Development of an automated order tracking method

J.R. Blough, J.P. DeClerck
Michigan Technological University, Mechanical Engineering- Engineering Mechanics Department
815 RL Smith ME-EM Building, 1400 Townsend Drive, Houghton, MI, USA
e-mail: **jrblough@mtu.edu**

Abstract
As computing power becomes more available and there is a desire to minimize analysis time or allow "non-expert" engineers and technicians to perform some analysis, the need develops to have a "smart" automated order tracking method.

This paper proposes a "smart" automated order tracking method that uses a combination of standard order tracking methods with some decision making. This approach can include computed order tracking based on angle domain sampling, the TVDFT approach, and/or the Vold-Kalman filter approach. The method can run fully automated or with limited to full user input as to how many decisions are made to obtain the best order tracking results possible. It is believed that this approach can significantly streamline rotating machinery analysis for noise and/or vibration.

Finally, a philosophical discussion on making decisions to optimize order tracking results is presented.

1 Introduction

This papers seeks to present a combination of order tracking methods which when used together can provide nearly fully automated order tracking for rotating machinery analysis. The approach presented relies on several different order tracking methods to provide a computationally efficient tool.

Several order tracking methods require user input and experience to achieve accurate order tracking results. Many papers have been written for instance which discuss the expertise and difficulties in using a Kalman or Vold-Kalman filter for order tracking, an example is [1].

The method presented in this paper seeks to make order tracking much easier to perform for engineers or technicians with limited information about how and why the order tracking methods perform the way they do. In many cases, this approach may be used in an almost totally automated fashion requiring very little user intervention, just some basic system information.

To automate the process of order tracking can require the use of several different order tracking methods, depending on type of result is desired. Automation also moves the burden of analysis from the user to a computer algorithm; this implies that it is not necessarily as computationally efficient as a knowledgeable and experienced analyst. Essentially, the user’s experience is replaced with additional computations.

Throughout this paper, the different order tracking methods used will be very briefly summarized so that the reader can understand the justification for each of the methods at each step in this process.

2 Automated Tachometer Processing

2.1 Tachometer Processing

As with any order tracking method the first step in the process is the processing of the tachometer signal [2]. The goal of the tachometer processing is to determine at each instant in time how fast the *"reference shaft"* of the machine being analyzed is rotating. It is assumed that the tachometer signal in this instance is a pulse type signal as that is the predominantly used method for acquiring a tachometer signal. If a voltage based signal whose amplitude was directly proportional to shaft rpm were used it an algorithm to include that ability would be quite straight forward to implement but in this paper it is assumed a pulse type signal is acquired as shown in Figure 1.

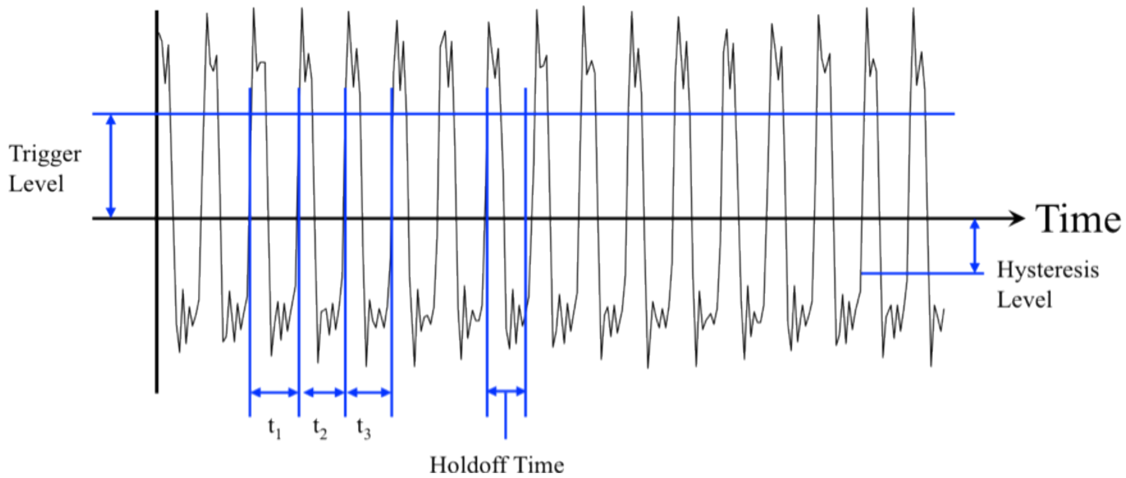

Figure 1: Example tachometer signal with processing parameters labeled.

To process a pulse type tachometer signal the user typically must choose a *Trigger Level,* that is the level at which to estimate signal crossings which are representative of pulse spacing. The time of each crossing is recorded and based on the period between each pair of crossings an instantaneous rpm is estimated. If the user does not wish to estimate a *Trigger Level,* the software can estimate an appropriate level. The approach used to estimate the *Trigger Level* is simply to find the average of the fifty highest values in the tachometer signal as these would typically be the tops of the pulses and then take a percentage of this average maximum as a trigger level, for example 30%. This approach works very well for *clean* tachometer signals that do not have many spurious pulses between the actual pulses that were generated by the tachometer sensor sensing a tooth or optical event.

If the tachometer data does not resemble a square wave with little noise, and hence would not be considered a *clean* signal, the tachometer algorithm has the ability to use a *Hold-off time* and *Hysteresis*. The *Hysteresis* works by requiring that the amplitude of the tachometer signal drop below a specified voltage before another trigger event will be recorded. This can be very useful for tachometer signals with significant noise or filtering on them, which causes each square pulse to have oscillation in its amplitude. The default value for the *Hysteresis* is half way between the lowest 50 values in the tachometer signal and the *Trigger Level*.

The *Hold-off time* requires that the software not identify the next amplitude crossing until after a specified period of time. Typically, this specified period of time is a percentage of the last estimated period between pulses. Again, this tool helps to achieve better rpm estimates when the signal contains oscillations between the desired pulse events which would lead to inaccurate period estimates based solely on the trigger level. In most software packages the hold-off time would be entered in percentage, for instance 80% would indicate not to identify the next trigger condition until (0.80*previous period estimate) seconds. In practice the *Hold-off time* works best when this percentage can be as large as possible so as to effectively filter out the largest portion of potentially noisy data. If the rpm is accelerating there is a limit as to how long the *Hold-off time* can be to not have it filter out a valid pulse crossing. To aid the user in the use of the *Hold-off time* the user can simply pick from the following options "Rapid rpm changes," "Moderate rpm changes," or "Nearly steady state." These choices are derived from the following table.

Hold-off Time Option	***% Used***	***Appropriate Conditions Examples***
Rapid rpm changes	50	Accelerating engine or vehicle
Moderate rpm changes	70	Shutdown or Start-up of large machine such as turbine, highway driving over uneven terrain.
Nearly steady state	90	Constant speed operation of engine or vehicle. Powerplant machinery while at steady state operating condition.

Table 1: *Hold-Off Time* options and implications.

2.2 Tachometer Spline Fit

Having found each of the pulse train crossings and estimated the periods between each pulse, the next step in the tachometer processing is to fit a spline to the raw data to smooth out the rpm vs. time estimate and to obtain an rpm estimate at each sample instance of the original data.

The spline fit is required because there will inherently be variance on the period estimates due to coarseness in the time axis originally used to sample the tachometer pulse train. Figure 2 shows how much error in the rpm estimate can exist if the tachometer is not sampled very fast relative to the pulse train frequency. Note the variation in the zoom axis is +/- 50 rpm at its worst!

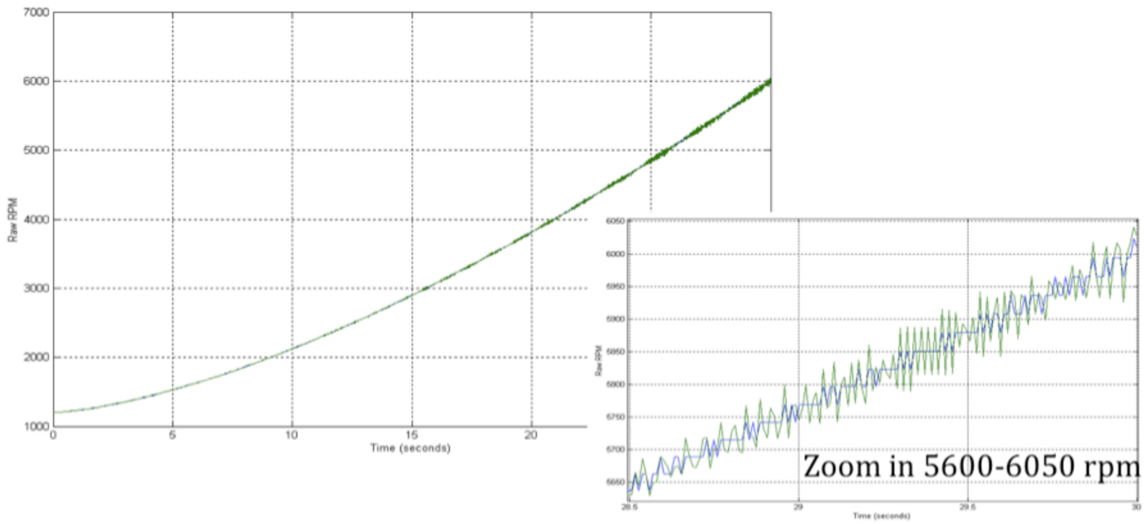

Figure 2: RPM variation due to sample rate limitations.

If the tachometer pulse train is sampled much faster than that shown in Figure 2 the variance will be reduced, however, for later processing it is still required to have an rpm estimate at each sample instance of the original data.

The justification for a spline fit vs. many other types of curve fits is that the machine of interest has inertia and therefore it is typically a safe assumption to assume that the rpm will not change instantly and hence some smoothing due to a spline fit is acceptable [3,4]. The spline fit algorithm used also allows for instance for a very rapid change in the slope of the rpm such as might be experienced during an automotive driving cycle where the transmission gears are shifted. A typical spline fit is shown below in Figure 3; note the slope changes due to gearshifts.

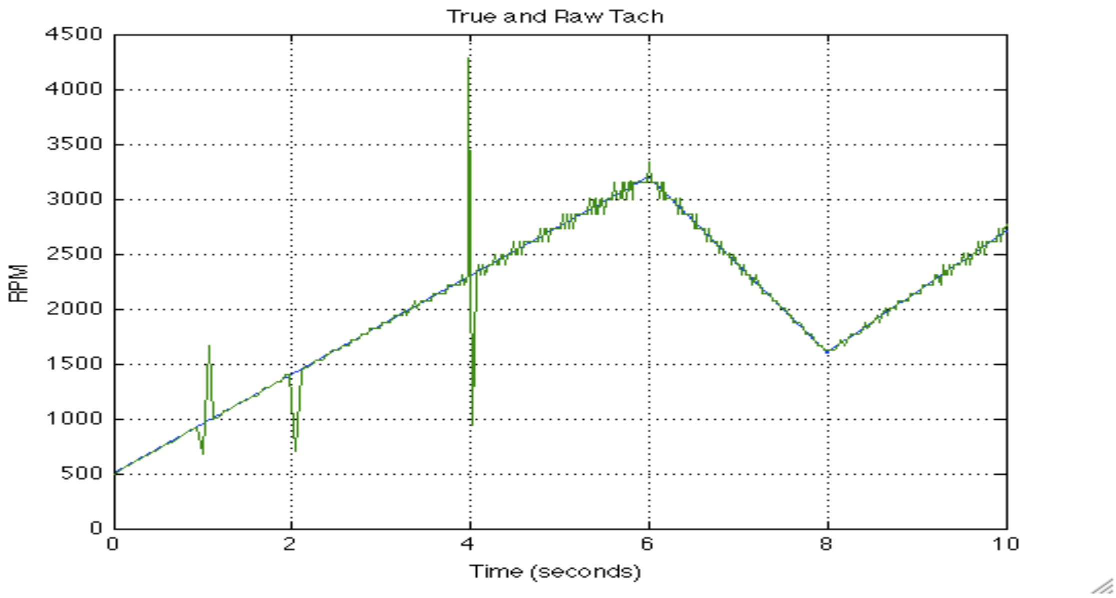

Figure 3: Raw and shaved spline fit RPM vs. time with gearshift event.

The user simply inputs whether they have gear changes or not and the software fits the splines. If there are no gear shifts, the default process is to divide the dataset into 15 even length parts and fit a spline to each part, preserving the spline boundary conditions between the sections to ensure a smooth continuous rpm function across the entire dataset.

If the user enters that there are gear changes, the user can either click with a cursor at the location of each gear change when the raw tachometer information is displayed or the software can attempt to find the gear changes through a moving average where the software is looking for a slope change over the average of the estimated raw rpm values. A moving average is required to smooth out the variation shown in Figure 2 above so that an actual gear change can be identified. The user then has the option to keep or discard these locations or enter their own before proceeding with the final spline fit.

The last process in the spline fit is to *Shave* the raw rpm estimates that are too far away from the spline fit. This is done in an attempt to eliminate outliers or points that appear to be in error. The default is to shave any locations that are farther than 3% away from the spline fit. Figure 3 also shows some outlier rpm estimates that in the final spline fit are *Shaved* from the dataset.

When the user is satisfied with the tachometer signal processing and an rpm estimated has been obtained for every sample instance of the original dataset the next step is order tracking. Note that in many cases the defaults of this series of algorithms will successfully process the tachometer signal with little to no user interaction.

3 Automated Determination of Dominant Orders

3.1 Step One: Angle Domain Resampling

The first step in any order tracking analysis after the tachometer signal has been processed and an rpm estimate has been obtained for each sample instance, is to determine which orders are of interest and therefore to track.

This approach has two options for this portion of the analysis, the first is for the user to input which orders they desire to analyze. In many cases, this will be the approach taken as the user will have sufficient experience with the type of machine being tested to know which orders are important or will have target orders that must be verified for example for a product specification.

If the user does not know which orders to analyze, the dominant orders from the dataset can be determined using an automated approach. The first step in this approach is to analyze the data in the order domain. This requires that time sampled data be resampled to the angle domain, if the data is already in the angle domain, no intermediate processing is required. This process is completely automated and does not require any user input unless there is a desire to reduce potential computation time or to have a specific sample rate in the resulting angle domain data.

One way to reduce computation time is to only resample one or a few channels of data to the angle domain and not necessarily the entire dataset, if it is a large channel count dataset. If the user wishes, they may choose which channels from a large dataset to use to determine which orders are dominant in the dataset.

If the user does not choose to have a specific sample rate in the resulting angle domain dataset, the program will analyze the original dataset sample rate and the rpm trace to determine the highest order present in the dataset. This order must then be treated as the *Nyquist* order and the angle domain sample rate set accordingly. Typically, the angle domain sample rate is rounded up to the next multiple of 360 so as to result in a sample rate with an integer number of samples both per degree and per revolution.

3.2 Step Two: Determination of Dominant Orders

The next step to determine the dominant orders is to perform *Fourier Transforms*, FFTs, on the angle domain data. The use of the FFT to determine the dominant orders is an iterative process. The initial FFT is performed with a blocksize of $1/100^{th}$ of the total dataset sample length rounded up so that the length of the transform is an integer number of revolutions. This blocksize is then used to perform 100 sliding FFTs on each channel of data with overlap processing as required. These FFTs are performed using a Uniform window to maximize the order domain resolution to allow the identification of relatively close orders from one another. The amplitudes of the resultant 100 spectra are then summed for each channel to result in one spectra per channel to analyze for the dominant orders. The resultant summed order domain spectra are then analyzed to determine the highest X amplitude orders for each channel being reported to the user, where X is the number of orders desired by the user with a default being 10 orders. The user has the option to create one summed amplitude spectra by summing all of the summed amplitude spectra from the different channels. This summation can be a straight summation or a weighted summation where a weighting factor is applied to each channel's summation before the final summation. The weighted summation can be very useful when not all channels have the same units, for instance sound pressure and acceleration as these amplitudes will be very different, in this case the summation can be the inverse of the calibration factor to effectively make them voltages or an arbitrary weighting factor to either weight a location(s) higher than other locations or to ensure that for instance the maximum amplitude is the same in each channel summation.

The next step in this process is to double the blocksize of the FFT and repeat the process. This is the first step of the iterative process to determine exactly which orders are to be tracked as an order could exist half way between two spectral lines and hence be split between two spectral bins or there could be multiple orders contributing to the same spectral line which should be separated from one another by increasing the frequency resolution. Other than the doubling of the FFT blocksize the processing performed is identical to that performed on the first iteration. Comparing the highest amplitude orders from this iteration to that obtained from the first iteration allows a decision to be made as to whether the orders identified are accurate estimates of the order number. This process can be repeated until a consistent order number is estimated for each of the highest orders.

The final step in this process to protect against errors due to leakage is to perform one final set of FFTs using the same blocksize as last used in the above iterative process except with the use of a Flattop window to ensure that the highest X orders still have the largest amplitudes when the leakage in the spectral calculations is minimized through the use of the Flattop window. Assuming that the order numbers remain constant, the user moves on to the next step in the process. If the order numbers do not remain constant the blocksize can be doubled again or the user may interrupt the iterations to make a judgment on the results.

An alternative approach to this method is that presented in reference [5] where the dominant orders are simply identified by the highest energy orders in the order domain spectra without the iteration to determine exactly what order number the order is. This approach may fail with the use of the *Vold-Kalman* filtering approach as there can be an error in the actual order number, this coupled with the very narrow bandwidth of the *Vold-Kalman* filter can result in an erroneous order extraction as the filter may not extract the actual order, but instead the energy nearby it defined by the order number which is in error.

4 Order Extraction

Having identified the largest amplitude orders the final step in the process is to extract the orders of interest. Several methods can be used to extract the orders of interest from the data depending on what the user wishes to do with the information, how many channels there are, and how many orders the user wishes to extract.

4.1 Amplitude and Phase Estimation of Orders

If the user is simply interested in the amplitude and phase estimates of the identified orders the orders can be tracked using the adaptive resampling, also called the *Computed Order Tracking* approach or the *Time Variant Discrete Fourier Transform, TVDFT* approach.

Another commonly used order tracking method is the *FFT based time domain* approach, however this method is not recommended as it yields the least accurate results.

The *Computed Order Tracking* approach can be used to extract the orders from all channels but is computationally demanding compared to the TVDFT approach [6,7]. However, if all channels were already transformed to the angle domain for order identification, there is no reason not to use this approach as the majority of the computations have already been performed. In this case, the orders are simply extracted from the resampled data using the desired blocksize and window to perform the necessary FFTs. The default would be the parameters used for the last iteration in the order identification process.

The TVDFT approach is much more computationally efficient and used when there are many channels of data which have not been transformed to the angle domain for the order identification process. In this case, the TVDFT is used to track the desired orders from all channels. The default signal processing parameters are those used in the last iteration of the order identification process. Again, if the user wishes they can use different signal processing parameters.

4.2 Time/Angle Domain Order Estimation

For some applications it is beneficial to have the time or angle domain representation of the orders of interest. In this case, the *Vold-Kalman Filters* or a FIR low pass filter coupled with an adaptive frequency demodulation algorithm can be used to extract the orders of interest from either the time domain data or the resampled angle domain data [2,8].

The difficulty with either method is in determining the proper filter bandwidth and type to use. The bandwidth has a large influence in the accuracy of the results.

If the Vold-Kalman filter is desired, an adaptive bandwidth filter can be used such as that shown in the flowchart below in Figure 4 [9].

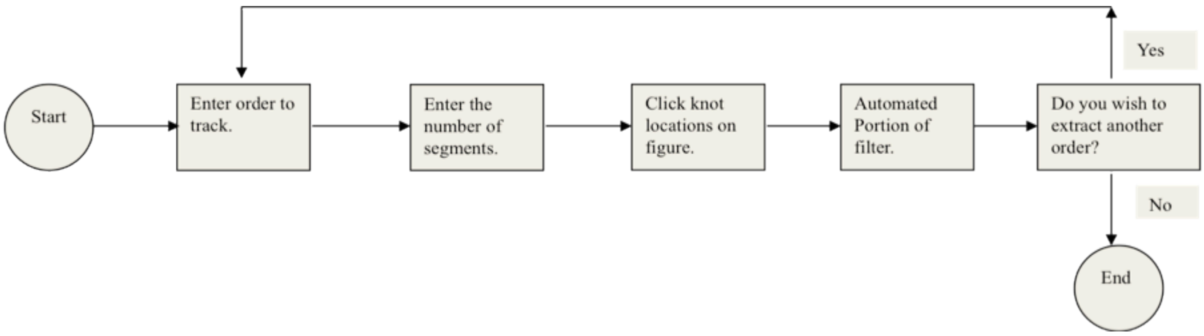

Figure 4: Flowchart of adaptive Vold-Kalman filtering algorithm.

A dataset was created to show the effectiveness of this adaptive Vold-Kalman algorithm. This algorithm is fully automated and does not require significant interaction to achieve excellent results relative to extracting and quantifying the energy associated with an order of interest. Figure 5 below shows how well this method can extract an order in the presence of a lightly damped resonance.

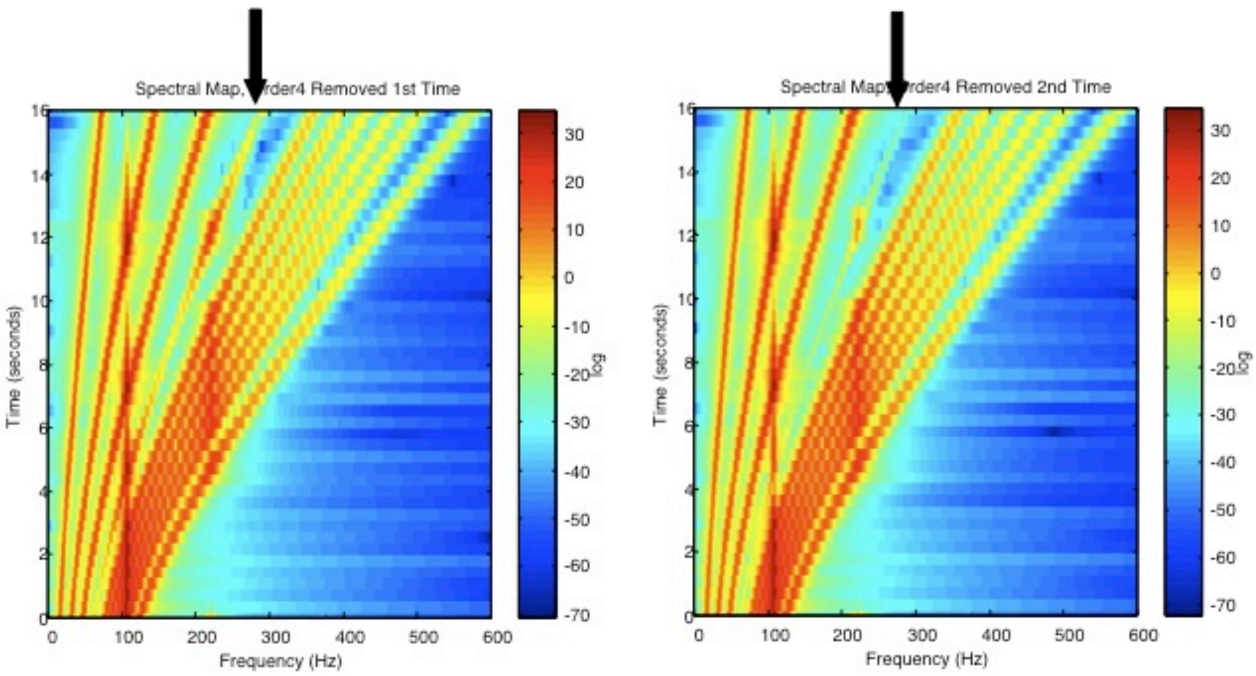

Figure 5: Adaptive Vold-Kalman filter tracking result, left side - constant bandwidth, right side - adaptive bandwidth.

The indication of the quality of the order track is how much of the high energy red color is left in the colormaps after the order extraction of the order indicated by the black arrows. Clearly, the right side colormap made using the adaptive Vold-Kalman filter shows superior results.

4.3 Order/Resonance Interaction

Regardless of the order tracking method used there is an important concept to consider when an order excites a resonance. The issue can be seen in Figure 5 above and is when a lightly damped resonance is excited by an order whose frequency is constantly varying that the amplitude modulation resulting from the interaction of the order and a resonance can be missed due to the order bandwidth being too narrow. In this case, the amplitude of the order will not increase as significantly as it would have if the frequency of the order were varying at a slower rate or the order bandwidth had been wider. In this situation, it is suggested that the resolution of the FFTs/TVDFTs be decreased so that the integration time associated with the transform is shorter and hence the order will estimate a higher amplitude more inline with what might be expected in this type of situation. If the *Vold-Kalman* filters are used the filter bandwidth may need to be increased to accurately capture this interaction. How exactly to process the data in a situation like this involves a philosophical discussion as to whether the energy associated with the order/resonance interaction should all be associated with the order or not. In the authors' opinions the energy would not be present had an order not excited the resonance and therefore the energy should be associated with the order.

5 Conclusions

This paper presented a step-by-step procedure that can be used to nearly fully automate order tracking of data obtained from rotating machinery. The paper discusses each of the critical decisions that must be made in the order tracking process and presents methods that can be used to make those decisions in an automated fashion. Combining each of the steps together leads to a fully automated order tracking process. It is believed that this process will be beneficial to those with limited order tracking experience and prove to be robust enough to be useful in many applications.

References

[1] Herlufson, H., Gade, S., Konstantin-Hansen, H., and Vold, H., *Characteristics of the Vold/Kalman Order Tracking Filter #212,* Proc. SPIE Vol. 3727, Proceedings of the 17th International Modal Analysis Conference, p1938. March, 1999. SPIE Document #: 1999SPIE.3727.1938H.

[2] Blough, J.R., *Improving the Analysis of Operating, Data on Rotating Automotive Components*, Ph.D., Dissertation, University of Cincinnati, 1998.

[3] Leuridan, J., Vold, H., Kopp, G., and Moshrefi, N., *High Resolution Order Tracking Using Kalman Tracking Filters - Theory and Applications,* Proceedings of the Society of Automotive Engineers Noise and Vibration Conference, Traverse City, MI., 1995, SAE paper no. 951332.

[4] Vold, H., Mains, M., and Blough, J.R., *Theoretical Foundations for High Performance Order Tracking with the Vold-Kalman Filter*, Proceedings of Society of Automotive Engineers 1997 Noise and Vibration Conference, SAE paper number 972007.

[5] K.Janssens, Z. Kollar, B. Peeters, S. Pauwels, H. Van der Auweraer, *Order-based Resonance Identification using Operational PolyMAX*, IMAC XXIV Conference Proceedings, *2006,* St. Louis, MO.

[6] Blough, J.R., Brown, D.L., and Vold, H., *The Time Variant Discrete Fourier Transform as an Order Tracking Method,* Proceedings of Society of Automotive Engineers 1997 Noise and Vibration Conference, SAE paper number 972006.

[7] Blough, J.R., Brown, D.L., and Vold, H., *Order Tracking with the Time Variant Discrete Fourier Transform*, Proceedings of the International Seminar of Modal Analysis - 21, Leuven, Belgium, 1996.

[8] Van Vlierberghe, P., Janssens, K., and Van der Auweraer, H., *Adaptive Amplitude Demodulation (AAD) as an order tracking method,* Proceedings of the IMAC-XXVI: A Conference & Exposition on Structual Dynamics Conference, Orlando, FL, paper number 94.

[9] Titulaer, J.J., *Development of an Automated Method for Determining the Correct Weighting Factor for Application of the First Order Vold-Kalman Tracking Filter*, Master of Science Thesis, Michigan Technological University, 2000, Houghton, MI.

Checklist

- paper size:
  DIN A4-paper (297mm x 210mm)
- margins:
  top 30mm, bottom 22mm, left 22mm, right 22mm
- title:
  Helvetica 17pt, bold, left aligned
- author:
  Times New Roman 11pt, bold, left aligned
- affiliation:
  Times New Roman 11pt, regular, left aligned
- first headings:
  Helvetica 14pt, bold, left aligned
- second headings:
  Helvetica 12pt, bold, left aligned
- third headings:
  Helvetica 11pt, bold, left aligned
- plain text:
  Times New Roman 11pt, regular, justified
- references:
  quotation with brackets [] and enumerated list with brackets
- font types in figures, tables, equations, etc.
- …