Local Frequency Based Estimators for Anomaly Detection in Oil and Gas Applications

Alexander Singh Alvarado [*], Lakshminarayan Choudur [#], Evan Kriminger [*], José C. Príncipe [*]

[*] *Department of Electrical and Computer Engineering, University of Florida*
Gainesville, Florida, USA
{asingh, ekriminger, principe}@cnel.ufl.edu

[#] *Hewlett-Packard Laboratories*
Palo Alto, California, USA
choudur.lakshminarayan@hp.com

***Abstract*— Modern industrial applications such as the smart grid and oil and gas are continuously monitored. The massive amounts of data collected is then processed, and analyzed to generate actions to ensure smooth operations to positively impact the bottom line. In the oil and gas industry, modern oil rigs are outfitted with thousands of sensors to measure the flow rates, as well as the physical and chemical characteristics that affect production from underground off-shore and on-shore reservoirs. Analytical methods packaged into a surveillance system and applied to the massive network of sensors track the state of the system and issue warning alerts about impending failures. In this setting, real time algorithms are needed to detect a diversity of event types, such as anomalies, trends or forewarn failure events to generate alerts for proactive engineering actions. In this paper, effective online algorithms drawn from the signal processing and statistics literature are applied to quickly detect anomalies, trends, and turbulence in the flow of oil in the bore well which is typical in oil production. The short time Fourier transform and dynamical systems were utilized to uncover structure in the data to apply methods based on local derivatives and estimation based on linear and non-linear methods. We compare the performance of the algorithms and make suitable recommendations for their application. Furthermore We apply non-linear time series models such as kernel adaptive filters for prediction purposes and compare their performance against standard linear methods such as the least mean square algorithm. Extensive experiments are conducted over a variety of stream patterns showing that our methods perform well both in terms of accuracy of detection and the short latency in the decision.**

I. Introduction

The supply of easily reachable and refinable petroleum is finite, which motivates producers to extract as much as possible from a given well. Ideally, production should flow 24/7 all year round. Any interruptions to the flow can result in lost income in the millions of dollars. Equipment can be highly specialized, or even custom manufactured for a site, so repair and replacement is expensive. This is particularly true of offshore assets. Demographics of the skilled labor supply are shifting unfavorably for the industry. Production may be located in a remote area; offshore, or in an extremely hot, cold or even dangerous area. For all the reasons cited, it is critical for the oil drilling and production industry to build automated surveillance systems that monitor various stages of production and aid the employees (operators) on the platform to ensure production with few interruptions [1]. Modern oil fields are equipped with thousands of sensors and gauges to measure various physical and chemical characteristics of oil and gas -from rock formations in the sea bed that contain deposits- to distribution systems. Continuous streams of sensor readings can be harnessed via analytical methods to paint a picture depicting the various stages of oil production.

Upon drilling to tap the oil deposits, the bore well is managed to yield maximum capacity. Several aspects come into play that introduce tremendous variability into the production process. Some factors that play a role are: fluid composition, oil viscosity, compressibility, specific gravity, specific gravity of water, solids, and others. These aspects can produce varying flow regimes. Two common occurrences in oil production that cause disruptions to flow rates are: *slugging* and *churn*. Slugging relates to turbulent flows where gas bubbles coalesce, expand, and collapse continuously. Changes in fluid composition from wholly liquid to wholly gaseous over time periods leads to churn which is a major cause of flow rates disruption. Detecting Some common flow rate patterns are 1. high amplitude, high oscillation, 2. low amplitude, high oscillation, 3. low oscillation with pseudo-periodic behavior, 4. normal flow rate followed by a jump, and anomalous flows that are some combination of (1-4). Figure 1 shows examples of the variation of the flow on four different time periods.

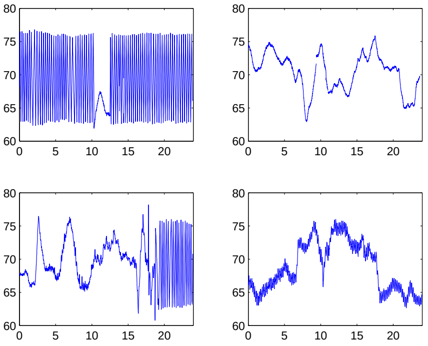


Fig. 1. Measurements on 4 different days. Left plots present an oscillatory behavior. The time axis is in hours for all 4 plots.

The most important variable in oil and gas production is the flow rate. Higher flow rates with fewer disruptions, result in greater yields. Therefore all our experiments are based on the flow rate variable. Due to the multiplicity of factors cited in the introduction, flow rates along a bore well tend to be oscillatory. Controlling the high frequency oscillation (churn) is critical and so, We focus on early detection and prediction of churn in flow rates.

II. Data description

The oscillations in the flow rates demonstrate a slew of behaviors that include high oscillation - high amplitude, high oscillation - low amplitude, low oscillation, linearity, discontinuities. When the high oscillation segment (Figure 2) is examined carefully, it resembles a triangular oscillation. But there may be a hidden periodic structure of the time series that is varying over time. This quasi-periodic phenomenon can be extracted by frequency domain methods. Although, flow rate time-series is quasi-periodic; it is a narrow band signal nonetheless. In contrast, the non-oscillatory regions do not have any discernible structure. Furthermore, different artifacts appear in the data, such as missing values, sudden drops to zero value or clipping of the signal due to sensor malfunction, or simply the sensor being turned off due to a preventive maintenance event.

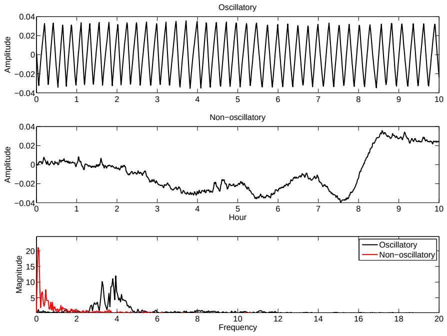


Fig. 2. Original data series in the oscillatory regime (a). Original data series in the Non-oscillatory regime (b). Frequency magnitude spectra of (a) and (b).

In order to fully comprehend the signal, We broke the time series into its oscillatory and non-oscillatory segments shown in Figure 2. It is clear that the flow rates are *periodic* in the oscillatory regions. Examining the time series carefully, it is suspected that there are different regions of varying frequencies. This suspected periodic structure in the flow-rate signal made a case for examining the time-series via frequency domain methods. Fourier analysis is the main-stay of frequency domain analysis. In classical Fourier analysis, *bandwidth* is defined in relation to the Fourier transform. Bandwidth is simply a measure of the range of frequencies (spectrum) usually measured in Hertz. The Fourier transform of a function $f(t)$ gives a view of the signature of the data known as the frequency spectrum, but it completely masks the relationship between the frequencies and time over which the data is analyzed. In other words bandwidth is a global characteristic of the function. To overcome this deficiency, the notion of Windowed Fourier transform was introduced to determine local bandwidth by analyzing the data over windows. This is the motivation underlying the short time Fourier transform (STFT). The STFT involves computing Fourier coefficients over windows of the time series. Figure 3(b) shows the *spectrogram* of the flow rates on one of the days. The spectrogram is merely a plot of time (horizontal axis) and frequencies (vertical axis) with a third dimension which is simply the magnitude of the Fourier coefficients over the window W. This enables us to grasp the frequency components within time windows. Notice that the major transitions in the regimes can be observed by the white vertical plumes. While the STFT provides a mechanism to analyze frequencies over time, We found that it is limited by the conflict between *time-frequency localization*. The Heisenberg uncertainty principle states that time and frequency resolutions are inversely related leading to the condition where analyzing a signal over longer windows compromises frequency resolution and vice-versa [2]. Also in the STFT implementation the window size is fixed which limits the frequency range. A description of the analysis and detection of the flow rate signal is given in the section on *Short Time Fourier Transform*. As We cannot uncover the time varying periodicities in the time-series data, an alternative approach based on dynamical systems was invoked to unravel the time-varying periodic structure in the oscillatory segments of the data. In addition to the time-frequency spectrum Figure 3(a) and (b), We examine the behavior of the oscillations by considering the underlying dynamical system which may have been generated the time series. The dynamical system is regulated by a set of parameters, and their evolution over time is known as the *phase space* of the process. The quasi-periodic regime is conceivably produced by a well organized *attractor*. An attractor in the parlance of dynamical systems is a set towards which the process converges over time. A detector based on this approach should be able to identify the quasi-periodic region by an attractor. Although We do not have

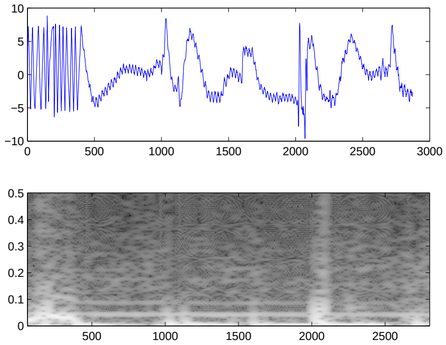


Fig. 3. (a) The original flow rate time series and (b) spectrogram of the time series.

access directly to the phase space of the original dynamical system We can still study the properties of attractors in a reconstructed phase space using Takens' time delay embedding theorem [3]. Takens' theorem reconstructs the phase space with m-dimensional vectors whose components are sampled from the univariate time series with a time spacing of ΔT. The parameters m and ΔT are known as the embedding dimension and time delay, respectively.

Both of these parameters are estimated by using segments of the flow-rate data that is known to contain oscillations. The parameter, time delay, ΔT, should be chosen such that components of the time delay vector are minimally correlated. It is common practice to find the mutual information between a time series and time series delayed by ΔT, as a function of ΔT, and choose the time delay that produces the first local minimum in mutual information, see Figure 4. The embedding dimension, m is larger than the expected *correlation dimension* of the attractor so that the time series is fully unfolded into phase space. The correlation dimension is a measure of the dimensionality of the space occupied by a set of points.

The correlation dimension can be estimated using the Grassberger-Procaccia algorithm [4]. In this algorithm We calculate the correlation sum, $C(r)$, which is the fraction of pairs of points that are within a distance r of each other. In D dimensional space, as r decreases, $C(r)$ will die off proportional to r^D, and therefore $D = \lim_{r \to 0} \frac{\log r^D}{\log r}$. Plotting $\log C(r)$ vs. $\log r$ reveals a linear relationship with slope equalling D for small r, as seen in Figure 5. In practice, with finite data the curve levels off for small r because each point is the only one in its neighborhood of radius r. These quantities can be used in generating the recurrence plot, which consists of a matrix of all the pairwise distances between the different vectors generated considering the lag and embedding dimension. Figure 6 shows an example for a purely quasi-oscillatory segment. In this case only distances less than five are shown as black wiggles. The embedding dimension vector is of size 5 (it is only required that the embedding dimension be larger than the correlation dimension) and ΔT equal to 3. Both axis correspond to time values, they denote the starting point of each vector created using the parameters ΔT and m. Therefore We are actually comparing different segments of the time series. As We move along the first row elements whose value is 1 denote future pieces of the time signal that are similar to the first segment. This can be interpreted as a periodic trajectory in the phase space. Therefore the distance between consecutive diagonals represents an estimate of the period. Figure 6 shows a specific window of the recurrence plot, in this case the distance between the diagonal lines is changing with time, which represents shifts in the period of the signal.

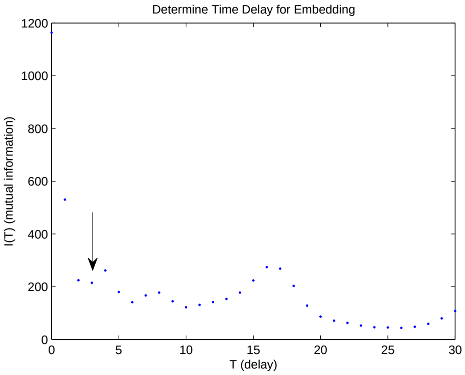


Fig. 4. Selection of time delay using mutual information as a dependency measure. The arrow denotes the value chosen $\Delta T = 3$

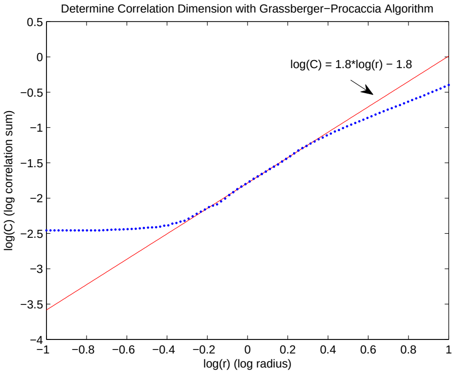


Fig. 5. Determination of the correlation dimension, D, which is the slope of the linear section before the plot levels off.

Thus the recurrence plot in Figure 6 resolves the resolution problem encountered with the STFT and allowed us to capture finer details, such as the time varying periodicities in the flow-rate signal. Also, the dynamical systems approach allowed us to characterize the time series in terms of the reconstructed phase space, which provided an estimate of the embedding dimension. The estimate of the embedding dimension allowed us to select the order (number of lags) of the linear model to approximate the flow rate time-series. The Least Mean Square (LMS) was fit to the data. LMS is an iterative version of ordinary least squares. The linearity assumption of LMS was limiting and it failed to capture the oscillatory behavior. To overcome the deficiency, We resorted to a non-linear modeling approach known as the kernel least mean square (KLMS) method. While it was able to detect the oscillatory and non-oscillatory regimes successfully, due to the non-stationary behavior of the non-oscillatory part of the signal, its implementation resulted in numerous false alarms. In order to reduce these errors, We developed non-parametric ad-hoc methods, handcrafted for the modeling of a time series interspersed with oscillatory and non-oscillatory regimes. These methods depend on the notion of the *local derivative*. The descriptions of the methods are outlined in the sections on *predictive approaches* and *local time and amplitude features*.

III. Short Time Fourier Transform

As We outlined in the section on data description, We applied the STFT to achieve time-frequency localization in order to separate the flow-rate regimes. This is achieved by breaking

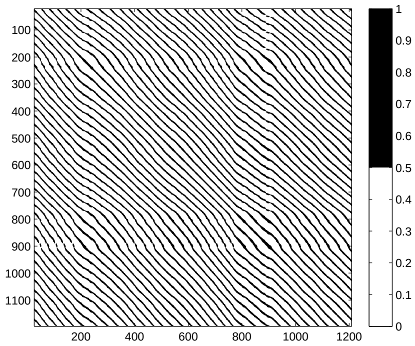


Fig. 6. Recurrence plot using a delay of 3 samples and an embedding dimension of 5.

the time-series into contiguous windows. We compute the Fourier coefficients in each window and choose the Fourier coefficient with largest magnitude over each window as our classification feature. This is a reasonable approach because the larger Fourier coefficients are highly correlated to the signal. This result can be derived on the basis of the Parseval's identity [5]. The window slides through the time series - each shifted by one time unit-producing one feature value over every window. Note that the window only considers past values. Since flow rate consists of two regions (oscillatory and non-oscillatory),We compute the empirical probability density functions (EPDF) over the two regions relative to the feature variable. The empirical densities are computed from the *training data*. The training data is simply the flow rates consisting of oscillatory and non-oscillatory regimes from our set of raw flow rates segmented by day collected from a certain oil well. A decision is made on any new value x_{new} by comparing the likelihood of the sample belonging to either the oscillatory or non- oscillatory regions denoted respectively by R_1 or R_2. The empirical probability density function for each region are given by; $\hat{f}(x_{new}|R_1)$ and $\hat{f}(x_{new}|R_2)$. The empirical densities are simply non-parametric density estimates obtained by applying the kernel smoothing density estimator to a set of discrete points $x_{i,j},\ i = 1, 2, \ldots, n$ and $j = 1, 2$. The subscripts, i and j index the number of points, and the class conditional densities. A new observation x_{new} is classified into R_1, if the class conditional probability of $\hat{f}(x_{new}|R_1)$ is greater than $\hat{f}(x_{new}|R_2)$. This approach is sensitive to the length of the window. For example a small window size will only provide frequency features that describe the rising and falling edge of the quasi triangular wave in the oscillations. This will also produce a number of false alarms since We are only sensitive to linear behavior. A larger window near the period of the triangular wave would reduce the number of false alarms because, We are covering the range of frequencies over the period of the signal. A detailed analysis of the performance of STFT and other methods given in section VII. Figure 11 shows the Receiver operating characteristic curves (ROC) for the short time Fourier and other methods. Departing from the classical Fourier type analysis of signal characterization and detection, and to resolve the time-frequency localization conflict, We will introduce in the next section, predictive approaches to detection, motivated by our results from the dynamical systems techniques.

IV. Predictive Model Approach

While the oscillatory part of the data is quasi-periodic, the non-oscillatory part consists of jumps, irregularities, low amplitude oscillations, and variety of other behaviors. We began by modeling the time-series oscillations by a linear model such that a change in the flow regime can be detected by tracking the prediction error. The coefficients of the linear model were determined by minimizing the mean squared error (MSE). An on-line method finds the optimal solution by using gradient descent which is a simple iterative version of the least squares estimation. This algorithm is popularly known as the Least Mean Squares (LMS) [6]. Assuming a stationary process, the optimal coefficients are given by the Wiener method. These are obtained by solving a linear system in terms of the correlation matrix of the input and the cross-correlation between the input and predicted value. The drawback of this method is that it is restricted to stationary time series and linear models.

Although the algorithm is simple and straight forward to implement, the linear assumption is limiting. The model does not capture the time varying periodicities in the time series. Therefore a linear relation between the past values and the future no longer holds. Therefore, to model the time series, We applied kernel adaptive filter (KAF) [7] in order to overcome the linearity model assumptions. The KAF models involve a nonlinear mapping from the input space to what is known as a feature space. The mapping is produced by means of the kernel function. It allows us to apply linear techniques in the feature space, whose solution would have required non-linear methods in the input space. In this setting, We use an extension to the LMS filter known as kernel LMS (KLMS) which is a member of the kernel adaptive family of filters [8]. The key idea is that the coefficient vector and the input vector no longer lie in the input space, but rather are projected into an infinite dimensional space by the use of the kernel. For more detailed treatment of kernel adaptive filters refer to [7]. Although We designed non-linear filters using basic linear techniques in the feature space, it introduces additional costs. The kernel is a bi-variate function which projects original samples into a new feature space, usually chosen to be of a much higher dimension since it simplifies the problem to the linear case. Upon projecting the samples to a higher dimensional space, the coefficient vector is a linear combination of the input series. Thus, KLMS requires storing the input series, in order to obtain the coefficient vector.

The details of the application of KLMS filter are presented in [7]. In contrast to the LMS algorithm, the KLMS specifically captured the structure of the oscillation and therefore the sample prediction error could be used to determine the switching between the oscillatory and non-oscillatory regimes

(Figure 7). The KLMS approach was applied to time series consisting of two states; oscillatory and non-oscillatory. In this case, it was able to detect the transitions from the two states successfully. Currently, We are conducting experiments to test its applicability to other types of signals. In the above formulation of the KAF, We used a Gaussian kernel with variance equal to 5, a step size of 0.9, and an input vector size of 15, which was approximately half the period. A range of values were tested but these settings gave us the best results.

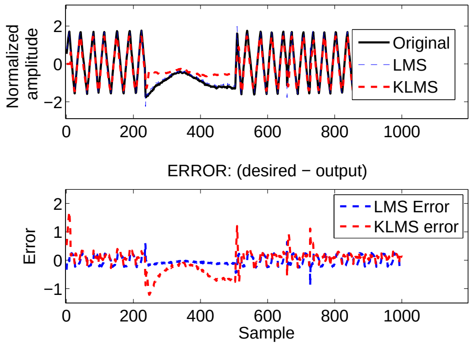


Fig. 7. Using the prediction error from Linear and non-linear models to detect oscillations.

V. Local Time and Amplitude Features

The time-frequency uncertainty can be circumvented by the definition of an instantaneous frequency as done by the Hilbert transform [9], but this will not be pursued here. Alternatively We will be developing methods to quantify the structure of the time series. These are based on locally determined time features particular to the structure of the oscillatory signal. In the time domain, frequency can be defined as the time between consecutive peaks (or troughs) or the time between pairs of *zero crossings*. Zero crossings are simply time instances when the time series crosses zero. This is an intuitive definition that is capable of recognizing a single cycle of a periodic signal. Furthermore, We strengthen the approach by considering amplitude-based features in addition to the local period estimates.

A. Peak-to-Peak Features

In the oscillatory region, the time between local extrema is the half-period of a single oscillation, and hence can be considered an estimate of local frequency. For simplicity, We will use the term “peak” to refer to both local maxima and minima. We will denote the i^{th} peak P_i, and the time of this peak as $t(P_i)$. The feature time between peaks is therefore $\Delta t(P_i) = t(P_i) - t(P_{i-1})$ In Figure 13(a) We see the empirical distributions for $\Delta t(P_i)$ based on training data of known classes (oscillatory and non-oscillatory regimes). A threshold θ based on likelihood ratio of empirical densities can be determined from the training data, and the samples between peaks are classified based on their *time* differences relative to the threshold. Moreover, We can also use the *peak-to-peak* heights as another discriminatory feature variable. The height of the i^{th} peak will be denoted $h(P_i)$ and the height between peaks is $\Delta h(P_i) = h(P_i) - h(P_{i-1})$. As seen in Figure 13(c), the classes have greater separation in this feature variable, thereby lowering the chances of misclassification. When We classify a *peak-to-peak* feature, We classify all of the samples between peaks to the same class. This grouping of sample classifications offers robustness over the sample-by-sample approach, since oscillatory behavior is persistent over time, but it requires constant amplitude. Obviously, these two features can be combined to create a two dimensional discriminant feature vector. We propose a simple on-line algorithm to detect local extrema and classify based on their features, delineated in Algorithm 1.

Algorithm 1 On-line classification based on peak features

```
  Given time series $s$, with $N$ samples
  $L$ = window length for peak detection
Require: $L$ odd
  $a = \frac{L-1}{2}$
  $Y = \{\mathbf{y}_j = (\Delta t(P_j), \Delta h(P_j) | j$ is the index of feature pairs
  in the training set$\}$
  Set $\theta$ = threshold on $|\Delta h(P)|$
  $i = 0$
  for $n = a + 1$ to $N - a$ do
    $W_P = s_{n-a}$ to $s_{n+a}$
    if $s_n == \max(W_P)$ or $s_n == \min(W_P)$ then
      $i = i + 1$
      $t(P_i) = n$
      $h(P_i) = s_n$
      if $i > 1$ then
        $\Delta t(P_i) = t(P_i) - t(P_{i-1})$
        $\Delta h(P_i) = h(P_i) - h(P_{i-1})$
        $\mathbf{x}_i = (\Delta t(P_i), \Delta h(P_i))$
        if 1-dimensional feature space then
          if $\Delta h(P_i) > \theta$ or $\Delta h(P_i) > -\theta$ then
            Assign samples from $P_{i-1}$ to $P_i$ to the oscillatory class
          else
            Assign samples from $P_{i-1}$ to $P_i$ to the non-oscillatory class
          end if
        else if 2-dimensional feature space then
          Find $\arg\min_j d_{Euc}(\mathbf{y}_j, \mathbf{x}_i)$
          Assign samples from $P_{i-1}$ to $P_i$ to the class of $\mathbf{y}_j$
        end if
      end if
    end if
  end for
```

In the training data set, a window length $L = 5$ was sufficient to correctly identify the peaks of the oscillatory signal without false alarms due to noise. See Figure 8 for

Algorithm 2 On-line classification based on sample differences

```
Given time series $s$, with $N$ samples
$W$ = window size for grouping
$M$ = number of samples in group that must be labeled oscillatory
$Y = \{\mathbf{y}_j = (\Delta s_j, \Delta s_{j-1}) | j$ is the index of feature pairs in the training set$\}$
Feature pairs are labeled with 1 if oscillatory and with 0 if non-oscillatory
Set $\theta$ = threshold on $|\Delta s|$
for $n = 3$ to $N$ do
    $\Delta s_n = s_n - s_{n-1}$
    $\Delta s_{n-1} = s_{n-1} - s_{n-2}$
    $\mathbf{x}_n = (\Delta s_n, \Delta s_{n-1})$
    if 1-dimensional feature space then
        if $\Delta s_n > \theta$ or $\Delta s_n < -\theta$ then
            Set $C_n = 1$
        else
            Set $C_n = 0$
        end if
    else if 2-dimensional feature space then
        Find $\arg\min_j d_{Euc}(\mathbf{y}_j, \mathbf{x}_n)$
        Set $C_n$ = binary label of training feature vector $\mathbf{y}_j$
    end if
    if $n > 2W$ then
        if $\sum_{i=n-2W+2}^{n-W+1} C_i \geq M$ or $\sum_{i=n-W+1}^{n} C_i \geq M$ then
            Classify $s_{n-W+1}$ as oscillatory
        else
            Classify $s_{n-W+1}$ as non-oscillatory
        end if
    end if
end for
```

an illustration of the parameters for the peak-to-peak method. Since the *peak-to-peak* method is based on time between *peaks*, We must wait for the next peak before classifying any samples. This induces *latency* in detection. We formally define latency as the number of samples that must elapse past the current sample before a decision is made on an event. For the peak-to-peak features, the latency is the number of samples between peaks, plus an additional $\frac{L-1}{2}$ samples, which is half the window size, required to detect the most recent peak. On average the peak-to-peak time is 16 samples, and if $L = 5$, the average latency will be 18.

B. Sample Differencing

Although the peak-to-peak height feature provides separation between these two classes, We can reduce the latency by considering the differences between adjacent samples. The training set distributions for the sample differences of each class are seen in Figure 13(d). Again We can select a threshold based on the likelihood ratio of the empirical probability density functions (EPDF) of the differences $\Delta s_n = s_n - s_{n-1}$, given a discrete time series s at time n. One drawback of *sample differencing* is that it is very sensitive to noise. But, as high oscillations persist over a long period in our specific application, We can strengthen the approach by requiring that a set of differences $\{\Delta s_n\}_{n=1}^{k}$ around a point in question exceed the threshold. Although this introduces latency as We must observe future samples before making a decision, the number of points observed are fewer compared to the *peak-to-peak* method. At the peaks of the oscillatory region, the consecutive differences of the signal decrease in magnitude. Therefore, these areas are prone to missed detections. We would like to mark these points with a warning, even though they fall below the threshold. To reduce these missed detections, We require a small number of samples M, in a window of W samples such that $M < W$, satisfy the threshold. At the onset of oscillation, only future samples will satisfy the threshold, while at the end of oscillation only past samples will satisfy it. Therefore, two windows must be considered to prevent failed detection at the edges. One window includes the sample in question and the next $W - 1$ samples, while the other includes the sample in question and the previous $W - 1$ samples. A warning is issued if either window contains M or more points above the threshold, with a latency of $W - 1$ samples. The selection of M is a tradeoff between missed detections and false alarms. If M is low, warnings can be triggered by only a few points, therefore it is recommended that M constitute a majority of points in the window. Requiring a $2/3$ majority worked well in our tests. The *sample differences* procedure is controlled by two parameters, window size W and the adjacent differences parameter M.

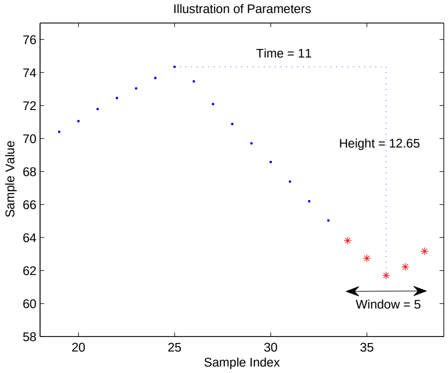


Fig. 8. Parameters for amplitude and interval based detectors.

VI. Multi-feature detection

Heretofore, detection was based on a single feature such as *time between peaks*, and *sample differences* of adjacent observations. Presumably, extending the feature set to more than one feature, the classifier performance can be improved. Once the peaks are detected it is a simple matter to calculate both the *time* and *height* features. Though *height* as a feature has good separation properties on its own, We explore the joint distributions of *peak height* and *peak time* respectively given by, $\Delta h(P_i)$ and $\Delta t(P_i)$ to increase the predictive capacity of the classifier. The benefit of a 2-dimensional feature space is seen in Figure 9. Clearly it can be seen from Figure 13(b), that there is considerable overlap between oscillatory and non-oscillatory regimes using $\Delta t(P_i)$. But, $\Delta t(P_i)$ in conjunction with $\Delta h(P_i)$ separates the oscillatory and non-oscillatory regimes into two distinctly separable classes. Note that the usefulness of this feature space for classification depends on the training set to be representative of the churn in general. It can be seen that the magnitudes of both *time* and *height* are small in the non-oscillatory class, and large in the oscillatory class. In the classification problem involving a single feature, the threshold is a scalar value. In the case of the 2-dimensional feature space, a non-linear decision boundary would be better suited to differentiate the two classes. We applied the *k*-nearest neighbor (KNN) method, with the neighborhood parameter $k = 1$. First, We create a training set which consists of pairs of $(\Delta t(P_i),\ \Delta h(P_i))$, i.e, *peak-to-peak* times and *peak-to-peak* heights. The training set is created using the data where regions of signals are labeled as oscillatory and non-oscillatory. Let the collection of feature vectors be denoted by $\mathbf{y}$. In the testing phase of the algorithm, We assign an example $\mathbf{x}$ belonging to an unknown class by finding the training set feature vector $\mathbf{y}$ closest in Euclidean distance and assign $\mathbf{x}$ to the same class as this "nearest neighbor." In the case of $k > 1$, the k closest $\mathbf{y}$ are found, and the unknown feature vector $\mathbf{x}$ is assigned to the majority class of the k nearest neighbors. The results of the performance of this approach are detailed in the results section. Compared to results using a single feature, the gains due to this approach is modest and We are refining our feature set to improve performance. The method of sample differences has the benefit of low latency, but the sample difference feature is not perfectly separable unlike in the method based on *peak-to-peak* heights. Classification based on a single sample difference feature can be made more effective in a 2-dimensional feature space. Consider now the difference Δs_n at time n, and the previous difference Δs_{n-1} at time $n-1$. The feature space created by these features can be seen in Figure 10. Comparing with the single-dimensional case in Figure 13(d), there is much better separation. This is achieved because the ambiguous differences occur at the peaks of the quasi-triangular wave where the differences change sign. The differences around the peak are small in magnitude and overlap with the differences of the non-oscillatory class. However, the consecutive differences $(\Delta s_n, \Delta s_{n-1})$ will be of opposite sign at the peak of the quasi-triangular waveform because a rising-to-falling transition is taking place. This sign change in the differences Δs_n and Δs_{n-1} as their magnitude becomes small keeps the oscillatory class separated from the non-oscillatory class in feature space. While not linearly separable, classification is achieved with the nearest neighbor method in this case. Algorithm 2 demonstrates classification with *sample differences* in the 1- and 2-dimensional cases.

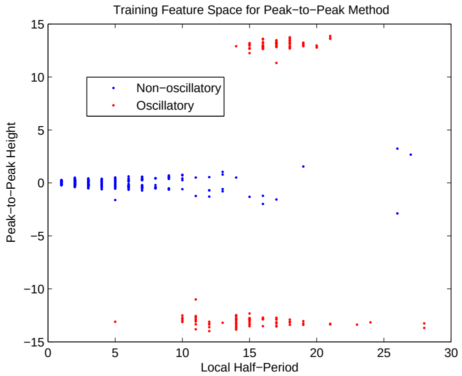


Fig. 9. Feature space constructed from the peak-to-peak time and height features based on the training set.

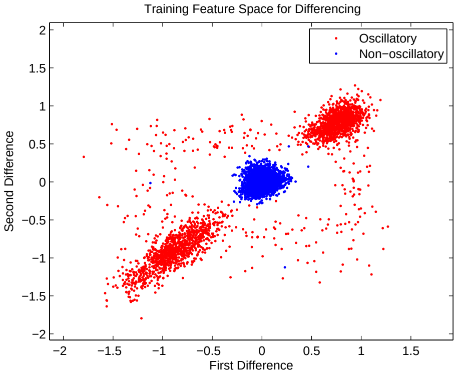


Fig. 10. Feature space constructed from the sample first and second differences.

VII. Results

The discussion heretofore, focused on the characterization of the high oscillation region of the signal via a single and bivariate features extracted from the data. These features include the amplitude of the Fourier coefficients in a local neighborhood (STFT), the *peak-to-peak* time $\Delta t(P)$ and height $\Delta h(P)$, and the *sample differences* Δs_n. The Fourier coefficients and *sample differences* were determined from current and previous data, and therefore can be used to classify on a sample-by-sample basis in real time. *Peak-to-peak* features induced latency in the classification because of the necessity to wait for the upcoming peak. Training sets of known class were used to generate a distribution of the desired feature for each class. Figure 13 shows the distributions for each of the features. The distributions are used to establish a threshold, θ on the feature. We see that the distribution of *peak-to-peak* height has the best separation between classes for this data set. In the single-feature case, We therefore use this feature rather than time between peaks, which has overlapping oscillatory and non-oscillatory distributions. The performance of the algorithms can be compared using the receiver operating characteristic (ROC) curves in Figure 11. These curves were obtained using labeled data over a week of flow-rate measurements, distinct from the training data. The ROC plots the true positive rate (TPR) against the false

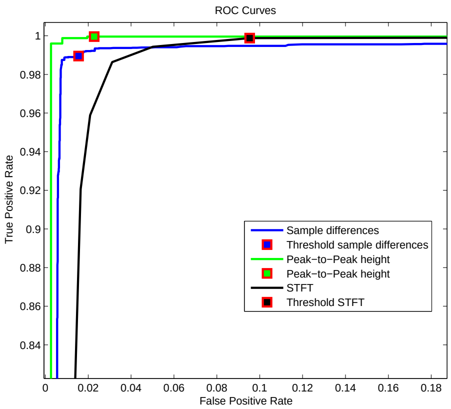


Fig. 11. Receiver operating characteristic curves generated by varying thresholds. The ML point represents the empirical distribution maximum-likelihood threshold.

TABLE I

PERFORMANCE COMPARISON FOR VARIOUS METHODS

Features	**Class. Method**	**Parameters**	**Error Rate**	**TPR**	**FPR**
Peak-to-Peak Features					
$\Delta h(P)$	Threshold	$\theta = 11$	1.9%	97.2%	1.7%
$\Delta h(P)$ $\Delta t(P)$	KNN	$k = 1$	2.9%	96.8%	2.8%
$\Delta h(P)$ $\Delta t(P)$	BF	$\sigma_h = 0.5$ $\sigma_t = 8$ $\eta = 0.01$	1.5%	97.3%	1.1%
Sample Differences					
Δs_n	Threshold	$\theta = 0.44$ $W = 1$ $M = 1$	5.2%	88.9%	3.1%
Δs_n Δs_{n-1}	KNN	$k = 1$ $W = 1$ $M = 1$	7.0%	95.9%	8.0%
Δs_n Δs_{n-1}	KNN	$k = 1$ $W = 10$ $M = 9$	2.9%	95.7%	2.3%

positive rate (FPR). The TPR is the ratio of true positives to actual positives (true positives and false negatives). The FPR is the ratio of false positives to actual negatives (false positives and true negatives). Points closest to the top left represent the best performance, with a high TPR and low FPR. The ROC curves were generated by testing through a range of thresholds, including the Maximum likelihood (ML) threshold obtained from the EPDF estimates. Note that the ML threshold does not offer the best performance, but provides a reasonably good estimate. From the ROC, in the optimal case, *peak-to-peak* height outperforms both *sample differences* and the maximum Fourier coefficient. This result clearly follows from the large separation between the distributions of *peak-to-peak* heights. The tradeoff is the latency involved in determining *peak-to-peak* features. The TPR, FPR, and total error rate are seen in Table I, which compares the performance of various methods. The values in Table 1 were calculated using A testing data consisting of approximately 62,000 samples. Table I compares the performance of single, and multi-feature classifiers. Column 1 is the feature(s) used, column 2 is the classification method, column 3 is the tuning parameter(s) of the classifier scheme, and column 4 is the error rate which is equal to the ratio of misclassifications of both types to total number of samples.

Table I also includes the results for combined feature spaces. The first two rows compare the threshold method using $\Delta h(P)$ to the KNN method in a 2-dimensional space composed of $\Delta h(P)$ and $\Delta t(P)$. Despite achieving the excellent separation in the 2-dimensional case, it is outperformed by simply thresholding $\Delta h(P)$. The threshold was set at 11, very close to the oscillatory heights as can be seen in Figure 13(c). In doing so We assume that the oscillatory training data will represent the testing values well, and by setting a tight threshold, false positives can be avoided. Over the course of testing, the non-oscillatory signal will produce *peak-to-peak* heights that are very different than those from the training set. These erratic parameters will often jump over a threshold evenly spaced between the oscillatory and non-oscillatory classes. The 2-dimensional space suffers because it employs KNN, which does not favor any particular class with its decision boundary. In order to improve the performance in 2-dimensional space, We need to use a classification algorithm that will produce a tighter decision boundary around the oscillatory class features.

Given the unpredictable nature of the non-oscillatory signal, We build a decision boundary around the features of the oscillatory class. Features that fall within the boundary will be classified oscillatory, and those that fall beyond the boundary as non-oscillatory. A simple way to do this is by the radial basis function (RBF) network [10]. The RBF network is an artificial neural network, of which the output φ is a weighted sum of radial basis functions $\rho(||\cdot||)$. Consider our classification problem, with feature vector $\mathbf{x}$ of unknown class and the set of N training set feature vectors $\{\mathbf{y}_i\}_{i=1}^{N}$ from the oscillatory class. The output of the RBF network, with weights set to 1, is

$$\varphi(\mathbf{x}) = \sum_{i=1}^{N} \rho(||\mathbf{x} - \mathbf{y}_i||) \tag{1}$$

The RBF is a function whose output only depends on the distance from a center point, in this case the $\mathbf{y}_i$. The RBF is often the Gaussian function, and the norm in its argument is the Euclidean norm. To apply this method, We put a Gaussian surface over every oscillatory training point in feature space. The sum of these Gaussian "bumps" produces a hill over the oscillatory features, that dies off as We move farther from them. Therefore, when an unknown feature $\mathbf{x}$ is close to the oscillatory region in feature space, the value of $\varphi(\mathbf{x})$ will be relatively large. Setting a threshold on φ, which We will denote η, produces a decision boundary in feature space. We can interpret η as an altitude on the surface created by the sum of basis functions. Features with $\varphi(\mathbf{x}) > \eta$ are classified as oscillatory.

Recall Figure 9, which displays the oscillatory training features. The *time* feature has greater variance than *height*, therefore We do not want to use a radial basis function. A RBF would produce a surface with too much width in respect to the *height* and again We would suffer from excess false positives. We would like a basis function with a sharp drop-off with respect to the *height*, but one with a gentler roll-off in the direction of the more sporadic *time* feature. A bi-variate Gaussian function, with principal axes aligned with the feature space axes, fits this requirement well. The marginal variance of *height*, denoted σ_h must be chosen small relative to the marginal variance of *time*, denoted σ_t. The decision boundary produced by the bi-variate Gaussian basis function is scene in Figure 12. The performance of the basis function (BF) approach is seen in Table I. With these parameters, it outperforms all other *peak-to-peak* methods, though the trade-off is an increase in the free parameters to be selected. The next section of Table I compares classification with a threshold on *sample differences* Δs_n, and with KNN in the 2-dimensional space formed by Δs_n and Δs_{n-1}. As a much noisier feature, both methods with *sample differences* are outperformed by the *peak-to-peak* methods. Despite a larger TPR, the 2-dimensional method has a larger total error rate. Again this is due to the false positives that arise when KNN establishes a decision boundary that does not closely encase the oscillatory features. Outliers of the non-oscillatory signal are often labeled oscillatory for this reason. In this case $W = 1$ and $M = 1$, in which a sample is classified exclusively on its own features, and not on the features of surrounding samples in its neighborhood. On the bottom row of Table I, We provide the results of 2-dimensional *sample differences*, but with $W = 10$ and $M = 9$. This imposes a much stricter condition on oscillation and reduces the false positives. In a neighborhood of 10 samples adjacent to the one in question, 9 of which must test as oscillatory under KNN, for the target sample to be officially labeled oscillatory. Under this condition, KNN in 2-dimensional space performs better. However, the latency in detection is now 9 samples.

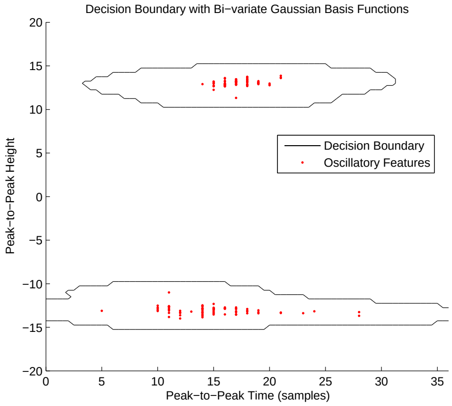


Fig. 12. Oscillatory training features and the decision boundary created with the use of the bi-variate Gaussian basis function. In this case $\sigma_h = 0.4$, $\sigma_t = 3$, and $\eta = 0.01$.

VIII. Conclusions

This paper addressed the important problem of modeling flow rate data in oil and gas production. Petroleum engineering despite being a mature discipline is found inadequate to mathematically model churn in the flow rates. We tackled the problem from the point of view of frequency domain analysis and pattern recognition. Our proposal to bring to bear the science of dynamical systems provided us the tools to uncover the hidden time varying periodicities in the flow rate which helped to invoke nonlinear modeling techniques to separate high oscillation regimes (churn) from non-oscillation regions. Frequency methods are more suitable for understanding and modeling complex data than time-domain based approaches. The application of kernel adaptive filters to detect onset of high oscillation overcame the limitation of linear methods such as the least mean square algorithm. Our methods are computationally light and can be implemented in an on-line setting. As data is being collected from trillions of sensors, large repositories are built to understand the data to harness the trove of information. The approaches We outline in this paper potentially give the ability to analyze the data in near real-time for process interdiction and correction. We plan to further investigate the techniques outlined to tackle other types of sensor data from oil bore wells such as pressures in the sea bed, and temperature gradients at various points in the well. As the sensors are networked, We plan to advance our methods to detect trends, patterns, affinities, and correlations across the network.

References

[1] M. Hill, M. Campbell, Y.-C. Chang, and V. Iyengar, “Event detection in sensor networks for modern oil fields,” in *DEBS ’08: Proceedings of the second international conference on Distributed event-based systems*. New York, NY, USA: ACM, 2008, pp. 95–102.

[2] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.

[3] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence*, D. A. Rand and L. S. Young, Eds. New York: Spinger-Verlag, 1980, pp. 366–381.

[4] P. Grassberger and I. Procaccia, “Measuring the strangeness of strange attractors,” *Physica D: Nonlinear Phenomena*, vol. 9, no. 1-2, pp. 189 – 208, 1983.

[5] T. M. Apostol, *Mathematical Analysis*. Addison Wesley, January 1974.

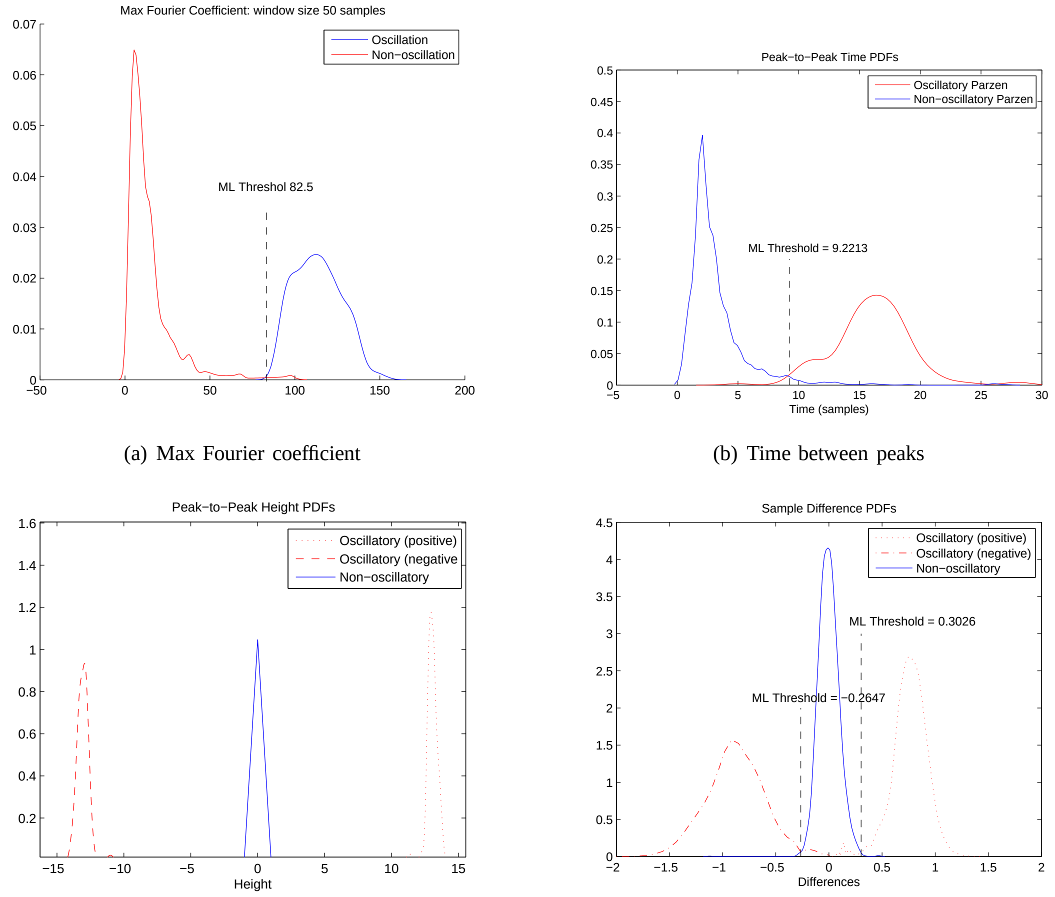
[6] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

[7] J. C. Principe, W. Liu, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. John Wiley & Sons, Inc., 2010.

[8] P. P. Pokharel, W. Liu, and J. C. Principe, “Kernel least mean square algorithm with constrained growth,” *Signal Process.*, vol. 89, no. 3, pp. 257–265, 2009.

[9] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing (4th Edition)*. Prentice Hall, March 2006.

[10] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations with CD-ROM*. New York, NY, USA: John Wiley & Sons, Inc., 1999.



(a) Max Fourier coefficient

(b) Time between peaks

(c) Height between peaks

(d) Sample differences.

Fig. 13. Empirical probability density function estimate by kernel smoothing.