07NVC-176

Implementation of the Time Variant Discrete Fourier Transform as a Real-Time Order Tracking Method

Brandon J. Dilworth and Jason R. Blough
Michigan Technological University

ABSTRACT

The Time Variant Discrete Fourier Transform was implemented as a real-time order tracking method using developed software and commercially available hardware. The time variant discrete Fourier transform (TVDFT) with the application of the orthogonality compensation matrix allows multiple tachometers to be tracked with close and/or crossing orders to be separated in real-time. Signal generators were used to create controlled experimental data sets to simulate tachometers and response channels. Computation timing was evaluated for the data collection procedure and each of the data processing steps to determine how each part of the process affects overall performance. Many difficulties are associated with a real-time data collection and analysis tool and it becomes apparent that an understanding of each component in the system is required to determine where time consuming computation is located.

INTRODUCTION

Digital order tracking techniques vary in complexity and computational load, in general, covering two different types of techniques: Fourier Transform based and adaptive filtering and/or resampling methods. With the power of personal computers steadily increasing in the last ten years, many of the techniques avoided in the past are now being explored as viable options. In addition, some of the techniques used as post-processing algorithms are now being employed as real-time algorithms. Careful consideration must still be made when developing a real-time algorithm with respect to computer system performance and algorithm efficiency. With the power and speed of today's computers there are many more options available to the noise and vibration engineer for analysis techniques.

A number of different order tracking methods have been implemented in commercial software with many offering both types of techniques in an order analysis package. Almost every commercial package has a Fast Fourier Transform method available, and some also include a resampling based method as part of a base package. There are also a number of vendors that provide an additional package that includes the ability to separate close and/or crossing orders in data with multiple reference tachometer signals. These methods, however, are implemented solely as post-processing techniques.

The time variant discrete Fourier transform (TVDFT) method is based upon a discrete Fourier transform which has a kernel whose frequency is not constant within the data block. The frequency of the kernel varies with the frequency of the order of interest, and is defined by a tachometer signal. The bandwidth of this technique may be either constant frequency or constant order bandwidth.

The application of the orthogonality compensation matrix (OCM) to the kernel of the TVDFT allows for the accurate separation of close and/or crossing orders present in a data set. In the past, the application of the OCM has been a post-processing technique as the computational load can be demanding as the delta order spacing is reduced for conditions where the OCM is ill-conditioned. Increasing the delta order resolution is one method to improve the conditioning of the OCM, but it increases the required block size for the computation which increases the computational load of the algorithm. The implementation of the OCM as part of a real-time TVDFT algorithm makes it the only real-time order tracking algorithm currently available in industry capable of separating close and/or crossing orders.

Although there are a few different definitions of the term "real-time" in industry, it is defined in this paper to be the ability to process data for a given block and present a result prior to the sampling of the next data block.

ORDER TRACKING THEORY

Orders are defined as rotational speed harmonics that relate the speed of a referenced rotating component to frequency. Order tracking is the estimation of the amplitude and phase of the response of a machine to a referenced rotating component that is allowed to vary in amplitude and frequency over time. Regardless of the method used for order tracking, the tachometer signal (either measured directly or generated virtually using a response channel) is the most important channel in the data set. Since the response amplitude and phase estimation of an order is referenced to a rotating component, it is apparent that the accuracy of the estimation is dependent on the accuracy of the referenced tachometer signal.

ORDER TRACKING METHODS

The Fourier transform methods are based on time domain data sampled with a constant time interval, t. The time domain Fourier transform follows Shannon's sampling criteria to determine the relationship between the sampling rate, frequency bandwidth, time sample period and frequency resolution as shown in Equation 1.

$$\begin{aligned} \Delta f &= \frac{1}{T} = \frac{1}{N * \Delta t} \\ T &= N * \Delta t \\ F_{\max} &< F_{nyquist} = \frac{F_{sample}}{2} \\ F_{sample} &= \frac{1}{\Delta t} \end{aligned} \tag{1}$$

Where: Δf is the frequency resolution of the resulting frequency spectrum
- T is the total sample time which is analyzed
- N is the total number of time points over which the transform is performed
- Δt is the time spacing of the time samples
- F_{sample} is the sample frequency of the data
- $F_{nyquist}$ is the Nyquist frequency
- $F_{\max}$ is the maximum frequency which can by analyzed

The Fourier transform makes two very distinct assumptions about the data it is transforming. One assumption is that the data has constant frequency across the length of the data block, and the other is that the data has constant amplitude and phase across the data block. Any variance in these two assumptions will induce leakage into the transform. Although using the Fourier transform with time data is very simple and not computationally demanding, it does have many cases where a large amount of error can be introduced to the results. Many of these errors have been studied and discussed [ref. 1-4].

The first computed order tracking method was developed and patented by Potter at Hewlett Packard [ref. 5]. The method is based on digitally resampling data acquired with constant t to obtain data with constant angular spacing, θ. The resampled data is transformed from the time/frequency domain into the angle/order domain. Completing this transform using digital resampling is computationally demanding. Resampling data into the angle/order domain has many benefits [ref. 6]. Orders that sweep across frequency bins in the frequency domain will remain constant in the order domain allowing for leakage free amplitude and phase estimates. Leakage free estimates are obtained when the transform is performed over an integral number of revolutions, and the order estimated falls on a o. The Fourier transform is applied to the angle domain data to transform it into the order domain. Thus, the time domain sampling parameters must be reformulated for the angle domain. These relationships are presented in Equation 2.

$$\begin{aligned} \Delta o &= \frac{1}{R} = \frac{1}{N * \Delta \theta} \\ R &= N * \Delta \theta \\ O_{\max} &< O_{nyquist} = \frac{O_{sample}}{2} \\ O_{sample} &= \frac{1}{\Delta \theta} \end{aligned} \tag{2}$$

Where: Δo is the order spacing of the resulting order spectrum
- R is the total number of revolutions which are analyzed
- N is the total number of time points over which the transform is performed
- $\Delta \theta$ is the angular spacing of the re-sampled samples
- O_{sample} is the angular sample rate at which the data is acquired
- $O_{nyquist}$ is the Nyquist order
- $O_{\max}$ is the maximum order which can be analyzed

Other resampling techniques have been implemented in industry since Potter, and their purpose is the same: transform the data from the time/frequency domain into the angle/order domain. Although these resampling techniques do provide better amplitude and phase estimates for order domain data as compared to standard FFT methods, errors become prevalent as the orders become tightly spaced or cross each other within a data set.

Kalman filtering was developed as a post-processing technique, as it requires information about "future" data points when calculating an order estimate. The Kalman filter allows for orders to be extracted from a data set so that "what if" games can be played to determine the effects of different orders. The Vold-Kalman filter, an extension to the Kalman filter, has expanded capabilities to include the ability to extract close and/or crossing orders [ref. 7-9]. The Vold-Kalman filter has been implemented in industry by a number of different companies as a means to extract close and/or crossing orders, but remains strictly as a post-processing method.

Another method developed as a post-processing technique that has the ability to extract orders from a data set in order to play "what if" games is the Gabor transform. The Gabor transform has been studied to compare its robustness to the Vold-Kalman filter, and it is noted that more research is needed to compare it to other order tracking methods [ref. 10].

The TVDFT method for order tracking is a special case of the chirp-z transform. The chirp-z transform is defined as a type of Fourier transform with a kernel that allows both frequency and damping to vary as a function of time [ref. 11]. The TVDFT is a special case that is defined as a discrete Fourier transform with a kernel that allows the frequency to vary in time defined by the reference tachometer, but amplitude of the data is assumed to remain constant across the data block. Like the resampling techniques the TVDFT is able to provide leakage free order estimates, but without the need to resample the data into the angle domain. The TVDFT is based on constant t sampled data, so Shannon's sampling theorem is applicable just as with the time domain Fourier transform methods. In addition, since the frequency is changing as a function of speed, the sampling theorems for angle domain data also apply. Due to the fact that that the TVDFT does not require any resampling, it is much less computationally demanding making it ideal as a real-time algorithm for higher slew rates and higher number of response channels as compared to the resampling methods. Slew rate, as defined in the field of order tracking, is the rate of change (slope) of RPM versus time during a measurement. Similar to other commercially available real-time methods, the TVDFT is capable of tracking multiple reference tachometers simultaneously but, unlike the commercially available methods, the TVDFT has the added capability of separating close and/or crossing orders with the application of the OCM.

The kernel of the TVDFT is provided in Equation 3. The kernel is a sine or cosine function of unity amplitude with an instantaneous frequency matching that of the tracked order at each instant in time.

$$
\begin{aligned}
a_m &= \frac{1}{N}\sum_{n=1}^{N} x(n\Delta t)\cos\left(2\pi\int_0^{n\Delta t}(o_m * \Delta t * rpm/60)dt\right) \\
b_m &= \frac{1}{N}\sum_{n=1}^{N} x(n\Delta t)\sin\left(2\pi\int_0^{n\Delta t}(o_m * \Delta t * rpm/60)dt\right)
\end{aligned}
\qquad (3)
$$

Where: $x(n\Delta t)$ is the nth discrete data sample of *x*

o_m is the order to be analyzed, $m\Delta o$

a_m is the estimated Fourier transform cosine coefficient for o_m

b_m is the estimated Fourier transform sine coefficient for o_m

rpm is the instantaneous RPM

The TVDFT is best suited to estimate orders using constant order bandwidth in order to reduce leakage as much as possible. Constant order bandwidth implements a constraint that the transform is performed over an integer number of revolutions of the referenced rotating component. The order bandwidth determines the number of revolutions that the transform needs to be applied to calculate a minimal leakage order estimate. The two relationships are derived from Equation 2 and are presented in Equation 4. It is then apparent that in order to achieve constant *o* bandwidth, then the block size used in the transform will become smaller as reference rpm increases; *f* frequency resolution becomes wider as the referenced speed increases.

$$
\begin{aligned}
\frac{1}{\Delta o} &= \text{integer} \\
\frac{\text{order tracked}}{\Delta o} &= \text{integer}
\end{aligned}
\qquad (4)
$$

These relationships restrict the actual order bandwidth that can be applied using the TVDFT and still have minimal leakage. The second relationship further restricts the method to the orders that can be traced with minimal error. However, it is also noted that the TVDFT can still be used to track orders that are typically of most interest even with these restrictions imposed.

Even with these restrictions there are still a few ways that leakage can be introduced into the computation. Leakage can be introduced if there are multiple orders present in the data near the frequency band of the order of interest. These orders can "leak" into the frequency band of analysis around the order. Using standard windows can help reduce this leakage effect if selecting a smaller order increment is not possible. Another way leakage can be introduced into the computation is based on the fact that the data is sampled at constant t. This type of data does not guarantee that the transform will be applied over an integer number of revolutions of the referenced rotating component. This leakage effect can be reduced by upsampling and interpolating the data so that the data block is guaranteed to be over an integer number of revolutions, by oversampling the data when it is collected so that the likelihood that a data sample falls on an integer number of revolutions is increased, and can also be reduced by using the OCM.

The OCM is used to reduce leakage due to non-orthogonality of the TVDFT kernels. With the OCM applied to the transform kernels, faster slew rates can be evaluated and close and/or crossing orders can be estimated using the TVDFT. The quality of the compensation is dependent on the quality of the uncompensated order estimates, so careful consideration must be taken to achieve accurate results. In many situations, applying a Hanning window will help increase out of band rejection and reduce leakage.

Another consideration that must be taken is that all orders with significant energy in the signal should be tracked prior to applying the OCM. Orders with significant energy that are not tracked will appear as added noise to the overall signal effectively increasing the amount of error in the TVDFT calculation of the estimated orders. The OCM can only compensate for orders tracked based on the order estimates provided by the TVDFT calculation. This requires previous knowledge of the system, as all orders with significant energy should be known before using the TVDFT applied with the OCM to obtain accurate order estimates.

The OCM is applied through a linear equations formulation as shown in Equation 5.

$$\begin{bmatrix} e_{11} & e_{12} & e_{13} & \cdots & e_{1m} \\ e_{21} & e_{22} & e_{23} & & \\ e_{31} & e_{32} & e_{33} & & \vdots \\ \vdots & & & \ddots & \\ e_{m1} & & \cdots & & e_{mm} \end{bmatrix} \begin{Bmatrix} o_1 \\ o_2 \\ o_3 \\ \vdots \\ o_m \end{Bmatrix} = \begin{Bmatrix} \tilde{o}_1 \\ \tilde{o}_2 \\ \tilde{o}_3 \\ \vdots \\ \tilde{o}_m \end{Bmatrix} \tag{5}$$

Where: e_{ij} is the cross orthogonality contribution of order *i* in the estimate of order *j*

o_i is the compensated value of order *i*

$\tilde{o}_i$ is the estimated value of order *i* obtained using the TVDFT

The cross orthogonality terms are calculated by applying the kernel of order *i* to the kernel of order *j* as shown in Equation 6. The same window applied to the original order estimate is applied to order *i* to compensate for any correction factor needed to scale the data correctly.

$$e_{ij} = \frac{1}{N}\sum_{n=1}^{N}\left\{\exp\left(2\pi\int_0^{n\Delta t}(o_i * \Delta t * rpm/60)dt\right)\times Window\right\}\times \exp\left(2\pi\int_0^{n\Delta t}(o_j * \Delta t * rpm/60)dt\right)^{*} \tag{6}$$

Each term in the OCM represents how much each kernel interacts with the others. If the tracked orders are perfectly orthogonal, then the matrix would be diagonal or banded with the inclusion of applied windows.

To solve for the compensated order estimates the OCM must be inverted and pre-multiplied to both sides of Equation 5. Inverting a matrix is not always an easy calculation, and can result in heavy computational load. One way to determine how well a matrix inversion can be performed is by determining the condition number of the OCM. A lower condition number means that the matrix is well-conditioned, and a matrix inversion would have little error. However, if the OCM condition number is high, then the matrix inversion can result in significant error which leads to an inaccurate order estimate.

The method explored in the research was the Gaussian elimination technique rather than direct matrix inversion to solve the linear system. Gaussian elimination is a better suited method for ill-conditioned matrices, and is also considered a much more efficient algorithm than direct matrix inversion in Matlab. Since Matlab was used to calculate the compensated order estimates, efficiency considerations were made specific to that environment.

In general, the OCM condition number is high for orders that are closely coupled together. As a result, one should be aware of the sensitivity of the OCM conditioning when reviewing the processed data. If the linear system is ill-conditioned, it may be of interest to explore different numerical techniques to solve the system or to use a smaller bandwidth in an effort to reduce numerical error.

EXPERIMENTAL SETUP

Controlled experimental data sets were generated using Sony AFG210 signal generators. Multiple signal generators were used to simulate both response and tachometer signals for the data sets. The output from the signal generators were square waves with varying frequencies. These controlled data sets allowed the results to be compared to expected values. The output of the signal generators were connected to response input channels and tachometer input channels on a Larson Davis Digital Sensing System (DSS).

The response channel inputs on the DSS started at the Digital Sensor Interface Transmitter (DSIT), which is a 24-bit ADC with anti-alias filtering capable of a number of different sampling rates. Each DSS Receiver module is capable of supporting 8 response channels and each DSS chassis is capable of supporting up to 8 receiver modules. Thus, each DSS chassis is capable of supporting up to 64 response channels which allowed the algorithm to be evaluated for many different number of response channels of synchronous measurement.

Each DSS Receiver module has a single tachometer input channel. The number of tachometer inputs is only limited by the number of DSS Receiver modules installed on a chassis. The tachometer input channel has a range of +/- 5 volts with 32-bit ADC resolution sampled at 62.9 MHz. The input voltage minimum slope is 10 volts/second which is used to define a pulse edge, and is capable of measuring signals accurately up to 100 kHz if the input levels are sufficient. The tachometer measures the period between negative edges and inverts the result to calculate frequency. The calculated frequencies are averaged and output at a rate of 100 Hz.

As the DSS collects the digitized data and stores it in its internal buffer during a measurement, the TVDFT algorithm polls the buffer to determine when a new data block is ready to be processed. Once the flag is triggered, the tachometer data is processed through a spline interpolation to match the number of samples in the response channel data block. The response channel data along with the interpolated tachometer data is then processed by the TVDFT algorithm. The processed data is presented to the user on the LabVIEW interface as new results become available. After the data collection is completed for a given set, the final processed results and the raw time histories are saved to file. Figure 1 summarizes the setup and presents it in a data flow block diagram.

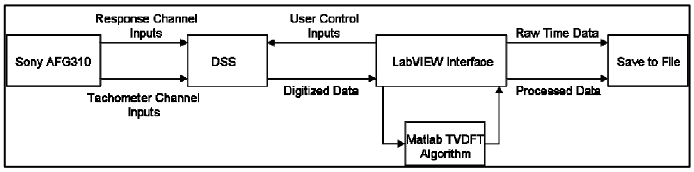

Figure 1: Data Flow Block Diagram

EVALUATION

Square wave signals were used to evaluate the TVDFT algorithm as the processed results are easy to compare to expected values as found in Equation 7. With multiple signal generators, it was possible to evaluate algorithm performance with crossing orders, again comparing the results with expected values. Another benefit of the square wave outputs was that it meets the slope requirements for pulse edges of the tachometer input channels on the DSS Receiver modules.

$$f(t) = \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{A}{(2n-1)}\sin((2n-1)\omega t) \qquad (7)$$

Where: A is the amplitude of the waveform
ω is the instantaneous frequency of a sine wave
t is time

The first iteration of the application was built to track only to a single reference tachometer. A single signal generator was used to generate a 1 Vrms output simulating a speed sweep from 1700-3500 RPM (about 28-58 Hz assuming single pulse per revolution). Orders 1, 3, 5 and 7 were tracked with an order bandwidth of 0.05. The expected output from the order tracking is shown in Table 1.

Order	Expected Output Value (Volt-rms)
Overall	1.0
1	0.9
3	0.3
5	0.18
7	0.129

Table 1: Expected Output Values from Order Tracks – Single Tachometer

Figure 2 shows how the results of the real-time application match the expected results from Table 1. This first iteration of the application proved that the method was implemented correctly as it compares favorably to computed values.

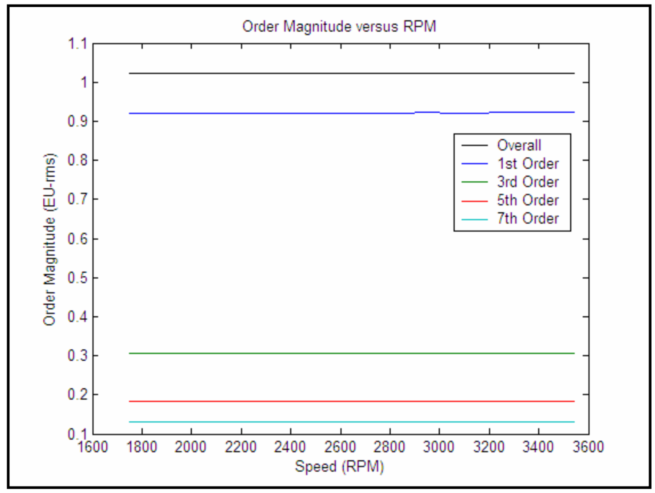

Figure 2: Results of the Real-Time Application – Single Tachometer

The next iteration of the order tracking application included the ability to track any number of tachometer channels. With multiple signal generators it was possible to simulate crossing orders in the data set. Crossing orders were simulated by using one signal generator to hold a constant frequency (simulate a component with constant rotating speed) and use another signal generator that sweeps through the frequency of the first signal generator. Same as with the test for a single tachometer, the sweeping signal output is 1 Vrms and simulates a sweep from 1700-3500 RPM. The other signal generator was set to 0.1 Vrms and held at a constant frequency of 50 Hz (simulating 3000 RPM). A product of the wiring used made the overall output equal to 0.5 Vrms. Table 2 shows the expected results for tracking orders 1, 3 and 5 for the primary tachometer and order 1 and 3 for the second tachometer.

Order	Expected Output Value (Volt-rms)
Overall	0.5
Tach 1: 1	0.45
Tach 1: 3	0.15
Tach 1: 5	0.09
Tach 2: 1	0.045
Tach 2: 3	0.015

Table 2: Expected Output Values from Order Tracks – Multiple Tachometers

An order bandwidth of 0.05 was used to ensure that the crossing orders could be separated easily. Figure 3 shows that the real-time results match the expected values and that the crossing orders were able to be separated accurately. The variation in the overall level can be explained by the additive effects due to the two signals at different frequencies. The separated orders do not vary over the duration of the sweep because the result is due only to the referenced signal and is effectively separated from the other signals.

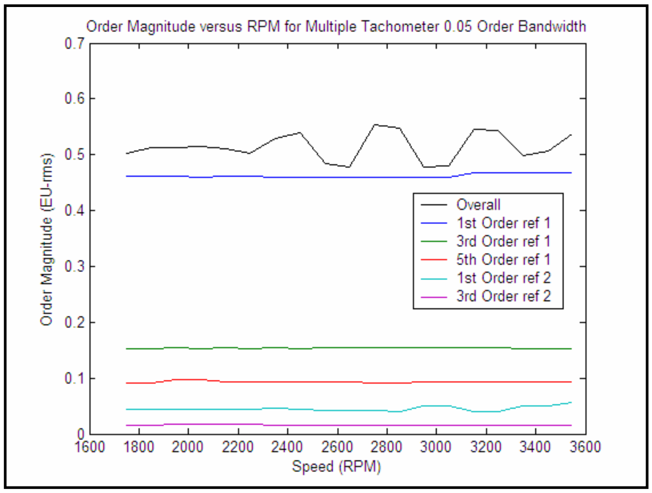

Figure 3: Results from the Real-Time Application – Multiple Tachometers

COMPUTATIONAL EFFICIENCY

Once the algorithm was tested for proper data handling and computational accuracy, the application was evaluated for effects to efficiency due to different parameters. The algorithm was monitored at different stages to determine the download time (time required to download the data from the DSS to the computer), processing time (time required for the TVDFT with the application of the OCM calculations) and interpolation time (time required to perform a spline interpolation to the tachometer signal). The parameters varied for evaluation were: number of response channels, number of orders to track, number of reference tachometers, order bandwidth, sampling rate and the type of computer used. All of the parameters were evaluated versus the number of samples per channel, which is dependent on the minimum trigger speed, sampling rate and order bandwidth used for the analysis.

Varying the number of response channels to process affects the amount of data required to download from the DSS and the number of times the TVDFT kernel must be applied. Figure 4 shows how download and processing times are affected by the number of response channels in the data set. The interpolation time remains constant for all the different number of response channels considered.

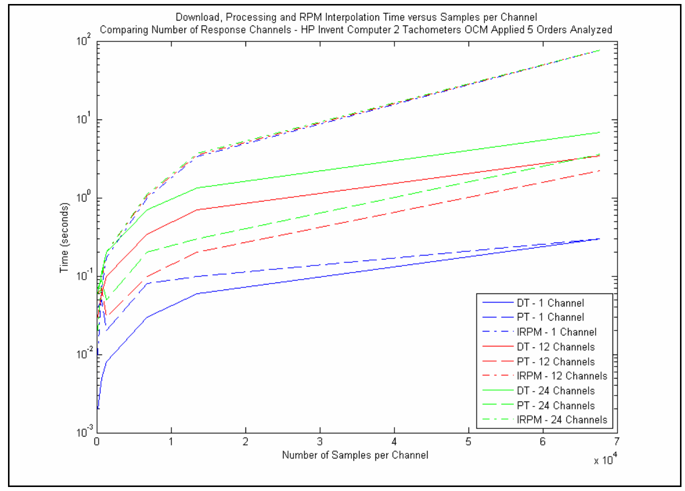

Figure 4: Download Time (DT), Processing Time (PT) and RPM Interpolation Time (IRPM) versus Samples per Channel Comparing Number of Response Channels – 2 Tachometers OCM Applied 5 Orders Analyzed

Varying the number of orders to track affects the size of the TVDFT kernel matrix. As the number of orders to track increases, the size of the linear system to be solved also increases requiring more computations. For close and/or crossing orders the interaction between the orders can become significant which can increase the condition number of the matrix. Figure 5 shows how the computation time is affected as the number of orders to track and download time and interpolation time remain constant.

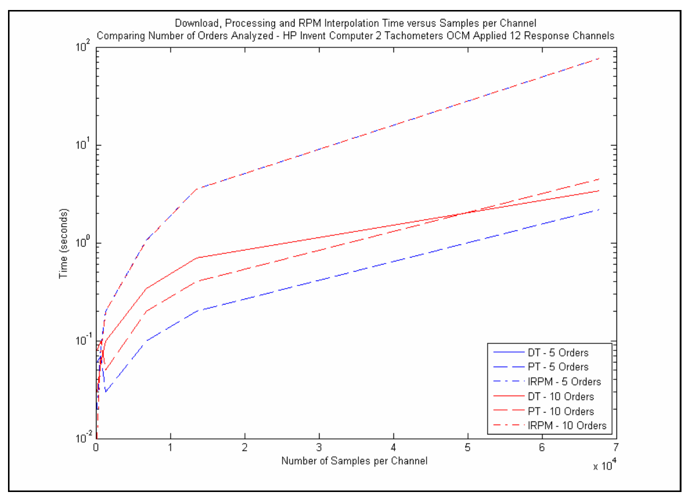

Figure 5: Download Time (DT), Processing Time (PT) and RPM Interpolation Time (IRPM) versus Samples per Channel Comparing Number of Orders in Analysis – 2 Tachometers OCM Applied 12 Response Channels

The number of reference tachometers affects the amount of data required to download from the DSS to the computer, the size of the TVDFT kernel matrix and the number of times the spline interpolation must be applied to the tachometer data. A single reference tachometer was compared to 2 reference tachometers. For a single tachometer, the OCM is not required to estimate the orders as there is no information available about any other rotating component. Thus, for this comparison the timing differences between the algorithm without the OCM applied to the algorithm with the OCM applied was evaluated. Figure 6 shows that the download time is the least significant of the factors and there are larger effects to the processing time and the interpolation time required.

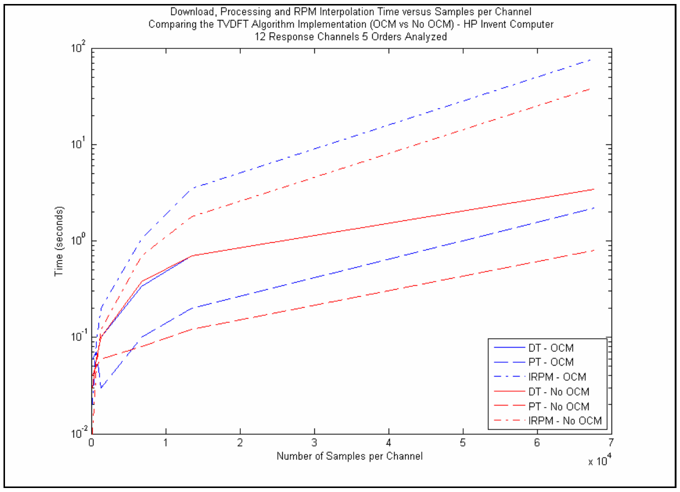

Figure 6: Download Time (DT), Processing Time (PT) and RPM Interpolation Time (IRPM) versus Samples per Channel Comparing the TVDFT Algorithm Implementation (OCM vs No OCM) – 12 Response Measurements 5 Orders Analyzed

The type of computer used also has affects algorithm performance. Two machines were compared, one with more memory and a faster processor than the other to determine the amount of the effect. The first computer used for the evaluation was a HP Invent Windows XP SP2 machine with an Intel Pentium M 1.60 GHz processor with 0.99 GB of RAM running LabVIEW 7.1. The second computer used for the evaluation was a HP Workstation xw8000 Windows XP SP2 machine with an Intel Xeon 3.06 GHz processor with 3.00 GB of RAM running LabVIEW 8.0. Figure 7 suggests that there is no significant difference between the two machines. The HP Workstation was configured as a network machine, requiring it to receive software license information over a network connection where the HP Invent was configured as a stand-alone machine. More work is needed to understand if configuring a computer as a network machine has an affect to the performance of the machine versus setting up as a stand-alone machine.

More work is needed to understand potential differences between the two different versions of LabVIEW used in the evaluation. It is possible that the internal functions within LabVIEW used for the TVDFT implementation, a specific example being the spline interpolation function used on the tachometer signals, has a significant effect on the overall system performance.

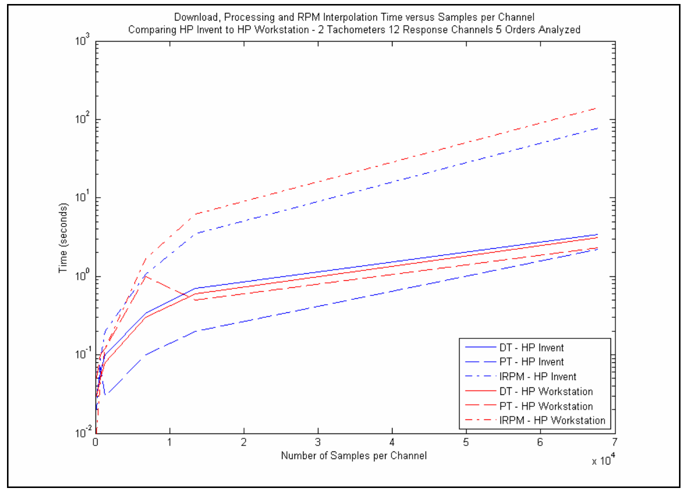

Figure 7: Download Time (DT), Processing Time (PT) and RPM Interpolation Time (IRPM) versus Samples per Channel Comparing HP Invent to HP Workstation – 2 Tachometers OCM Applied 12 Response Measurements 5 Orders Analyzed

Order bandwidth and sampling rate and the minimum RPM trigger (the lowest speed to be analyzed in the data set) combined determine the block size required to be downloaded from the DSS to the computer. As it can be seen in all of the previous figures, the timing required for each process increases as the number of samples per channel increases.

CONCLUSION

The Time Variant Discrete Fourier Transform with the Orthogonality Compensation Matrix has been successfully implemented as a real-time application. Many parameters play a role to the overall performance of the application including the algorithm itself. The TVDFT with OCM application has been proven using controlled experimental data sets for many different scenarios including effectively separating crossing orders. Future work should include using this application with measured data with practical applications.

For the experimental setup evaluated, the tradeoffs between number of response channels, number of orders to track, type of TVDFT algorithm implemented and the number of samples per channel to computational efficiency are understood. The factors that affect computational efficiency for this implementation are the interpolation of the tachometer channels and the number of samples per channel in each data block. Minimizing the number of samples per channel will also minimize the interpolation time required for a given number of tachometer inputs. The overall timing of this implementation is also minimized when the number of samples per channel is minimized. As with any other real-time algorithm, one must understand the tradeoffs and limitations between input parameters and computational efficiency in order to make reliable calculations without overloading the algorithm.

Continued advancements in computer technologies will continue to open the door for more real-time data analysis algorithms. This case shows that an algorithm once only considered as a post-processing algorithm can be implemented as a real-time application as long as careful considerations are made regarding code, algorithm and data management efficiencies.

REFERENCES

[1] Van de Ponseele, P., Van der Auweraer, H., and Mergeay, M., “A Global Approach to the Acquisition and Analysis of Harmonic Waveforms,” Proceedings of International Modal Analysis Conference 7, Las Vegas, Nevada., pp.1290-1299, 1989.

[2] Van de Ponseele, P., Van der Auweraer, H., and Mergeay, M., “Performance Evaluation of Advanced Signature Analysis Techniques,” Proceedings of International Modal Analysis Conference 7, Las Vegas, Nevada., pp.154-158, 1989.

[3] Bandhopadhyay, D.K. and Griffiths, D., “Methods for Analyzing Order Spectra,” Proceedings of Society of Automotive Engineers 1995 Noise and Vibration Conference, Traverse City, Michigan, SAE paper number 951273.

[4] Harris, C.M., “Shock and Vibration Handbook,” McGraw-Hill, 4th Edition, pp.14.26-14.32, New York, New York, 1995.

[5] Potter, R. and Gribler, M., “Computed Order Tracking Obsoletes Older Methods,” Proceeding of the Society of Automotive Engineers Noise and Vibration Conference, Traverse City, Michigan, 1989, SAE paper number 891131.

[6] Blough, J.R., “Adaptive Resampling – Transforming From the Time to the Angle Domain,” Proceedings of International Modal Analysis Conference 24, St. Louis, Missouri, pp.315-329, 2006.

[7] 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.

[8] Vold, H. and Deel, J., “Vold-Kalman Order Tracking: New Methods for Vehicle Sound Quality and Drive-Train NVH Applications,” Proceedings of Society of Automotive Engineers 1997 Noise and Vibration Conference, SAE paper number 972033.

[9] Vold, H., Herlufson, H., Mains, M., and Corwin-Renner, D., “Multi Axle Order Tracking with the Vold-Kalman Tracking Filter,” Sound and Vibration Magazine, May 1997, pp.30-34.

[10] Qian, S., “Application of Gabor expansion for Order Analysis,” LabVIEW application notes, National Instruments, Austin, Texas, 2003.

[11] Rabiner, L. and Gold, B., “Theory and Application of Digital Signal Processing,” Prentice Hall International, London, pp.390-399, 1975.