A FREQUENCY TRACKER BASED ON A KALMAN FILTER UPDATE OF A SINGLE PARAMETER ADAPTIVE NOTCH FILTER

Randall Ali and Toon van Waterschoot *

Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Belgium
randall.ali@esat.kuleuven.be | toon.vanwaterschoot@esat.kuleuven.be

ABSTRACT

In designing a frequency tracker, the goal is to follow the continual time variation of the frequency from a particular sinusoidal component in a noisy signal with a high accuracy and a low sample delay. Although there exists a plethora of frequency trackers in the literature, in this paper, we focus on the particular class of frequency trackers that are built upon an adaptive notch filter (ANF), i.e. a constrained bi-quadratic infinite impulse response filter, where only a single parameter needs to be estimated. As opposed to using the conventional least-mean-square (LMS) algorithm, we present an alternative approach for the estimation of this parameter, which ultimately corresponds to the frequency to be tracked. Specifically, we reformulate the ANF in terms of a state-space model, where the state contains the unknown parameter and can be subsequently updated using a Kalman filter. We also demonstrate that such an approach is equivalent to doing a normalized LMS filter update, where the regularization parameter can be expressed as the ratio of the variance of the measurement noise to the variance of the prediction error. Through an evaluation with both simulated and realistic data, it is shown that in comparison to the LMS-updated frequency tracker, the proposed Kalman-updated alternative, results in a more accurate performance, with a faster convergence rate, while maintaining a low computational complexity and the ability to be updated on a sample-by-sample basis.

1. INTRODUCTION

Frequency estimation is a well-known problem in signal processing with a long history [1, 2] and continues to be relevant for a number of audio-related applications[1] including acoustic feedback detection [3, 4], automatic music transcription [5], tuning of musical instruments, and audio effects such as pitch-shifting just to name a few. Specifically, it refers to the problem of estimating the frequency of a sinusoidal component from a set of noisy observations. In cases where the periodic component of the noisy observations consist of harmonically-related sinusoidal components and it is the lowest frequency component that is being estimated, the problem is often referred to as fundamental frequency (f_0) estimation or pitch estimation [1, 2].

In this paper we are concerned with following the continual time variation of the frequency pertaining to a particular sinusoidal component of an audio signal, and hence we refer to the type of frequency estimation as frequency tracking. The main consideration for a frequency tracker is that it needs to have a low sample delay, with the ideal case being zero delay, and can be updated on a sample-by-sample basis. More concretely, the problem of frequency tracking can be summarized as finding an updated estimate of the frequency given a new set of samples or simply just one (in the real-time scenario) and a prior estimate of the frequency [6]. Several approaches for this have been proposed in the literature [6, 7, 8, 9], where the problem is referred to as pitch tracking.

The frequency tracker investigated in this work is one based on adapting the coefficients of a constrained bi-quadratic (biquad) infinite impulse response (IIR) filter [3, 4, 10, 11, 12], which functions as an adaptive notch filter (ANF). In a nutshell, the centre frequency of the ANF is continually updated so as to cancel a high-energy sinusoidal component in an attempt to minimize the mean-square of the output signal power. One main advantage of using the constrained biquad filter is that only one parameter needs to be adapted in order to obtain a frequency estimate. Furthermore, by expressing the filter in its direct form II, this single parameter can be updated very efficiently [12] such as with a least-mean-square (LMS) algorithm [3, 4], resulting in a frequency tracker that can be updated on a sample by sample basis. Moreover, the computational complexity is very low, making the algorithm suitable for real-time applications.

Our contribution in this paper is a subtle but powerful extension of the aforementioned approach, whereby we reformulate the ANF in terms of a state-space model, with the state containing the unknown parameter to be estimated. In such a formulation, a Kalman filter [13] can then be used for updating the state and hence for frequency estimation and tracking. We subsequently refer to this frequency tracker[2] as a Kalman ANF (KalmANF). We will demonstrate that the KalmANF is equivalent to a normalized LMS (NLMS) filter update [14, 15], where the regularization parameter can be expressed as the ratio of the variance of the measurement noise to the variance of the prediction error [14], both of which can be tuned accordingly. This results in a more accurate frequency tracker as compared to one that uses an ANF updated with an LMS algorithm, while maintaining a low computational

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956369. The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

[1] Although certainly not limited to audio as it is also relevant in biomedical signal processing, power-line monitoring, and seismology for instance.

[2] As a consequence of using the ANF approach, we stick to the term of frequency tracking as opposed to pitch estimation since an estimation of the frequency will not necessarily correspond to a fundamental frequency, but to the frequency of the sinusoidal component contributing to most of the energy in the audio signal.

complexity and the ability to be updated on a sample-by-sample basis.

In relation to prior work, a state-space approach of the ANF has also been considered in [16], however an alternative formulation was used, and a relation with the normalized LMS was not established. In [17], an extended Kalman filter was used to update a single parameter adaptive comb filter (i.e. multiple ANFs), whereas we consider a single ANF in this paper, which allows us to have a linear state-space model. Several other Kalman filtering-based approaches to frequency tracking also exist [18, 19, 20, 21, 22], but they are not built upon an ANF.

The remainder of this paper is organized as follows. In Section 2, we review the ANF, i.e. the constrained biquad IIR filter, and how the LMS algorithm is used to update the filter coefficients. In section 3, we reformulate the problem in terms of a state-space model where the state contains the relevant filter coefficients to be updated. By applying a Kalman filter, it is then shown how the KalmANF is equivalent to using an NLMS algorithm with a well-defined time-varying regularization parameter. In section 4, we evaluate the KalmANF in comparison to its LMS-based counterpart on both simulated and realistic data, where it is demonstrated that the KalmANF outperforms the ANF frequency tracker that uses an LMS algorithm in terms of its accuracy and convergence speed.

2. LEAST-MEAN-SQUARE ADAPTIVE NOTCH FILTER

Let us consider the following signal model in the discrete-time domain, with n being the discrete-time index:

$$y(n) = A_o(n)\sin\left[n\,\omega_o(n) + \phi_o(n)\right] + g(n) \tag{1}$$

where $y(n)$ is a measured signal consisting of a sinusoidal component, $A_o(n)\sin\left[n\,\omega_o(n) + \phi_o(n)\right]$, with time-varying parameters: amplitude $A_o(n)$, phase $\phi_o(n)$, digital angular frequency $\omega_o(n) = 2\pi f_o(n)/f_s$, where $f_o(n)$ is the frequency (Hz), and f_s is the sampling frequency (Hz). This model is very broad in the sense that the remaining component, $g(n)$ can be representative of a number of signals such as a broadband desired signal, additional harmonics, or simply noise depending on the application. Given the measurement, $y(n)$, our goal is to track the time-variation of $f_o(n)$. The approach that we follow is to design an ANF that can be applied to $y(n)$ to effectively suppress the sinusoidal component and by consequence will also result in a frequency tracker.

A well-known technique of designing an ANF is to adaptively compute the parameters of a constrained IIR filter [10, 11, 12], which can be done quite efficiently using an LMS algorithm [3]. In this paper, since our signal model only consists of one sinusoidal component, we will simply consider a constrained biquad IIR filter, i.e. with two-zeros and two-poles. Firstly, let us recall the biquad filter in the z-domain without any constraints:

$$H(z^{-1}) = \frac{b_o + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2}$$

which for $b_o = 1$, can be expressed in polar coordinates (in the complex plane) in terms of a zero radius, ζ, and zero angle ω_z, and a pole radius, ρ, and pole angle, ω_p as follows

$$H(z^{-1}) = \frac{(1 - \zeta e^{j\omega_z} z^{-1})(1 - \zeta e^{-j\omega_z} z^{-1})}{(1 - \rho e^{j\omega_p} z^{-1})(1 - \rho e^{-j\omega_p} z^{-1})} \tag{3}$$

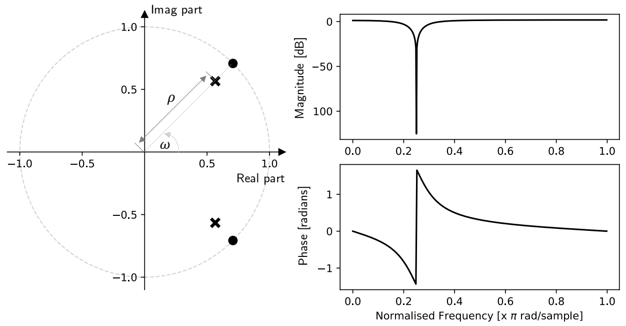

Figure 1: *(Left) Pole-zero plot of a constrained biquad IIR filter configured as a notch filter. The poles and zeros lie on the same radial line defined by $\omega = \pi/4$, where the zeros are placed on the unit circle and the poles at a distance $\rho = 0.8$. (Right) The corresponding magnitude and phase response. A notch is clearly visible at $\omega = \pi/4$ with a very narrow bandwidth due to $\rho = 0.8$.*

where $b_1 = -(\zeta e^{j\omega_z} + \zeta e^{-j\omega_z}) = -2\zeta\cos(\omega_z)$, $b_2 = \zeta^2$, $a_1 = -2\rho\cos(\omega_p)$, and $a_2 = \rho^2$. In order to convert this filter into a more suitable form where its coefficients can be adapted, two constraints need to be subsequently introduced.

The first of these constraints as proposed in [10] is to make the poles and zeros lie on the same radial line, defined by angle ω in the complex plane (see Fig. 1), i.e. $\omega_z = \omega_p = \omega$. These poles and zeros must also lie completely within the unit circle, where the zeros would be in between the poles and the unit circle in order to define a notch filter. The intuition behind this is that placing a zero near to the unit circle would attenuate all the frequency components in the neighbourhood of the angular frequency, ω, defining that particular radial line. Placing a pole on the same radial line then creates a resonance at ω, with the bandwidth of the notch filter becoming narrower as $\rho \to \zeta$.

The second constraint on the biquad filter is to let the zeros all lie on the unit circle [11] so that $\zeta = 1$. In this case the frequency component at ω would be completely attenuated and the pole at the same radial line would once again create a resonance at ω, with the bandwidth of the notch filter becoming narrower as $\rho \to 1$.

Imposing these constraints on the biquad filter of (3), results in the constrained biquad filter:

$$\begin{aligned} H(z^{-1}) &= \frac{(1 - e^{j\omega} z^{-1})(1 - e^{-j\omega} z^{-1})}{(1 - \rho e^{j\omega} z^{-1})(1 - \rho e^{-j\omega} z^{-1})} \\ &= \frac{1 - 2\cos(\omega) z^{-1} + z^{-2}}{1 - 2\rho\cos(\omega) z^{-1} + \rho^2 z^{-2}} \\ &= \frac{1 - a z^{-1} + z^{-2}}{1 - \rho a z^{-1} + \rho^2 z^{-2}} \end{aligned} \tag{4}$$

where $a \triangleq 2\cos(\omega) = 2\cos(2\pi f/f_s)$ is the only parameter we need to estimate (since it appears in both the numerator and denominator) and is directly related to the centre frequency, f, of the notch filter. Consequently, by adapting the a coefficient, the centre frequency of the notch filter also changes resulting in an ANF. The pole-zero plot and corresponding magnitude and phase response of an example constrained biquad filter is shown in Fig. 1.

In order to estimate the frequency, f_o, of the sinusoid in (1), we need to find the parameter, a, such that when the ANF is applied to the input (or measured) signal, $y(n)$, the output signal power of

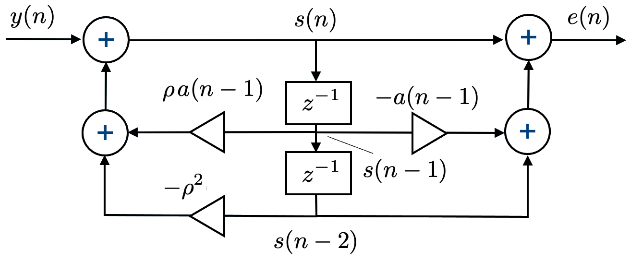

Figure 2: *Direct form II of the constrained biquad filter.*

the filter is minimal in the mean-squared sense. This would imply that the centre frequency of the ANF would have been updated to be f_o, thereby cancelling the high-energy sinusoid, resulting in a minimum mean-square output signal power.

An efficient method to estimate a can be derived by considering the direct form II of the constrained biquad filter [12] as illustrated in Fig. 2. The implementation equations are given as

$$s(n) = y(n) + \rho a(n-1)s(n-1) - \rho^2 s(n-2) \tag{5}$$

$$e(n) = s(n) - a(n-1)s(n-1) + s(n-2) \tag{6}$$

where $y(n)$ is the input to the constrained biquad filter (the measured signal from (1)), $e(n)$ is the output, and $s(n)$ is introduced as an auxiliary variable. In this form, the biquad filter is explicitly split into two sections. The first is a two-pole resonance IIR filter illustrated on the left side of Fig. 2 corresponding to the denominator of (4) and whose difference equation is given by (5). The second section is a finite impulse response (FIR) two-zero notch filter illustrated on the right side of Fig. 2, corresponding to the numerator of (4) and whose difference equation is given by (6).

We can now proceed to estimate a by minimizing the mean-squared output signal power of the filter, i.e. minimizing the mean-square of $e(n)$. In [12], it was proposed to only update the FIR section of the biquad filter, i.e. estimate the coefficient a in the FIR section, and since this a occurs in both the numerator and denominator in (4), this estimate can be simply copied to the IIR section of the biquad filter. An LMS algorithm can then be used to estimate a by making use of (6) as follows [3, 4]

$$\begin{aligned}\hat{a}(n) &= \hat{a}(n-1) + \mu\left(-\frac{\partial e^2(n)}{\partial a(n-1)}\right)\\ &= \hat{a}(n-1) + 2\mu\, s(n-1)e(n)\end{aligned} \tag{7}$$

where μ is the step size parameter. As opposed to estimating a, we could alternatively attempt to directly estimate $\omega_o = 2\pi f_o$, however this would result in a nonlinear update equation that will not have the properties of an LMS algorithm, and hence we stick to estimating a from which we can then obtain an estimate for f_o.

The algorithm for computing frequency estimates using such an LMS update is given in Algorithm 1 and we subsequently refer to the resulting frequency tracker as LMS-ANF. In Algorithm 1, N is the length of the signal, $y(n)$, and due to the $s(n-2)$ term, we simply start the for loop from $n = 2$ and initialize $s(0)$ and $\hat{a}(1), s(1)$. Since $\arccos(\hat{a}(n)/2)$ does not exist for $|\hat{a}(n)| > 2$, we have additionally imposed a constraint on the values of $\hat{a}(n)$ such that $\hat{a}(n)$ is re-initialized to zero when $|\hat{a}(n)| > 2$, i.e. we restart the algorithm[3] with an initial frequency estimate at half of the Nyquist frequency. By defining the computational complexity as the number of multiplications per recursion, from Algorithm 1, we can deduce that the LMS-ANF has a computational complexity of 11 multiplications per recursion. It should also be noted that in addition to obtaining a sample-by-sample update of the estimated frequency, we also obtain a sample-by-sample update of the output (i.e. adaptive notch-filtered input signal), $e(n)$, however we are only concerned with the former as it pertains to the frequency tracker.

As previously mentioned, by thinking of the biquad filter as consisting of a two-pole IIR resonance filter, followed by a two-zero FIR notch filter as depicted in Fig. 2, we can give the following interpretation to the algorithm. The two-pole IIR resonance filter amplifies the frequency component in $y(n)$ corresponding to the initial value of $\hat{a}(n-1)$ according to (5) so that the signal $s(n)$ would have a fairly dominant component[4] at $\hat{f}_o(n-1)$. The two-zero FIR notch filter then attempts to reduce the error, $e(n)$ by cancelling this same frequency component that was amplified in $s(n)$ as evident by (6). If the true sinusoidal component in $y(n)$ was not amplified in $s(n)$, then the two-zero FIR notch filter would yield a mean-square of $e(n)$ that remains sufficiently large. Consequently a step size according to (7) is taken to update $\hat{a}$, which corresponds to "trying" another frequency to be amplified and notched. This procedure repeats until the frequency corresponding to the true sinusoidal component in $y(n)$ is found, since applying a notch to this component will minimize the mean-square of $e(n)$.

Algorithm 1 *LMS Update of the ANF (LMS-ANF)*

```
Initialize $\hat{a}(1), s(0), s(1) = 0$
Set $\mu, \rho$
for $n = 2$ to $N-1$ do
    $s(n) = y(n) + \rho\hat{a}(n-1)s(n-1) - \rho^2 s(n-2)$
    $e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)$
    $\hat{a}(n) = \hat{a}(n-1) + 2\mu s(n-1)e(n)$
    if $|\hat{a}(n)| > 2$ then
        $\hat{a}(n) = 0$
    end if
    $\hat{f}_o(n) = (f_s/2\pi)\arccos(\hat{a}(n)/2)$
end for
```

3. KALMAN-BASED ADAPTIVE NOTCH FILTER (KALMANF)

In this section, we derive an alternative algorithm for the estimation of a in (4) using a Kalman filter. We will refer to this algorithm as KalmANF and demonstrate that it is an example within the family of normalized LMS algorithms that are based on the Kalman filter [14]. Let us firstly recall the vector form of the state-space model [23, 24]:

$$\mathbf{x}(n) = \mathbf{C}\mathbf{x}(n-1) + \mathbf{w}(n) \tag{8}$$

$$\mathbf{z}(n) = \mathbf{H}\mathbf{x}(n) + \mathbf{v}(n) \tag{9}$$

where $\mathbf{x}(n) \in \mathbb{R}^L$ is the state vector at time n, $\mathbf{C} \in \mathbb{R}^{L\times L}$ is the state-transition matrix, $\mathbf{w}(n) \in \mathbb{R}^L$ is the process noise vector,

[3] This is certainly not the only strategy to deal with out of range values for $\hat{a}(n)$, but an investigation into this aspect of the ANF is out of the scope of this work.

[4] This of course depends on the value of ρ chosen as well as the signal-to-noise ratio of $y(n)$.

which is modelled as a zero-mean, Gaussian process with covariance matrix $\mathbf{Q} \in \mathbb{R}^{L\times L}$, $\mathbf{z}(n) \in \mathbb{R}^{M}$ is the measurement vector, $\mathbf{H} \in \mathbb{R}^{M\times L}$ is the measurement matrix, and $\mathbf{v}(n) \in \mathbb{R}^{M}$ is the measurement noise vector, also modelled as a zero-mean, Gaussian process but with covariance matrix $\mathbf{R} \in \mathbb{R}^{M\times M}$.

For dynamical systems which can be described in the state-space form of (8) and (9), the state-vector at time n can be estimated using a Kalman filter. The Kalman filter consists of two steps: (i) a prediction (or update) stage and, (ii) an estimation (or measurement) stage, which are performed in a recursive manner, and are given by the following equations [23, 24]

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{C}\hat{\mathbf{x}}(n-1) \tag{10}$$

$$\hat{\mathbf{P}}(n|n-1) = \mathbf{C}\hat{\mathbf{P}}(n-1)\mathbf{C}^T + \mathbf{Q} \tag{11}$$

$$\mathbf{K}(n) = \hat{\mathbf{P}}(n|n-1)\mathbf{H}^T\left(\mathbf{H}\hat{\mathbf{P}}(n|n-1)\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{12}$$

$$\mathbf{v}(n) = \mathbf{z}(n) - \mathbf{H}\hat{\mathbf{x}}(n|n-1) \tag{13}$$

$$\hat{\mathbf{x}}(n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)\mathbf{v}(n) \tag{14}$$

$$\hat{\mathbf{P}}(n) = \left[\mathbf{I} - \mathbf{K}(n)\mathbf{H}\right]\hat{\mathbf{P}}(n|n-1) \tag{15}$$

where $\mathbf{K}(n)$ is the Kalman gain, the notation $\hat{\mathbf{x}}(n|n-1)$ denotes a prediction of $\mathbf{x}(n)$ based on measurement samples up to time $n-1$, and the prediction error is defined as $\mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1)$ with a covariance matrix, $\mathbf{P}(n)$, whose estimate is denoted as $\hat{\mathbf{P}}(n)$. The first two equations, (10) and (11), are the prediction equations, which update the state and the covariance matrix of the prediction error from measurement samples up to time $n-1$. The subsequent equations are the estimation equations. $\mathbf{v}(n)$ in (13) is also referred to as the innovation signal, which is the error between the new measurement at time n and the prediction based on measurement samples up to time $n-1$, and is used to update the state-vector estimate at time n in (14) along with the Kalman gain, $\mathbf{K}(n)$, computed in (12). The prediction error covariance matrix at time n is finally updated in (15), and the entire sequence of equations is repeated for the next time index.

By following the strategy of estimating a in the FIR section of the constrained biquad filter from Fig. 2 and copying the estimate to the IIR section, we can use (5) and (6) to define a state-space model corresponding to the form of (8) and (9) as follows:

$$\underbrace{\begin{bmatrix} a(n) \\ 1 \end{bmatrix}}_{\mathbf{x}(n)} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} a(n-1) \\ 1 \end{bmatrix}}_{\mathbf{x}(n-1)} + \underbrace{\begin{bmatrix} w(n) \\ 0 \end{bmatrix}}_{\mathbf{w}(n)} \tag{16}$$

$$\underbrace{s(n)}_{\mathbf{z}(n)} = \underbrace{\begin{bmatrix} s(n-1) & -s(n-2) \end{bmatrix}}_{\mathbf{H}(n)} \underbrace{\begin{bmatrix} a(n) \\ 1 \end{bmatrix}}_{\mathbf{x}(n)} + \underbrace{e(n)}_{\mathbf{v}(n)} \tag{17}$$

Focusing firstly on (17) and comparing with (9), it is evident that we have defined $s(n)$ as our measurement, which is a scalar. Although we do not explicitly measure $s(n)$, it is a function of the input signal, $y(n)$, and therefore we can use (5) with $a(n-1) = \hat{a}(n-1)$ to obtain a value for $s(n)$. We can also observe that the measurement matrix, $\mathbf{H}(n)$, is now time-varying and depends on two previous measurement samples. The measurement noise vector is also simply a scalar and is the error, $e(n)$, we want to minimize in the ANF context.

Finally we can observe that the state vector, $\mathbf{x}(n)$ is a function of $a(n)$, which is the parameter that we want to estimate. With the measurement equation defined, the state equation of (16) follows directly from (8), where $\mathbf{C}$ is simply an identity matrix and $\mathbf{w}(n)$ has one non-zero value, $w(n)$, since it is only $a(n)$ that needs to be updated.

We can simply proceed to use the equations (10) - (15) to obtain an estimate for $a(n)$. However, because of the low dimensionality of the state-space equations defined in (16) and (17), we can also substitute them into (10) - (15) to obtain simpler and more intuitive expressions to understand how $a(n)$ is being estimated.

Since $\mathbf{C}$ is an identity matrix, (10) is simply

$$\hat{\mathbf{x}}(n|n-1) = \begin{bmatrix} \hat{a}(n|n-1) \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{a}(n-1) \\ 1 \end{bmatrix} \tag{18}$$

We initialize the estimate of the covariance matrix of the prediction error, $\hat{\mathbf{P}}(n)$, and the covariance matrix of the process noise, $\mathbf{Q}$ with only one non-zero entry so that (11) reduces to

$$\hat{\mathbf{P}}(n|n-1) = \begin{bmatrix} \hat{p}(n|n-1) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{p}(n-1) + q & 0 \\ 0 & 0 \end{bmatrix} \tag{19}$$

where q is the variance of $w(n)$, which is a hyperparameter of the proposed algorithm.

Since the measurement equation of (17) is scalar, the covariance of the measurement noise boils down to the variance of $e(n)$, which we denote as r, another hyperparameter. Using the time-varying measurement matrix, $\mathbf{H}(n)$ from (17) and $\hat{\mathbf{P}}(n|n-1)$ from (19), the Kalman gain follows from (12) as

$$\mathbf{K}(n) = \frac{s(n-1)}{s^2(n-1) + \frac{r}{\hat{p}(n|n-1)}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{20}$$

From (14) we then obtain the update equation for the state vector. Since the second element in the state vector is always 1 and the Kalman gain is zero for this entry, we will in fact just have a scalar update equation as follows:

$$\hat{a}(n) = \hat{a}(n-1) + \frac{s(n-1)}{s^2(n-1) + \frac{r}{\hat{p}(n|n-1)}} e(n) \tag{21}$$

where using (13), $e(n)$ is given by

$$e(n) = s(n) - s(n-1)\hat{a}(n-1) + s(n-2) \tag{22}$$

which is identical to (6) but with $a(n-1) = \hat{a}(n-1)$.

Finally from (15), the update of the first and only non-zero element of the covariance matrix of the prediction error is

$$\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1) + \frac{r}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1) \tag{23}$$

It can now be seen that (21) is indeed in the form of a normalized LMS (NLMS) filter update [14, 15], with a time-varying step size of $1/\left[s^2(n-1) + r/\hat{p}(n|n-1)\right]$, where the term $s^2(n-1)$ provides the normalization and $r/\hat{p}(n|n-1)$ acts as a time-varying regularization parameter. As opposed to having to choose this regularization parameter in a heuristic manner [25], an optimal value is now defined in the Kalman filter context as the ratio of the variance of the measurement noise to the variance of the prediction error obtained from measurements samples up to time $n-1$ [14]. We also note that a similar, yet time-invariant expression for the NLMS parameter was obtained in a Bayseian framework in [26, 27].

A summary of the KalmANF frequency tracker is given in Algorithm 2. As with the LMS-ANF, due to the $s(n-2)$ term, we simply start the for loop from $n = 2$ and initialize $s(0), \hat{a}(1), s(1)$, and $\hat{p}(1)$. The constraints are also imposed on $\hat{a}(n)$ to ensure $\arccos(\hat{a}(n)/2)$ exists. We can also deduce that the KalmANF has a computational complexity of 14 multiplications per recursion (in line 7, only two multiplications are counted since $k(n)$ would have been computed in line 4), which is the same order of magnitude as the LMS-ANF.

Algorithm 2 *Kalman-based/NLMS update of the ANF (KalmANF)*

```
    Initialize $s(0), s(1), \hat{a}(1), \hat{p}(1) = 0$
    Set $r, q, \rho$
 1: for $n = 2$ to $N-1$ do
 2:     $\hat{p}(n|n-1) = \hat{p}(n-1) + q$
 3:     $s(n) = y(n) + \rho\hat{a}(n-1)s(n-1) - \rho^2 s(n-2)$
 4:     $k(n) = \frac{s(n-1)}{s^2(n-1)+\frac{r}{\hat{p}(n|n-1)}}$
 5:     $e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)$
 6:     $\hat{a}(n) = \hat{a}(n-1) + k(n)e(n)$
 7:     $\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1)+\frac{r}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1)$
 8:     if $|\hat{a}(n)| > 2$ then
 9:         $\hat{a}(n) = 0$
10:     end if
11:     $\hat{f}_o(n) = (f_s/2\pi)\arccos(\hat{a}(n)/2)$
12: end for
```

4. EVALUATION

We evaluate the performance of the KalmANF in relation to the LMS-ANF using both simulated and realistic data. We firstly use simulated data so that we can compare frequency estimates to a ground truth, and consequently make observations on the general performance of the KalmANF in relation to the LMS-ANF. We then apply the algorithms to realistic acoustic data and compare how well a dominant frequency component is tracked. For both the LMS-ANF and the KalmANF, it was always the case that $-2 \leq \hat{a}(n) \leq 2$ so that the constraint of $\hat{a}(n) = 0$ when $|\hat{a}(n)| > 2$ was never executed. We do not consider an evaluation of the KalmANF against other types of pitch/frequency trackers that are not based upon the ANF as this is beyond the scope of this work. The code used to generate all of the results that follow is available at [28].

4.1. Simulated Data

4.1.1. Instantaneous change in frequency

In this first simulation, we observe the performance of the KalmANF and LMS-ANF for the situation where there is an instantaneous change in frequency of the sinusoidal component of the input signal, so as to initially gauge how well the algorithms are suited for rapidly changing sinusoidal components. We synthesized an input signal of duration 4 s consisting of a sinusoid embedded in white Gaussian noise at a sampling frequency of 8 kHz and with a signal to noise ratio of 2 dB. For the first 2 s, the frequency of the sinusoid was 1500 Hz, after which the frequency was instantaneously changed to 500 Hz for the remainder of the signal. The amplitude of the sinusoid was 0.5, and its initial phase was set to zero.

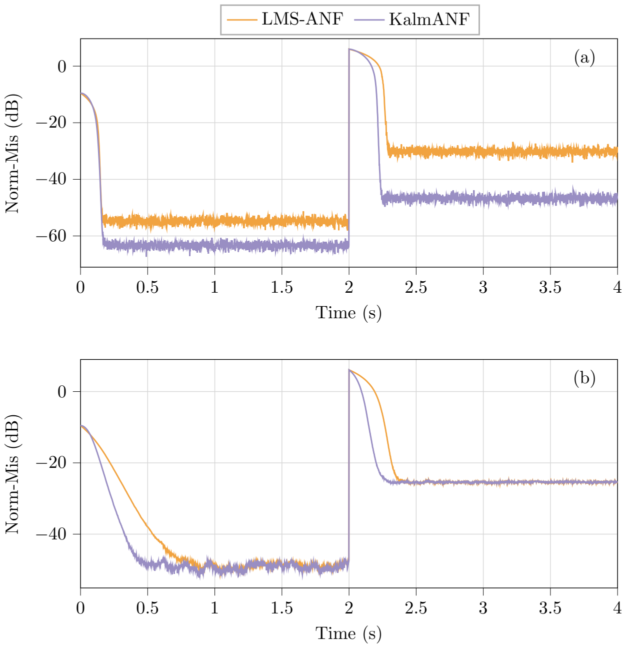

Figure 3: *Averaged Norm-Mis from the LMS-ANF and KalmANF across 100 different signal realizations consisting of a sinusoid whose frequency instantaneously changes in frequency at 2* s *and white Gaussian noise at an SNR of* 2 dB. *(a) Using an ANF with* $\rho = 0.95$*, (b) Using an ANF with* $\rho = 0.7$*. In both cases,* $\mu = 1 \cdot 10^{-3}$ *for the LMS-ANF and* $q = 8 \cdot 10^{-5}$*, and* $r = 10$ *for the KalmANF.*

Therefore, at any point in time, this signal corresponded to the signal model of (1), where $A_o(n) = 0.5$, $\phi_o(n) = 0$, and $f_o(n)$ varied according to the aforementioned frequency of the sinusoid.

We applied both the LMS-ANF and KalmANF algorithms to this input signal to estimate the frequency, $\hat{f}(n)$, of the sinusoid over time. In order to quantify the performance of both algorithms, we computed the normalized misalignment (error) between the estimated frequency and the true frequency as follows:

$$\text{Norm-Mis}(n) = 20\log_{10}\frac{|f_o(n) - \hat{f}_o(n)|}{f_o(n)} \tag{24}$$

We repeated this procedure for 100 different realizations of white Gaussian noise and averaged the normalized misalignment across the different realizations. Fig. 3 shows this averaged Norm-Mis for two values of ρ. In Fig. 3 (a), $\rho = 0.95$ for a narrow bandwidth notch filter, and in Fig. 3 (b), $\rho = 0.7$ for a wider bandwidth notch filter. In both simulations, $\mu = 1 \cdot 10^{-3}$ for the LMS-ANF and $q = 8 \cdot 10^{-5}$, and $r = 10$ for the KalmANF. These parameters were chosen such that the initial convergence rates for $\rho = 0.95$ of both algorithms were approximately similar as shown in Fig. 3 (a).

Despite the similar initial convergences rates, however, in Fig. 3 (a) we can firstly observe that the KalmANF converges to a lower steady state Norm-Mis than the LMS-ANF both during the first 2 s and after the instantaneous change in the frequency of the input signal. We can also observe that the KalmANF converges faster than the LMS-ANF after this instantaneous change in frequency, without any change to the aforementioned algorithm parameters.

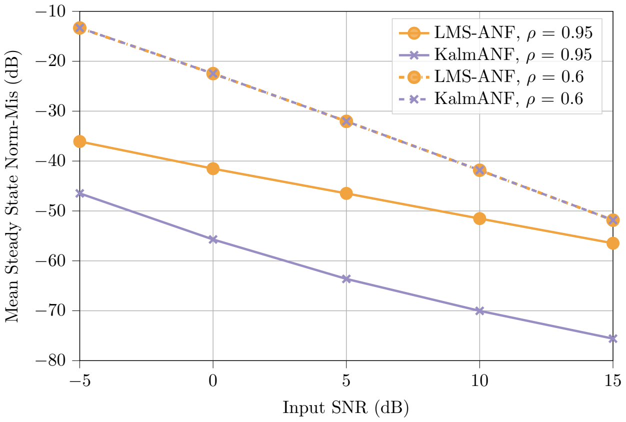

Figure 4: *Mean of the steady state Norm-Mis for the LMS-ANF and KalmANF (after averaging over 100 different realizations) as a function of the input SNR. The signal was 4* s *and consisted of a sinusoid embedded in white Gaussian noise.*

In Fig. 3 (b) where $\rho = 0.7$, the KalmANF now converges to a similar steady state Norm-Mis as the LMS-ANF, but however at a faster rate both during the first 2s and after the instantaneous change in the frequency.

It should be noted that since we have initialized $\hat{a} = 0$, the initial frequency estimate is half of the Nyquist frequency ($f_s/4$). Hence the tracking of low frequencies at higher sampling frequencies can result in longer convergence times, which would be particularly problematic for the LMS-ANF due to its fixed step size. If there is some prior knowledge of the dominant frequency content of the signal however, the sampling frequency could be chosen so that the initial frequency estimate is close to this dominant frequency.

4.1.2. Influence of ρ and input SNR

In order to observe the influence of ρ and the input SNR on the performance of the algorithms, in this simulation, we used a 4 s input signal consisting of a single sinusoidal component with $A_o(n) = 0.5$, $\phi_o(n) = 0$, and $f_o(n) = 868$Hz embedded in white Gaussian noise at $f_s = 8$kHz. We ran the KalmANF and the LMS-ANF for $\rho = 0.6$ and $\rho = 0.95$ for a range input SNRs = $\{-5, 0, 5, 10, 15\}$ dB. For each input SNR and ρ, we ran the algorithms using 100 different realizations of white Gaussian noise and averaged the Norm-Mis across the different realizations. We then computed the mean of the steady state Norm-Mis using the last 2s of the averaged Norm-Mis across the different realizations. In other words, we computed the mean of a converged region of the Norm-Mis such as that between 3 s and 4 s in Fig. 3 (a). For the LMS-ANF, $\mu = 1 \cdot 10^{-3}$, and for the KalmANF, $r = 10$, but q was varied such that for the different values of ρ and all input SNRs, the convergence rates of both the LMS-ANF and KalmANF were approximately the same. For $\rho = 0.95$, $q = \{10, 4.5, 2.5, 1.9, 1.7\} \cdot 10^{-5}$, and for $\rho = 0.6$, $q = \{4, 2.5, 2, 2, 2\} \cdot 10^{-5}$, where each value of q corresponds to the particular input SNR in the range $\{-5, 0, 5, 10, 15\}$ dB. As can be seen from (21), the ratio $r/\hat{p}(n|n-1)$ directly affects the time-varying step size, and hence can be tuned to obtain a desired convergence speed by varying r and/or q. In general, when there is more uncertainty in the system model, such as in low SNR conditions, q should be set to a larger value (as was done in this simulation), which would consequently yield larger step sizes, allowing the algorithm to accommodate for larger deviations from the system model.

Fig. 4 shows the mean of the steady state Norm-Mis plotted against the input SNR for the different algorithms and different values of ρ. We can observe that for the larger value $\rho = 0.95$, the KalmANF outperforms the LMS-ANF by achieving a lower steady state Norm-Mis, and hence a more accurate frequency estimate, whereas for the smaller value of $\rho = 0.6$, both algorithms achieve the same the steady state Norm-Mis. In general however, for both algorithms, the results suggest that larger values of ρ are a preferable choice as they result in a lower steady state Norm-Mis. As expected, we can also observe that the performance of all algorithms degrades as the SNR decreases, and is particularly poor for the smaller values of ρ at very low input SNRs.

4.2. Real data

In this section, we evaluate the KalmANF using two realistic acoustic signals, both of which conform to the model in (1), but where the signal represented by $g(n)$ is different for each case.

4.2.1. Musician Wren

The first signal we consider is that of a musician wren (Cyphorhinus arada), a bird in the family of brown passerine birds, known for its melodious birdsong that spans a considerable frequency range. Fig. 5 (a) shows the spectrogram of an excerpt of musician wren recorded in Uiramutã, Brazil taken from www.xeno-canto.org[5]. As can be observed, the birdsong is fairly sinusoidal and spans a range of about 2 kHz, with substantial and almost instantaneous jumps in frequency. The remaining component of the signal, $g(n)$, in this case is the noise of the outdoor environment, which is not fully white or Gaussian, and in fact consists of impulsive-type sounds presumably due to rainfall.

In Fig. 5 (b), the estimated frequency tracks at each sample using the LMS-ANF and the KalmANF, both with $\rho = 0.95$ are overlaid on the same spectrogram of 5 (a), so as to visualize how well the frequency of the birdsong is being tracked. For the LMS-ANF, $\mu = 0.3$ and for the KalmANF, $q = 8 \cdot 10^{-3}$ and $r = 1$. The LMS-ANF could not be tuned to match the convergence rate of the KalmANF since larger values of μ resulted in an unstable filter. This demonstrates that firstly, the KalmANF is able to converge faster than the LMS-ANF. Secondly, it can also be observed that the KalmANF is able to quickly adapt to the rapid changes in frequency of the birdsong. In most cases, the KalmANF appears to yield a more accurate frequency estimate as compared to the LMS-ANF. In particular, around 2.3 s, it can be seen that the impulsive background noise negatively impacts the frequency estimation of the LMS-ANF, whereas the KalmANF estimate remains fairly stable by comparison. It is also noted that there were no sinusoidal components to be tracked in the first 1.2s and between approximately 3.5s and 4.7s, and hence frequency estimates during these times are meaningless. It nevertheless does provide some insight into the convergence of the filters when there is no dominant sinusoidal component.

[5] This recording has a catalogue number XC 513058 and was recorded by Gabriel Leite.

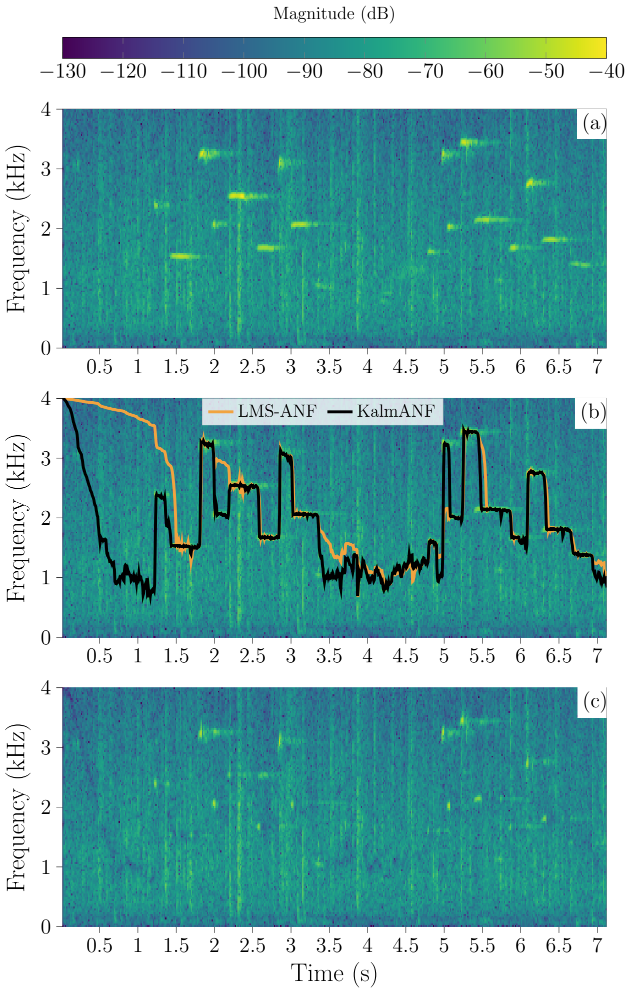

Figure 5: *(a) Spectrogram from an excerpt of a musician wren (Cyphorhinus arada). (b) Overlaid frequency tracks, i.e. sample-by-sample frequency estimates for the LMS-ANF and KalmANF with $\rho = 0.95, \mu = 0.3, q = 8 \cdot 10^{-3}, r = 1$. (c) Output signal from the KalmANF.*

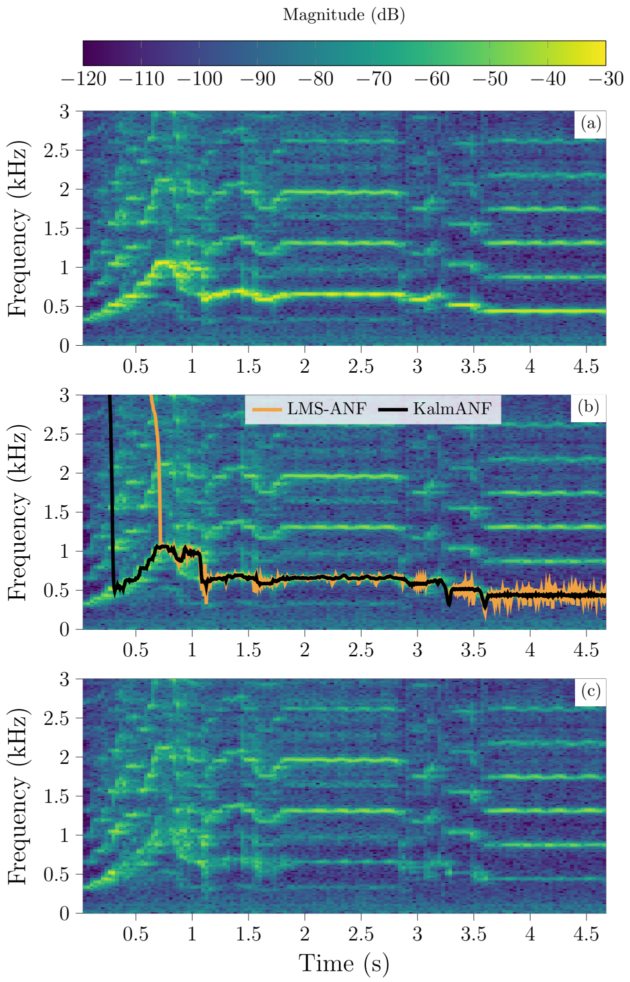

Figure 6: *(a) Spectrogram from a short flute passage. (b) Overlaid frequency tracks for the LMS-ANF and KalmANF with $rho = 0.93, \mu = 5 \cdot 10^{-3}, q = 5 \cdot 10^{-4}, r = 10$. (c) Output signal from the KalmANF.*

Finally, to give a better impression of the performance of the KalmANF, Fig. 5 (c) shows the spectrogram of the error signal (output) of the KalmANF defined in (22). Upon comparison with Fig. 5 (a), it can be seen that the sinusoidal component of the birdsong has been significantly attenuated, implying that the frequency of the birdsong has been accurately tracked.

4.2.2. Flute

Here we consider a short passage from a flute with both rapid and slow frequency changes taken from freesound[6]. Fig. 6 (a) shows the spectrogram of this signal (which was converted to a mono signal and resampled to 16 kHz). There is a strong sinusoidal component along with several harmonics. Hence the signal still conforms to the model of (1) with a dominant sinusoidal component, but the remaining part of the signal, $g(n)$, would now consist of the harmonics and any background noise.

In Fig. 6 (b) the estimated frequency tracks at each sample using the LMS-ANF and the KalmANF, both with $\rho = 0.93$ are overlaid on the spectrogram of 6 (a). For the LMS-ANF, $\mu = 5 \cdot 10^{-3}$ and for the KalmANF, $q = 5 \cdot 10^{-4}$ and $r = 10$. Just as in the birdsong example, the LMS-ANF could not be tuned to match the convergence rate of the KalmANF since larger values of μ resulted in an unstable filter. Hence, the KalmANF has a faster convergence rate than the LMS-ANF, and is able to quickly adapt to the frequency changes with a fairly accurate frequency track. Once the LMS-ANF has converged, it generally follows a similar frequency track to that of the KalmANF, but with a much larger variance around the fundamental frequency, particularly toward the end of the signal. Similar to Fig. 5 (c), Fig. 6 (c) shows the spectrogram of the error signal output of the KalmANF, where we can observe a significant attenuation of the dominant sinusoidal component.

[6] "Flute trill" by juskiddink (https://freesound.org/people/juskiddink/) licensed under CCBY 4.0.

5. CONCLUSION

We have developed a fast frequency tracker that is based on updating a single parameter of an adaptive notch filter (ANF) with a Kalman filter (KalmANF). Whereas this parameter is conventionally updated using a least-mean-square (LMS) algorithm, in this work, we reformulate the ANF (which is a constrained biquadratic filter) in terms of a state-space model, where the state contains the parameter to be updated. By using a Kalman filter to update the state, we have also demonstrated that such an update is equivalent to a normalized LMS (NLMS) filter update where the regularization parameter can be expressed as the ratio of the variance of the measurement noise to the variance of the prediction error. Using both simulated and realistic data, it was shown that in comparison to the ANF-based frequency tracker using an LMS algorithm, the KalmANF resulted in a more accurate performance, with a faster convergence rate, while maintaining a low computational complexity and the ability to be updated on a sample-by-sample basis.

6. REFERENCES

[1] D. Gerhard, *Pitch Extraction and Fundamental Frequency: History and Current Techniques*, Technical Report TR-CS 2003-06, Department of Computer Science, University of Regina, Regina, Canada, Nov. 2003.

[2] M. G. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Morgan and Claypool Publishers, 2009.

[3] J. M. Kates, "Feedback Cancellation in Hearing Aids: Results from a Computer Simulation," *IEEE Trans. Signal Process.*, vol. 39, no. 3, pp. 553–562, 1991.

[4] J. A. Maxwell and P. M. Zurek, "Reducing Acoustic Feedback in Hearing Aids," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 304–313, 1995.

[5] E. Benetos, S. Dixon, Z Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, 2019.

[6] M. G. Christensen, "A method for low-delay pitch tracking and smoothing," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '12)*, 2012, pp. 345–348.

[7] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds., chapter 14, pp. 495–518. Elsevier Science, Amsterdam, The Netherlands, 1995.

[8] J. Tabrikian, S. Dubnov, and Y. Dickalov, "Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model," *IEEE Trans. Speech Audio Process.*, vol. 12, pp. 76 – 87, 2004.

[9] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, and M. G. Christensen, "Robust Bayesian pitch tracking based on the harmonic model," *IEEE Trans. Audio Speech Lang. Process.*, vol. 27, no. 11, pp. 1737–1751, 2019.

[10] D.V. Bhaskar Rao and S.Y. Kung, "Adaptive Notch Filtering for the Retrieval of Sinusoids in Noise," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 4, pp. 791–802, 1984.

[11] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 983–996, 1985.

[12] J. M. Travassos-Romano and M. Bellanger, "Fast Least Squares Adaptive Notch Filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 158–161, 1988.

[13] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[14] P. A. C. Lopes and J. B. Gerald, "New Normalized LMS Algorithms Based on the Kalman Filter," in *2007 IEEE Int. Symp. Circuits Syst.*, may 2007, vol. 1, pp. 117–120.

[15] S. Haykin, *Adaptive filter theory*, Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.

[16] R. Panchalard, J. Koseeyaporn, and P. Wardkein, "State-Space Kalman Adaptive IIR Notch Filter," in *Proc. 2006 Int. Conf. on Commun., Circuits Syst.*, 2006, vol. 1, pp. 206–210.

[17] N. G. Chernoguz, "A single-parameter adaptive comb filter," in *Proc. 2001 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '01)*, 2001, vol. 6, pp. 3749–3752.

[18] H. Hajimolahoseini, S. Gazor, and R. Amirfattahi, "A robust and fast method for estimating and tracking the instantaneous fundamental frequency of audio signals," in *Proc. 2016 IEEE Canadian Conf. Elect. and Comput. Eng. (CCECE)*, may 2016, vol. 2016-Octob, pp. 1–4.

[19] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, and M. G. Christensen, "A Kalman-based fundamental frequency estimation algorithm," in *Proc. 2017 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '17)*, 2017, pp. 314–318.

[20] S. Bittanti and S. M. Savaresi, "On the parameterization and design of an extended Kalman filter frequency tracker," *IEEE Trans. Automatic Control*, vol. 45, no. 9, pp. 1718–1724, 2000.

[21] B. F. La Scala, R. R. Bitmead, and B. G. Quinn, "An extended Kalman filter frequency tracker for high-noise environments," *IEEE Trans. Signal Process.*, vol. 44, no. 2, pp. 431–434, 1996.

[22] O. Das, J. Smith, and C. Chafe, "Improved real-time monophonic pitch tracking with the extended complex kalman filter," *J. Audio Eng. Soc.*, vol. 68, no. 1/2, pp. 78–86, 2020.

[23] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1997.

[24] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, John Wiley & Sons, 2008.

[25] J. Benesty, C. Paleologu, and S. Ciochină, "On Regularization in Adaptive Filtering," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 6, pp. 1734–1742, 2011.

[26] T. van Waterschoot, G. Rombouts, and M. Moonen, "MSE optimal regularization of APA and NLMS algorithms in room acoustic applications," in *Proc. 2006 Int. Workshop Acoustic Echo Noise Control (IWAENC '06)*, 2006.

[27] T. van Waterschoot, G. Rombouts, and M. Moonen, "Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement," *Signal Processing*, vol. 88, no. 3, pp. 594–611, 2008.

[28] R. Ali, "KalmANF: A frequency tracker based on a kalman filter update of a single parameter adaptive notch filter," *GitHub repository*, 2023, `https://github.com/randyaliased/KalmANF`.