional techniques not known *a priori*, such as the mean length of the final segments or the number of these segments.

6. Experiments

6.1. *Quality measurements and algorithm configuration*

In the previous sections of this work the importance of the number of segments has been repeatedly stated, along with the absence of its value which is usually found in available references. But, for our presented domain, there are additional considerations which must be included in the quality indicators to perform an accurate comparison.

Evidently, as was introduced, the objective of any segmentation process is, usually, reducing the amount of information while keeping a record as similar as possible, compared to the original data (even though additional processes with different objectives can be performed over the transformed data). This means that, in a noisy domain such as the one presented, we would like to reduce the effect of the noise as much as possible (whenever it can be different from the actual data of our aircraft). Considering the MM's presented, if we divide uniform segments into several different sub-segments, that division is performed due to the noise position changes, and thus, we are including additional segments which are a waste of data (along with the additional problems for any processing which might be performed afterwards).

That fact can lead to misleading values in the $total\_error$ metric, which contains the deviation of the regression line with respect to the noisy samples. Oversegmentation would reduce the residual, with an evident effect of over-fitting to the noise contained in the series. Figure 8 presents an example of a uniform time series to which Gaussian noise has been added, with $\mu = 0$, $\sigma^2 = 1$ along with its ideal segmentation (based on the original time series previous to the noise addition) and a segmentation result based on interpolation using segments with length = three measurements.

The ideal segmentation in figure 8 identifies correctly the time series as a single segment, presenting a $total\_error$ value of 14.91, while the interpolation segmentation, which oversegmentates the trajectory into different segments, presents a $total\_error$ value of 10.69. This simple example shows that the quality of a segmentation on noisy time series should not be measured by means of a $total\_error$ metric (at least over uniform segments). According to this, we will introduce two different metrics, one related to the segmentation quality over non-uniform segments and a different one for those who were performing a uniform MM in the original trajectory.

For non-uniform segments, we will include the $total\_non\_uniform\_error$, which is the $total\_error$ metric but only applied to those measurements where the aircraft was performing a non-uniform MM (lacking a better quality metric for those non-uniform segments). This metric assesses the behavior of segmentation algorithm under a situation in which the series should be divided to avoid the deviation produced by a linear model in situations in which it is not applicable.

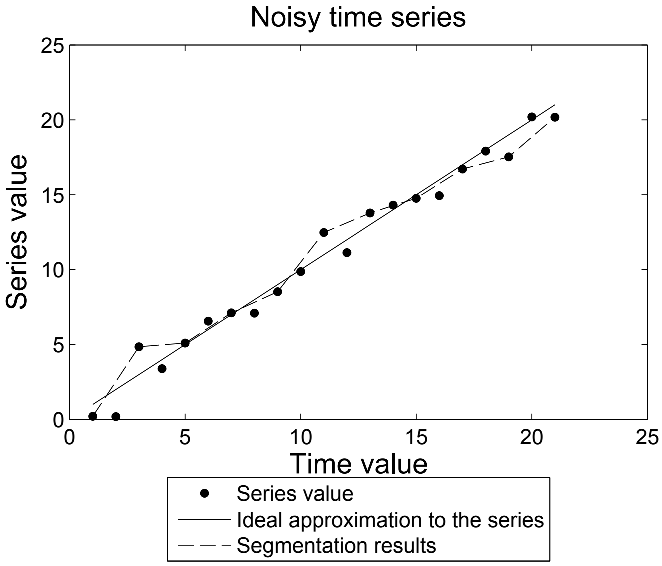


Fig. 8. Two example segmentations for a completely uniform noisy time series.

On the other hand, for the quality assessment of uniform segments, we will introduce the USR (uniform segmentation ratio), which, dividing the ideal number of uniform segments (those performed by the aircraft) by the number of segments obtained by the technique, tries to measure the level of over-segmentation obtained during uniform MMs. This quality metric is defined in (13).

$$USR = \frac{\text{number of original uniform segments}}{\text{resultant segments during uniform MMs}} . \tag{13}$$

The ideal value for this indicator is 1. Lower values indicate oversegmentation on those uniform MM's, while a higher value is only possible if segments exhibiting a non-uniform MM are approximated in a single segment, with a severe increase in the approximation error. The segments taken into account to be computed in the previous ratio are those which have any measurement recorded while the aircraft was performing a uniform MM. It is interesting to realize the complementary nature of the two previous figures of merit. An algorithm prone to oversegmentation will have a very low uniform segmentation ratio and, conversely, an algorithm prone to keep long segments through the whole series will have an unfeasible error value during maneuver sections of the time series.

The final value included for this comparison is the *running_time* of the technique. Obviously, the actual value of this metric depends on different factors, such as the programming language chosen, and cannot be interpreted as an absolute value, but it can be used as a comparative value between different techniques or different configurations over the same technique.

Four different quality metrics will be used to measure the performance of a given algorithm: $total\_non\_uniform\_error$ (accuracy in the representation of those segments which lack a uniform model), *number of segments* (overall cost of the segmentation results), USR (accuracy during uniform segments), and *running time* (computational cost of the segmentation results, an indicator of the feasibility of the application to long time series).

There are configuration issues to be faced which will involve the quality of the results. During the presentation of the traditional techniques, the only shared configuration value was *max_segment_error* (sliding window cannot provide a *total_error* boundary due to its online nature), so we will choose the value for this parameter for the three techniques. An approach could be to determine different values for the different trajectories, in order to optimize the performance of the algorithms in each particular case. This can be achieved by means of the knowledge we have, for simulated trajectories, of the typical durations of maneuvers performed by aircraft following different trajectories.

If we took such an approach each trajectory and algorithm would require their own configuration values, meaning that they would be inapplicable to real trajectories afterwards (where we would have no *a priori* knowledge). To prevent this, a single *max_segment_error* value will be determined to be applied to any trajectory or algorithm, in order to test their performance as a whole.

Once that decision has been taken, the choice of that threshold is not trivial either. We have different techniques, different trajectories and, most importantly, different metrics which have to be optimized jointly. There is also an additional consideration. Choosing a fixed *max_segment_error* tends to set a threshold on the maximum length the obtained segments (considering that every measurement of the time series carries an error), leading the algorithm to obtain shorter segments. To prevent this behavior, we will use the *max_mean_segment_error* value instead. The idea for this parameter is to allow segments to be as long as possible, by setting a threshold over the mean value of the different errors of the measurements belonging to a segment, eliminating the implicit length boundary which *max_segment_error* exhibits. According to this, the three traditional segmentation techniques will be provided with only one parameter, the *max_mean_segment_error*, and different values will be tested regarding this parameter, in order to compare their applicability.

For the proposed technique, the time length of the window (according to Eq. (5) has to be set. As introduced in the domain section, this value is chosen based on the non-uniform MM characteristics. The value chosen for the data set proposed is 60 seconds. The bottom-up technique also requires a *max_mean_segment_error* value which, in this case, is set to 300 meters.

Finally, statistical tests are required to determine the quality of the different compared techniques. This introduces the difficulty of quantitatively determining the quality of different multi-objective solutions and their comparison for quality assessment purposes.[31] Basically, this issue can be approached by the use of a quality indicator, which can reduce the different objectives to a single value and performing a statistical test to determine whether the different result sets can be considered to belong to the same distribution. Since this approach is not the focus of this work, but a tool to determine the significance of the results obtained, we will not analyze in-depth the different tools available for this purpose, but rather choose among them to present the results.

Quality indicators were designed for the comparison of different Pareto Fronts, but, in our case, we will have only one solution for each trajectory in the data set, which simplifies the difficulties of the comparison. Also, we will use three objective functions (USR, *number of segments* and $total\_non\_uniform\_error$ quality measurements). Considering these simplifications we will stick to a unary hypervolume quality indicator[32] for the individual estimation of the quality of the obtained solutions. This estimator requires the choice of a *nadir* point, which is the worst possible solution for the problem. The choice of these points is itself an issue. The $total\_non\_uniform\_error$ value of our chosen nadir points will be a theoretical maximum error obtained by joining the first and last points of the current time series with a segment and calculating the error of the different points in the time series as the distance to that segment. The highest number of segments considered will be the number of points in the time series minus one (representing the worst oversegmentation situation possible, where a segment joins every pair of adjacent points).

The USR value for the nadir points is a little harder to obtain since we may degrade its value oversegmenting segments with uniform MM or introducing into them segments with a non-uniform MM. Considering only the oversegmentation, the nadir point value for its USR component would be zero, but there is not such a boundary for the possible values of this indicator considering the possible errors in the segmentation of non-uniform MMs. In the results for the dataset presented (tables 1–3) the worst possible result obtained regarding this error source is 2, so these values will be converted to the $[0, 1]$ interval considering a worst value of 2.01 (Eq. (14)). This means that a USR value of 2 is treated in a similar way in the results as an oversegmentation value of 0.01, and the nadir point value for USR is 0. In order to normalize the hypervolume values, the $total\_non\_uniform\_error$ and *number of segments* values for the different techniques will be normalize according to the worst possible results presented, so that the nadir point values for both of them will be 1 (Eq. (15)).

$$\textit{normalized USR} = \begin{cases} USR & \text{if } USR \leq 1 \\ 2.01 - USR & \text{if } USR > 1 \end{cases}, \tag{14}$$

$$\textit{nadir point} \rightarrow \begin{cases} USR = 0 \\ t\_n\_u\_e = 1 \\ \textit{number of segments} = 1 \end{cases}. \tag{15}$$

Considering the normalized values presented, the hypervolume indicator (which, in this case, is a three dimensional volume) can be calculated with (16). Over the hypervolume values for the dataset, the Wilcoxon test[33] test will be applied to determine their statistical significance.

$$hypervolume = (t\_n\_u\_e_{nadir} - norm.t\_n\_u\_e) * (norm.\ USR - USR_{nadir}) \\ * (n\_o\_s_{nadir} - norm.n\_o\_s_{)} . \quad (16)$$

6.2. *Data set used*

The data set used is based on eight trajectories covering the different MM described in the segmentation issues section. The complete dataset used is shown in figure 9.

These simulations cover the casuistry of the domain, with the specified characteristics presented, and allow us to determine the validity of the different included techniques. For the computation of the proposed ratio, completely uniform trajectories (trajectories 3 and 4) show a difficulty, as all their measurements belong to a uniform MM, so that the $total\_non\_uniform\_error$ value, regardless of the segmentation performed, will always be 0.

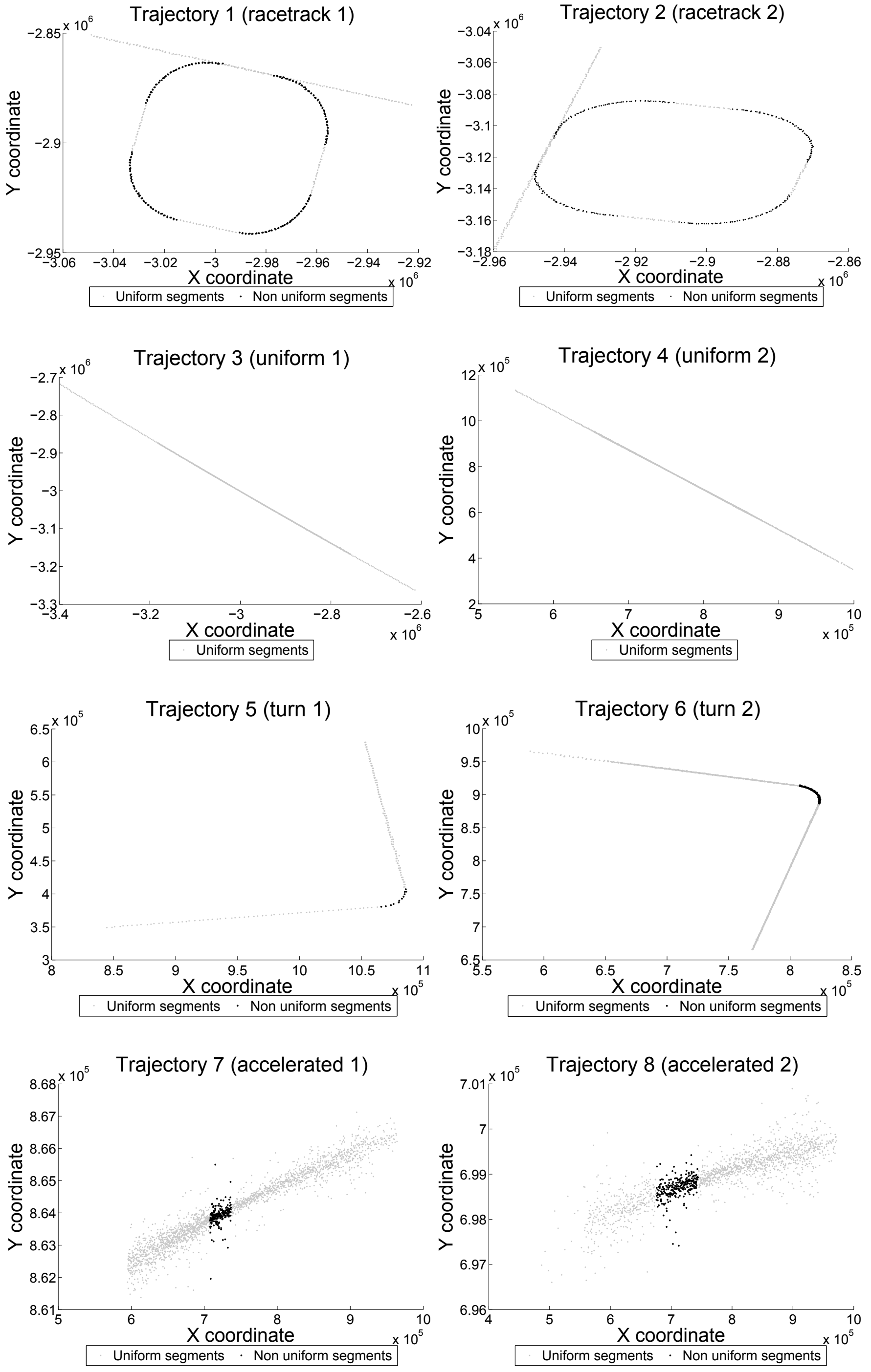
6.3. *Traditional techniques results*

Tables 1 and 2 show the results of the presented classical segmentation techniques applied to the proposed data set. The tray column shows the trajectory number (according to figure 9), $m.m.e.s$ stands for $max\_mean\_segment\_error$.

There are some interesting observations to be made regarding the results exposed in tables 1 and 2. First of all, the effect of the $max\_mean\_segment\_error$ is opposite in the two introduced quality indicators: choosing higher values allows the technique to improve the segmentation results in the uniform segments (reflected in the USR values) but introduces poorer results in the segmentation of $non\_uniform\_segments$. This introduces irresolvable configuration issues, due to the lack of mechanisms in these techniques to differentiate uniform and non_uniform segments.

Regarding the previous configuration issue, it is also noticeable that these algorithms are not able to correctly segmentate the uniform trajectories (trajectories 3 and 4) with any of the tested $max\_mean\_segment\_error$ values. The highest value for this parameter, 800, lead to a segmentation of completely uniform trajectories into two segments (probably an acceptable result) but made the techniques obtain very inaccurate results in accelerated trajectories (7 and 8), obtaining only one final segment in them.

Offline algorithms, reported to be the most accurate ones due to their global knowledge of the time series, reach inadmissible running time levels in some trajectories when faced with values which affect greatly their complexity. The top down algorithm has difficulties dealing with low $max\_mean\_segment\_error$ values



Trajectory 1 (racetrack 1)

Trajectory 2 (racetrack 2)

Trajectory 3 (uniform 1)

Trajectory 4 (uniform 2)

Trajectory 5 (turn 1)

Trajectory 6 (turn 2)

Trajectory 7 (accelerated 1)

Trajectory 8 (accelerated 2)

Fig. 9. ATC trajectory dataset used for evaluation purposes.

Table 1. Sliding window and Bottom-up segmentation techniques results for different *max_mean_segment_error* values.

		Sliding Window				Bottom Up			
Tray	m.m.s.e.	n.u.error	n. segm	USR	Run. time	n.u.error	n. segm	USR	Run. time
1	200	55680	93	0.12	0.06	42019	92	0.13	15.68
1	500	152973	27	0.45	0.06	109954	24	0.63	16.20
1	800	263000	13	1.00	0.06	184770	13	0.83	16.10
2	200	55650	255	0.03	0.07	46422	246	0.03	14.47
2	500	158135	58	0.12	0.06	93688	57	0.13	16.68
2	800	257286	20	0.42	0.06	197702	16	0.71	16.85
3	200	0	168	0.01	0.08	0	114	0.01	27.08
3	500	0	15	0.07	0.09	0	2	0.50	27.54
3	800	0	2	0.50	0.09	0	2	0.50	27.52
4	200	0	556	0.00	0.21	0	539	0.00	149.83
4	500	0	54	0.02	0.25	0	36	0.03	159.56
4	800	0	3	0.33	0.32	0	1	1.00	160.77
5	200	1995	148	0.01	0.03	2000	138	0.02	1.70
5	500	9487	114	0.02	0.03	8834	110	0.02	1.96
5	800	23122	93	0.02	0.03	11877	93	0.02	2.09
6	200	54463	328	0.01	0.25	34698	288	0.01	284.44
6	500	203155	19	0.14	0.34	182421	5	1.00	286.63
6	800	335674	4	1.00	0.37	445360	3	1.00	286.33
7	200	32381	49	0.04	0.29	32718	38	0.05	179.64
7	500	33512	1	2.00	0.35	33512	1	2.00	180.18
7	800	33512	1	2.00	0.36	33512	1	2.00	180.53
8	200	38162	62	0.03	0.18	39467	1	2.00	103.50
8	500	39467	1	2.00	0.23	39467	1	2.00	103.54
8	800	39467	1	2.00	0.23	39467	1	2.00	103.42

(trajectories 4 and 6 with MMSE=200), while the bottom up technique increases its running time noticeably in the presence of a large number of measurements (trajectories 4, 6, 7 or 8). These high running times may make them inapplicable to long real trajectories. There are additional issues related to the extremely high recursion level that the Top Down algorithm has to reach in order to perform its segmentation in trajectories with a high number of measurements, which may lead to the algorithm malfunction.

6.4. *Hybrid local residue analysis segmentation results*

Table 3 presents the results obtained by the proposed algorithm. The main handicaps which were detected in the results presentation of traditional techniques have been properly corrected: those trajectories which were originally completely uniform are now correctly segmentated into a single segment (trajectories 3 and 4, where traditional techniques showed a minimum number of 2 segments), accelerated trajectories are detected to include non-uniform segments and segmentated accordingly (where certain configurations of the bottom-up and sliding window algorithms

Table 2. Top down segmentation technique results for different max_mean_segment_error values.

		Top Down			
Tray	m.m.s.e.	n.u.error	n. segm	USR	Run. time
1	200	57788	94	0.12	0.90
1	500	94136	38	0.29	0.77
1	800	151262	15	0.71	0.63
2	200	65193	199	0.04	3.23
2	500	112310	116	0.07	2.86
2	800	160489	43	0.18	2.58
3	200	0	223	0.00	4.97
3	500	0	2	0.50	0.19
3	800	0	2	0.50	0.19
4	200	0	786	0.00	59.08
4	500	0	5	0.20	2.60
4	800	0	2	0.50	0.65
5	200	9419	87	0.02	1.09
5	500	9419	87	0.02	1.09
5	800	26722	78	0.03	1.07
6	200	27816	769	0.00	162.41
6	500	209334	4	1.00	1.83
6	800	209334	4	1.00	1.83
7	200	29359	294	0.01	20.13
7	500	33504	2	1.00	0.71
7	800	33504	2	1.00	0.71
8	200	39467	2	1.00	0.46
8	500	39467	2	1.00	0.46
8	800	39467	2	1.00	0.46

Table 3. HLRA segmentation technique results for the complete dataset.

	HLRA algorithm			
Tray	n.u.error	n.segm	U.S.R.	Run. time
1	58979	47	0.29	2.20
2	70874	51	0,45	2,27
3	0	1	1,00	1,34
4	0	1	1,00	8,32
5	23471	8	0,67	0,18
6	52488	23	0,25	18,57
7	32282	7	1,00	10,24
8	36748	5	1,00	7,69

obtained a USR value of 2, bypassing the accelerated MM) and the running time remains at an allowable maximum value (18.57 seconds, while the bottom-up algorithm showed a maximum value of 286.33 seconds and the top-down approach a maximum value of 162.41). It is also noticeable that the trade-off among the different values of the metrics (even though its configuration parameters are fixed, while traditional techniques have been tested with a different set of values for their configuration) is consistently better than the one present in traditional techniques.

6.5. *Results comparison*

Even though the individual analysis of the results has already been presented in the two previous sections, it is necessary to compare some of the quality indicators results for the different techniques graphically, in order to complete this results presentation section. With this approach we want to present a general analysis of the performance achieved by the different techniques to support the choice made for the most promising alternative, followed by the complete comparison versus our presented technique in order to validate its results.

This graphical overview will present firstly the comparison of the different techniques for a concrete trajectory (one of the racetracks, trajectory 2). In figures 10 and 11 it can be observed that the proposed technique achieves much better results according to the quality metrics, especially regarding (as was commented after the presentation of the results tables) the trade-off in their different values. To obtain a better result in terms of *total_non_uniform_error* or *uniform segmentation ratio* traditional techniques must degrade the value of the complementary evaluated metric, obtaining unfeasible solutions. Using intermediate configurations (parameter MMSE set to 500), the proposed technique obtains better solutions for the two metrics, also obtaining a smaller number of segments as its output. Among the

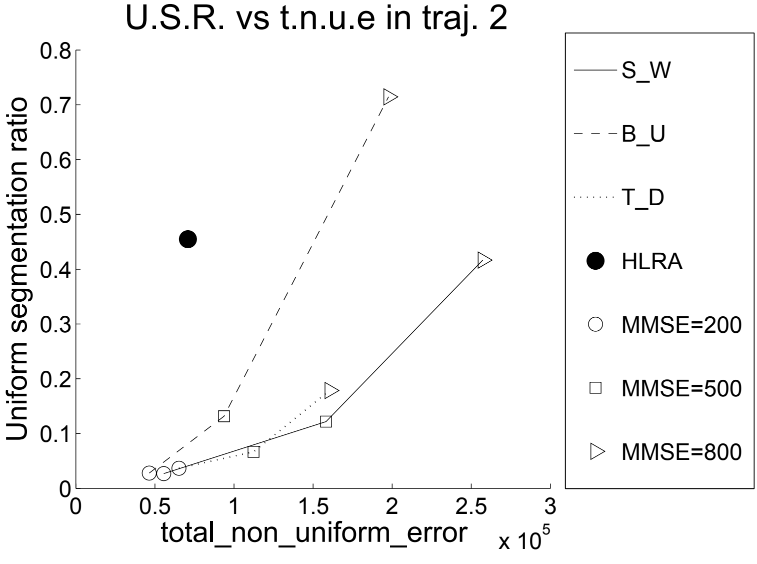


Fig. 10. *Uniform segmentation ratio* and *total_non_uniform_error* values comparison in trajectory 2.

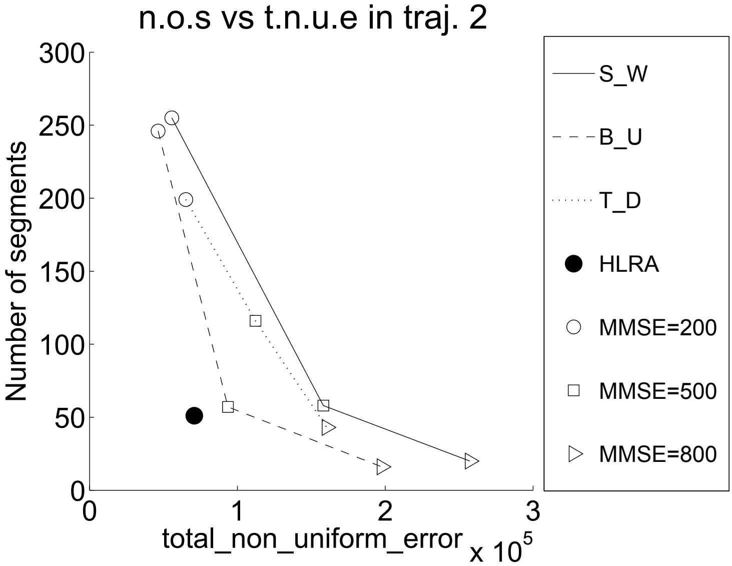


Fig. 11. *Number of segments* and *total_non_uniform_error* values comparison in trajectory 2.

traditional techniques, bottom up segmentation seems to achieve the best results, so we will choose it with a MMSE value of 500 as the most promising technique.

Once the most promising traditional technique has been chosen, we present the results for the whole set of trajectories comparing the bottom-up technique with the indicated configuration with HLRA's results. These comparisons are shown in figures 12 and 13. The results seem to be conclusive: in all the different trajectories presented, the proposed technique achieves better results that the bottom up algorithm, being specially remarkable in some cases, such as the uniform trajectories (HLRA-T3, HLRA-T4, where the degree of oversegmentation with bottom up technique is very high, while HLRA detects correctly a single segment) or the turn ones (where the bottom up technique presents extreme values in either the number of segments, HLRA-T5, or the *total_non_uniform_error* value, HLRA-T6). It is also interesting to highlight that the results of the proposed technique are satisfactory for all the different trajectories, being suitable for any of them.

Finally, the statistical significance of the results must be proved. To do so, as explained in section 6.1, we will calculate the hypervolume indicator values for the results over the different trajectories in the data set for the already chosen most promising technique and HLRA, in order to apply the Wilcoxon test to state whether the result improvements of HLRA are significant or not. The results of the normalized values for the chosen quality measurements, along with their associated hypervolumes are presented in tables 4 and 5.

Running the Wilcoxon test over the hypervolume values in tables 4 and 5, the p-value obtained is 0.0368 (resorting to Matlab's ranksum function for this purpose).

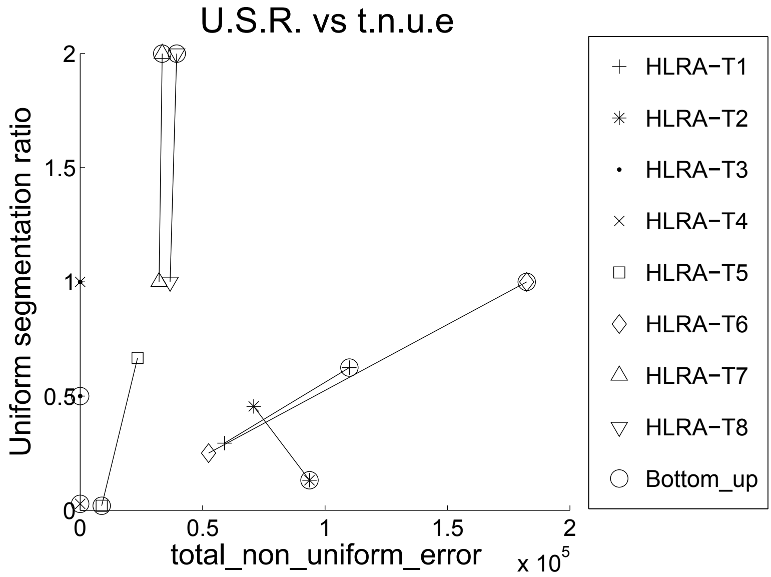


Fig. 12. *Uniform segmentation ratio* and $total\_non\_uniform\_error$ values comparison for the bottom-up technique with MMSE = 500 and the proposed technique applied to the whole data-set.

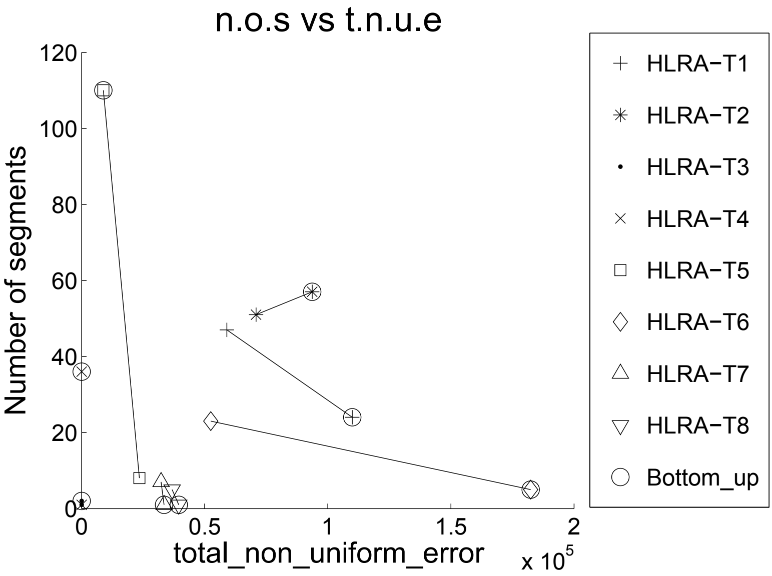


Fig. 13. *Number of segments* and $total\_non\_uniform\_error$ for the bottom-up technique with MMSE = 500 and the proposed technique applied to the whole data-set.

Table 4. Normalized quality measures and associated hypervolume values for bottom up technique with mmse = 500.

Trajectory			Bottom Up mmse = 500			
id	max. error	max. segm.	norm. t.n.u.e.	norm. n.o.s.	norm. USR	Hypervolume
1	9,05E+11	651	1,22E-07	0,036866359	0,625	0,601958452
2	1,77E+13	665	5,30E-09	0,085714286	0,131579	0,120300799
3	3,27E+10	851	0	0,002350176	0,5	0,498824912
4	6,38E+09	2056	0	0,017509728	0,0277778	0,027291418
5	6,17E+12	252	1,43E-09	0,436507937	0,0192308	0,010836403
6	1,15E+14	2747	1,58E-09	0,001820167	1	0,998179831
7	3,26E+08	2178	1,03E-04	0,000459137	0,01	0,009994382
8	3,49E+08	1650	1,13E-04	0,000606061	0,01	0,009992808

Table 5. Normalized quality measures and associated hypervolume values for HLRA technique.

Trajectory			HLRA			
id	max. error	max. segm.	norm. t.n.u.e.	norm. n.o.s.	norm. USR	Hypervolume
1	9,05E+11	651	6,52E-08	0,072196621	0,294118	0,272883657
2	1,77E+13	665	4,01E-09	0,076691729	0,454545	0,419685156
3	3,27E+10	851	0	0,001175088	1	0,998824912
4	6,38E+09	2056	0	0,000486381	1	0,999513619
5	6,17E+12	252	3,80E-09	0,031746032	0,666667	0,645502966
6	1,15E+14	2747	4,55E-10	0,00837277	0,25	0,247906807
7	3,26E+08	2178	9,90E-05	0,003213958	1	0,996687373
8	3,49E+08	1650	1,05E-04	0,003030303	1	0,996864615

This means that with the usual 5% significance level, the null hypothesis that both datasets came from the same distribution can be rejected. In fact, the significance level can be lowered down to a 4% value and still reject the null hypothesis. This result implies that the improvements are statistically significant.

7. Conclusions and Future Work

This work has introduced the difficulties faced by time series segmentation algorithms on domains with long time series exhibiting noisy measurements. Noise degrades the segmentation performance over uniform sections of the time series (which should be packed into a single segment), while the large number of measurements prevents the application of techniques based on global approaches (due to the running time or the recursion level required). These difficulties are faced with the proposed Hybrid Local Residue Analysis technique, based on two phases: the first one differentiates uniform and non-uniform segments in the trajectory, while the second one segmentates the identified non-uniform segments one by one. Noise information is introduced in the initial separation into uniform and non-uniform segments, and the individual approach to each non-uniform segment separately allows the techniques to deal with time series which were not approachable without this pre-segmentation (due to the huge decrease in the number of measurements in each of those individual non-uniform segments). A modification of traditional error indicators is performed in order to deal with the noise in pure traditional techniques (basically establishing a threshold over the max error in mean over the window, instead of absolute values), and performance metrics are introduced in order to measure the quality of the different compared techniques. The results obtained with the Air Traffic Control domain dataset show that the HLRA technique can take advantage of the noise information in order to perform the initial division accurately and afterwards apply bottom up segmentation to obtain a fine segmentation over the non-uniform sections, providing considerably better results that traditional techniques for the different quality indicators presented.

Along with the quantitative objectives of the work, represented by the segmentation results already commented, this work presents the application of artificial intelligence tools to improve the heuristic guided pattern recognition issue which is at the core of segmentation problems, along with the use of quality metrics obtained from the multi-objective evolutionary algorithms domain to determine statistical significance of the results. Also, the results obtained may be used as a data source to improve the performance of reconstruction approaches in the air traffic control domain, and, at the same time, noise handling techniques are introduced to the general PLR segmentation issue. Future lines include the formulation of hybrid techniques (traditional techniques guided by residue values similar to the one used by HLRA) and the study of portability issues to alternative domains.

Acknowledgments

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02

References

1. C.-H. Yeang and T. Jaakkola, Time series analysis of gene expression and location data, *International Journal on Artificial Intelligence Tools* **14**(5) (2005) 755–770.
2. S. Taylor, *Modelling Financial Time Series* (World Scientific Pub. Co. Inc., 2008).
3. G. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods and Tools for ECG Data Analysis* (Artech House, 2006).
4. X. Ge and P. Smyth, Segmental semi-Markov models for endpoint detection in plasma etching, *IEEE Transactions on Semiconductor Engineering* (2001).
5. G. Yu and Z. Schwartz, Forecasting short time-series tourism demand with artificial intelligence models, *Journal of travel Research* **45**(2) (2006) 194.
6. A. Gionis and H. Mannila, Segmentation algorithms for time series and sequence data, *Tutorial on SIAM International Conference in Data Mining* (2005).

7. P. Brockwell and R. Davis, *Time Series: Theory and Methods* (Springer Verlag, 2009).
8. D. Percival and A. Walden, *Wavelet Methods for Time Series Analysis* (Cambridge Univ. Press, 2006).
9. A. Balzanella, A. Irpino, and R. Verde, Dimensionality reduction techniques for streaming time series: A new symbolic approach, *Classification as a Tool for Research* (2010), pp. 381–389.
10. E. Keogh, S. Chu, D. Hart, and M. Pazzani, Segmenting time series: A survey and novel approach, *Data Mining in Time Series Databases* (2003), pp. 1–21.
11. X. Liu, Z. Lin, and H. Wang, Novel online methods for time series segmentation, *IEEE Transactions on Knowledge and Data Engineering* **20**(12) (2008) 1616–1626.
12. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, *Knowledge and Information Systems* **3**(3) (2001) 263–286.
13. Y. Zhu, D. Wu, and S. Li, A piecewise linear representation method of time series based on feature points, in *Knowledge-Based Intelligent Information and Engineering Systems* (Springer, 2010), pp. 1066–1072.
14. M. Machos, D. Gunopulos, and G. Das, Indexing time-series under conditions of noise, *Data Mining in Time Series Databases* (2004), pp. 67–100.
15. J. Garcia, J. Besada, A. Soto, and G. de Miguel, Opportunity trajectory reconstruction techniques for evaluation of ATC systems, *International Journal of Microwave and Wireless Technologies* **1**(3) (2009) 231–238.
16. O. Baud, P. Gomord, N. Honoré, L. Ostorero, O. Taupin, and P. Tubery, Ch. 6: Multi sensor data fusion architectures for air traffic control applications (*Sensor and Data Fusion*, I-Tech Education and Publishing, 2009).
17. J. L. Guerrero and J. García, Domain transformation for uniform motion identification in air traffic trajectories, in *Advances in Soft Computing* (J. M. Corchado, S. Rodríguez, J. Llinas, and J. M. Molina, eds.), Vol. 50 (Springer, 2008), pp. 403–409.
18. J. L. Guerrero, J. Garcia, and J. M. Molina, Air traffic control: A local approach to the trajectory segmentation issue, in *Proceedings for the Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, Part III. Lecture Notes in Artificial Intelligence*, Vol. 6098, pp. 498–507 (Springer, 2010).
19. M. M. West and T. L. McCluskey, The application of machine learning tools to the validation of an air traffic control domain theory, *International Journal on Artificial Intelligence Tools* **10**(4) (2001) 613–637.
20. M. Skolnik, *Radar Handbook* (New York: McGraw-Hill, 2008), 3rd edition.
21. G. Williams, "GPS for the Sky: A survey of Automatic Dependent Surveillance-Broadcast (ADS-B) and its implementation in the United States, *Journal of Air Law and Commerce* **74** (2009) 473–701.
22. M. Ehrgott, *Multicriteria Optimization* (Springer Verlag, 2005).
23. U. Appel and A. V. Brandt, Adaptive sequential segmentation of piecewise stationary time series, *Information Science* **29**(1) (1983) 27–56.
24. H. Vullings, M. Verhaegen, and H. Verbruggen, ECG segmentation using time-warping, in *Advances in Intelligent Data Analysis. Reasoning about Data: Second International Symposium, IDA-97, London, UK, August 1997. Proceedings* (Springer, 1997), pp. 275–285.
25. R. Agrawal, C. Faloutsos, and A. Swami, Efficient similarity search in sequence databases, *Foundations of Data Organization and Algorithms* (1993), pp. 69–84.

26. D. H. Douglas and T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *The Canadian Cartographer* **10**(2) (1973) 112–122.
27. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, 1973).
28. S. Park, D. Lee, and W. Chu, Fast retrieval of similar subsequences in long sequence databases, in *3rd IEEE Knowledge and Data Engineering Exchange Workshop* (1999).
29. P. Heckbert and M. Garland, Survey of polygonal surface simplification algorithms, *Multiresolution Surface Modeling Course, ACM Siggraph Course Notes*, 1997).
30. P. Meyer, Introductory Probability and Statistical Applications (Addison Wesley, 1970).
31. E. Zitzler, J. Knowles, and L. Thiele, Quality assessment of pareto set approximations, *Multiobjective Optimization*, (2008), pp. 373–404.
32. E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *Evolutionary Computation, IEEE Transactions* **7**(2) (2003) 117–132.
33. M. Hollander and D. Wolfe, *Nonparametric Statistical Methods* (Wiley-Interscience, 1999).