SAS3141-2019

# Condition-Based Monitoring Using SAS® Event Stream Processing

Anya McGuirk, Yuwei Liao, Byron Biggs, Deovrat Kakde, and Joseph Costin

SAS Institute Inc.

## ABSTRACT

Condition-based maintenance (CBM) of critical high-valued machines is a typical Internet of Things (IoT) scenario. Key to CBM is monitoring important machine health parameters, such that maintenance can be based on the perceived condition of the machine, rather than performing preventive maintenance at intervals, where the likelihood of failures between repairs is very small, or performing corrective maintenance by running the machine until it fails and repairing it. Vibration data analysis is a key tool for understanding the internal condition of a machine, often when it is running continuously. This paper illustrates how to use the new digital filters, time-frequency analysis tools, machine learning algorithms available in SAS® Visual Data Mining and Machine Learning, and SAS® Event Stream Processing to monitor the real-time condition of a "live" variable-speed rotating machine by analyzing vibration sensor measurements obtained at sampling rates higher than 10,000 Hz.

## INTRODUCTION

Measurement and monitoring one or more health parameters is a critical component of condition-based maintenance (CBM), the practice of performing maintenance based on the perceived health of a machine. It is now generally accepted that CBM, where possible, is economically superior to the practices of preventative maintenance—conducting maintenance in predetermined, fixed time intervals where the likelihood of failures between repairs is very small—and the practice of *run-to-failure* or corrective maintenance, which refers to running the machine until it fails. CBM is particularly more economical in the case of high-valued machines that are typical in many IoT scenarios. Understanding the internal condition or health of a machine is not trivial, especially when the machine is constantly in operation, as is often the case.

One popular way to monitor machine health is by analyzing vibration data. Vibration analysis requires a vibration sensor on the machine that can take multiple readings per second. The real-time analysis of such data requires analytical tools that can handle a large volume of data and turn this data into actionable insights in seconds to sub-second rates.

The main purpose of this paper is to investigate the following research questions:

- Can our more advanced analytical methods handle the large data volume typical of streaming sensor data?
- What SAS tools are best to detect faults using vibration data?
- Which is more effective: time or frequency domain analysis?
- If the motor runs at different speeds, do we need different models?
- Can we detect faults and use these tools in real time with data streaming at very high sampling rates?

## EXPERIMENTAL SETUP

The importance of vibration data analysis for effective CBM and the data requirements for thorough vibrational analysis make this a perfect test case to address our research questions. To conduct the research, we use the variable speed, three-phase Induction Squirrel Cage Motor equipped with three accelerometer vibration sensors owned by SAS (Figure 1). Using this machine, we run three experiments with the motor running at 25, 35, and 50 revolutions per minute (RPM), respectively.

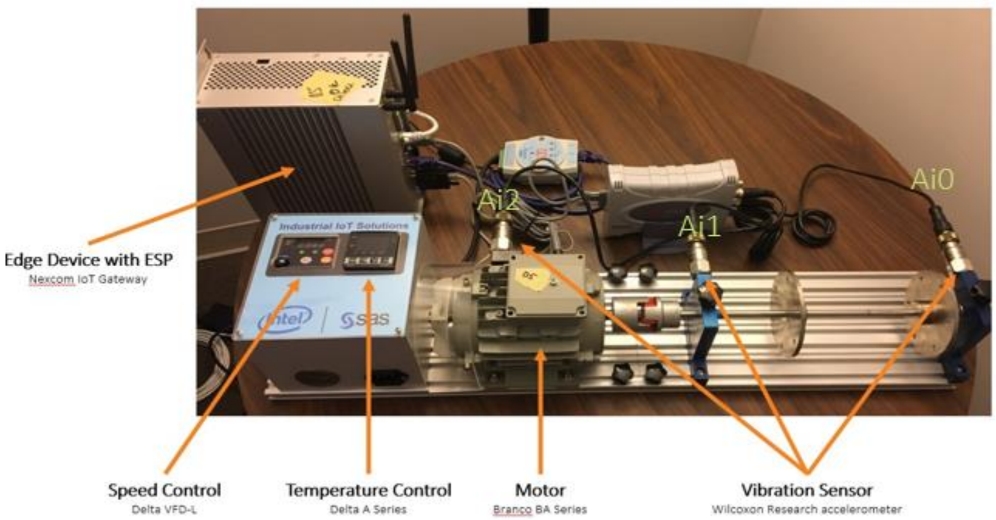



**Figure 1. Variable Speed Squirrel Cage Motor**

While running each experiment, we collect data from the vibration sensors at 12,800 Hz (that is, 12,800 data reads per second). Initially we collect approximately 10 minutes of data with the machine running under its current, presumed non-fault "normal" state. After approximately 10 minutes, we artificially change the state of the motor from this normal condition. To do this, we adjust the black knob on top of the blue plate near sensor Ai1. (Figure 1.) This adjustment effectively changes the alignment of the main motor shaft ever so slightly from its "balanced" state. To understand what happens as this knob is adjusted, see Figure 2. A twist of the knob raises the center rectangular section up slightly, changing the alignment of the shaft that runs through the hole illustrated in the center of this piece. In this way, we cause the shaft to become unbalanced. We then let the machine run in this new state with an artificially induced misalignment.

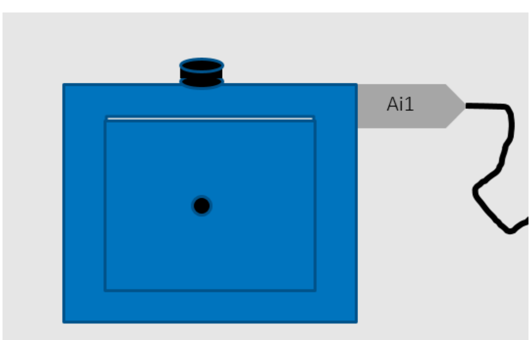



**Figure 2. Knob to Adjust Shaft Balance**

## TIME DOMAIN ANALYSIS OF VIBRATION DATA

In this paper, we focus on the first experiment where the motor ran at 25 RPM. We also focus on the data from the Ai1 sensor as it is located on the blue plate where we intervened and distorted the balance of the motor shaft. Figure 3 shows the time-domain waveform from the "normal" state (blue) followed by the data obtained after we adjust the state of the motor (red). Time (in minutes) is measured on the horizontal axis, and displacement due to vibration, referred to as the amplitude, is measured on the vertical axis.

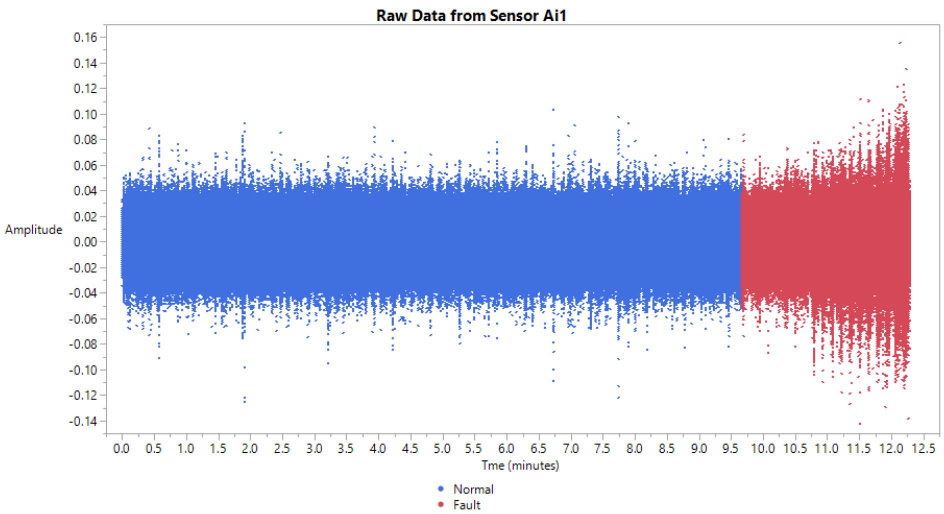



**Figure 3. Raw Sensor Data from Ai1 over Time**

As evident in Figure 3, in the normal state the amplitude of the time waveform is relatively stable. Once the new state is introduced (red), the pattern slowly changes. By minute 11, the change of state appears obvious to the eye.

We start our vibration data analysis by first focusing on the signal waveform, performing analysis in the time domain. When monitoring vibration, the root mean square amplitude (RMSA) of the signal is often tracked. If we let $x_i$ denote the $i^{th}$ vibration sensor measurement, then RMSA is defined as: $RMSA = \frac{1}{N}\sqrt{\sum_{i=1}^{N} x_i^2}$, where *N* is the total number of measurements.

The data in Figure 3 suggests that in addition to tracking RMSA, we might also want to track the kurtosis of the vibration data. Kurtosis, the fourth moment of a distribution, measures the thickness/thinness of the distribution tails (the proportion of the observations away from the mean of the series). In Figure 3, for example, we can observe the tails getting fatter the longer the machine runs in the altered state. Here, we use the standard formula to calculate kurtosis.

To track both the kurtosis and RMSA over time, we calculate both statistics within a one-second window of the signal (*N* = 12800 observations), and then recalculate their values by sliding the window half a second (window overlap ratio = 0.5). The windowed estimates of RMSA and kurtosis over time are shown in Figure 4. To determine where an alert might be thrown, we use the first 75% of the "normal" data to estimate the standard deviation of each series. We then draw the +/- 3-sigma boundaries on the respective graphs. (See red lines in Figure 4.) Based on the 3-sigma thresholds, an alert would likely be thrown at 10.78 minutes as illustrated by the vertical black line in Figure 4.

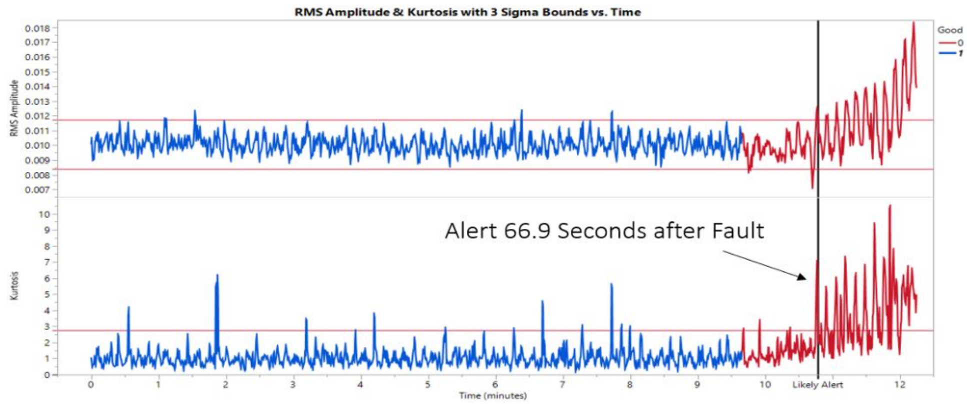



**Figure 4. Root Mean Square Amplitude and Kurtosis over Time with 3 Sigma Bounds**

If we overlay the time domain alert on the original data (Figure 5) we can see that the alert happens just before our eye would have noticed a problem.

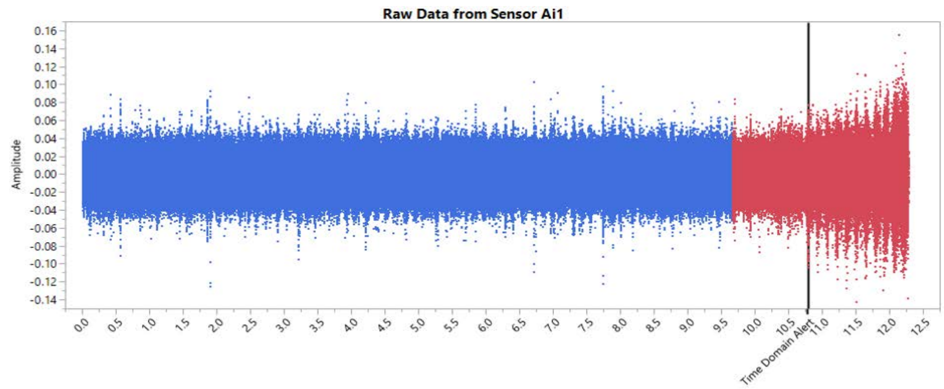



**Figure 5. Time Domain Alert**

## MONITORING SPECIFIC FREQUENCIES USING DIGITAL FILTERS

Some signals are better understood in the frequency, rather than time, domain. Vibration signals from rotating equipment are a perfect example. Rhythmicity or cycles arise from components or parts that rotate. While an in-depth discussion of the relationship between the time and frequency domain is beyond the scope of this paper, Figure 6 provides the intuition that is needed to understand the analysis that follows. Thus far, we have focused on analyzing the amplitude of the waveform in time-domain as seen in Figure 6. As illustrated in this same figure, the time waveform can be decomposed into the frequencies that it consists of to obtain its spectrum.

The (discrete) Fourier spectrum describes the magnitude of the various frequencies comprising the signal. The frequencies that one expects to see in the data depend on the characteristics of the machine being monitored. Motor characteristics like the speed of rotation or RPM, the diameter of the rotating components such as the shaft, and the size and number of ball bearings all affect what frequencies will be seen in the data. For example, to monitor the state of the inner-race that the ball bearings of our squirrel cage motor ride on, you should monitor frequencies around 53 Hz (assuming that the motor is running at 25 RPM). This frequency is derived based on the fact that the machine has seven ball-bearings that are 12mm in diameter, with a pitch diameter of 32mm, and a contact angle of 15 degrees. Similarly, you can use the motor design and specifications to derive which frequencies need to be monitored to verify that other components are working properly. Once the frequencies of interest are identified, they can be monitored using a digital filter.

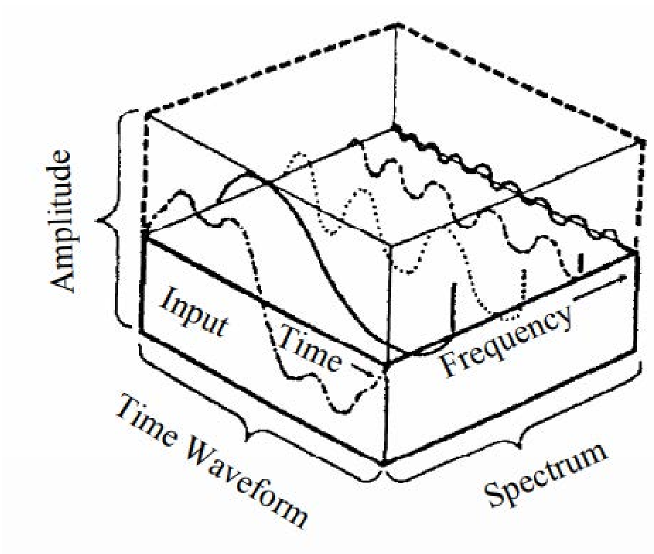



**Figure 6. The Relationship between the Time and Frequency Domains**

A digital filter takes a signal (input data), selectively chooses certain frequency components from the input, and produces a new filtered output series. The Butterworth and Chebyshev Type I and II digital filters are currently available in SAS. For our shaft imbalance experiment, we use a highpass Butterworth filter. The highpass filter we implement is designed to capture the frequencies 3840 Hz and higher from our original raw sensor data ($x_t$). The new filtered data is denoted $y_t$.

Why did we choose this frequency band for our experiment? Because we did not know the exact consequences of our intervention on the state of the machine, we first analyzed the raw sensor data (in Figure 3) using Short-Time Fourier Transform (STFT) available in the Time Frequency Analysis (TFA) package in the SAS® Visual Forecasting product on SAS® Viya®. STFT takes a window of data (hence short-time) and computes the discrete Fourier transform of the windowed signal, which tells us which frequencies are most dominant in the data in that window of time. Then the window slides in time, and the Fourier spectrum of the next window of data is computed. For our analysis, we used a window size of 4096 observations and then advance 2048 observations before computing the next STFT. The results of the STFT are shown in Figure 7.

The colors in the figure indicate the power/magnitude of the various frequencies (vertical axis) versus time (horizontal axis). The brighter the yellow color, the higher the power/magnitude. The blue vertical line in Figure 7 indicates where we intervened to change the state of the motor. The horizontal line indicates the cutoff frequency for our highpass filter (3840 Hz or 0.6 normalized frequency). A significant change in the magnitude of the higher frequencies after our intervention is clear in Figure 7. We chose our filter cutoff frequency based on this observation. The output series from the digital filter ($y_t$, in yellow) and the original sensor data ($x_t$, in green) are shown in Figure 8. The stability of both the original sensor data and the filtered series before the intervention are evident. The change in model state following our intervention is evident almost immediately.

To compare the change in the filtered (high frequency) series relative to the original series, we calculate the ratio (R) of their relative root mean square amplitude as:

$\boldsymbol{R} = \frac{\frac{1}{N}\sqrt{\sum_{i=1}^{N} y_i^2}}{\frac{1}{N}\sqrt{\sum_{i=1}^{N} x_i^2}}$ using windows of size $N$= 512 observations. See Figure 9 for the full ratio series and Figure 10 for a zoomed view of the series, where we show where a likely fault would have been thrown. Figure 10 illustrates how the alert based on the ratio RMS amplitude is much earlier than the alert based solely on the time domain analysis (kurtosis and RMSA). Both the STFT results (Figure 7) and the results based on the filtered series (Figure 8 and Figure 10) indicate that considering the frequency domain can be very fruitful, especially if you know what frequency range should be monitored.

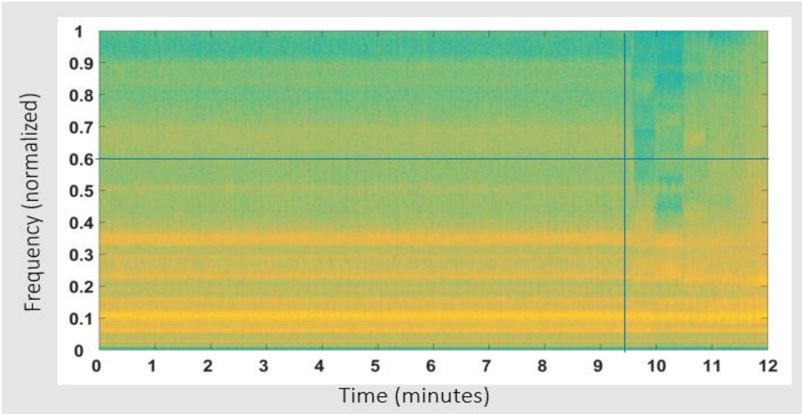



**Figure 7. Short-time Fourier Transform on Raw Sensor Data (Ai1)**

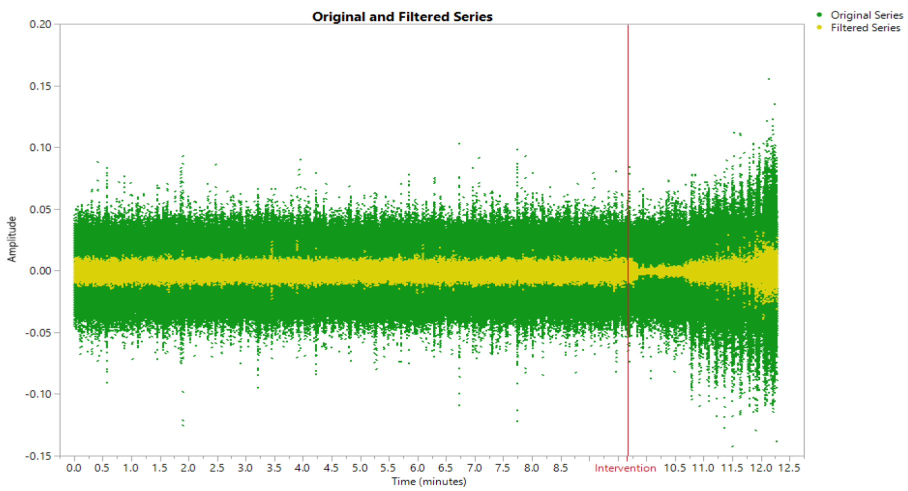



**Figure 8. Raw Sensor Data (Green) and Filtered Series (Yellow)**

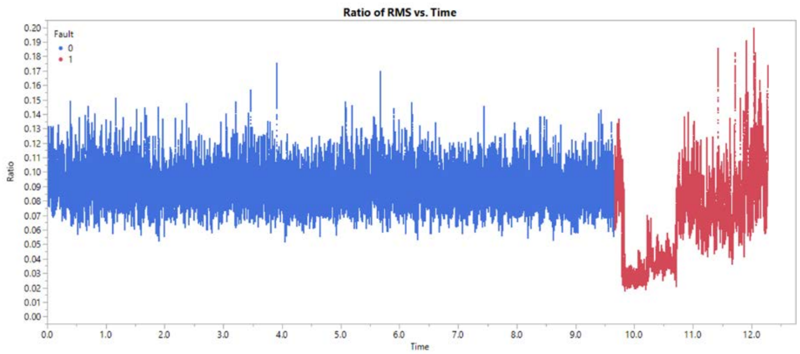



**Figure 9. Ratio of RMS Amplitude: Filtered to Original Series**

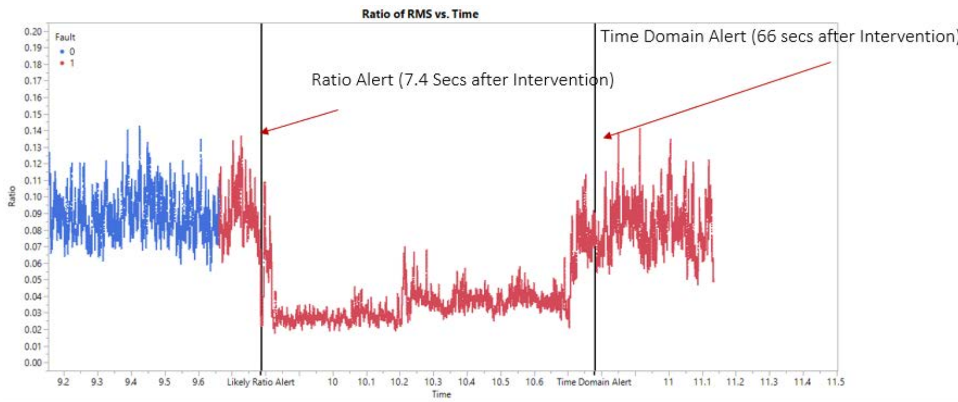


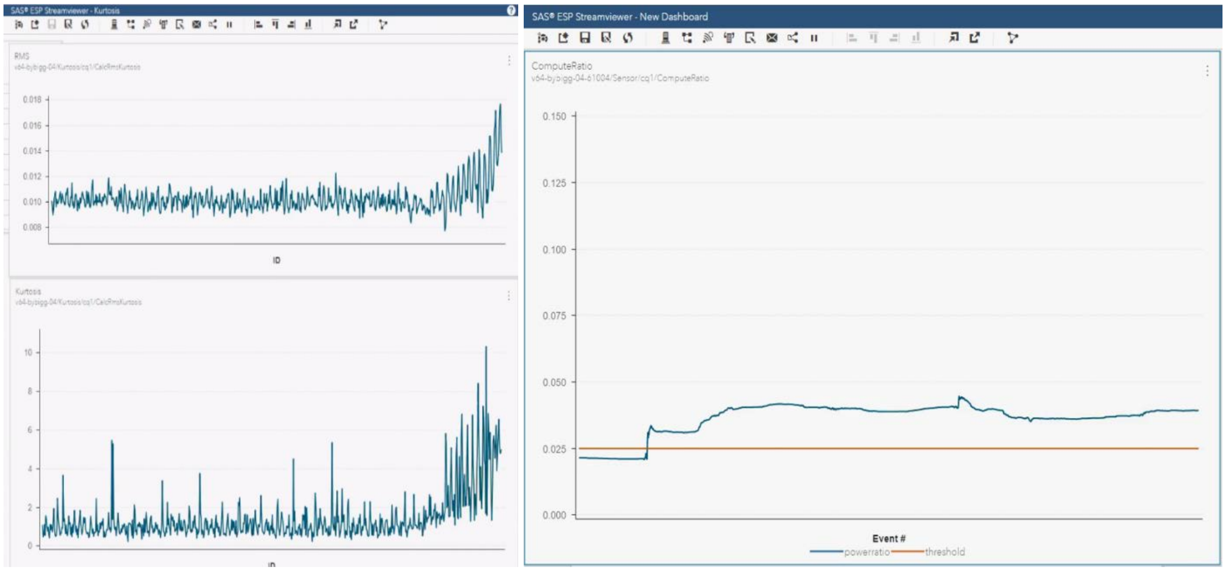

**Figure 10. Likely Alert Based on Ratio of RMS Amplitude**

## REAL-TIME ANALYSIS USING SAS EVENT STREAM PROCESSING

Analysis of the window estimates of both RMS amplitude and kurtosis is possible in SAS Event Stream Processing even when data is streaming at very high frequency such as 12,800 Hz. While it is impossible to show a demonstration of this happening in real time here, Display 1 (left panel) shows a screen capture of a working Streamviewer window. Streamviewer is a graphical user interface that allows the user to visualize streaming events.

You can easily add threshold values and program SAS Event Stream Processing to throw an alert if the threshold is exceeded. Note though that the Streamviewer window statistics are not calculated using overlapping windows. Consider the case here where we use a window size of 12800 observations. In this case, 12800 observations are collected as the data are streaming in, the statistic is calculated, and the data discarded. The next 12800 observations are then collected, and the statistics recalculated. As Display 1 (left panel) shows, the graph is not much different from the overlapping window statistics calculated in SAS Viya (see Figure 4).

The Butterworth digital filter is available in SAS Event Stream Processing, so we also implement the highpass filter described above. We then calculate and display the ratio of RMS amplitude of the filtered to unfiltered series on streaming data in Streamviewer. (See Display 1 right panel.) In SAS Event Stream Processing, it is easy to have an alert triggered once the ratio exceeds a specified threshold. In both cases, demonstrated in Display 1, the high data streaming rate is easily handled.




**Display 1. SAS® ESP Streamviewer: Streaming RMS Amplitude and Kurtosis (Left), Ratio RMS Amplitude Filtered to Non-Filtered (Right)**

## MONITORING THE WHOLE FOURIER SPECTRUM

Instead of, or in addition to, monitoring a frequency band, you can monitor the whole Fourier spectrum to look for *any* changes in the spectrum. To do this, observe the motor operating in a normal or fault-free state, compute a "normal" or reference Fourier spectrum, and then compute the spectrum for incoming data and look for differences from the reference.

The process of determining the reference spectrum is outlined in Figure 11. The reference spectrum will be computed offline by analyzing vibration sensor data under "normal" conditions.

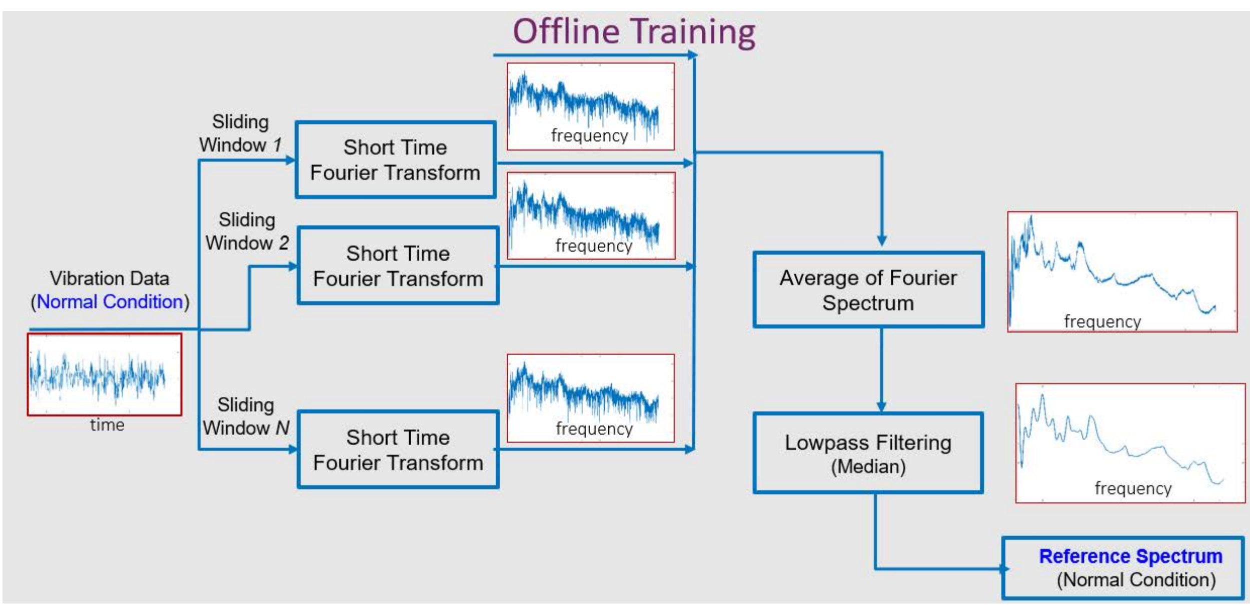



**Figure 11. Estimating a Reference Spectrum**

For our experiments, we used the first minute of data under normal operation at each speed to derive a reference spectrum. For the 25 RPM data, we calculate the STFTs as described earlier (windows of length 4096, with overlap 2048). We then average the STFTs computed over one minute (375 windows) to get the average Fourier spectrum. We follow by smoothing this spectrum using a median filter. The result is our 25 RPM reference spectrum. The averaged and final reference spectrum are shown in Figure 12.

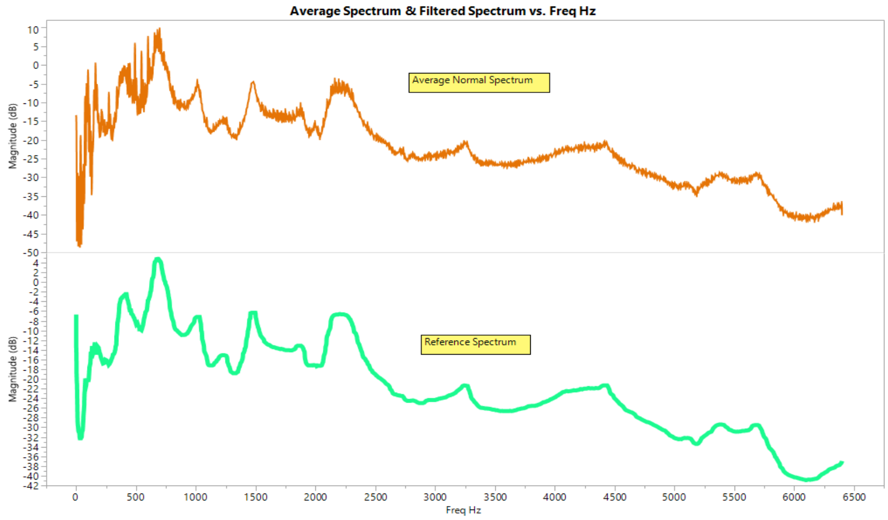



**Figure 12. Average and Reference (Smoothed) Fourier Spectrum**

A procedure very similar to that outlined in Figure 11 is performed as the data streams in. The only difference is that we average Fourier spectrums over 20 seconds rather than a minute. These average spectrums are then smoothed with a median filter and compared to the reference spectrum.

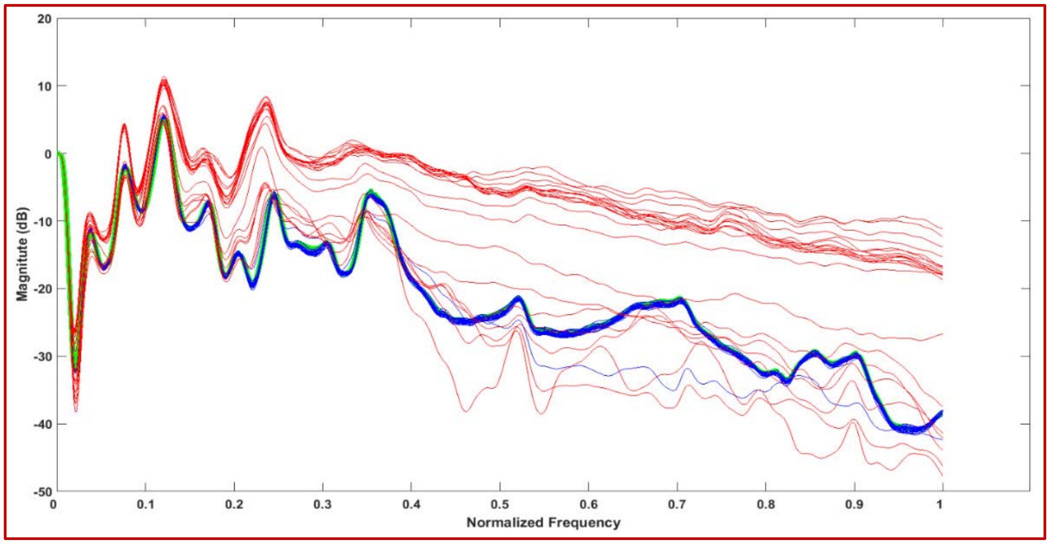



**Figure 13. Averaged Fourier Spectrum (20 secs.) Versus Reference (Blue=Normal, Red=Intervention)**

All the averaged spectrums of the streaming data are shown, along with the reference spectrum in Figure 13. Note that the reference spectrum is shown in green, all the spectrums computed during the normal operation are shown in blue, and those computed after the intervention are shown in red. The reference spectrum is quite hard to see because so many of the normal data spectrums lie directly on top of it. Note also that only one normal (blue) spectrum lies away from the reference, whereas most if not all the red non-normal spectrums lie clearly away from the reference.

To effectively monitor the differences in spectrum on streaming data, we need a useful measure of distance from the reference spectrum. Here we introduce two measures:

- Mean Absolute Distance: $\boldsymbol{D_{mean}} = \boldsymbol{Mean}(|\boldsymbol{S_{in}} - \boldsymbol{S_{ref}}|)$
- Hausdorff Distance: $\boldsymbol{D_{Hausdorff}} = \boldsymbol{D_H}\ (\boldsymbol{S_{in}}, \boldsymbol{S_{ref}})$ where $\boldsymbol{D_H}(\boldsymbol{X}, \boldsymbol{Y}) =$

$$\boldsymbol{max}\left(\sup_{x\in X}\inf_{y\in Y}\boldsymbol{d}(\boldsymbol{x},\boldsymbol{y}), \sup_{y\in Y}\inf_{x\in X}\boldsymbol{d}(\boldsymbol{x},\boldsymbol{y})\right)$$

where $\boldsymbol{S_{in}}$ and $\boldsymbol{S_{ref}}$ are points on a streaming spectrum and the reference spectrum, at a given frequency, respectively. (See Figure 14.) Note that the Hausdorff distance is simply the largest of all the distances from a point on one spectrum to the closest point on the other.

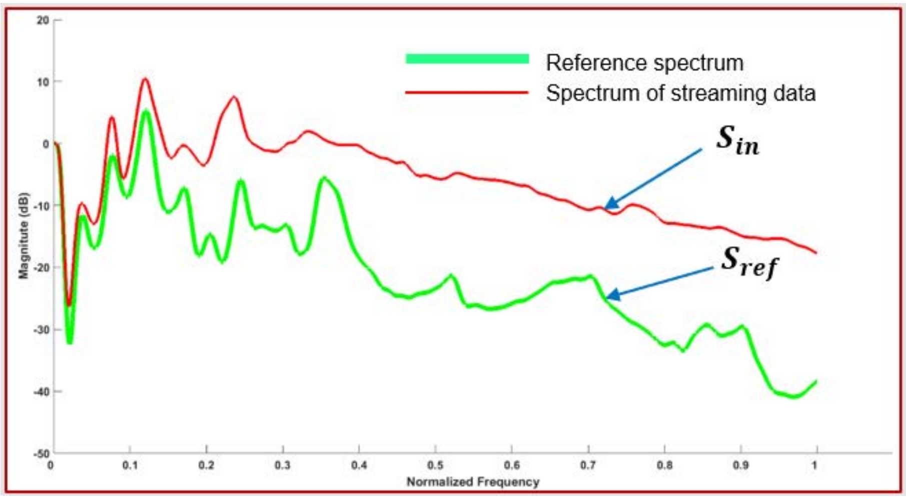



**Figure 14. Calculating Distance from Reference Spectrum**

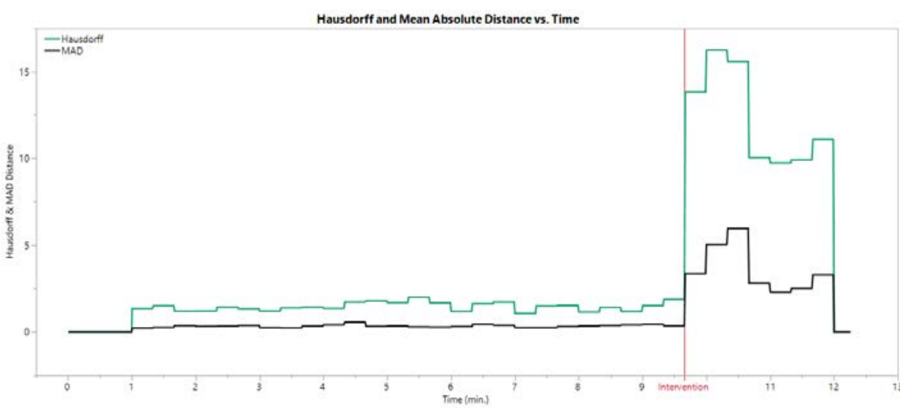



**Figure 15. Distance Measure Results**

Figure 15 shows the results of the distance measures for all the 25 RPM data. The Hausdorff distance is shown in green and the mean absolute distance in black. Both distance measures detect the change in model state only a half a second after the intervention! As with the time domain analysis and the digital filter, we successfully implemented the entire workflow just outlined on SAS Event Stream Processing. Because the streaming version shows a figure identical to Figure 15, we do not show it here.

## MONITORING THE WHOLE FOURIER SPECTRUM BY SEGMENTS

In practice, once a change in the spectrum is flagged, the machine operator would likely want to investigate further to figure out which frequencies or frequency bands were the likely source of the shift in the spectrum. Rather than whole-spectrum monitoring at once as we did above, they might be interested in segment-by-segment monitoring of the whole spectrum. We illustrate how this can be done.

To monitor the entire Fourier spectrum by frequency band segments, our first task is to determine the frequency segments. There are several ways this can be done. First, the segmentation can be done based on frequencies most relevant given the motor design. Second, we could look at the average spectrum under "normal" or fault-free, stable conditions and segment it based on common sense. Because we are not vibration or machine experts, we decided to use statistics to help us come up with the segments.

We tried segmenting the "normal" Fourier spectrum based on several different representations of it (such as power, magnitude, cumulative power over frequency, and cumulative magnitude over frequency). The segmentation based on cumulative magnitude gave us what seemed to be the most logical segmentation results. First, sections of the spectrum with higher power were largely in the same segment. Further, the segmentation results were quite similar across the three different speed experiments. Lastly, our ability to detect the change in model state was similar no matter which experiment's segmentation results were used. Thus, we segment cumulative (over frequency) magnitude.

To segment the cumulative magnitude, we used the SEGMENTATION function in the Time Series Analysis (TSA) package in the TSMODEL procedure (Leonard and Trovero 2014). We approximated the cumulative magnitude curve for the 50 RPM experiment using thirteen linear segments. Figure 16 shows these 13 segments on both the cumulative magnitude and power of the normal spectrum for the 25 RPM experiment.

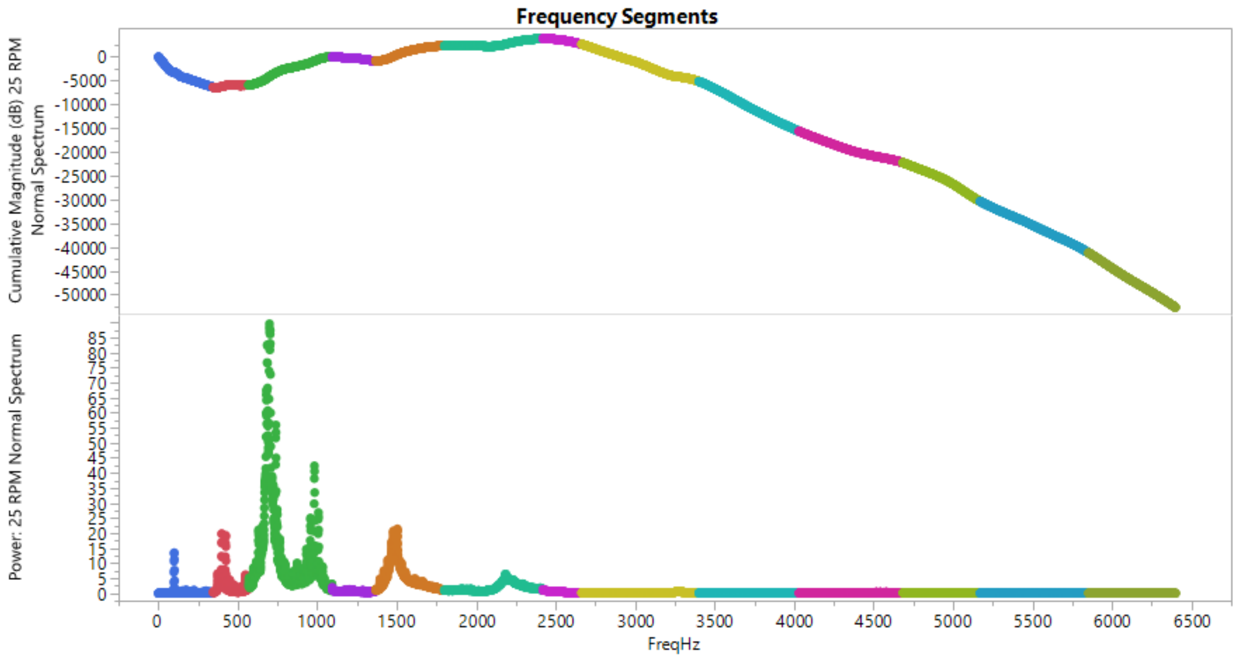



**Figure 16. Frequency Segments of Cumulative Magnitude (dB) 25 RPM Experiment**

Once the frequency segments to monitor were identified, we began our offline training. The offline training here begins like the offline training for monitoring the whole spectrum at once. (See Figure 17.)

1. We begin by performing STFT on the "normal" data using a window size of 4096 observations and then skipping ahead 2048 observations.
2. We then calculate a moving average of 125 STFT windows (20 seconds). For each of these 20 second average spectrums, we calculate the average spectrum magnitude in each of the 13 frequency segments.
3. Finally, we evaluate how the magnitude in each segment varies across time under the "normal" condition.

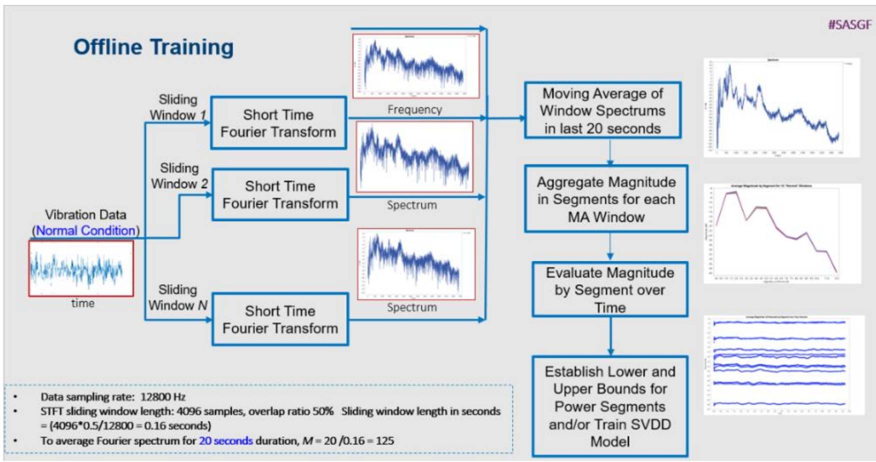



**Figure 17. Offline Training: Monitoring Frequency Segments**

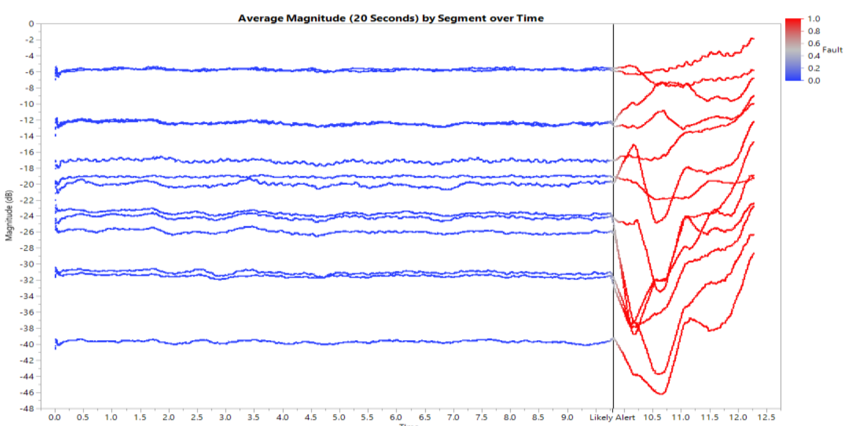



**Figure 18. Average Magnitude in 13 Frequency Segments over Time**

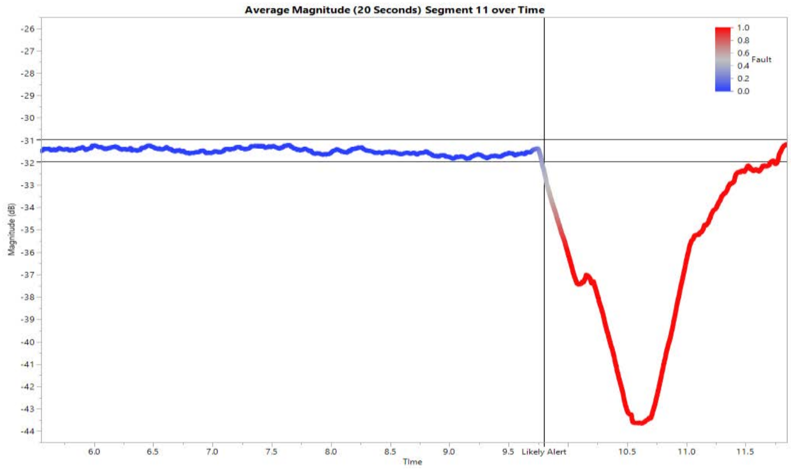



**Figure 19. Average Magnitude in 5170-5852 Hz Segment with +/- 3 Sigma Bounds**

The bottom-right small graph in Figure 17 shows the relative stability of the magnitude of the averaged spectrums in the 13 segments over time under the "normal" condition. Upper and lower bounds are easy to establish, using, for example, a +/- 3-sigma rule for each segment. Figure 18 illustrates how the magnitude in these segments varies over the whole 25 RPM experiment. Changes in many of the segments immediately following the intervention are obvious. Note that the transition from blue (normal) to red is gradual as the samples in between will initially (20 seconds) have a mix of the "normal" and post intervention data. Figure 19 shows monitoring results from segment 11 (5170-5852 Hz) using the +/- 3-sigma rule. An alert is triggered in just over 7 seconds following the intervention.

Is there a way to monitor all the segments together? Yes, we did this when we monitored the whole spectrum earlier!

What happens if we run the motor at another speed? Will the whole spectrum and spectrum segment monitoring approaches change? Clearly, as we change the rotational speed of the motor, we expect the Fourier spectrum to change. Thus, the average magnitude in the various segments will also change. This does not render the approaches invalid. It just means that you have to compute a new reference spectrum and/or determine new bounds on the frequency segments. This is a relatively simple thing to do.

Is there any way we can avoid having to change reference spectrum and/or bounds as the machine conditions change? One possible approach is Support Vector Data Description (SVDD) (Tax and Duin 2004). SVDD can theoretically handle multiple "operating modes". Briefly, SVDD is a one class classifier. In our case, the one class is the "normal" operating condition and in our experiments, normal includes running the machine at 25, 35, and 50 RPM (multiple operating modes) when everything is "normal". Intuitively, SVDD finds a minimum radius boundary around the data characterizing the normal condition(s).

Then as additional data are generated, the model assesses whether the new data are close enough to the "normal" operating condition description. SVDD can handle multivariate data and the boundaries around the data can be very flexible if a kernel function is used. To see this, consider Figure 20, where we illustrate SVDD results for a system that is assumed to have three different operating modes and only two sensors.

- The normal operating data are shown in the left graph.
- The SVDD results are shown in the right graph.

- The areas of normal operation as characterized by the SVDD model are shaded in dark gray. Any new data streaming in that lie outside these gray areas would be considered NOT "normal".

In theory, a single SVDD model could capture the normal condition of our motor running at multiple speeds. How well the model works in practice depends on whether the non-normal modes at the different speeds are also outside all the normal modes (at all speeds).

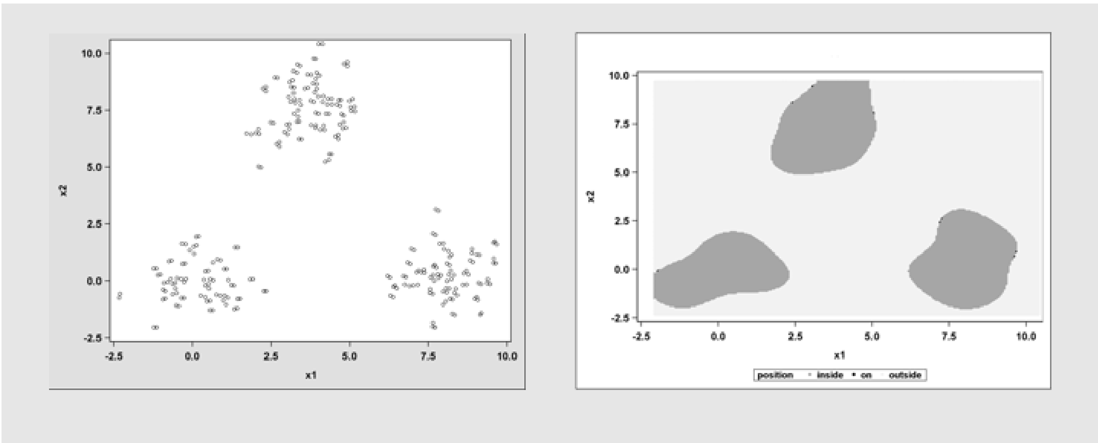



**Figure 20. SVDD Under Multiple Modes of Normal Operation**

To train an SVDD model for our three experiments, we use 75% of the normal data for each speed. The value of Gaussian bandwidth parameter plays an important role in getting a good training data description. We used the SVDD procedure available in SAS Visual Data Mining and Machine Learning. The SVDD procedure supports Mean (Chaudhuri et.al 2017), Mean2 (Liao et.al 2018) and Trace (Chaudhuri et.al 2018) criterion for bandwidth selection. We use the Trace bandwidth selection criterion in this analysis. We then score the model on all three data sets to assess model performance to see how quickly it can find where we intervened at each speed. Note that since we model all three experiments together, we just have one critical value for all the speed experiments.

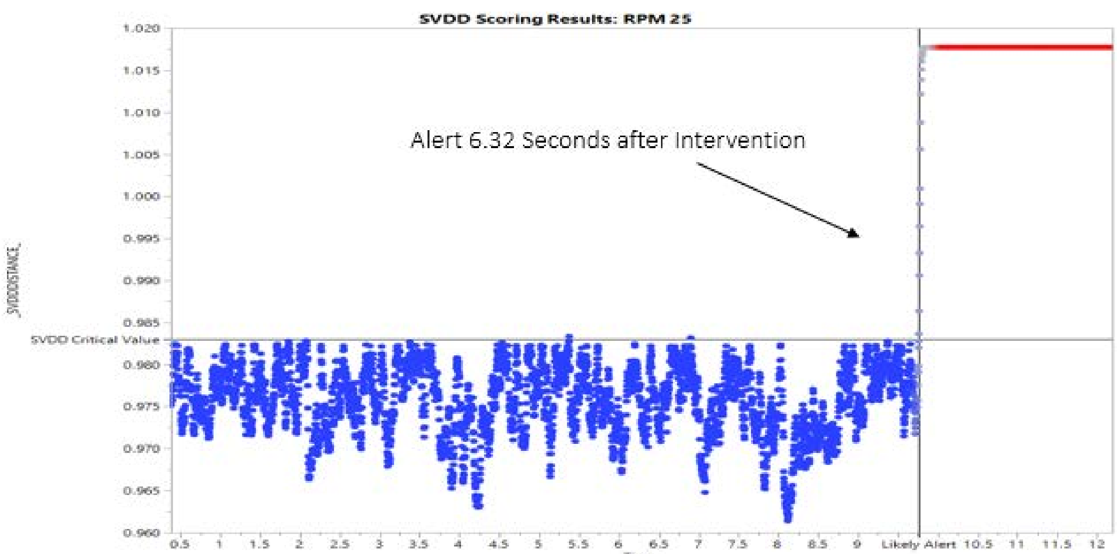



**Figure 21. SVDD Results 25 RPM Experiment**

Figure 21 shows the performance of our SVDD model for the 25 RPM experiment.  It alerts that there is a change from the normal condition 7.6 seconds after the intervention.  Notice that the alert comes when the spectrums being averaged contain a mix of the "normal" and "changed" condition!

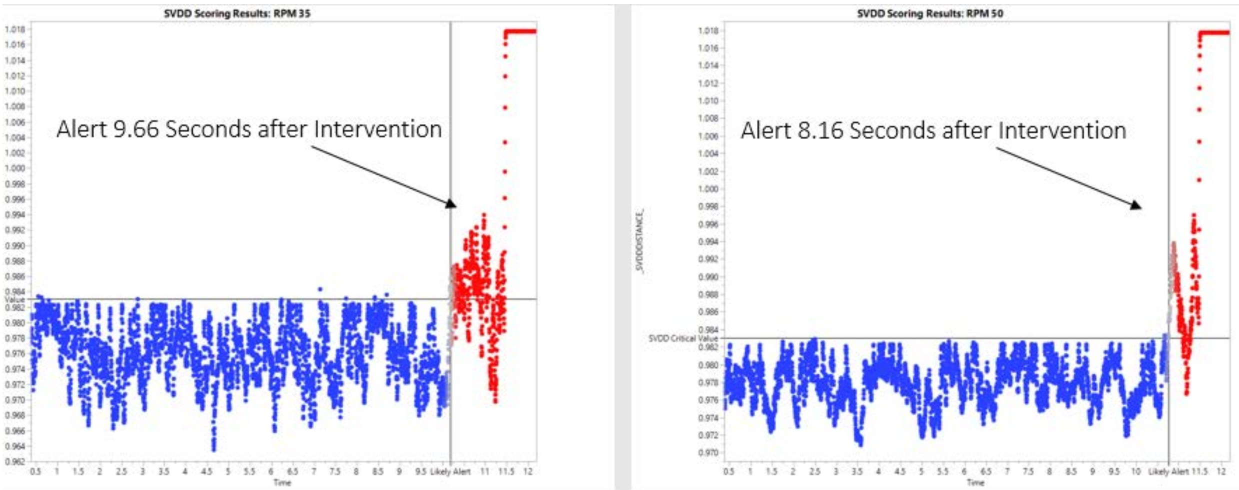



**Figure 22. SVDD Results 35 RPM (Left) and 50 RPM (Right)**

The SVDD results for the other two experiments are shown in Figure 22.  The results are like those in the 25 RPM experiment—quite impressive.  These results suggest that if you train an SVDD model using normal conditions from all speeds at which the motor is to be run, the model should be able to detect changes in normal conditions.  Whether the model remains successful for other non-normal conditions depends on how different from the normal operation at any speed these non-normal conditions are. How could one assess whether this was true or not (presuming other fault data are available)? ALL we need to do is to visualize our 13-dimensional data!

Fortunately, there have been some interesting developments in tools designed to help visualize multidimensional data. The TSNE procedure in SAS VDMML implements t-distributed Stochastic Neighbor Embedding—one such method of visualizing multidimensional data (Van der Maaten and Hinton 2008). Looking at the 13 variables used to train and score our SVDD model will help understand the performance of the SVDD model shown above.  Recall these 13 variables describe the average magnitude of the Fourier spectrum in the different frequency segments.  We use TSNE procedure to visualize these 13 variables in 2 dimensions. The results are shown in Figure 23.

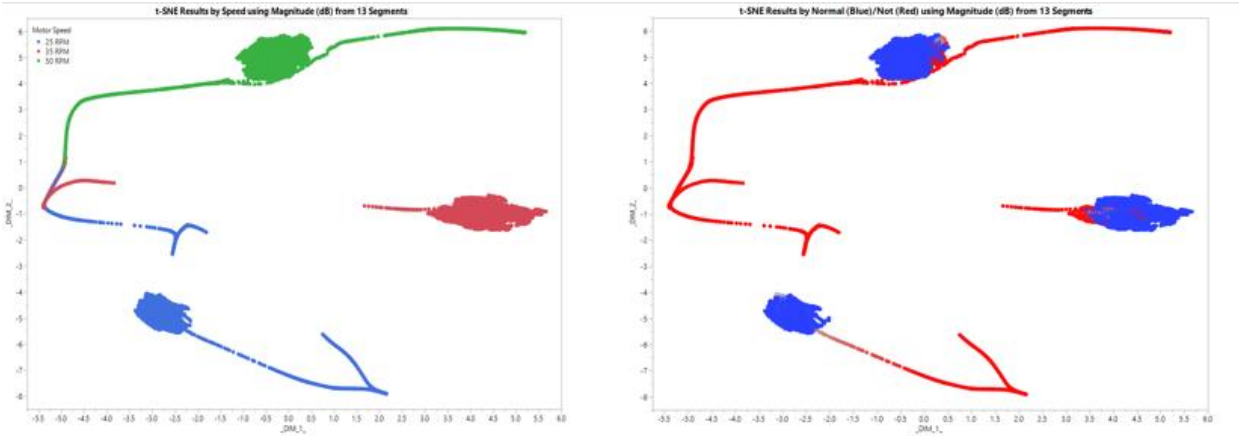
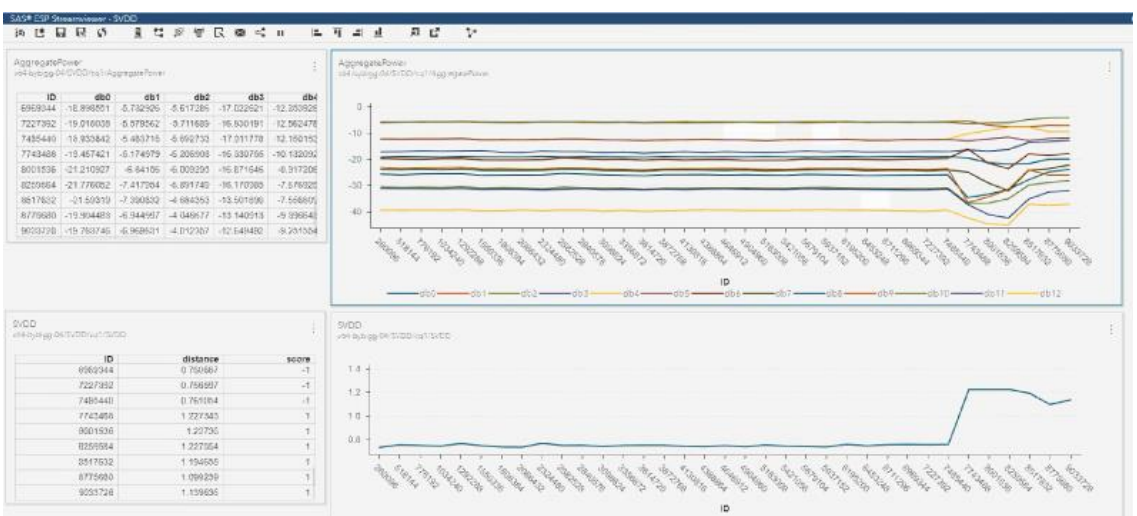



**Figure 23. t-SNE Results by RPM (25 Blue, 35 Red, 50 Green) on Left; Normal (Blue)/Not (Red) on Right**

The left panel in Figure 23 shows the 2-D visualization of our data by speed: Blue = 25 RPM, Red=35 RPM, and Green=50 RPM. Interestingly, most of the data from each of the RPM experiments lie in an almost circular data cloud. But then there are some odd tails! A look at the right panel shows exactly what the tails are about. They are the data points post intervention! Although the data points are not completely visible, you can see that some of the post intervention data for the 35 and 50 RPM experiment do lie in the "normal" data cloud—explaining the slightly worse SVDD results for these experiments. Although we cannot show an animation here, if you animate the above graphs by time, you can see an almost circular motion in the dense circles. Then once the intervention happens, the data shoots away from the "normal" area. The TSNE procedure provides valuable insight into what the normal or stable, fault-free motor conditions look like for different motor speeds and why the SVDD model works well for the intervention in these experiments.

Once again, we can do all the above real-time analysis using the SAS Event Stream Processing with data streaming at 12800 reads per second. Display 2 is a screen capture of the Streamviewer of the Frequency Segment Monitoring and the SVDD model scoring results as the data stream in. The SVDD model is trained offline and the incoming data scored using an ASTORE.



**Display 2. ESP Streamviewer: Frequency Segment Monitoring and SVDD Score Results**

## CONCLUSION

This paper explores the effectiveness of some of the more advanced SAS analytical tools to analyze sensor data streaming at high sampling rates. Given the large volume of data needed to study machine vibrations and the importance of vibration analysis in assessing the health of high-valued machines for condition-based maintenance, we focused on vibration data. We used a variable speed, three-phase Induction Squirrel Cage Motor equipped with three accelerometer vibration sensors. Using this machine, we ran three experiments with the motor running at 25, 35, and 50 revolutions per minute (RPM). First, the motor was run in its usual stable state or under "normal" conditions, and then we intervened to slightly change the adjustment on the motor shaft. We then proposed several methods to analyze the vibration data with the goal of early detection of the changed state due to the intervention.

Initially we focused on time domain analysis and monitored the Root Mean Square of the Amplitude (RMSA) and the kurtosis of the raw vibration sensor series using moving windows of data over time. We detected the change in motor state just over one minute after the intervention in real time using SAS Event Stream Processing. Next, we illustrated how we could detect the change in model state much faster with a combination of frequency and time domain analysis. We demonstrated how to isolate the frequency band of interest using the new Butterworth digital filter and then used the ratio of RMS amplitude of the filtered time series relative to the original signal to detect change. This ratio was able to detect the machine change of state in under eight seconds. This vibration analysis technique, demonstrated in real time in SAS Event Stream Processing, will be very useful in those cases where analysts know exactly which frequencies need to be monitored based on the machine design and its operating parameters.

Next, we switched entirely to the frequency domain. Here we proposed a method of monitoring changes in the entire Fourier spectrum using Short-Time Fourier Transform (STFT) and a novel, relatively simple distance measure (Hausdorff). Both STFT and the distance measures were easy to implement on streaming data in SAS Event Stream Processing, and this method detected the change in model state under one second after the intervention. This method has the slight inconvenience of requiring recalibration of the reference spectrum whenever the motor condition (speed or otherwise) changes.

To help analysts understand the specific frequency segments that seem to be changing, we propose another method of monitoring the full Fourier spectrum, but now we monitor the spectrum in frequency segments. We propose several techniques for segmenting the spectrum, including using the SEGMENTATION function in the Time Series Analysis (TSA) package in the TSMODEL procedure. Once the segments are determined, the monitoring technique uses STFT. The results from STFT are aggregated by segment to greatly reduce the data required in analysis. We demonstrate how to monitor the average magnitude of the frequency in each segment using simple threshold limits and/or train an SVDD model using the "normal" operation data and then score on incoming data. We show how both the individual segment monitoring and SVDD model can be implemented in real time in SAS Event Stream Processing and how they both detect the change in motor state very quickly. A nice advantage of the SVDD model is that the same model can be used to monitor the motor running at multiple speeds if normal operating data at each of these speeds are used in training.

The surprisingly effective performance of the SVDD model at all motor speeds examined in this study made us curious about how disparate the normal-state data clouds were at the different speeds and where the data from the altered model state were relative to the normal states. To help us understand what was happening in the 13-dimensional space occupied by our data, we used a new TSNE procedure at SAS that allows you to look at high-dimensional data in two or three dimensions. The results showed clearly why SVDD worked so well. An obvious extension of this work is to expand the experiments to include more speeds and include various load conditions and other faults to see if the success of the current analysis generalizes under more scenarios.

The clear conclusion of this paper is that some very advanced analytical techniques can be used with SAS Event Stream Processing to successfully monitor machine health with data streaming online at high sampling rates.

## REFERENCES


Chaudhuri, Arin, et al. 2017. "The Mean and Median Criterion for Automatic Kernel Bandwidth Selection for Support Vector Data Description." In *2017 IEEE 17th International Conference on Data Mining Workshops* (ICDMW). Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Chaudhuri, Arin, et al. 2018. "Sampling Method for Fast Training of Support Vector Data Description." In *Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS)*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Chaudhuri, Arin, et al. 2018. "The Trace Criterion for Kernel Bandwidth Selection for Support Vector Data Description" Available https://arxiv.org/abs/1811.06838.

Leonard, Michael, and Michele Trovero. 2014. "How to Separate Regular Prices from Promotional Prices." *Proceedings of the SAS® Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings14/SAS212-2014.pdf.

Liao, Yuwei, et al. 2018. "A New Bandwidth Selection Criterion for Using SVDD to Analyze Hyperspectral Data." In *Proceedings of the 2018 Defense and Security Symposium*. Bellingham, WA: SPIE.

Oppenheim, Alan V, and Ronald W. Schafer. 2009. *Discrete-Time Signal Processing,* 3rd ed. New York: Pearson.

Randall, Robert Bond. 2011. *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. West Sussex, UK: John Wiley and Sons, Ltd.

Sejdić, Ervin, Igor Djurović, and Jin Jiang. 2009. "Time-frequency feature representation using energy concentration: An overview of recent advances". *Digital Signal Processing*. 19 (1): 153–183.

Tax, David M. J., and Robert P. W. Duin. 2004. "Support Vector Data Description." *Machine Learning* 54 (1):45–66.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research* 9:2579–2605.


## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:


Anya McGuirk
SAS Institute, Inc.
919.531.0934
Anya.McGuirk@sas.com

## ACKNOWLEDGMENTS

The authors would like to thank Gul Ege, Arin Chaudhuri, Michael Leonard, Jorge Silva, Seunghyun Kong, Shawn Ellis, Abel Henson, and Daniel Wilkins for all their help and insights on this project. They are also grateful to Carolyn Sutton for her editorial assistance.