Real-Time Pitch Tracking in Audio Signals with the Extended Complex Kalman Filter

Das, Orchisama; Smith, Julius O. III; Chafe, Chris

Published in:
Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)

Publication date:
2017

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version:
Das, Orchisama; Smith, Julius O. III; Chafe, Chris (Accepted/In press). Real-Time Pitch Tracking in Audio Signals with the Extended Complex Kalman Filter. In Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17) (pp. 118-124).

REAL-TIME PITCH TRACKING IN AUDIO SIGNALS WITH THE EXTENDED COMPLEX KALMAN FILTER

Orchisama Das, Julius O. Smith III, Chris Chafe

Center for Computer Research in Music and Acoustics,
Stanford University
Stanford, USA
[orchi|jos|cc]@ccrma.stanford.edu

ABSTRACT

The Kalman filter is a well-known tool used extensively in robotics, navigation, speech enhancement and finance. In this paper, we propose a novel pitch follower based on the Extended Complex Kalman Filter (ECKF). An advantage of this pitch follower is that it operates on a sample-by-sample basis, unlike other block-based algorithms that are most commonly used in pitch estimation. Thus, it estimates sample-synchronous fundamental frequency (assumed to be the perceived pitch), which makes it ideal for real-time implementation. Simultaneously, the ECKF also tracks the amplitude envelope of the input audio signal. Finally, we test our ECKF pitch detector on a number of cello and double bass recordings played with various ornaments, such as vibrato, portamento and trill, and compare its result with the well-known YIN estimator, to conclude the effectiveness of our algorithm.

1. INTRODUCTION

Pitch detection in music and speech has been an active area of study. In [1], Gerhard gives a history of pitch recognition techniques. He also establishes in the importance of pitch in carrying much of the semantic information in tonal languages, which makes it useful in the context of speech recognition. In music, its obvious application is in automatic transcription. It is also to be noted that pitch is a perceptual feature [2], whereas most pitch-detectors detect fundamental frequency, which corresponds to perceived pitch for periodic signals.

Algorithms for pitch detection can be classified into three broad categories – *time domain methods, frequency domain methods* and *statistical methods*. Time domain methods based on the zero-crossing rate and autocorrelation function are particularly popular. The best example of this is perhaps the YIN [3] estimator, which makes use of a modified autocorrelation function to accurately detect periodicity in signals. Among frequency domain methods, the best known techniques are cepstrum [4], harmonic product spectrum [5] and an optimum comb filter algorithm [6]. Statistical methods include the maximum likelihood pitch estimator [5, 7], more recent neural networks [8] and hidden Markov models [9].

In [10] Cuadra et al. discuss the performance of various pitch detection algorithms in real-time interactive music. They establish the fact that although monophonic pitch detection seems like a well-researched problem with little scope for improvement, that is not true in real-time applications. Some of the most common issues in real time pitch tracking are optimization, latency and accuracy in noisy conditions. The ECKF pitch detector proposed in this paper can be easily implemented on hardware with less computational power, has a maximum latency of 20 ms (a latency of 30–40 ms is tolerable) and has excellent performance in presence of a high amount of noise. It also yields pitch estimates on a fine-grained, sample-by-sample basis, resulting in very accurate pitch tracking. Instruments like the cello and flute which have strong harmonics, pose an additional challenge in pitch detection, which makes testing on cello recordings a reasonable way to check the performance of our algorithm.

The Kalman filter [11] has several applications in power system frequency measurements, and one such brilliant application inspired this work. For example, in [12], the extended Kalman filter was used to track the harmonics of the 60Hz power signal. Several models of the extended Kalman filter exist for tracking the fundamental frequency in power signals, but the one we use here was proposed by Dash et al. [13] The extended complex Kalman filter (ECKF) developed here is ideal for use in real-time, with a high tolerance for noise. The complex multiplications can be carried out on a floating point processor. The ECKF simultaneously tracks fundamental frequency, amplitude and pitch, in presence of harmonics and noise. The assumption is that the strength of the harmonics is less than that of the fundamental. Of course, several modifications need to be made before applying it to audio signals, in which correct pitch detection has to happen within milliseconds and which can have large variations in pitch in a short amount of time.

The rest of this paper is organized as follows – in Section 2 we give details of the model used for ECKF pitch tracking. In Section 3, details of its implementation are given, including calculation of initial estimates for attaining steady state values quickly, resetting the error covariance matrix based on silent frame detection, and an adaptive process noise variance based on the measurement residual. In Section 4, we give the results of testing our algorithm on audio data. In Section 4.1, we note some limitations of our algorithm and delineate scope for improvement. Finally we conclude the paper in Section 5, and talk about the scope for future work.

2. ECKF MODEL AND EQUATIONS

We make use of the sines+noise model for music [14] (that matches the model used in [13]) to derive our state space equations. The non-linear nature of the model calls for an extended Kalman filter [15], which linearizes the function about the current estimate by using its Taylor series expansion. Only the first order term is kept in the Taylor series expansion, and higher order terms are ignored. In vector calculus, the first derivative of a function is found by computing its *Jacobian*.

Let there be an observation y_k at time instant k, which is a sum

of additive sines and a residual noise.

$$y_k = \sum_{i=1}^{N} a_i \cos\left(\omega_i t_k + \phi_i\right) + v_k \tag{1}$$

where a_i, ω_i and ϕ_i are the amplitude, frequency and phase of the ith sinusoid and v_k is a normally distributed Gaussian noise $v \sim N(0, \sigma_v^2)$. Of course in music signals, the residual is never precisely a Gaussian white noise but we make that assumption for this model. σ_v^2 is known as the measurement noise variance. We also assume that the fundamental is considerably stronger than the partials, and 1 reduces to

$$y_k = a_1 \cos\left(\omega_1 k T_s + \phi_1\right) + v_k \tag{2}$$

where a_1, w_1 and ϕ_1 are the fundamental amplitude, frequency and phase respectively and T_s is the sampling interval. The state vector is constructed as

$$x_k = \begin{bmatrix} \alpha \\ u_k \\ u_k^* \end{bmatrix} \tag{3}$$

where

$$\begin{aligned} \alpha &= \exp(j\omega_1 T_s) \\ u_k &= a_1 \exp(j\omega_1 k T_s + j\phi_1) \\ u_k^* &= a_1 \exp(-j\omega_1 k T_s - j\phi_1) \end{aligned} \tag{4}$$

This particular selection of state vector ensures that we can track all three parameters that defines the fundamental – frequency, amplitude and phase. The relative advantage of choosing this complex state vector has been described in [13]. The state vector estimate update rule x_{k+1} relates to x_k as

$$\begin{aligned} \begin{bmatrix} \alpha \\ u_{k+1} \\ u_{k+1}^* \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \frac{1}{\alpha} \end{bmatrix} \begin{bmatrix} \alpha \\ u_k \\ u_k^* \end{bmatrix} \\ x_{k+1} &= f(x_k) \\ f(x_k) &= \begin{bmatrix} \alpha & \alpha u_k & \frac{u_k^*}{\alpha} \end{bmatrix}^T \end{aligned} \tag{5}$$

y_k relates to x_k as

$$\begin{aligned} y_k &= H x_k + v_k \\ H &= \begin{bmatrix} 0 & 0.5 & 0.5 \end{bmatrix} \end{aligned} \tag{6}$$

where H is the observation matrix. We can see that

$$\begin{aligned} H x_k &= \frac{a_1}{2}\left[\exp\left(j\omega_1 k T_s + j\phi_1\right) + \exp\left(-j\omega_1 k T_s - j\phi_1\right)\right] \\ &= a_1 \cos\left(\omega_1 k T_s + \phi_1\right) \end{aligned} \tag{7}$$

2.1. Kalman Filter equations

The recursive Kalman filter equations aim to minimize the trace of the error covariance matrix. The EKF equations are given as follows

$$K_k = \hat{P}_{k|k-1} H^{*T} \left[H \hat{P}_{k|k-1} H^{*T} + 1\right]^{-1} \tag{8}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - H \hat{x}_{k|k-1}) \tag{9}$$

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) \tag{10}$$

$$\hat{P}_{k|k} = (I - K_k H) \hat{P}_{k|k-1} \tag{11}$$

$$\hat{P}_{k|k+1} = F_k \hat{P}_{k|k} F_k^{*T} + \sigma_w^2 I \tag{12}$$

where F_k is the Jacobian given by

$$F_k = \left.\frac{\partial f(x_k)}{\partial x_k}\right|_{x_k = \hat{x}_{k|k}} = \begin{bmatrix} 1 & 0 & 0 \\ \hat{x}_{k|k}(2) & \hat{x}_{k|k}(1) & 0 \\ -\frac{\hat{x}_{k|k}(3)}{\hat{x}_{k|k}^2(1)} & 0 & \frac{1}{\hat{x}_{k|k}(1)} \end{bmatrix} \tag{13}$$

- $\hat{x}_{k|k-1}, \hat{x}_{k|k}, \hat{x}_{k|k+1}$ are the *a priori*, current and *a posteriori* state vector estimates respectively.
- $\hat{P}_{k|k-1}, \hat{P}_{k|k}, \hat{P}_{k|k+1}$ are the *a priori*, current and *a posteriori* error covariance matrices respectively.
- K_k is the Kalman gain that acts as a weighting factor between the observation y_k and *a priori* prediction $\hat{x}_{k|k-1}$ in determining the current estimate.
- σ_w^2 is modeled as the process noise variance and I is an identity matrix of dimensions 3×3.
- Initial state vector and error covariance matrix are denoted as $\hat{x}_{1|0}$ and $\hat{P}_{1|0}$ respectively

The fundamental frequency, amplitude and phase estimates at instant *k* are given as

$$\begin{aligned} f_{1,k} &= \frac{\ln\left(\hat{x}_{k|k}(1)\right)}{2\pi j T_s} \\ a_{1,k} &= \sqrt{\hat{x}_{k|k}(2) \times \hat{x}_{k|k}(3)} \\ \phi_{1,k} &= \frac{1}{2j} \ln\left(\frac{\hat{x}_{k|k}(2)}{\hat{x}_{k|k}(3) \hat{x}_{k|k}(1)^{2k}}\right) \end{aligned} \tag{14}$$

The state vector multiplied with the observation matrix essentially gives the low passed observation signal with the cutoff frequency of the LPF approximately at the signal's fundamental frequency.

3. IMPLEMENTATION DETAILS

For best performance, our proposed ECKF pitch estimator needs to be modified in many ways. This includes keeping track of silent regions in the signal, resetting the error covariance matrix whenever there is a transition from silence to transient, giving it correct initial estimates for a low settling time and calculating an adaptive process noise variance.

3.1. Detection of Silent Zones

It is important to keep track of note-off events (unpitched moments) in the signal because the estimated frequency for such *silent* regions should be zero. Moreover, whenever there is a transition from note-off to note-on, the Kalman filter error covariance matrix needs to be reset. This is because the filter quickly converges to a steady state value, and as a result the Kalman gain K_k and error covariance matrix $\hat{P}_{k|k}$ settle to very low values. If there is a sudden change in frequency of the signal (which happens at note onset), the filter will not be able to track it unless the covariance matrix is reset.

To keep track of *silent* regions, we divide the signal into frames of 20 ms. One way to determine if a frame is silent or not is to calculate its energy. The energy of the ith frame, E_i is given as the

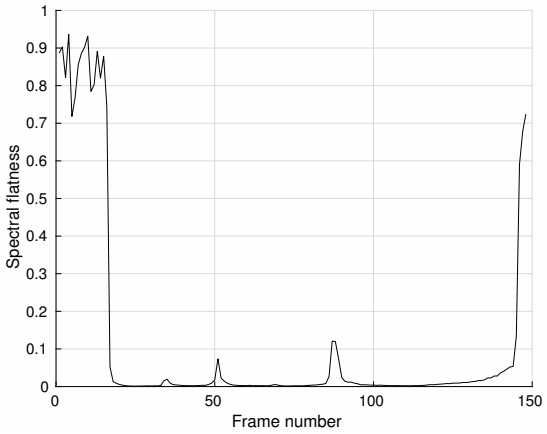


Figure 1: *Spectral flatness varying across frames. A high value indicates silent frame.*

sum of the square of all the signal samples in that frame. If the energy is below -60dB, then the frame is classified to be silent. If the frame has *N* samples, then

$$E_i = 20\log_{10}\sum_{n=0}^{N-1} y(n)^2. \tag{15}$$

However, for noisy input signals, the energy in silent frames is significant. To find silent frames in noisy signals, we make use of the fact that noise has a fairly flat power spectrum. Therefore, for each frame, we calculate the spectral flatness [16]. If spectral flatness ≈ 1, then the frame is classified to be silent. Figure 1 shows spectral flatness v/s frame number for an audio file containing a single note preceded and followed by some silence.

The power spectral density (PSD) of the observed signal, Φ_{yy}, is given as

$$\Phi_{yy}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \phi_{yy}(n)e^{-j\omega n} \tag{16}$$

where $\phi_{yy}(n)$ is the autocorrelation function of the input signal *y*, given as

$$\phi_{yy}(n) = \sum_{m=-\infty}^{\infty} \overline{y(m)}y(n+m). \tag{17}$$

The power spectrum is the DTFT of the autocorrelation function and one way of estimating it is Welch's method [17] which makes use of the periodogram. The power spectral density can be calculated in `Matlab` with the function `pwelch`. The spectral flatness is defined as the ratio of the geometric mean to the arithmetic mean of the PSD.

$$\text{spf} = \frac{\sqrt[K]{\prod_{k=0}^{K-1}\hat{\Phi}_{yy}(e^{j\omega_k})}}{\frac{1}{K}\sum_{k=0}^{K-1}\hat{\Phi}_{yy}(e^{j\omega_k})} \tag{18}$$

where $\hat{\Phi}_{yy}(e^{j\omega_k})$ is the estimated power spectrum for *K* frequency bins covering the range $[-\pi, \pi]$.

For white noise $v \sim N(0, \sigma_v^2)$ corrupting the measurement signal, *y*, the silent frames of *y* will have pure white noise with the following properties

$$\begin{aligned} \phi_{yy}(n) &= \sigma_v^2\delta(n) \\ \Phi_{yy}(e^{j\omega}) &= \sigma_v^2 \,\forall\, \omega \;\in\; [-\pi, \pi] \\ \text{spf} &= \frac{\sqrt[K]{\sigma_v^{2K}}}{\frac{1}{K}(K\sigma_v^2)} = 1 \end{aligned} \tag{19}$$

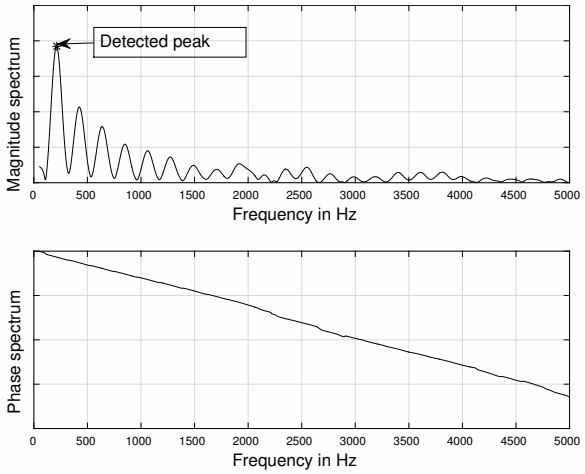


Figure 2: *Frequency spectrum used for calculating initial estimates of the state vector.*

3.2. Calculating Initial State and Resetting the Error Covariance Matrix

It has already been established that resetting the error covariance matrix is necessary whenever there is a change in the frequency content of the signal (i.e., whenever a new note is played). Along with resetting the covariance matrix, we need to recalculate our initial estimates for the state vector. This is because the rate of convergence of the Kalman filter depends on the accuracy of its initial estimates. Basically, we need to calculate $\hat{x}_{1|0}$ and $\hat{P}_{1|0}$ whenever there is a note onset, i.e., whenever there is a transition from silent frame to non-silent frame.

Depending on how strong the attack of the instrument is, we may need to skip a few audio frames until we reach steady state to accurately estimate initial states. This is because the transient is noisy which makes frequency estimates go haywire, and we must wait for the signal to settle. The number of frames to skip after detecting a transition from silent to non-silent frame can be a user-defined parameter.

To calculate the initial estimates of the state vector $\hat{x}_{1|0}$ we take an FFT of the first non-silent frame following a silent frame, after multiplying it with a *Blackman* window and zero-padding it by a factor of 4. Zero padding increases the FFT size which increases the sampling density along the frequency axis. We calculate the magnitude and frequency of the peaks in the magnitude spectrum, and take $f_{1,0}$ as the minimum of the frequencies corresponding to the largest peaks. $a_{1,0}$ is the magnitude corresponding to $f_{1,0}$ normalized by the mean of the window and number of points in the FFT. $\phi_{1,0}$ is the corresponding phase. Next, we perform parabolic interpolation on $f_{1,0}, a_{1,0}, \phi_{1,0}$ to get more accurate estimates. A typical plot of the spectrum used to calculate initial

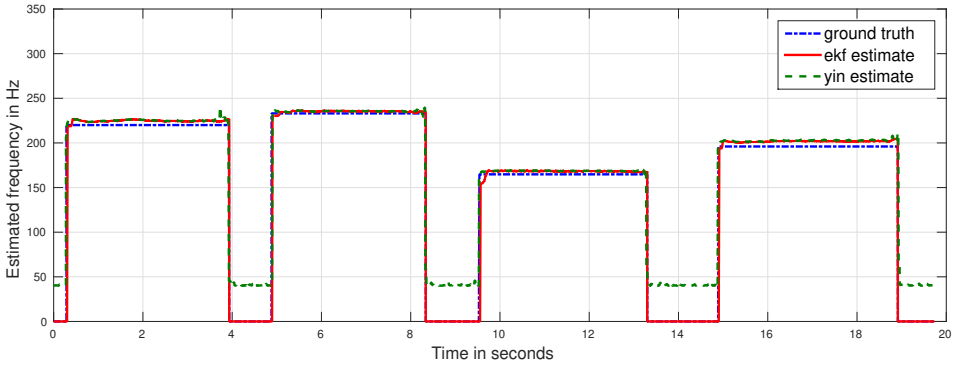


Figure 3: *Plot of a) Ground Truth (blue) b) ECKF pitch detector (red) c) YIN estimator (green)*

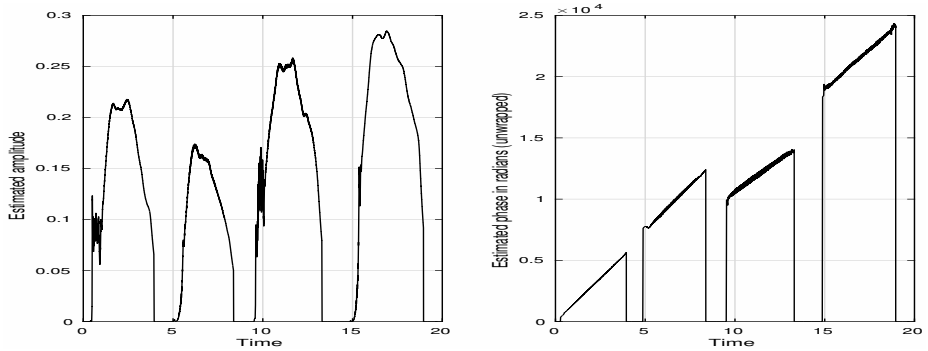


Figure 4: *ECKF: Estimated amplitude and unwrapped phase for same input*

estimates is given in Figure 2. The initial states are as follows

$$\begin{aligned}\hat{x}_{1|0} &= \mathbb{E}[x_1] \\ &= \begin{bmatrix} e^{(2\pi j f_{1,0} T_s)} \\ a_{1,0} e^{(2\pi j f_{1,0} T_s + j\phi_{1,0})} \\ a_{1,0} e^{(-2\pi j f_{1,0} T_s - j\phi_{1,0})} \end{bmatrix} \\ \hat{P}_{1|0} &= \mathbb{E}[(x_1 - \hat{x}_{1|0})(x_1 - \hat{x}_{1|0})^{*T}] \\ &= \mathbf{0}_{(3,3)}\end{aligned} \tag{20}$$

where $\hat{P}_{1|0}$ is a null matrix of order 3×3.

3.3. Adaptive Process Noise

We only reset the covariance matrix when there is a transition from silent to non-silent frame. However, new notes may be played without any rest in between, and dynamics such as vibrato also need to be captured. To track these changes, an additional term is added to equation 12. σ_w^2 is modeled as the process noise variance given as

$$\log_{10}(\sigma_w^2) = -c + |y_k - H\hat{x}_{k|k}| \tag{21}$$

where $c \in \mathbb{Z}^+$ is a constant. [1] The term $y_k - H\hat{x}_{k|k}$ is known as the *innovation*. It gives the error between our predicted value and the actual data. Whenever the innovation is high, there is a significant discrepancy between the predicted output and the input, which is probably caused by a change in the input that the ECKF needs to track. In that case, σ_w^2 increases and there's a term added to the *a posteriori* error covariance matrix $\hat{P}_{k|k+1}$. This increase in the error covariance matrix causes the Kalman gain K_k to increase in the next iteration according to equation 8. As a result, the next state estimate $\hat{x}_{k|k+1}$ depends more on the input, and less on the current predicted state $\hat{x}_{k|k}$. Thus, the innovation reduces in the next iteration, and so does σ_w^2. In this way, the process noise acts as an error correction term that is adaptive to the variance in input.

4. RESULTS

The ECKF pitch detector was tested on cello notes downloaded from the **MIS** database, which contains ground truth labels in the form of annotated note names, and on cello and double bass notes played with various ornaments that we recorded ourselves.[2] The data was summed with white noise normally distributed with 0 mean and 0.01 variance, to test the performance of our algorithm with noisy input.

Figure 3 shows the output when the notes *A3-Bb3-E3-G3* were played on the cello one after another with pauses in between. The pitch detected by the ECKF is compared with the ground truth and YIN estimator output. Figure 4 shows the corresponding estimated amplitude and phase plots for the same input given by the ECKF estimator. The mean and standard deviation of absolute er-

[1] for this paper, c lies in the range 7–9 but its value can be tuned according to the input

[2] The `Matlab` implementation can be cloned from `https://github.com/orchidas/Pitch-Tracking`

ror is given in Table 1. A zoomed in plot can be seen in Figure 5 which explains the cause of higher standard deviation of error with the ECKF. Unlike the YIN estimator which yields a single pitch estimate for an entire block of data, the ECKF yields a unique pitch value for every sample of data, and it fluctuates about a mean value. The frequency of these oscillations is approximately equal to the fundamental frequency of the note being played. The oscillations maybe caused due to the resonance of the bridge of the instrument or some artifact introduced by the tracker. However, the amplitude of the oscillations is small, so in reality it is perceptually insignificant, hence we neglect it. It would be interesting to explore the cause of these fluctuations in a future work.

	Mean	Std. Dev
YIN	5.195	9.637
ECKF	5.181	14.778

Table 1: Error statistics for Figure 3

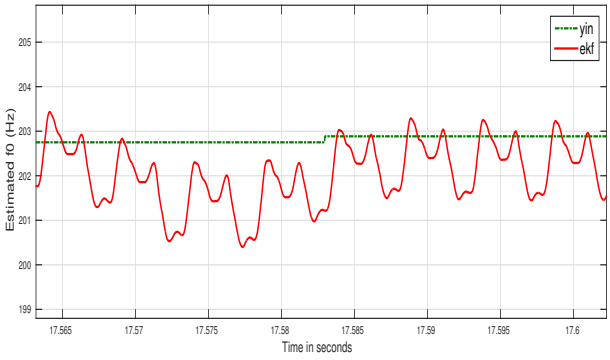
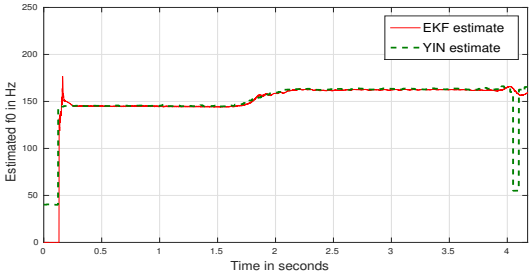


Figure 5: *Higher variance of ECKF is caused by rapid fluctuations of estimated pitch about a mean value.*

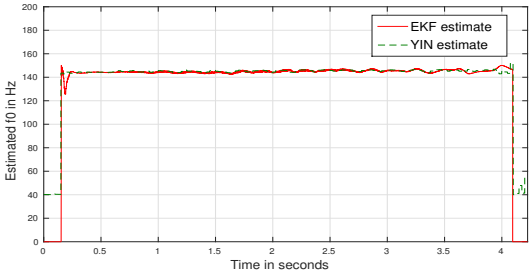
Figure 6a shows the pitch detector output when the input is a portamento played on the note *D3* and glided to *E3*. Figure 6b shows the output when the input is a vibrato on the note *D3*, and Figure 6c shows the output when the input is a vibrato trill on the notes *D3-Eb3*. Figures 6a and 6b were notes played on the cello whereas Figure 6c was a note played on the double bass. All three plots show excellent agreement with what we expect and the output of the YIN estimator. In fact, in Figure 6c the output of the ECKF is much smoother than that of the YIN estimator and shows less drastic fluctuations. Moreover, the YIN estimator gets the pitch wrong in a number of frames where it dips and peaks suddenly. It can be concluded that the ECKF pitch follower is ideal for detecting ornaments and stylistic elements in monophonic music. It could also be used successfully in tracking minute changes in speech formants.

4.1. Limitations

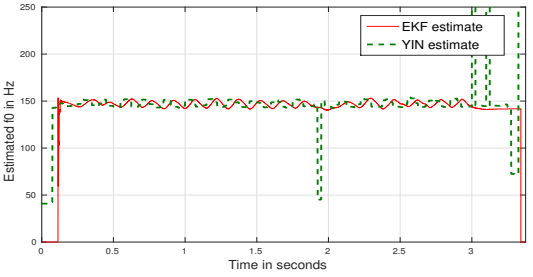
Although our proposed pitch detector performs well in many cases, it has certain drawbacks. Firstly, the additional processing that includes detecting silent frames and estimating initial state makes its implementation more computationally expensive than the one in [13], which is a drawback in real-time processing. This is because estimating the power spectrum with Welch's method requires computing an FFT for each block of data, which is of complexity $\mathcal{O}(N \log_2 N)$. However, depending on the noise environment, a



(a) D3 to E3 portamento



(b) D3 vibrato



(c) D3-Eb3 vibrato trill

Figure 6: *ECKF estimated pitch for various ornaments played on the cello and double bass*

cheaper algorithm can be used to distinguish between non-silent and silent frames. Calculating initial estimates also requires computing an FFT but we only need to do that whenever there is a transition from silent to non-silent frame, not for every block of data.

Secondly, if the initial states estimated from the FFT are off by 20 Hz or more, then the filter is slow to converge to its steady state value. In Figure 6a, it is observed that the ECKF gets the pitch wrong during the transient, but that is expected since there is no well-defined pitch during the transient. To avoid getting a spurious value of pitch, the method of skipping a few buffers to wait until the steady state can be used as described in Section 3.2. Perhaps the ECKF pitch tracker's biggest limitation is tracking notes played quickly together without any pause, as demonstrated in Figure 7. ECKF is slow in tracking very rapid and large changes. The faster the notes are played, the higher the latency in convergence. A solution to this could be to observe note onsets and estimate initial state and reset the covariance matrix whenever there is an onset

detected. However, we leave this problem open for future work.

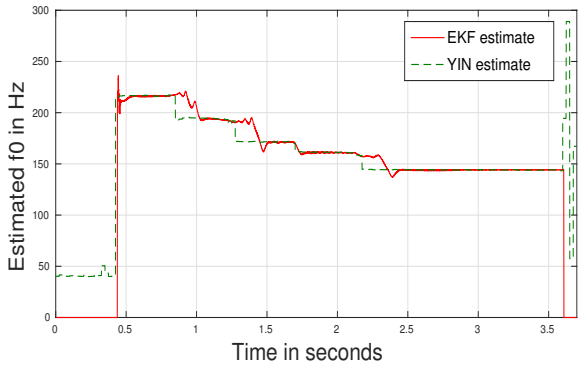


Figure 7: *ECKF is slow in tracking fast note changes. Notes played are descending fifths from the A string on the double bass.*

5. CONCLUSION

In this paper, we have proposed a novel real-time pitch detector based on the Extended Complex Kalman Filter (ECKF). Several adjustments have been made for optimum tracking. An algorithm based on spectral flatness has been proposed to detect silence in incoming noisy audio signal. The importance of accurate initial state estimates and resetting the error covariance matrix has been explained. A correction factor, σ_w^2, has been included to track fast, small changes in input signal.

After all these changes have been incorporated into the ECKF pitch detector, the results match those of robust and successful pitch detectors like the YIN estimator. Perhaps its greatest advantage is the fact that it estimates pitch on a sample-by-sample basis, against most methods which are block-based. Moreover, its performance is robust to the presence of noise in the measurement signal. The ECKF has been observed to be well suited for tracking fine changes in playing dynamics and ornamentation, which makes it an excellent candidate for real-time transcription of solo instrument music. Additionally, the ECKF also yields the amplitude envelope and phase of the signal, along with its fundamental frequency.

5.1. Future Work

We hope this work will encourage new uses of the Kalman filter in audio and music. In recent years, the Kalman filter has been used in music for online beat tracking [18] and partial tracking [19, 20]. It has also been used for frequency tracking in speech [21]. Since the Kalman filter is such a powerful tool that can work for any valid model with the right state-space equations, we believe it can have many more potential real-time applications in music. Some of the other topics we wish to explore include partial tracking and real-time onset detection with the Kalman filter. A real-time pitch detector along with an onset detector lays the groundwork for real-time transcription, which remains an exciting and advanced problem.

6. REFERENCES

[1] David Gerhard, "Pitch extraction and fundamental frequency : History and current techniques," Tech. Rep., University of Regina, 2003.

[2] Joseph Carl Robnett Licklider, "A duplex theory of pitch perception," *The Journal of the Acoustical Society of America*, vol. 23, no. 1, pp. 147–147, 1951.

[3] Alain De Cheveigné and Hideki Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[4] A. Michael Noll, "Cepstrum pitch detection," *The Journal of the Acoustical Society of America*, vol. 41, pp. 293–309, 1967.

[5] A. Michael Noll, "Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate," in *Proceedings of the symposium on computer processing communications*, 1969, vol. 779.

[6] James A Moorer, "On the transcription of musical sound by computer," *Computer Music Journal*, pp. 32–38, 1977.

[7] J Wise, J Caprio, and T Parks, "Maximum likelihood pitch estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 418–423, 1976.

[8] Etienne Barnard, Ronald A Cole, Mathew P Vea, and Fileno A Alleva, "Pitch detection with a neural-net classifier," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307, 1991.

[9] F Bach and M Jordan, "Discriminative training of hidden markov models for multiple pitch tracking," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2005, vol. 5.

[10] Patricio De La Cuadra, Aaron S Master, and Craig Sapp, "Efficient pitch detection techniques for interactive music.," in *ICMC*, 2001.

[11] Rudolph E. Kalman et al., "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[12] Adly A Girgis, W Bin Chang, and Elham B Makram, "A digital recursive measurement scheme for online tracking of power system harmonics," *IEEE transactions on Power Delivery*, vol. 6, no. 3, pp. 1153–1160, 1991.

[13] Pradipta Kishore Dash, G Panda, AK Pradhan, Aurobinda Routray, and B Duttagupta, "An extended complex kalman filter for frequency measurement of distorted signals," in *Power Engineering Society Winter Meeting, 2000. IEEE.* IEEE, 2000, vol. 3, pp. 1569–1574.

[14] Xavier Serra and Julius Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[15] Gabriel A Terejanu, "Extended kalman filter tutorial," Tech. Rep., University of Buffalo, 2008.

[16] A Gray and J Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 3, pp. 207–217, 1974.

[17] Peter Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[18] Yu Shiu, Namgook Cho, Pei-Chen Chang, and C-C Jay Kuo, "Robust on-line beat tracking with kalman filtering and probabilistic data association (kf-pda)," *IEEE transactions on consumer electronics*, vol. 54, no. 3, 2008.

[19] Andrew Sterian and Gregory H Wakefield, "Model-based approach to partial tracking for musical transcription," in *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*. International Society for Optics and Photonics, 1998, pp. 171–182.

[20] Hamid Satar-Boroujeni and Bahram Shafai, "Peak extraction and partial tracking of music signals using kalman filtering.," in *ICMC*, 2005.

[21] Özgül Salor, Mübeccel Demirekler, and Umut Orguner, "Kalman filter approach for pitch determination of speech signals," *SPECOM*, 2006.