O. Das, J. O. Smith III 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 January/February.).
DOI: https://doi.org/10.17743/jaes.2019.0053

Improved Real-Time Monophonic Pitch Tracking with the Extended Complex Kalman Filter

ORCHISAMA DAS, *AES Student Member*, **JULIUS O. SMITH III,** *AES Fellow*, **AND CHRIS CHAFE**
(orchi@ccrma.stanford.edu) (jos@ccrma.stanford.edu) (cc@ccrma.stanford.edu)

Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA, USA

This paper proposes a real-time, sample-by-sample pitch tracker for monophonic audio signals using the Extended Kalman Filter in the complex domain (Extended Complex Kalman Filter). It improves upon the algorithm proposed by the same authors in a previous paper [1] by fixing the issue of slow tracking of rapid note changes. It does so by detecting harmonic change in the signal and resetting the filter whenever a significant harmonic change is detected. Along with the fundamental frequency, the ECKF also tracks the amplitude envelope and instantaneous phase of the input audio signal. The pitch tracker is ideal for detecting ornaments in solo instrument music—such as slides and vibratos. The improved algorithm is tested to track pitch of bowed string (double-bass), plucked string (guitar), and vocal singing samples.

0 INTRODUCTION

Pitch is a perceptual feature [2] that roughly relates to the fundamental frequency of the sound wave under consideration. Pitch tracking algorithms aim to track the evolution of fundamental frequency with time. Pitch tracking in speech and music has been an active area of research [3], with applications in speech recognition and automatic music transcription.

Algorithms for monophonic 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. One example of this is the YIN [4] 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 [5], harmonic product spectrum [6], and an optimum comb filter algorithm [7]. Statistical methods include the maximum likelihood pitch estimator [6, 8], more recent neural networks [9, 10], and hidden Markov models [11]. Combinations of these methods include pYIN (probabilistic YIN) [12], which uses multiple pitch candidates from YIN with associated probabilities and feeds it to a Hidden Markov model, and the Extended Kalman filter based pitch trackers proposed in [1, 13] that are a combination of statistical methods and frequency domain methods. The state-of-art in monophonic pitch tracking is CREPE [10], which uses a deep convolutional network. It is to be noted that multi-pitch estimation in a polyphonic context is a more complex problem, and the tools used to tackle it are different [14].

In [15] 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 Extended Complex Kalman filter (ECKF) based pitch tracker developed in [1] overcomes some of these limitations. It has low latency, is robust to the presence of noise, and yields pitch estimates on a fine-grained sample-by-sample basis.

The ECKF was originally proposed by Dash et al. in [16] to track frequency fluctuations in a 60 Hz power signal. The ECKF 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 phase in the presence of harmonics and noise. Of course, several modifications need to be made before applying it to track pitch in audio signals, such as detecting silent frames that have no pitch, initializing the filter for fast convergence, and an adaptive process noise to accurately detect fine changes in pitch. Perhaps, the biggest application of the ECKF pitch tracker is detecting ornaments—such as glissando, vibrato, trill, etc. Such techniques are essential in adding expression to a musical performance. Playing technique detection [17] is an important task in MIR in which pitch detection is usually the first step.

However, the biggest disadvantage of the method proposed in [1] was that it could not track fast note changes. This is due to the inherent latency of convergence in Kalman tracking [18]. In this paper we propose an improvement by devising a method to detect harmonic change and reinitializing the filter every time a significant change in pitch is detected. This improves performance significantly and gets rid of the lag in estimated pitch. We also propose a more accurate fundamental frequency estimation method to initialize the filter.

The rest of this paper is organized as follows: in Sec. 1 we give details of the model used for ECKF pitch tracking. In Sec. 2 details of its implementation are given, including methods for silent frame and harmonic change detection, calculation of initial estimates for attaining steady state values quickly, resetting the error covariance matrix, backtracking to avoid transient errors, and an adaptive process noise variance calculation based on the measurement residual. Sec. 3 has the results of running the ECKF pitch tracker on double bass, guitar, and vocal singing samples. It is also compared to YIN and CREPE pitch trackers. We discuss some parameter selection details and conclude the paper in Sec. 5 and delineate the scope for future work.

1 MODEL AND EQUATIONS

1.1 Model

We make use of the sines+noise model for music [19] to derive our state space equations. Since the model is nonlinear, we use the extended Kalman filter [20], which linearizes the function about the current estimate by using its Taylor series expansion up to the first order. Higher order terms are ignored.

Let there be an observation y_k at time instant k, which is a sum of additive sines and a measurement noise.

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

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 $v \sim N(0, \sigma_v^2)$, which is the *measurement noise*—usually background noise picked by the microphone. σ_v^2 is the measurement noise variance—typically given by the SNR. If we ignore the partials and only take into account the fundamental, Eq. (1) reduces to

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

where a_1, ω_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 define the fundamental—frequency, amplitude, and phase. The relative advantage of choosing this complex state vector has been described in [16]. The state vector estimate update rule x_{k+1} relates to x_k as

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

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 and w_k is the *process noise*. In our model, the process noise can be assumed to be filtered white noise that represents the residual that remains after removing the harmonic content of the signal. We can see that

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

1.2 EKF Equations

The recursive Kalman filter equations aim to minimize the trace of the error covariance matrix. Each iteration reduces the variance of $\hat{x}_{k|k-1}$ until it converges. At each time-step k, the EKF equations are as follows

$$K_k = \hat{P}_{k|k-1} H^{*T} [H \hat{P}_{k|k-1} H^{*T} + \sigma_v^2]^{-1} \tag{8}$$

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

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

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

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

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_v^2 is the measurement noise variance, which is fixed to be 1, as in [16]. Ideally, this should depend on the SNR of the measurement. A lower SNR would give a higher measurement noise variance.

- σ_w^2 is modeled as the process noise variance and $I \in \mathbb{C}^{3\times 3}$ is an identity matrix.
- Initial state vector and error covariance matrix are denoted as $\hat{x}_{1|0}$ and $\hat{P}_{1|0}$ respectively.

From Eq. (4) the fundamental frequency, amplitude, and phase estimates at instant k can be calculated as

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

2 IMPLEMENTATION DETAILS

2.1 Detection of Silent Frames

It is important to keep track of pitch-off events in the signal because the estimated frequency for such *silent* regions should be zero. Moreover, whenever there is a transition from pitch-off to pitch-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 non-overlapping frames. In real-time processing, this is easily achieved by working with audio buffers. One way to determine if a frame is silent or not is to calculate its energy. The energy of a frame is given as the sum of the square of all the signal samples in that frame. If the energy is below –50 dB, then the frame is classified to be silent.

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

$$\Phi_{yy}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \phi_{yy}(n)e^{-j\omega n} \tag{15}$$

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

$$\phi_{yy}(n) = \sum_{m=-\infty}^{\infty} \overline{y(m)}y(n+m). \tag{16}$$

The power spectrum is the DTFT of the autocorrelation function and one way of estimating it is Welch's method [21] which makes use of the periodogram. Since we are dealing with very short frames here of the order of a few milliseconds, an alternative is to estimate the autocorrelation function and take its FFT after zero-padding by a factor ≥ 5. The spectral flatness is defined as the ratio of the geometric mean to the arithmetic mean of the PSD.

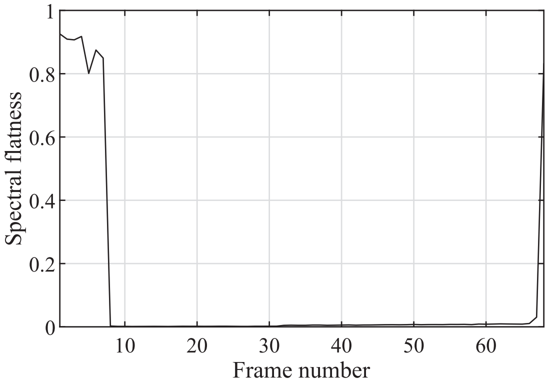


Fig. 1. Spectral flatness varying across frames. A high value indicates a silent frame.

$$\text{spf} = \frac{\sqrt[K]{\prod_{k=0}^{K-1} \hat{\Phi}_{yy}(e^{j\omega_k})}}{\frac{1}{K}\sum_{k=0}^{K-1} \hat{\Phi}_{yy}(e^{j\omega_k})} \tag{17}$$

$\hat{\Phi}_{yy}(e^{j\omega_k})$ 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 the following properties

$$\begin{aligned} \phi_{yy}(n) &= \sigma_v^2\delta(n) \\ \Phi_{yy}(e^{j\omega}) &= \sigma_v^2 \forall\omega \in [-\pi, \pi] \\ \text{spf} &= \frac{\sqrt[K]{\sigma_v^{2K}}}{\frac{1}{K}(K\sigma_v^2)} = 1 \end{aligned} \tag{18}$$

Therefore, for each frame, we calculate the spectral flatness [22]. If spectral flatness $\geq th$[1], then the frame is classified to be silent. Fig. 1 shows spectral flatness v/s frame number for an audio file containing a single note preceded and followed by some silence.

2.2 Resetting the Kalman Filter

In [1] the error covariance matrix was reset and initial estimates were re-calculated only when there was a transition from pitch-off to pitch-on event (silent frame to non-silent frame). However, the error covariance matrix should be reset and the filter re-initialized whenever there is a significant change in pitch. Otherwise, the filter cannot track fast pitch changes; this was one of the main drawbacks of the method proposed in [1]. To fix this problem we implement a harmonic change detector and reset the filter whenever a significant harmonic change is detected.

2.2.1 Initial F0 Estimate

In [1] we simply looked at the first peak in the magnitude spectrum of a frame to calculate the initial F0 estimate. We use a more sophisticated method here by looking at the first few peaks (*NP*) in the magnitude spectrum. We multiply the magnitude spectrum with a slowly tapering Poisson

[1] *th* is a threshold ≥ 0.5

window, $w(n)$ [23], with $\alpha = 10$ and $M =$ FFT size to weigh the lower partials more than the higher partials (which may be spurious and noisy).

$$w(n) = \exp\left(\frac{-0.5\alpha n}{M-1}\right) \tag{19}$$

Peaks in the magnitude spectrum typically represent harmonics, and the differences between their frequencies should remain constant. We utilize this fact and find the mode of the difference between the adjacent frequencies of the *NP* peaks, and use that to detect fundamental frequency and set that as the initial F0 estimate, $\hat{f}_0$. The amplitude, $\hat{a}_0$, and phase, $\hat{\phi}_0$ are found by quadratically interpolating the magnitude and phase spectrum around the peak at estimated F0. The initial state and error covariance matrix are

$$\hat{x}_{1|0} = \begin{bmatrix} e^{(2\pi j \hat{f}_0 T_s)} \\ \hat{a}_0 e^{(2\pi j \hat{f}_0 T_s + j\hat{\phi}_0)} \\ \hat{a}_0 e^{(-2\pi j \hat{f}_0 T_s - j\hat{\phi}_0)} \end{bmatrix}$$
$$\hat{P}_{1|0} = \mathbb{E}[(x_1 - \hat{x}_{1|0})(x_1 - \hat{x}_{1|0})^{*T}] = \mathbf{0} \tag{20}$$

2.2.2 Detecting Harmonic Change

To detect harmonic change we need to keep track of both the previous frame and the current frame. We find the *NP* largest peaks in the magnitude spectra of the consecutive frames. For frames $k-1$ and k, the *NP* peak frequencies can be represented as $[f_{1,\ k-1}, f_{2,\ k-1}, \cdots f_{NP,k-1}]$ and $[f_{1,\ k}, f_{2,\ k}, \cdots f_{NP,k}]$ respectively. We then find the first order difference between adjacent peaks in frames k and $k-1$. This typically represents the fundamental frequency.

$$f_{i+1,k} - f_{i,k} = d_{i,k} \ \forall \ i = 1, \cdots, NP-1 \tag{21}$$

We calculate the mode of the distribution given by

$$m = \max_i \mid d_{i,k} - d_{i,k-1} \mid \tag{22}$$

The fundamental frequency deviation among consecutive frames is calculated in *cents* as

$$f0_{dev} = 1200\log_2\left(\frac{\hat{f}_0}{\hat{f}_0 + m}\right) \tag{23}$$

where $\hat{f}_0$ is the calculated initial fundamental frequency. If the same note is being played in the current frame and the previous frame, then m would typically be zero. If there is a large enough frequency deviation between consecutive frames, i.e., $f0_{dev} \geq n_{st}$ (specified in number of semitones), then the filter is re-initialized with a null covariance matrix and the estimated initial state.

We show the detected peaks in two consecutive frames when there is a harmonic change, along with a histogram of frequency deviation between them in Fig. 2.

2.3 Backtracking to Avoid Transient Errors

For an instrument with a strong transient attack, such as a guitar or a piano, the F0 estimation is likely to fail during a note change. This is because the transient is broadband noise-like in nature and the frequency peaks do not have harmonic relationships between them. To avoid this, whenever there is a transition from a silent to a non-silent frame or whenever a harmonic change is detected, a few frames are skipped before estimating the initial state. The number of frames to skip depends on the instrument whose pitch is to be tracked. Once a stable F0 estimate is found, the filter is initialized with that estimate, and we backtrack the number of frames we skipped and start tracking the pitch. This introduces latency (which can be up to 130 ms depending on the number of frames skipped) but ensures that an accurate pitch is detected for all frames in the signal.

Table 1. Pitch detection errors with frequency modulated sawtooth wave

SNR	Mean Absolute Error (Hz)	Standard Deviation (Hz)
5	5.5673	19.0500
10	5.6536	19.0715
15	5.7145	19.0937
20	5.9741	19.3449

2.4 Adaptive Process Noise

We only reset the covariance matrix when there is a transition from silent to non-silent frame or when a new note with a different pitch is played. However, dynamics such as vibrato also need to be captured. To track these changes, σ_w^2 in Eq. (12) is modeled as the process noise variance.

$$\log_{10}(\sigma_w^2) = -c + |y_k - H\hat{x}_{k|k}| \tag{24}$$

where $c \in \mathbb{Z}^+$ is a constant. 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 Eq. (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.

3 RESULTS

The MATLAB code for the following examples is available at https://github.com/orchidas/Pitch-Tracking.

3.1 Synthesized Signal

We test our pitch estimator on an artificially synthesized signal—a sawtooth wave at 440 Hz, with a sinusoidal vibrato of frequency 5 Hz, with added white noise at different SNRs. The results showing the mean absolute error and standard deviation are given in Table 1. The tracked pitch lags slightly behind the actual pitch as observed in Fig. 3. Higher variance of detected pitch is caused by rapid fluctuations of estimated pitch about a mean value.

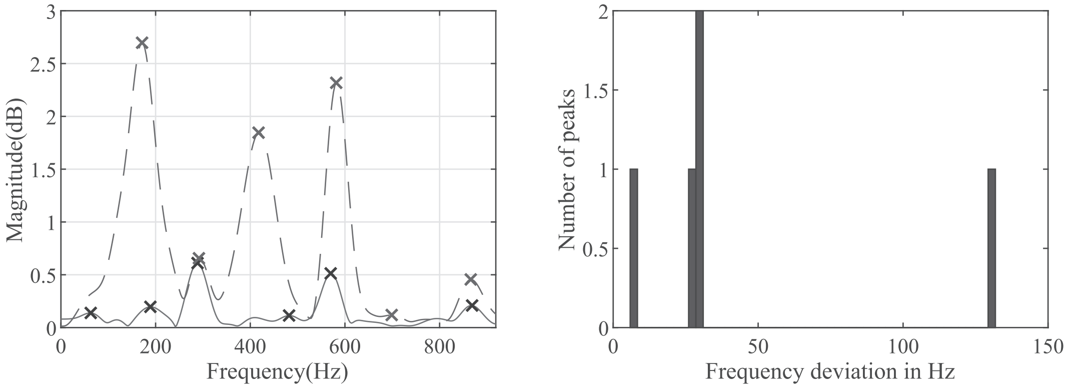


Fig. 2. Different notes played in consecutive frames (dashed line—previous frame, solid line—current frame). On the right is a histogram of the distribution given by Eq. (22).

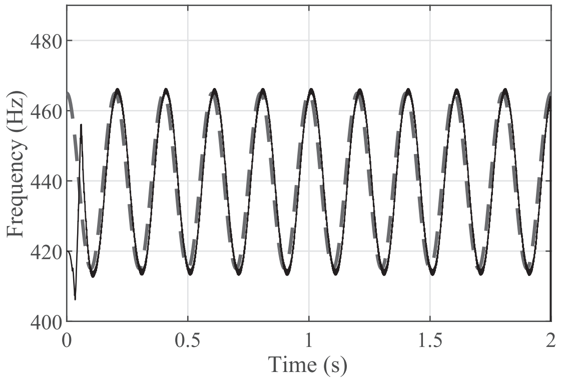


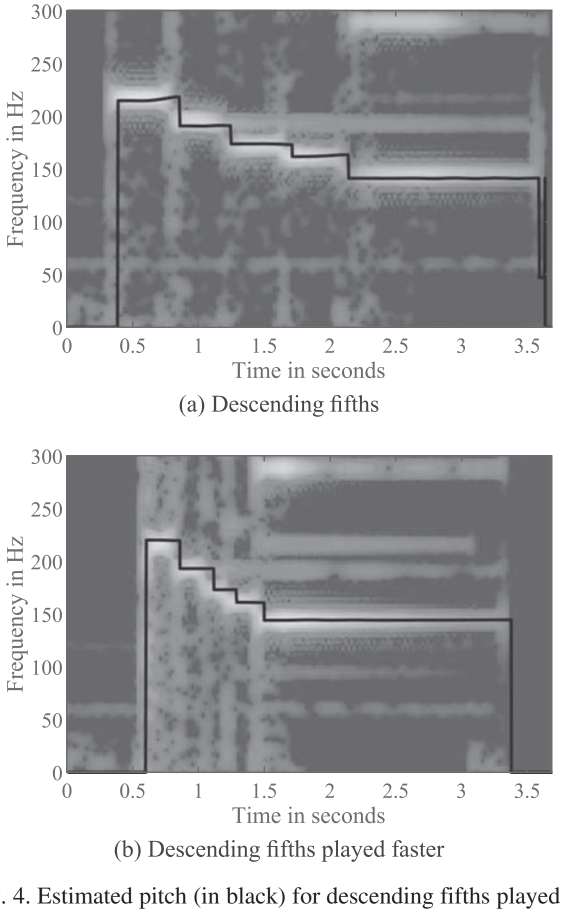
Fig. 3. Tracking pitch in a sawtooth wave with 5 Hz sinusoidal vibrato. Dashed line—actual pitch, solid line—ECKF detected pitch.

3.2 Tracking Dynamic Note Changes

To address the main drawback of [1], we track ECKF pitch trajectories for rapid note changes. The results are shown in Fig. 4, where the spectrogram of the fundamental is plotted, along with the estimated pitch trajectory with a black solid line. The trajectory is smooth and accurate and follows the changing notes without any significant lag.

3.3 Tracking Ornaments

One of the potential applications of a sample-by-sample pitch tracker is to get smooth trajectories of ornaments and embellishments. We track portamento (smooth glide from one note to another), vibrato (frequency modulation), and trill (hammering on adjacent note) on two string instruments— the double bass and electric guitar. Tracking pitch in a bowed string instrument such as the double-bass is easier, since it is a driven oscillator, whereas a plucked string instrument will have a strong transient where pitch detection goes haywire. We recorded the samples into Audacity through a TASCAM 2x2 interface using a Beyer Dynamic omni mic and direct line input at a sampling rate of 48 kHz. The results comparing the ECKF pitch tracker to YIN and CREPE are shown in Fig. 5. There is an overshoot/undershoot in the initial pitch detected by the ECKF in some plots, but it quickly converges to the correct pitch. This is due to errors in initial F0 estimation. The ECKF trajectory is smoother and follows YIN and CREPE trajectories closely in most cases, except in Fig. 5d, where the YIN pitch trajectory becomes unstable, and in Fig. 5f, where it slightly lags behind the YIN and CREPE trajectories.



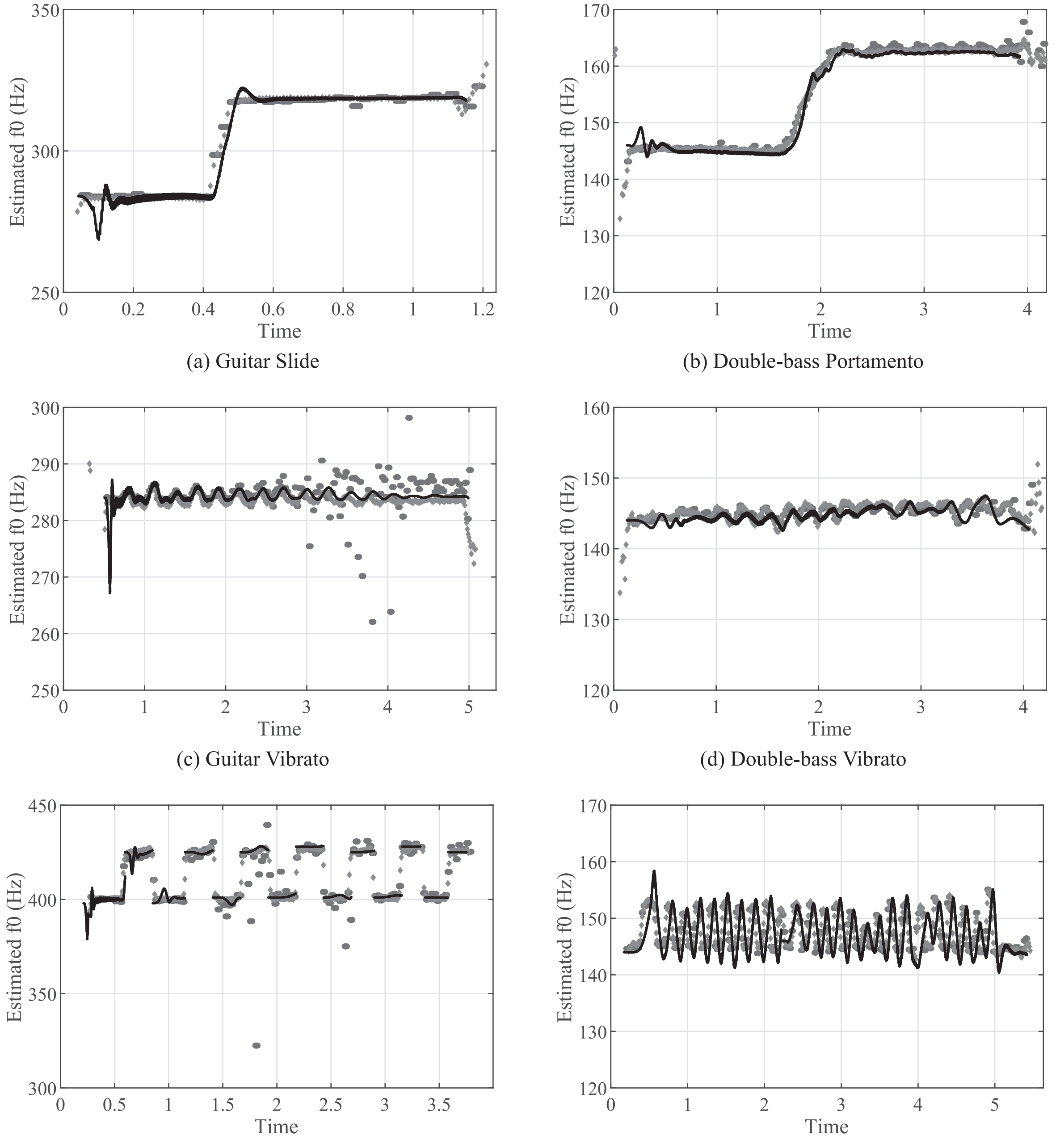
(a) Descending fifths

(b) Descending fifths played faster

Fig. 4. Estimated pitch (in black) for descending fifths played on the A string on the double bass.

3.4 Tracking Singing Voice

Putting it all together, we compare the performance of our proposed pitch tracker against YIN and CREPE in a real-



(a) Guitar Slide

(b) Double-bass Portamento

(c) Guitar Vibrato

(d) Double-bass Vibrato

(e) Guitar Trill

(f) Double-bass Trill

Fig. 5. Estimated pitch for various ornaments played on the guitar. Circles—YIN, Diamonds—CREPE, Black line—ECKF.

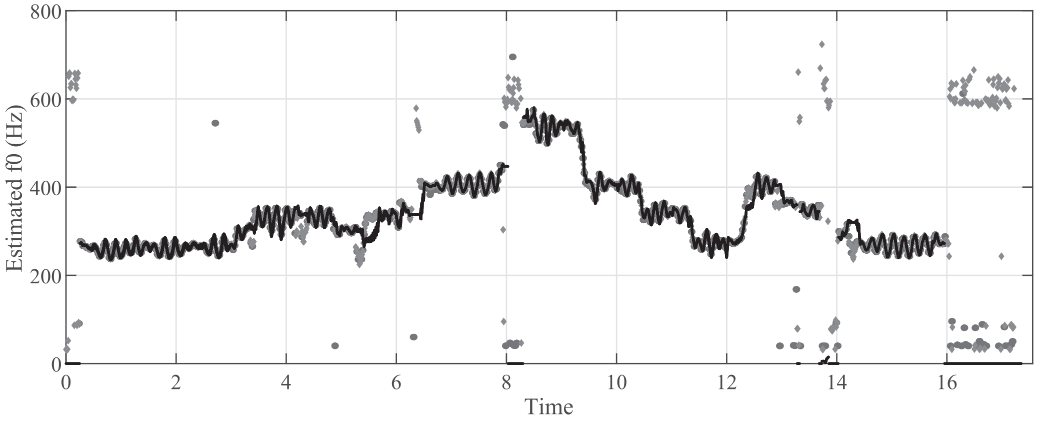


Fig. 6. Pitch track of a female voice excerpt singing with vibrato from VocalSet. Circles—YIN, Diamonds—CREPE, Black line—ECKF.

world example. We use a track from the VocalSet dataset [24]: a female soprano singer singing an excerpt from the song "Row your boat gently down the stream" with vibrato. The results are given in Fig. 6. The pitch track generated by ECKF is comparable to that of CREPE. It is more densely sampled (one f0 estimate per sample), with the additional advantage of recovering the amplitude envelope and the phase track of the fundamental. Essentially, with the ECKF pitch tracker, we can extract the fundamental waveform from the rest of the signal[2]. This is an advantage it has over other pitch trackers proposed in the literature. The run-times in ascending order are YIN, ECKF, and CREPE.

4 DISCUSSION

The proposed improved ECKF pitch tracker gives a sample-by-sample pitch estimate, along with amplitude envelope and phase, which yields the extracted fundamental. The smooth pitch trajectory obtained could be used to make subtle pitch corrections during the mixing process. Although the Kalman filter is inherently robust to the presence of observation noise, it affects the accuracy of the F0 estimate calculated to reset the filter. Moreover, a number of parameters need to be carefully selected, depending on the type of instrument whose pitch is to be tracked.

4.1 Selection of Parameters

The following parameters need to be carefully selected for optimum results with the ECKF pitch tracker.

- **Frame size.** For tracking very fast pitch changes, frame size should be small. However, choosing a shorter frame would lead to errors in initial F0 estimation. Therefore, it is a trade-off. In this paper we chose a frame size of 2048 samples.
- **Number of peaks (*NP*).** The number of peaks used to calculate an initial F0 estimate and detect harmonic change depends on the instrument whose pitch is to be tracked. If the partials are harmonically related, then a large number of peaks would give a more stable estimate. However, if the partials are inharmonic, then selecting a small *NP* works better. In this paper we picked $NP = 5$ or 6 for the double bass signals and $NP = 2$ or 3 for guitar, and $NP = 4$ for vocal signals.
- **Number of frames to skip during transient.** If the instrument to be tracked has a strong attack, such as a piano or a guitar, then more frames need to be skipped before initial pitch is calculated from the steady state. This introduces latency. However, for instruments with a softer attack, such as woodwind instruments, skipping one frame should be adequate in getting a good initial estimate. In this paper the number of frames to be skipped $\in [1, 3]$.
- **Adaptive process noise coefficient (c).** This coefficient determines how smooth the pitch trajectory is. A higher value of this coefficient reduces the process noise variance σ_w^2 and gives a smoother pitch trajectory. In this paper $c \in [7, 11]$.
- **Harmonic change threshold.** The threshold beyond which a harmonic change is detected, n_{st}, is specified in number of semitones ($n_{st} = 2$ is default and indicates a whole tone). Small fluctuations in pitch can be tracked by the ECKF, but for larger fluctuations the filter needs to be reset. In case of discrete pitch trajectories, such as in guitar and piano, a smaller value of n_{st} (quarter or eighth tone) works better, whereas for more continuous trajectories, like vocal or bowed string instruments, n_{st} should be equal to a few semitones.

5 SUMMARY

In this paper we have improved upon the proposed Kalman filter based pitch tracker in [1]. The ECKF pitch tracker gives a sample-by-sample estimate of the fundamental frequency, amplitude, and phase and is robust to the presence of measurement noise. We have discussed in detail when and how to reset the filter so that fast note changes as well as ornaments and embellishments can be tracked. We have compared the performance of the pitch tracker to YIN and CREPE pitch detectors and found the results to be comparable. Additionally, this method has the ability to extract the fundamental from the waveform of the signal as it yields the amplitude envelope and a phase track, along with the pitch track. However, parameter selection for the ECKF pitch tracker requires knowledge of the type of signal whose pitch is to be tracked. That is a potential drawback of this method. In future, it would be interesting to automatically pick the optimum set of parameters given an audio signal by training on instrument-specific datasets.

6 ACKNOWLEDGMENT

Part of this research was conducted at the Sound Analysis-Synthesis team at IRCAM during a residency at the Cité Internationale des Arts, Paris, in early 2019. The authors would like to thank Axel Röebel, who heads the Analysis-Synthesis team at IRCAM.

7 REFERENCES

[1] O. Das, J. O. Smith, and C. Chafe, "Real-Time Pitch Tracking in Audio Signals with the Extended Complex Kalman Filter," *Int. Conf. on Digital Audio Effects (DAFx 17)*, pp. 118–124 (2017).

[2] J. C. R. Licklider, "A Duplex Theory of Pitch Perception," *J. Acoust. Soc. Amer.*, vol. 23, no. 1, pp. 147–147 (1951).

[3] D. Gerhard, "Pitch Extraction and Fundamental Frequency: History and Current Techniques," Tech. Rep., University of Regina (2003).

[4] A. De Cheveigné and H. Kawahara, "YIN, a Fundamental Frequency Estimator for Speech and Music," *J. Acoust. Soc. Amer.*, vol. 111, no. 4, pp. 1917–1930 (2002).

[2] Sound examples are available at https://ccrma.stanford.edu/~orchi/Kalman_pitch/EKF.html

[5] A. M. Noll, "Cepstrum Pitch Detection," *J. Acoust. Soc. Amer.*, vol. 41, pp. 293–309 (1967).

[6] A. M. Noll, "Pitch Determination of Human Speech by the Harmonic Product Spectrum, the Harmonic Sum Spectrum, and a Maximum Likelihood Estimate," *Proceedings of the Symposium on Computer Processing Communications*, vol. 779 (1969).

[7] J. A. Moorer, "On the Transcription of Musical Sound by Computer," *Computer Music J.*, vol. 1, no. 4, pp. 32–38 (1977 Nov.).

[8] 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).

[9] E. Barnard, R. A. Cole, M. P. Vea, and F. A. Alleva, "Pitch Detection with a Neural-Net Classifier," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307 (1991).

[10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A Convolutional Representation for Pitch Estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165 (2018).

[11] F. Bach and M. Jordan, "Discriminative Training of Hidden Markov Models for Multiple Pitch Tracking," presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5 (2005).

[12] M. Mauch and S. Dixon, "pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663 (2014).

[13] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, and M. G. Christensen, "A Kalman-Based Fundamental Frequency Estimation Algorithm," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 314–318 (2017).

[14] T. Tolonen and M. Karjalainen, "A Computationally Efficient Multipitch Analysis Model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716 (2000).

[15] P. De La Cuadra, A. S. Master, and C. Sapp, "Efficient Pitch Detection Techniques for Interactive Music," presented at the *International Computer Music Conference (ICMC)* (2001).

[16] P. K. Dash, G. Panda, A. Pradhan, A. Routray, and B. Duttagupta, "An Extended Complex Kalman Filter for Frequency Measurement of Distorted Signals," *Power Engineering Society Winter Meeting, 2000. IEEE*, vol. 3, pp. 1569–1574 (2000).

[17] P.-C. Li, L. Su, Y.-h. Yang, A. W. Su, et al., "Analysis of Expressive Musical Terms in Violin Using Score-Informed and Expression-Based Audio Features," *ISMIR*, pp. 809–815 (2015).

[18] M. Boutayeb, H. Rafaralahy, and M. Darouach, "Convergence Analysis of the Extended Kalman Filter Used as an Observer for Nonlinear Deterministic Discrete-Time Systems," *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 581–586 (1997).

[19] X. Serra and J. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition," *Computer Music J.*, vol. 14, no. 4, pp. 12–24 (1990).

[20] G. A. Terejanu, "Extended Kalman Filter Tutorial," Tech. Rep., University of Buffalo (2008).

[21] P. 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).

[22] 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).

[23] J. O. Smith III, *Spectral Audio Signal Processing* (W3K Publishing, 2011).

[24] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "VocalSet: A Singing Voice Dataset," *International Symposium on Music Information Retrieval (ISMIR)*, pp. 468–474 (2018).

THE AUTHORS

Orchisama Das

Julius O. Smith

Chris Chafe

Orchisama Das received the B. Eng. degree in Instrumentation and Electronics engineering from Jadavpur University, India, in 2016. She is currently a Ph.D. candidate at the Center for Computer Research in Music and Acoustics at Stanford University. She is also a teaching assistant for various signal processing courses at CCRMA. In 2015, Orchisama worked at the University of Calgary funded by a Mitacs Globalink fellowship. She interned at Tesla Motors in 2018 doing DSP for the Noise, Vibration and Harshness team. In 2019 she was a visiting researcher in the Sound Analysis-Synthesis team at IRCAM.

•

Julius O. Smith received the B.S.E.E. degree from Rice University, Houston, TX, in 1975 (control, circuits, and communication). He received the M.S. and Ph.D. degrees in E.E. from Stanford University, Stanford, CA, in 1978 and 1983, respectively. His Ph.D. research was devoted to improved methods for digital filter design and system identification applied to music and audio systems. From 1975 to 1977 he worked in the Signal Processing Department at ESL, Sunnyvale, CA, on systems for digital communications. From 1982 to 1986 he was with the Adaptive Systems Department at Systems Control Technology, Palo Alto, CA, where he worked in the areas of adaptive filtering and spectral estimation. From 1986 to 1991 he was employed at NeXT Computer, Inc., responsible for sound, music, and signal processing software for the NeXT computer workstation. After NeXT, he became an Associate Professor at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford, teaching courses and pursuing research related to signal processing techniques applied to music and audio systems. Continuing this work, he is presently Professor of Music and (by courtesy) Electrical Engineering at Stanford University. For more information, see http://ccrma.stanford.edu/∼jos/.

•

Chris Chafe is a composer, improvisor, and cellist developing much of his music alongside computer-based research. He is Director of Stanford University's Center for Computer Research in Music and Acoustics (CCRMA). At IRCAM (Paris) and The Banff Centre (Alberta), he pursued methods for digital synthesis, music performance, and real-time internet collaboration. Online collaboration software including jacktrip and research into latency factors continue to evolve. An active performer either on the net or physically present, his music reaches audiences in dozens of countries and sometimes at novel venues. A simultaneous five-country concert was hosted at the United Nations a decade ago. Gallery and museum music installations involve "musifications" resulting from collaborations with artists, scientists, and MD's. Recent work includes the Brain Stethoscope project, PolarTide for the Venice Biennale, Tomato Quintet for the transLife:media Festival at the National Art Museum of China, and Sun Shot played by the horns of large ships in the port of St. Johns, Newfoundland.