

Understanding the Kalman/Vold-Kalman Order Tracking Filters' Formulation and Behavior

J.R. Blough

Michigan Technological University

Copyright © 2007 SAE International

ABSTRACT

The Kalman and Vold-Kalman order tracking filters have been implemented in commercial software since the early 90's. There are several mathematical formulations of filters that have been implemented by different software vendors. However, there have not been any papers that have been published which sufficiently explain the math behind these filters and discuss the actual implementations of the filters in software. In addition, upon generating the equations represented by these filters, solving the equations for datasets in excess of several hundred thousand datapoints is not trivial and has not been discussed in the literature. The papers which have attempted to cover these topics are generally vague and overly mathematically eloquent but not easily understandable by a practicing engineer. This paper will attempt to solve that problem by explaining the filters' formulations and behavior in very straight forward and practical terms through the use of both equations and example datasets.

INTRODUCTION

An order tracking method that overcomes many of the limitations of order resolution is the Kalman filter based order tracking. The Kalman filter methods allow the extraction of the time history of the order as well as the estimate of the amplitude and phase of an order. The Kalman filter was first adapted to order tracking by Vold and Leuridan [1,2]. Since this original implementation, Vold has continued to develop more advanced filters with additional capabilities [3-5].

Several papers have been written by both Vold and applications engineers at Bruel & Kjaer which describe many of the characteristics of the Vold-Kalman filters and how to apply them to problems [12-14]. However, none of these papers presents the mathematical formulations of the filters or solution techniques to the generated equations in a manner which a practicing

engineer can easily follow. It is hoped that this paper will complete the body of work necessary for a practicing engineer to implement and use the Vold-Kalman filters in a software environment such as Matlab. Several different solution methods are presented and their computational efficiencies discussed for both shorter and longer time histories.

MATHEMATICAL FORMULATIONS

This section presents the development of the equations which define the Kalman and Vold-Kalman filters and explicitly develops the equations to the point that there are clearly understandable.

KALMAN ORDER TRACKING FILTER

The Kalman filter approach to estimation requires that apriori information of some type be known [6,7]. To use the Kalman filter to extract order information from data requires information about the order to be extracted. In this case, the frequency is known from processing of the tachometer signal.

This apriori information is used to formulate the structural equation of the Kalman filter. The structural equation is an equation that describes the mathematical characteristics of the order to be extracted. The structure equation that can be used to mathematically describe a sampled sine wave, and used in the original second order Kalman order tracking filter, is given in Equation 1.

$$x(n\Delta t) - 2\cos(\omega\Delta t)x((n-1)\Delta t) + x((n-2)\Delta t) = 0 \quad (1)$$

where: $x(n\Delta t)$ is the n^{th} discrete time sample.

ω is the instantaneous frequency of the sine wave.

Equation 1 describes a sine wave whose frequency and amplitude is constant over three consecutive time points. The frequency of an order is allowed to vary with time, which implies that the frequency of the sine wave is not constant. The structure equation is then re-written to account for this, shown in Equation 2.

$$x(n\Delta t) - 2\cos(\omega\Delta t)x((n-1)\Delta t) + x((n-2)\Delta t) = \varepsilon(n) \quad (2)$$

where: $\varepsilon(n)$ is the non-homogeneity term.

The non-homogeneity term is used to describe the amplitude and frequency variations from a perfect sine wave. Mathematically, if a sine wave is amplitude modulated there must be other frequency components present in the data. These additional frequencies are sidebands which allow the amplitude of the sine wave to change with time. If the amplitude is to change quickly, then more frequency information must be allowed to pass through the filter and the non-homogeneity term must be larger.

The second equation that the Kalman filter is based on is the data equation. The data equation describes the relationship between the order, $x(n)$, and the measured data, $y(n)$. The measured data contains not only the order of interest but all orders generated by the machine and other random noise present in the data. This equation is written in Equation 3.

$$y(n) = x(n) + \eta(n) \quad (3)$$

where: $\eta(n)$ is the nuisance component.

The nuisance component, $\eta(n)$, is the portion of the signal containing the non-tracked orders and random noise. If the nuisance term is large it indicates that a significant portion of the measured signal, $y(n)$, is attributable to non-tracked orders and random noise.

The structure and data equations are combined into a set of linear equations to solve for the amplitudes of the order of interest, as shown in Equation 4. Normally, the least squares formulation is formulated in a matrix form to solve for all points of a time history simultaneously [1,10,11]. This formulation is technically a Kalman smoothing algorithm, as opposed to a filtering algorithm, because it can use time points both before and after the desired time point to obtain the order estimate [6,7].

$$\begin{bmatrix} 1 & -2\cos(\omega\Delta t) & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{Bmatrix} = \begin{Bmatrix} \varepsilon(n) \\ y(n) - \eta(n) \end{Bmatrix} \quad (4)$$

A weighted solution to this problem is formed by ratioing the standard deviations of the structure and data equations. This ratio is what is referred to as the *Harmonic Confidence Factor*, HCF, in commercial

implementations of this filter. The value of this parameter is what determines the tracking characteristics of the filter. This ratio is shown in Equation 5.

$$r(n) = \frac{s_\varepsilon(n)}{s_\eta(n)} \quad (5)$$

Applying the ratio as a weighting function to the solution of the matrix problem, results in Equation 6.

$$\begin{bmatrix} 1 & -2\cos(\omega\Delta t) & 1 \\ 0 & 0 & r(n) \end{bmatrix} \begin{Bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{Bmatrix} = \begin{Bmatrix} \varepsilon(n) \\ r(n)(y(n) - \eta(n)) \end{Bmatrix} \quad (6)$$

Assuming an isotropic error that is consistent with a minimum variance unbiased solution with locally zero mean error terms gives Equation 7. Assuming a different error model will alter the tracking characteristics of the filter.

$$\begin{bmatrix} 1 & -2\cos(\omega\Delta t) & 1 \\ 0 & 0 & r(n) \end{bmatrix} \begin{Bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{Bmatrix} = \begin{Bmatrix} 0 \\ r(n)y(n) \end{Bmatrix} \quad (7)$$

1ST ORDER VOLD-KALMAN ORDER TRACKING FILTER

Vold both simplified and extended the original Kalman order tracking filter into the Vold-Kalman order tracking filter [3]. This extended filter can be formulated with different numbers of poles to alter its band-pass characteristics. The filter may also be applied in either an iterative or direct solution to separate the contributions of very close or crossing orders.

Vold realized that the second order formulation for the structure equation, presented in Equations 2 and 3, could be written in an equivalent complex first order form. The complex formulation for the structure equation is shown in Equation 8.

$$x(n+1) - x(n)\exp(i\omega\Delta t) = \varepsilon(n) \quad (8)$$

The exponential term in this equation is the angle that the order rotates through in one time sample, Δt . If the amplitude of the order does not change in the sample period, $\varepsilon(n)$, the non-homogeneity term, will be 0. This form of the structure equation has several advantages over the original second order formulation. The data may be pre-multiplied by the exponential function, or phasor, which varies with time. The phasor's frequency exactly matches the frequency of the order at all time values. This pre-multiplication is shown in Equation 9.

$$y_{DC}(t) = y(t) \exp\left(-ik \int_0^t \omega(u) du\right) \quad (9)$$

where: $y_{DC}(t)$ is the frequency shifted time history.
 k is the order of interest.

Equation 9 shows the phasor in time-varying form. Multiplying the data by the time-varying phasor centers the order of interest about DC. This operation simplifies the structure equation to the form shown in Equation 10.

$$x(n+1) - x(n) = \varepsilon(n) \quad (10)$$

This simplified structure equation for the order is simply a relationship where $\varepsilon(n)$ represents the amplitude change of the order from one time sample to the next. The order is then simply a DC amplitude profile, in other words, the extracted order at this phase is the demodulated complex amplitude profile.

One key advantage to this form of the structure equation is that frequency is not present in the equation, and therefore, it is obvious that there is absolutely no frequency or slew rate limitations.

The Vold-Kalman filter uses the same data equation as the original Kalman filter, except in this case the data, $y(n)$, is actually the phasor shifted data, $y_{DC}(n)$. This filter also uses the same definition for the weighting function and error terms. The matrix formulation of the structure and data Equations is shown in Equation 11.

$$\begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ r(n) & & \\ & r(n+1) & \end{bmatrix} \begin{Bmatrix} x(n) \\ x(n+1) \\ x(n+2) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ r(n)y_{DC}(n) \\ r(n+1)y_{DC}(n+1) \end{Bmatrix} \quad (11)$$

The order amplitude/phase profile that is estimated by Equation 8 is not a time-varying phasor that represents the order, as was estimated by the original Kalman order tracking filter. This first order formulation will estimate the amplitude/phase envelope that must be re-modulated to remove the tracked order from the original data. Re-modulation is performed by multiplying the estimated amplitude/phase profile by the negative time-varying phasor of Equation 9, used to de-modulate the original data. The estimate is only the positive frequency component of the order. The negative frequency component must also be accounted for. The complete re-modulation equation is shown in Equation 12 for both the positive and negative frequency order components.

$$\tilde{x}(t) = 2 * \text{Real}\left(x(t) \exp\left(ik \int_0^t \omega(u) du\right)\right) \quad (12)$$

where: $\tilde{x}(t)$ is the re-modulated order function.

This re-modulated order function can be subtracted from the original time history of the data to analyze the data without the presence of the extracted order.

HIGHER ORDER VOLD-KALMAN ORDER TRACKING FILTER

Vold developed the higher order Vold-Kalman filters at the same time the 1st order Vold-Kalman filters were developed [3]. The higher order filters' structure equations are generated through computing products of the 1st order filter kernels for time shifts. The generic equation used to generate the structure equations as shown by Vold in publications is shown in Equation 13.

$$\nabla^s A_j(n) = \bar{\varepsilon}(n) \quad (13)$$

The 2nd and 3rd order filter derivations are shown in detail below. From these derivations it should become obvious how to develop higher order implementations.

2nd Order Vold-Kalman Implementation

The 2nd order Vold-Kalman filter's structural equation is explicitly generated by computing the product of two 1st order kernels, with a single time shift between them, as shown in Equation 14.

$$(1 - \theta_n)(1 - \theta_{n-1}) \quad (14)$$

$$\text{where: } \theta_n = \exp\left(-ik \sum_0^n \omega(n) \Delta t\right)$$

i = square root of -1

k = order number

n = time index

Δt = time sample spacing ($=1/F_{\text{sample}}$)

ω = instantaneous frequency of reference shaft rotation in radians/second

This multiplication yields the structural equation shown below in Equation 15. Notice that while the 1st order structural equation had two terms, the 2nd order structural equation contains 3 terms.

$$x(n-2) - (\theta_{n-1} + \theta_n)x(n-1) + (\theta_n \theta_{n-1})x(n) = \varepsilon(n) \quad (15)$$

3rd Order Vold-Kalman Implementation

The 3rd order Vold-Kalman filter's structural equation is explicitly generated by computing the product of three 1st order kernels, with a single time shift between each of them, as shown in Equation 16.

$$(1-\theta_n)(1-\theta_{n-1})(1-\theta_{n-2}) \quad (16)$$

This multiplication yields the structural equation shown below in Equation 17. Notice that while the 2nd order structural equation had 3 terms, the 3rd order structural equation contains 5 terms.

$$\begin{aligned}
& x(n-2) - (\theta_n + \theta_{n-1} + \theta_{n-2})x(n-1) + \\
& \quad (\theta_n \theta_{n-1} + \theta_n \theta_{n-2} + \theta_{n-1} \theta_{n-2})x(n) - \\
& \quad \quad (\theta_n \theta_{n-1} \theta_{n-2})x(n+1) = \varepsilon(n)
\end{aligned} \tag{17}$$

NUMERICAL IMPLEMENTATION AND SOLUTION

Equation 7 can be expanded into a least squares solution if multiple estimates of the order amplitude and phase are of interest, this formulation is used to extract the time history of an order. The least squares formulation shown in Equation 8 will solve for m points of the order time history for the 1st order Vold-Kalman filter.

$$\begin{bmatrix}
 1 & -2\cos(\omega_n\Delta t) & & & & \\
 & 1 & -2\cos(\omega_{n+1}\Delta t) & & & \\
 & & 1 & -2\cos(\omega_{n+2}\Delta t) & & \\
 & & & \ddots & \ddots & \\
 & & & & 1 & -2\cos(\omega_{n+m}\Delta t) \\
 & & & & & 1
 \end{bmatrix}
 \begin{bmatrix}
 r(n) \\
 r(n+1) \\
 r(n+2) \\
 \vdots \\
 r(n+m)
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 \vdots \\
 0 \\
 r(n)y(n) \\
 r(n+1)y(n+1) \\
 r(n+2)y(n+2) \\
 \vdots \\
 r(n+m)y(n+m)
 \end{bmatrix}
 \quad (14)$$

Equation 14 is then solved by pre-multiplying both sides of the equation by the inverse of the combined structure/data matrix seen in the left side of the equation. The multiplication will result in a tightly banded matrix which then must be inverted to solve for the order amplitudes $x(n)$. The amplitude of the order will be estimated for each time point included in the calculation. A complete mathematical summary of the Vold-Kalman solution is presented by Feldbauer in reference [15].

Realizing that the matrix to be inverted will be a square matrix, with row and column dimensions equal to the number of datapoints to be estimated, it becomes clear that this inverse may not be trivial to calculate due to its

size. This becomes the case for instance with time histories that may be several million data points long. For this reason several other solution methods in addition to the matrix inverse solution must be considered.

MATRIX INVERSE SOLUTION, SPARSE MATRIX

An important realization about the matrix form that can be made is that it is a sparse matrix equation, meaning that the matrix to be inverted is tightly banded with all of its values on the main diagonal with only the sub and super diagonals adjacent to it populated. The 1st order filter has a tri-diagonal matrix form, the 2nd order filter has a 5 banded matrix, and the 3rd order filter has a 7 banded matrix.

Having the ability to use Sparse matrix solution methods allows the large matrix equation typically required for a complete time history to be solved using an efficient algorithm such as an LU decomposition [8]. The solutions for a given matrix size increase with increasing filter order. On the same computer, for the same dataset in Matlab, the cpu times were measured for the solution of the 1st, 2nd, and 3rd order filter applications with the following results, the 1st order filter had a solution time of 1.39 seconds, the 2nd order filter had a solution time of 2.05 seconds, and the 3rd order filter had a solution time of 2.67 seconds for a 200,000 point time history. This simulation was performed using Matlab's sparse matrix commands and an LU decomposition technique to solve the equations. It can be seen that the 3rd order filter requires approximately twice the solution time of the 1st order filter. Simulations were also performed with varying size time histories and it was determined that the solution time increases linearly with the time history size for all three filter implementations.

MATRIX INVERSE WITH OVERLAPPING WINDOWS

In cases where either the computer memory is limited or the length of the time history to be filtered is excessively long other solution methods must be considered. One method is to use the Sparse matrix/LU decomposition approach described above along with dividing the long time history into smaller pieces and solving each piece independently from each other piece. This approach requires less memory space however if the computer can solve the original problem in its memory space without swapping to disk this method will result in longer solution times. For these reasons this solution method is application and computer dependent as to whether it is more or less efficient as solving the entire problem at once.

While this approach seems straight forward and obvious it must be realized that the Vold-Kalman filters may have significant startup and ending transients as shown by the plots in Figures 1 and 2 below for the 1st and 2nd order implementations respectively.

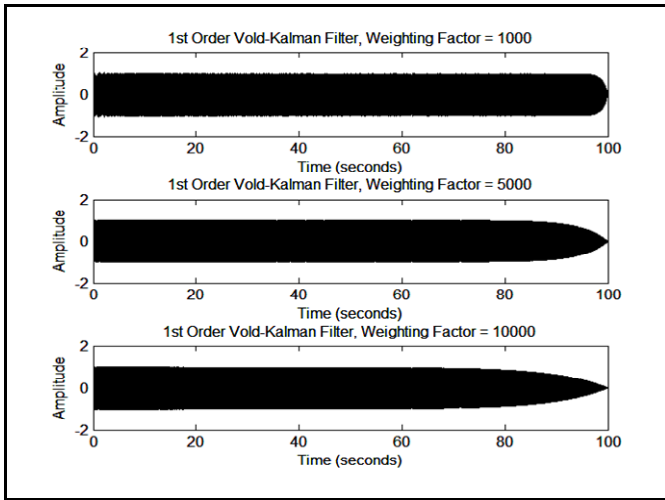


Figure 1: Filtered responses using 1st Order Vold-Kalman Filter, weighting factors of 1000, 5000, and 10,000.

Note that in the 1st Order filter results seen in Figure 1, that the main transient events can be seen at the end of the time histories. The higher the weighting factor used the longer the transients become.

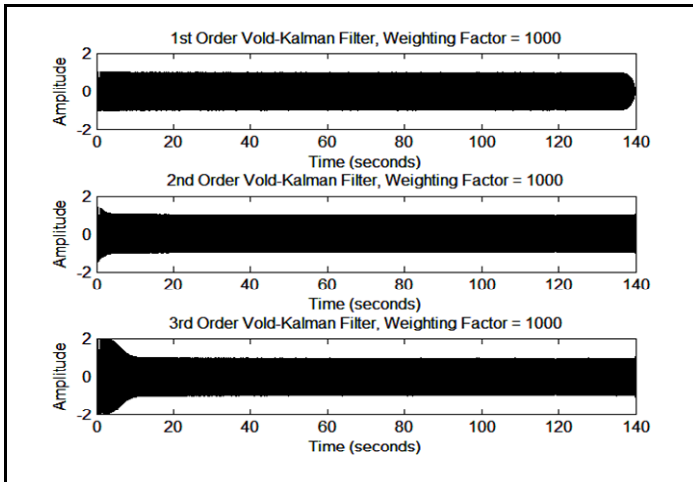


Figure 2: Filtered responses of 1st, 2nd, and 3rd Order Vold-Kalman Filter, weighting factors of 1000.

Figure 2 shows that the filter transients of the filters can appear either at the start or the end of the filtered time history. The amplitudes and duration of these transients vary with both the weighting factor chosen and the order of the filter chosen.

Due to the behavior shown in Figures 1 and 2 it is not straight forward to implement a block solution approach and not introduce any startup or ending transients in the process. An example of one implementation however is shown in Figure 3. The top of Figure 3 is the filtered order time history, the bottom of Figure 3 shows the

difference between solving the entire time history at one time and the block solved time history.

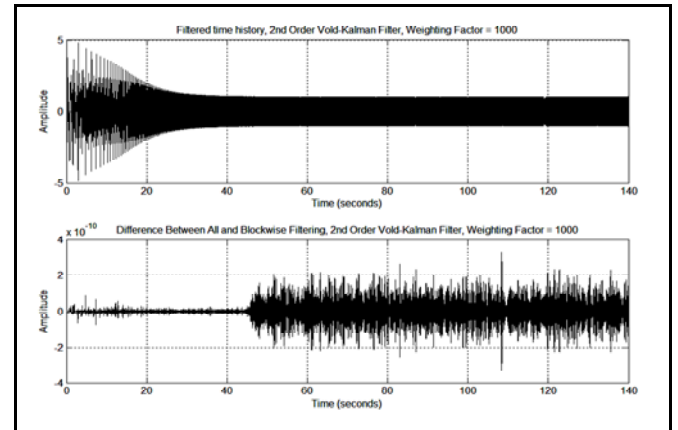


Figure 3: Top plot, 2nd Order Vold-Kalman Filtered Order. Bottom Plot, Difference between the block wise filtered results and the whole time block solution.

It can be seen in Figure 3 that while the order has an amplitude of approximately 5, the difference between the two solution methods is on the order of 10^{-10} or effectively zero. This example shows that the blockwise solution will give the same results as solving the entire time history in one calculation if it is implemented correctly.

Figure 4 below shows the intermediate filter results computed for each block. Since the time history is partitioned into blocks and each block of time will have startup and ending transition zones the data is broken up into overlapping blocks. In this case there is a 15% of the blocksize overlap between blocks. The percentage overlap may have to be changed for each of the different filters used to account for the transients, experience has shown that the 1st order filter will need the most overlap and the 3rd order filter the least in most cases. Each block is then solved independently and the final time history derived by adding the intermediate block results back together. However, to accomplish this without introducing transients between the blocks, the blocks are weighted at both ends by a cosine taper function. Note that the blocks are offset along the y-axis for plotting purposes; they are actually centered about zero.

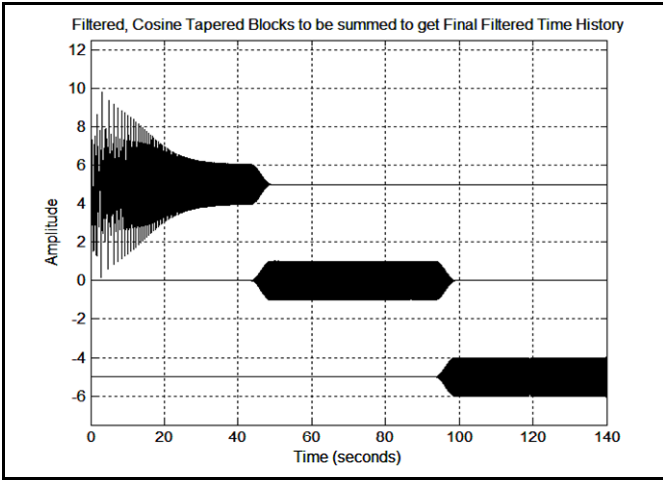


Figure 4: Cosine tapered intermediate filter results from block processing.

Summing the intermediate results shown in Figure 4, results in the filtered time history shown in the top of Figure 3.

Since there is overlap required in the solution, the block solution method will typically take slightly more computation time when the entire problem can be solved in the computer's memory, however it does allow longer time histories to be solved when the entire solution cannot be computed simultaneously in memory.

FORWARD/BACKWARD SUBSTITUTION SOLUTION

The normal equations form of the 1st order filter is a matrix equation which is a tri-diagonal set of equations that may be solved very efficiently using a forward/backward substitution method [8]. The forward/backward substitution method can be formulated to solve for the solution in place, meaning that the result of the computation is computed in the same memory space which held the original matrix. An advantage of this type of approach is the memory efficiency which is achieved since the large matrix does not have to be copied to a different memory space for solution and there are no additional memory requirements for the LU decomposition.

Simulations run to determine the computational efficiency of the LU matrix inverse approach versus the Forward/Backward solution approach determined that the forward/backward substitution approach will run in approximately ¼ of the time of the LU solution. This indicates that the forward/backward substitution is substantially more computationally efficient for solving the 1st order Vold-Kalman filter.

If a forward/backward substitution method is used for solution, further computational advantages can be gained if the same weighting factor is used for either multiple orders or multiple channels of data based on the formulation using Equation 9 to shift each order or channel to DC [ref. 3,4,10]. The forward substitution

portion of the solution is independent of the data and relies only on the normal equations matrix of the left hand side of the equation. The forward substitution may be performed and stored for repeated application of the backward substitution to solve for the multiple orders or channels of data. This computational advantage means that for additional orders or channels after the initial order or channel, the number of calculations necessary to solve for the orders is reduced by a factor of 2! An example of how the equation is setup to accomplish this type of solution is shown in Equation 15.

$$\begin{bmatrix} 1 & -1 & \\ & 1 & -1 \\ r(n) & & r(n+1) \end{bmatrix} \begin{Bmatrix} x_1(n) & x_2(n) \\ x_1(n+1) & x_2(n+1) \\ x_1(n+2) & x_2(n+2) \end{Bmatrix} = \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ r(n)y_{1DC}(n) & r(n)y_{2DC}(n) \\ r(n+1)y_{1DC}(n+1) & r(n+1)y_{2DC}(n+1) \end{Bmatrix} \quad (15)$$

where: $x_1(n)$ = order amplitude for channel or order #1
 $x_2(n)$ = order amplitude for channel or order #2
 $y_{1DC}(n)$ = order or channel #1 shifted to DC
 $y_{2DC}(n)$ = order or channel #2 shifted to DC

Realize that this formulation using the forward/backward substitution to reduce computation time only works when all orders and/or channels being tracked use the same weighting factor with the 1st order Vold-Kalman filter.

KALMAN/VOLD-KALMAN APPLICATIONS

Choosing a relatively high value for the HCF weights the structure equation more heavily in the solution process and results in a filter shape that is very narrow. This allows very little sideband information to pass through the filter and hence only allows the amplitude to change slowly. A relatively high HCF then gives a very sharp filter with very good frequency discrimination; mathematically this solution is weighting the structure equation much higher than the data equation. Since the structure equation describes the characteristics of a sine wave, this implies that the order of interest is a sine wave which is stationary with respect to frequency and amplitude over the points which it is applied.

Choosing a relatively low value for the HCF has the effect of weighting the data equation more heavily in the solution process and results in a filter that does not possess as sharp a rolloff and therefore is not as frequency discriminating as the high HCF filter. Using a low HCF allows the amplitude of the filtered order to vary much more quickly than the high HCF filter. This behavior is necessary around lightly damped resonances or in fast speed sweeps where the

frequency and amplitude of the order must be allowed to change quickly. The various filter characteristics and how their tracking characteristics vary relative to the HCF are well documented [1,10-14].

The formulations, derivations, and discussions of both the original and the Vold-Kalman filters are centered about a weighting factor that is constant as a function of time. This is not a requirement of either filter's formulation. In fact, all commercial implementations of either filter allow the weighting factor to vary as a function of time or rpm. If the weighting factor is varied as a function of rpm, a pseudo-constant order bandwidth filter may be obtained. Another strategy to vary the weighting factor of the filter is based on instants of known transient activity in the data. Examples of this transient activity are gear shifts or clutch engagements. In these areas of transient activity, it is assumed that the amplitude of a tracked order may change very quickly, thus the weighting factor is reduced in these regions to allow more sideband energy to pass through the filter. The additional sideband energy allows the amplitude of the order to change very quickly in these regions, allowing a realistic amplitude profile to be estimated.

The best frequency discrimination that can be expected from the Kalman filters is the same as that of the Fourier transform. Frequencies closer together than the inverse of the total length of time of the data cannot be effectively separated, as defined by Rayleigh's criteria.

While the Kalman order tracking methods have many advantages over the traditional order tracking methods, including better dynamic range and the time domain order extraction, they do have some disadvantages. Computational complexity is one disadvantage of the adaptive filters. The largest disadvantage, however, is the experience required to get valid accurate results for each order extraction. The experience is required in choosing the appropriate weighting factor to extract the order with a minimal bandwidth while tracking the amplitude profile accurately. The weighting factor that is necessary for an accurate extraction is a function of the other orders present in the data, the sweep rate, and the properties of excited resonances. All of these items, which can vary from channel to channel, change the rate at which the amplitude of an order changes and therefore affect the width of the filter necessary for an accurate extraction. A very complete discussion of what aspects of the data to consider when choosing a weighting factor and/or bandwidth for the Vold-Kalman filters is presented in references [12-14].

CONCLUSION

The mathematical formulations for the implementation of both Kalman and Vold-Kalman filters were presented. Included in these formulations were the necessary equations to implement the Vold-Kalman filter using either the 1st order formulation or a higher order formulation. Formulations above a 3rd order filter are

possible and should be easily derived from the equations provided.

Solution methods for all of the filter implementations were presented. It was found that for the 1st order Vold-Kalman filter the most computationally efficient solution was the forward/backward substitution. The 2nd and 3rd order implementations are efficiently solved by taking advantage of the fact that the matrices which must be inverted are sparse in nature and using the LU Decomposition. It was also found that the 3rd order filter requires about twice the compute time of the 1st order filter if both are solved using the LU Decomposition. The number of computations for all filters scales linearly with the length of the time history which the filter is being applied to.

The solution of very long time histories was handled through dividing the time history into shorter overlapping blocks and applying the filter to each block. The resulting filtered blocks were then weighted with a cosine taper at both their ends and summed together to achieve a complete transient free filtered time history. This solution method was shown to provide the same result as filtering the time history all in one solution.

It is believed that by coupling this paper with the many papers already written about the Vold-Kalman filters, that they can be easily implemented by practicing engineers in a software environment such as Matlab.

REFERENCES

1. Vold, H. and Leuridan, J., "High Resolution Order Tracking at Extreme Slew Rates, Using Kalman Tracking Filters", Proceedings of the SAE Noise and Vibration Conference, Traverse City, MI., 1993, SAE paper no. 931288.
2. 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.
3. 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.
4. 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.
5. 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.
6. Grewal, M.S. and Andrews, A.P., "Kalman Filtering - Theory and Practice", Prentice Hall Inc., Englewood Cliffs, New Jersey, 1993.

7. Haykin, S., "Adaptive Filter Theory", Second Edition, Prentice Hall Inc, Englewood Cliffs, New Jersey, 1991.
8. James, M.L., Smith, G.M., and Welford, J.C., "Applied Numerical Methods For Digital Computation", Third Edition, Harper & Row Publishers Inc., New York, New York, 1985.
9. Flannery, B.P., Press, W.H., Teukolsky, S.A., and Vetterling, W.T., "Numerical Recipes in Fortran – The Art of Scientific Computing", Second Edition, Cambridge University Press, New York, New York, 1992.
10. Blough, J.R., "Improving the Analysis of Operating Data on Rotating Automotive Components", Ph.D. Dissertation, University of Cincinnati, 1998.
11. Blough, J.R., and Gwaltney, G.D., "Summary and Characteristics of Rotating Machinery Digital Signal Processing Methods", Proceedings of the Society of Automotive Engineers 1999 Off-Highway Conference, SAE paper number 99F-40.
12. Gade, S., Herlufson, H., Konstantin-Hansen, H., and Vold, H., Brüel & Kjaer Technical Review, Characteristics of the Vold-Kalman Order Tracking Filter, No. 1 – 1999, Brüel & Kjaer Sound&Vibration Measurement A/S, DK-2850 Nærum, Denmark.
13. Herlufson, H., Gade, S., Konstantin-Hansen, H., and Vold, H., "Characteristics of the Vold/Kalman Order Tracking Filter", Sound and Vibration Magazine, April 1999, pp.30-34.
14. 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.
15. Feldbauer, C. and H"oldrich, R., "Realization of a Vold-Kalman Tracking Filter – A Least Squares Problem," Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9, 2000.

CONTACT

The author may be contacted with questions or comments either by phone at 1-906-487-1020 or email at: jrblough@mtu.edu.