REALIZATION OF A VOLD-KALMAN TRACKING FILTER — A LEAST SQUARES PROBLEM

Christian Feldbauer, Robert Höldrich

Institute of Electronic Music and Acoustics
University of Music and Dramatic Arts, Graz, Austria
feldbauer@iem.kug.ac.at, robert.hoeldrich@kug.ac.at
iem.kug.ac.at/˜feldbauer/, www.kug.ac.at/iem/info/personal/rh.htm

ABSTRACT

The aim of this work was the implementation of the so-called Vold-Kalman filter. This filter was introduced by Vold and Leuridan in 1993 [1], it is a heterodyne filter for tracking the sinusoidal components of a noisy signal. The formulation of the Vold-Kalman filter leads to a least squares problem. The great advantage of this time-varying filter is that all sinusoids of a signal can be extracted simultaneously yielding a suppression of beating phenomena of close or crossing frequency trajectories. In this paper, we propose a realization for offline processing using the preconditioned conjugate gradient method. Furthermore, we present trivial expressions for the bandwidth and the transition time of first and second order filters.

1. INTRODUCTION

Let us consider a real valued signal as a sum of sinusoidals plus noise:

$$sig(n) = \sum_k x_k(n)\, e^{i\varphi_k(n)} + noise(n) \tag{1}$$

A sine wave in this signal model can be represented as a modulated carrier wave where the frequency of the carrier is usually not constant. The slowly time-varying (complex) amplitude $x_k(n)$ modulates the carrier $e^{i\varphi_k(n)}$. If the phase sequence $\varphi_k(n)$ is known exactly the envelope $x_k(n)$ is real valued (modulation of amplitude only), but in most practical problems just an estimate of the instantaneous frequency is available resulting in a complex valued modulator (modulation of amplitude and phase).

The aim of a tracking filter is to extract selected components that means to determine the complex envelope $x_k(n)$ for a number of sinusoidals and given estimates for the instantaneous frequencies. We will specify the sinusoids of interest with the subscripts $1 \dots K$.

The base of the Vold-Kalman filter is formed by two equations: the data and the structural equation [1] [2].

1.1. Data Equation

In order to extract the K partials simultaneously, the so-called data equation is

$$sig(n) - \sum_{k=1}^{K} x_k(n) e^{i\varphi_k(n)} = \delta(n) \tag{2}$$

$\delta(n)$ is an error sequence or noise which should be minimal. It is also possible to track the sine waves independently — the data equation is simplified to

$$sig(n) - x_k(n) e^{i\varphi_k(n)} = \delta_k(n) \tag{3}$$

The advantage of coupling in equation (2) is the suppression of beating phenomena of close or crossing frequency trajectories, the drawback is a much larger system of equations.

1.2. Structural Equation

The so-called structural equation (4) results in a smooth modulation sequence $x_k(n)$ if the error sequence $\varepsilon_k(n)$ is minimized. It works like a lowpass filter.

$$\nabla^{p+1} x_k(n) = \varepsilon_k(n) \tag{4}$$

∇^s is the difference operator of order s and p is the order of our filter. For example the structural equation of a first order filter is

$$x_k(n-1) - 2\, x_k(n) + x_k(n+1) = \varepsilon_k(n) \tag{5}$$

1.3. Combination

The following realization uses a non-causal filter yielding a complex envelope without phase bias. It is not a realtime application.

Using vector notation, the data equation (2) can be written as

$$\mathbf{sig} - \sum_{k=1}^{K} C_k \mathbf{x}_k = \delta \tag{6}$$

where $C_k \mathbf{x}_k = \mathrm{diag}(\mathbf{c}_k)\mathbf{x}_k = \mathbf{c}_k \circ \mathbf{x}_k$ ($\circ$ means element-by-element multiplication). The column vector $\mathbf{c}_k$ contains the N samples of the complex valued carrier $c_k(n) = e^{i\varphi_k(n)}$ of the k^{th} partial, $\mathbf{x}_k$ stands for the N unknown samples of the complex envelope and $\mathbf{sig}$ for the N samples of the real valued signal. The structural equation (4) reads as

$$Str_p\, \mathbf{x}_k = \varepsilon_k \tag{7}$$

with the matrix Str_p which equals for a first order filter

$$Str_1 = \begin{bmatrix} -2 & 1 & 0 & & 0 \\ 1 & -2 & 1 & & 0 \\ 0 & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & 0 & & 1 & -2 \end{bmatrix} \tag{8}$$

Furthermore, the error vectors ε_k for $k = 1 \dots K$ are linked to $\varepsilon = \left[(R_1\varepsilon_1)^\mathsf{T}, \dots, (R_K\varepsilon_K)^\mathsf{T}\right]^\mathsf{T}$ where R_k is the weighting factor of the k^{th} structural equation. Finally we join ε and δ together to $\mathbf{res} = \left[\varepsilon^\mathsf{T}, \delta^\mathsf{T}\right]^\mathsf{T}$ and minimize the norm of $\mathbf{res}$:

$$\min_{\mathbf{x}} \|\mathbf{res}\|^2 = \min_{\mathbf{x}} \mathbf{res}^\mathsf{H}\mathbf{res} \tag{9}$$

The purpose of the weighting factor is to balance the influence of the k^{th} structural equation in the minimization of $\mathbf{res}$. If R_k is large, $\mathbf{res}$ is strongly influenced by the structural equation yielding a very smooth modulator. That means that the bandwidth of the filter for the k^{th} sinusoid is very small (see section 3.1). R_k can be a real scalar or a diagonal matrix.

$$R_k = \begin{cases} r_k & , \quad \text{constant bandwidth} \\ \operatorname{diag}(\mathbf{r_k}) & , \quad \text{variant bandwidth} \end{cases} \tag{10}$$

The later can be used for a bandwidth proportional to the frequency of the sinusoid (constant percentage bandwidth) as an example.

2. THE LEAST SQUARES PROBLEM

The minimization in expression (9) is equivalent to find the optimal solution (in the sense of least squares) of the following over-constrained system

$$A\mathbf{x} \approx \mathbf{b} \tag{11}$$

which consists of $(K+1)\,N$ equations and KN unknown variables. The vector $\mathbf{x}$ contains the $\mathbf{x}_k$ for $k = 1, \dots, K$: $\mathbf{x} = \left[\mathbf{x}_1^\mathsf{T},\, \mathbf{x}_2^\mathsf{T}\, \dots\, \mathbf{x}_K^\mathsf{T}\right]^\mathsf{T}$. The matrix is

$$A = \begin{bmatrix} R_1 Str_p & 0 & \cdots & 0 \\ 0 & R_2 Str_p & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_K Str_p \\ C_1 & C_2 & \cdots & C_K \end{bmatrix} \tag{12}$$

with the sparse band matrix Str_p obtained from the structural equation and the diagonal matrix C_k from the data equation. The vector on the right side of (11) is

$$\mathbf{b} = \left[0,\, 0,\, \dots,\, 0,\, \mathbf{sig}^\mathsf{T}\right]^\mathsf{T} \tag{13}$$

2.1. The Normal Equations

Using the Normal Equations

$$A^\mathsf{H} A\, \mathbf{x} = A^\mathsf{H}\, \mathbf{b} \tag{14}$$

one obtains a system of equations

$$\begin{bmatrix} B_1 & C_{1,2} & \cdots & C_{1,K} \\ C_{2,1} & B_2 & \cdots & C_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ C_{K,1} & C_{K,2} & \cdots & B_K \end{bmatrix} \mathbf{x} = \begin{bmatrix} \overline{C}_1\mathbf{sig} \\ \overline{C}_2\mathbf{sig} \\ \vdots \\ \overline{C}_K\mathbf{sig} \end{bmatrix} \tag{15}$$

which has KN equations and KN unknown variables with the complex diagonal matrices

$$C_{u,v} = \operatorname{diag}(\overline{\mathbf{c}}_u \circ \mathbf{c}_v) \tag{16}$$

and symmetric band matrices.

$$B_k = Str_p^\mathsf{T} R_k^2 Str_p + I \tag{17}$$

A^H means transposition and conjugation $A^\mathsf{H} = \overline{A}^\mathsf{T}$ and I is the unity matrix.

In order to extract the partials independently, one has to solve the K smaller systems

$$B_k\, \mathbf{x}_k = \overline{C}_k \mathbf{sig} \qquad\qquad k = 1, \dots, K \tag{18}$$

Because N is usually very big and the problems (15) or (18) are very ill-conditioned, we use the preconditioned conjugate gradient method for solving the systems.

2.2. The Preconditioned CG Method

I'd like to present the preconditioned conjugate gradient method for solving $A\mathbf{x} = \mathbf{b}$ now [3]. The superscript $^{(j)}$ indicates the iteration number.

choose $\mathbf{x}^{(0)}$ set $\mathbf{g}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{b}$

solve $M\hat{\mathbf{g}}^{(0)} = \mathbf{g}^{(0)}$ set $\mathbf{d}^{(0)} = \hat{\mathbf{g}}^{(0)}$

for $j = 0,\, 1,\, \dots$

$$\alpha^{(j)} = \frac{\left(\hat{\mathbf{g}}^{(j)}\right)^\mathsf{H} \mathbf{g}^{(j)}}{\left(\mathbf{d}^{(j)}\right)^\mathsf{H} A\mathbf{d}^{(j)}} \tag{19}$$

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} - \alpha^{(j)}\mathbf{d}^{(j)} \tag{20}$$

$$\mathbf{g}^{(j+1)} = \mathbf{g}^{(j)} - \alpha^{(j)} A\mathbf{d}^{(j)} \tag{21}$$

$$\text{solve } M\hat{\mathbf{g}}^{(j+1)} = \mathbf{g}^{(j+1)} \tag{22}$$

if convergence stop loop

$$\beta^{(j)} = \frac{\left(\hat{\mathbf{g}}^{(j+1)}\right)^\mathsf{H} \mathbf{g}^{(j+1)}}{\left(\hat{\mathbf{g}}^{(j)}\right)^\mathsf{H} \mathbf{g}^{(j)}} \tag{23}$$

$$\mathbf{d}^{(j+1)} = \beta^{(j)}\mathbf{d}^{(j)} + \hat{\mathbf{g}}^{(j+1)} \tag{24}$$

There are two criteria to choose the preconditioning matrix M:

- The matrix M should be similar to A.
- In every step the equation (22) has to be solved, therefore the inversion of M should be easy.

The problem is that these two points contradict each other.

Fortunately, we found an ideal matrix M by chance. Concerning the first criterion, one can say that $Str_p^\mathsf{T} Str_p$ is the dominant part of A especially for large values of r_k (small filter bandwidths).

$$M = (LU)^{p+1} \approx Str_p^\mathsf{T} Str_p \approx A \qquad \text{if } r_k >> 1 \tag{25}$$

where

$$L = \begin{bmatrix} 1 & & & 0 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix} \tag{26}$$

and

$$U = L^\mathsf{T} = \begin{bmatrix} 1 & -1 & & 0 \\ & 1 & \ddots & \\ & & \ddots & -1 \\ 0 & & & 1 \end{bmatrix} \tag{27}$$

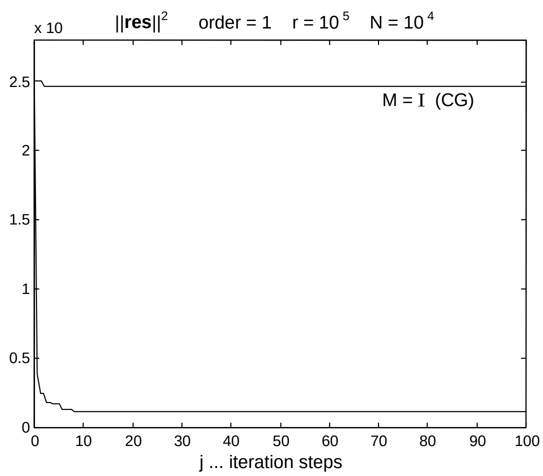
The second criterion is that the inversion of M should be easy: L and U can be inverted without any problems.

$$L^{-1} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (28)$$

and

$$U^{-1} = \left(L^{-1}\right)^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \cdots & 1 \\ & & \ddots & \vdots \\ 0 & & & 1 \end{bmatrix} \quad (29)$$

Thus the matrix M is optimal for the PCG algorithm and an enormous speedup will be achieved compared to the standard CG method where $M = I$. Figure 1 illustrates the effect of this preconditioning. The cost function $||\mathbf{res}||^2$, which should be minimized, is calculated after every step j and plotted.



M^{-1}

(22) in every step. This is easily done by

$$\begin{aligned} M^{-1}\mathbf{v} &= \ldots U^{-1}L^{-1}\mathbf{v} \\ &= \ldots \mathrm{flip}\left(\mathrm{cumsum}\left(\mathrm{flip}\left(\mathrm{cumsum}(\mathbf{v})\right)\right)\right) \end{aligned} \quad (30)$$

where $\mathrm{flip}(\mathbf{v})$ reverses the vector $\mathbf{v}$ and $\mathrm{cumsum}(\mathbf{v})$ results in a vector containing the cumulative sums of the elements of $\mathbf{v}$.

3. CHARACTERISTICS OF THE VOLD-KALMAN FILTER

We examined the Vold-Kalman filter to find trivial expressions for the bandwidth (-3 dB) and the transition time (10 % – 90 %) of first and second order filters as functions of the weighting factor. Concerning the filter shape one can say that the selectivity increases with growing filter order (see [4] and [5] for more information).

3.1. Bandwidth

Vold proposed in [2] to construct an empirical table of weighting factors and corresponding bandwidth values to obtain a desired bandwidth by interpolating in this table. We found trivial mathematical expressions (31) and (32) so we no longer need such a table.

$$B_{3dB}(r) = 1.58\, r^{-\frac{1}{2}} \quad (31)$$

This is the -3 dB bandwidth of the first order Vold-Kalman bandpass filter in radians. One has to multiply it by $\frac{f_s}{2\pi}$ to get Hz where f_s is the sampling rate in Hz. For the second order filter we got:

$$B_{3dB}(r) = 1.70\, r^{-\frac{1}{3}} \quad (32)$$

Figure 2 shows these two functions.

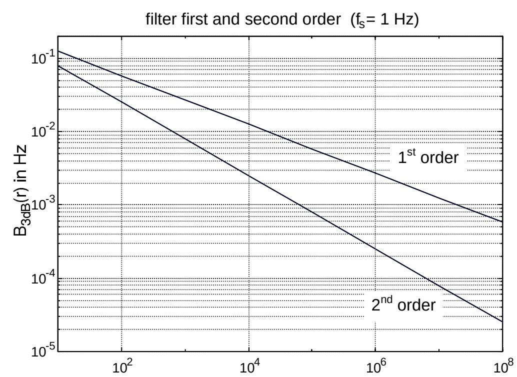


Figure 2: *The Bandwidth as a function of the weighting factor.*

3.2. Transition Time

Concerning the time response we measured the 10 % – 90 % transition time and found the following expression for a first order filter:

$$T(r) = \frac{4.51}{B_{3dB}(r)} = 2.85\, r^{\frac{1}{2}} \quad (33)$$

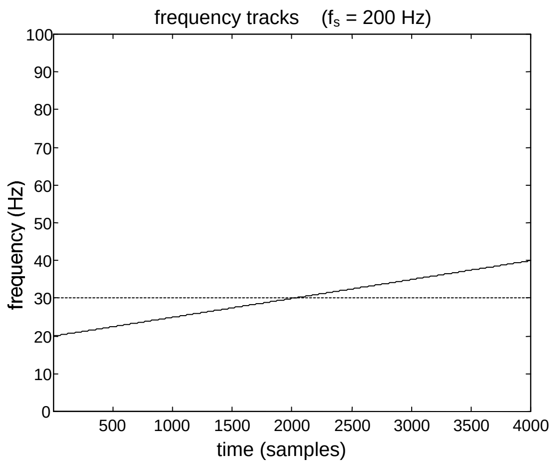
and for a second order filter:

$$T(r) = \frac{4.75}{B_{3dB}(r)} = 2.80\, r^{\frac{1}{3}} \quad (34)$$

The product $T\, B_{3dB}$ is a constant. It's important to mention that the measurements yielded results with a big variance. So equations (31) – (34) should not be regarded as the ultimate results.

3.3. The Effect of Coupling

As already mentioned, solving the larger system of equations (15) instead of the set of the K smaller ones (18) leads to a suppression of beating phenomena of close or crossing frequency tracks. Figures 3 – 5 illustrate this great advantage. The signal in this example consists of two sinusoids. The first one (labeled with 1) has a constant frequency of 30 Hz and a constant amplitude of 2. The frequency of the second one (labeled with 2) goes linearly from 20 Hz to 40 Hz, the amplitude linearly from 0.5 to 1. Figure 4 shows the results of (18) where the amplitudes are extracted independently. In the area of the crossing of the frequency tracks an increase can be observed because the distribution of the energy between the sinusoids is not well defined. Any other conventional heterodyne filter produces comparable results. In figure 5 where the results of (15) are plotted, this problem is eliminated.



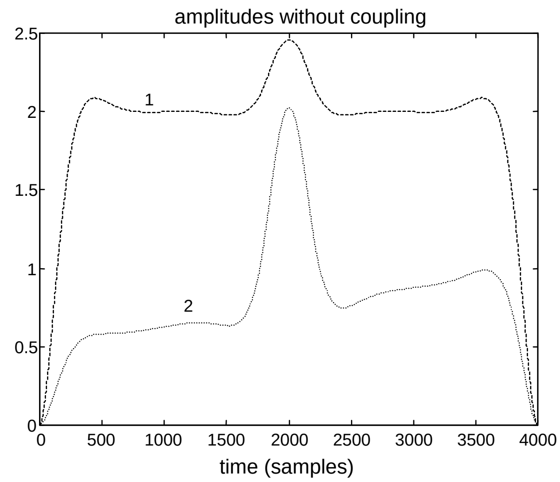


Figure 4: *The independently extracted amplitudes.*

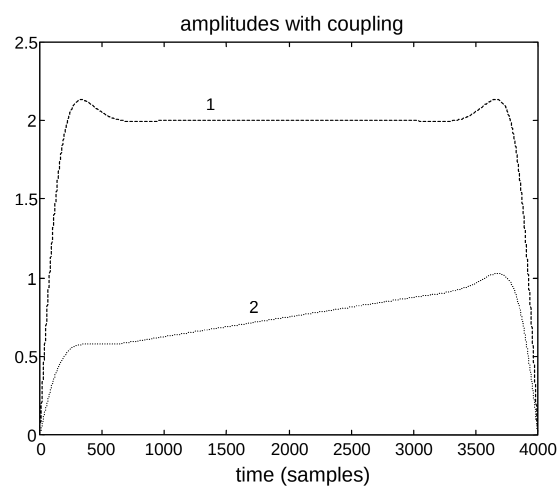


Figure 5: *The simultaneously extracted amplitudes.*