Department of Electronic and Telecommunication Engineering
University of Moratuwa, Sri Lanka

Mini Project - II

Kalman Filters and Extended Kalman Filters for ECG Signal Filtering

K. W. T. S. Fonseka 200173J

submitted in partial fulfillment of the requirements for the course module

EN4594 - Autonomous Systems

December 13, 2024

Contents

1 Background

Electrocardiograms (ECGs) are critical tools in modern medicine for diagnosing and monitoring cardiovascular health. By recording the electrical activity of the heart, ECGs provide valuable insights into heart rhythm, conduction abnormalities, and potential pathologies such as arrhythmias and ischemia. However, the reliability of ECG interpretation is heavily influenced by signal quality. ECG signals are often contaminated by various noise sources, such as baseline wander, powerline interference, motion artifacts, and muscle contractions. These noise components necessitate the use of advanced signal processing techniques for effective denoising.

Traditional filtering methods, such as frequency domain filtering, have been widely employed to address noise in ECG signals. Techniques like low-pass and high-pass filters are useful for removing specific frequency components, such as baseline wander or powerline interference. However, these methods often fail to preserve critical signal features, especially when noise overlaps with the signal's frequency band. Wavelet transform-based denoising has gained popularity for its ability to analyze signals at multiple scales, allowing effective suppression of noise while retaining time-frequency characteristics. Weiner filter gives the optimal filter weights given that the noise is stationery and the noise characteristics are known. But these techniques assume stationary signal and noise characteristics hence fail to address non-stationary noises.

The Kalman filter, a state-space approach, offers a compelling alternative for ECG denoising. As an adaptive filtering technique, it models the ECG signal as a dynamic system, estimating the underlying clean signal by minimizing the Bayesian mean square error between predicted and observed data. The Kalman filter's ability to incorporate both temporal and statistical characteristics of the signal makes it well-suited for handling non-stationary noise and variations in the ECG signal. Its recursive nature also ensures efficient real-time processing, making it ideal for continuous monitoring applications. Beyond the standard Kalman filter, the Extended Kalman Filter (EKF) extends these capabilities to handle non-linear systems, which are common in physiological systems. The EKF linearizes the system dynamics around the current estimate, allowing it to better model the complex, non-linear nature of ECG signals and noise.

This report explores the application of the Kalman filter and extended Kalman filter for ECG signal denoising and their limitations.

2 Scaler Kalman Filter

2.1 Input Signal

noise removed normal ECG signal is used to evaluate the performance of filters with added noise. The sampling frequency of the signal is 128Hz.

A 50Hz sinusoidal noise (imitationg the powerline noise) and a additive white Gaussian noise with 10dB signal to noise ratio (SNR) is added to contaminate the signal. Covariance of the total noise is $0.0027mV^2$. Figure 2.1 shows the original noise free signal and the contaminated signal.

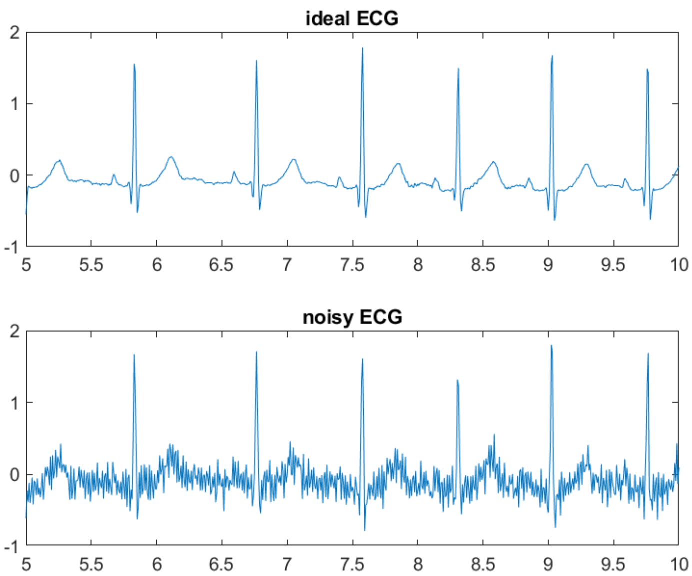

Figure 2.1: Original signal and the noisy signal

2.2 Filter Algorithm

When it comes to single channel bio signals state vector usually reduces to a scaler which is the underlying clean signal. Since we do not have a control input, the prediction stage can be written as,

$$s[n] = as[n-1] + u[n]$$

Where s[n] is the clean signal and u[n] is the process noise with standard error of σ. (R $= \sigma^2$)

The Observation is the current data point of the ECG recording. The variance of the observation noise (v[n]) is Q.

$$x[n] = cs[n] + v[n]$$

The Kalman filter algorithm can be derived as follows.

- **Prediction:**

$$\hat{s}[n|n-1] = a\hat{s}[n-1|n-1]$$

- **Minimum Prediction MSE:**

$$M[n|n-1] = a^2 M[n-1|n-1] + R$$

- **Kalman Gain:**

$$K[n] = \frac{M[n|n-1]}{Q + M[n|n-1]}$$

- **Correction:**

$$\hat{s}[n|n] = \hat{s}[n|n-1] + K[n]\big(x[n] - \hat{s}[n|n-1]\big)$$

- **MMSE:**

$$M[n|n] = \big(1 - K[n]\big)M[n|n-1]$$

2.3 Implementation

The Kalman filter algorithm explain above is implemented in MATLAB using the following script.

```
A = 1; % State transition matrix
H = 1; % Observation matrix
x_est = 0; % Initial state estimate
P = 1; % Initial estimate covariance
R  % Process noise covariance
Q  % Measurement noise covariance

% Apply the Kalman filter
for k = 1:length(nECG)
    % Prediction step
    x_pred = A * x_est;
    P_pred = A * P * A' + Q;

    % Update step
    K = P_pred * H' / (H * P_pred * H' + R); % Kalman gain
    x_est = x_pred + K * (nECG(k) - H * x_pred);
    P = (1 - K * H) * P_pred;

    % Store the filtered estimate
    filteredECG(k) = x_est;
end
```

2.4 Results

The parameters R and Q have to be adjusted to obtain desired output. The following section describes the results obtained for deferent parameter combinations.

When the process covariance is too low filter tries to follow the prediction, in this case the previous state.

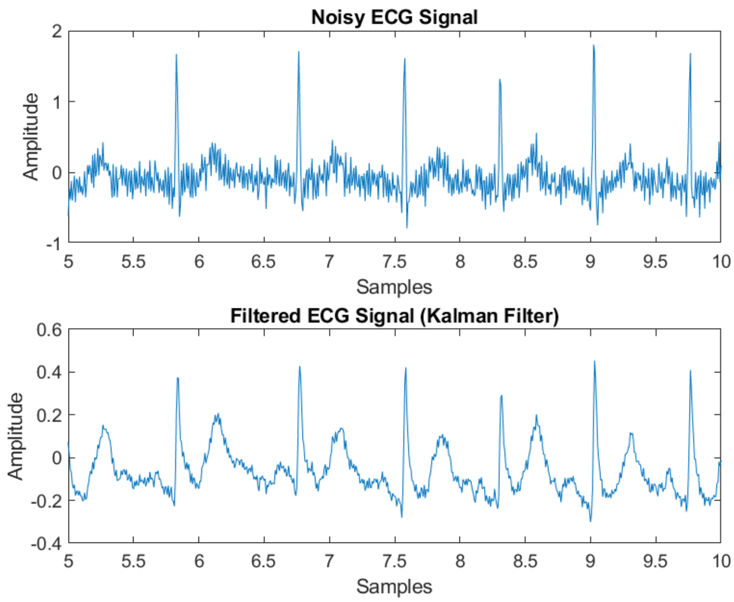

Figure 2.2: Filtered signal using scalar KF.(R=0.001, Q=0.027)

When the process covariance is too high, filter outputs a signal almost similar to the observation.

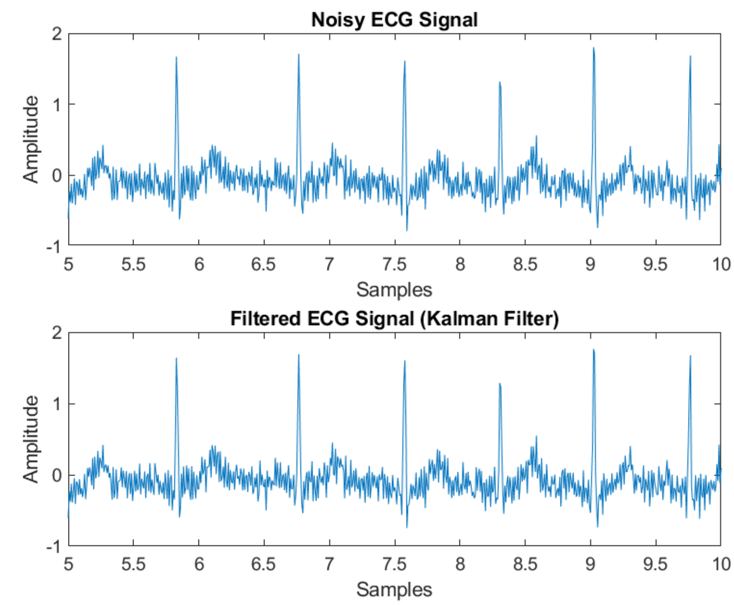

Figure 2.3: Filtered signal using scalar KF.(R=1, Q=0.027)

The filter gives a satisfactory result for the parameter combination R=0.044 & Q=0.027 with a mean square error of 0.0149.

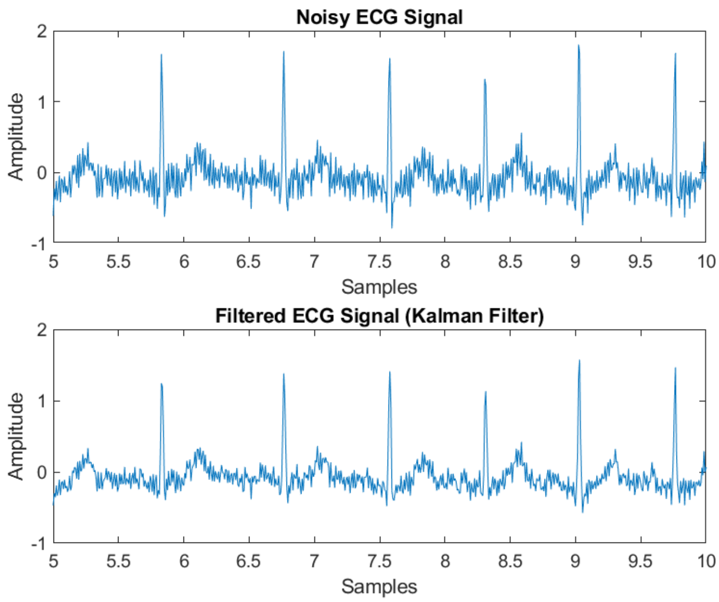

Figure 2.4: Filtered signal using scalar KF.(R=1, Q=1)

The Figure 2.5 shows the power spectrum densities of the ideal signal, noisy signal and the filtered signal.

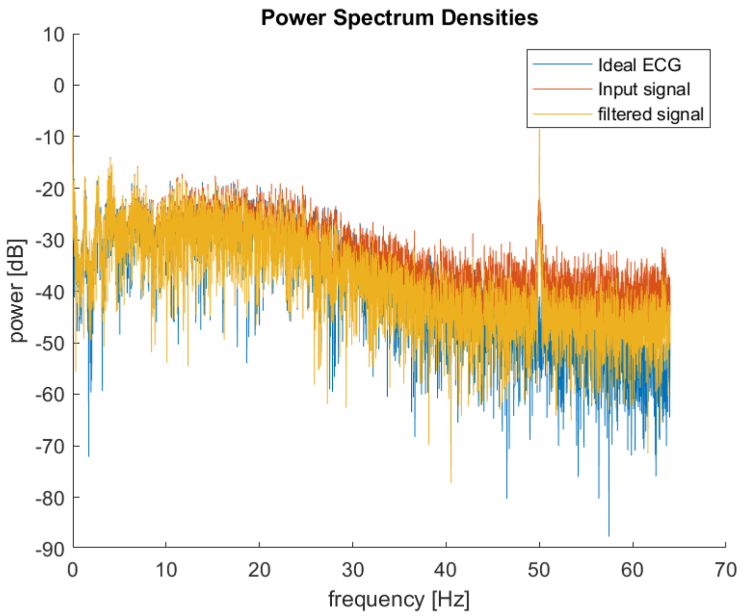

Figure 2.5: PSD of filtered signal using scalar KF.(R=0.044, Q=0.027)

3 Extended Kalman Filter

In the previous section we predicted the current signal state as same as previous state. In other words we assumed the signal is purely random in prediction stage. Basically we did not incoperate any additional information in that stage. But ECG signals have a typical non-linear waveform. To incoperate that for prediction an extended Kalman filter should be used.

3.1 Synthetic ECG model

Sameni *et al.* [1] [2] have used the synthetic ECG generator, which is based on a nonlinear dynamic model proposed by MCSharry *et al.* [3]. They have modeled the ECG signal as a trajectory in Cartesian coordinate frame with 3 dimensional state equations. This can be further simplified into 2 dimensianal vector space in polar coordinate frame. The model consist of severeal parameters (a_i, b_i, θ_i) coresponding to P, Q, R, S, T Waveforms in ECG signal.

$$r' = r(1-r)$$

$$\theta' = \omega$$

$$z' = -\sum_{i\in\{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\frac{\Delta\theta_i^2}{2b_i^2}\right) - (z - z_0) \tag{3.1}$$

Here, $-\pi < \theta < \pi$ and $\Delta\theta_i = (\theta - \theta_i)$. The ω is the angular velocity of the trajectory which is generated by 3.1, consists of a circular limit cycle which is pushed up and down when it approaches one of the P, Q, R, S or T points. In fact, each of the components of the ECG waveform has been modeled with a Gaussian function, which is located at a specific angle. By varying parameters (a_i, b_i, θ_i and ω) we can model varias morphologies of ECG signal including abnormalities.

Figure 3.1 shows the synthetic ECG signal generated according to the typical parameters for a normal person presented in the original paper [3] which are presented in table 3.1.

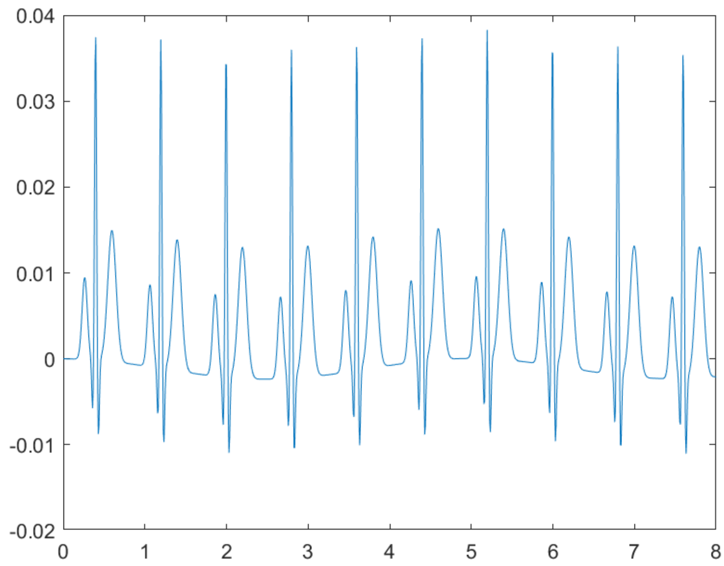

Figure 3.1: Synthetic ECG signal

Table 3.1: Typical parameters of the synthetic ECG model [3]

Index (i)	P	Q	R	S	T
Time (Sec.)	-0.2	-0.05	0	0.05	0.3
$\boldsymbol{\theta_i}$ **(rads.)**	$-\pi/3$	$-\pi/12$	0	$\pi/12$	$\pi/2$
$\boldsymbol{a_i}$	1.2	-5.0	30.0	-7.5	0.75
$\boldsymbol{b_i}$	0.25	0.1	0.1	0.1	0.4

3.2 Implementation

The baseline wandering component $(z - z_0)$ in equation 3.1 is replaced with a random noise. The equation 3.2 shows the resuling state transition functions.

$$\hat{\theta}[n] = \theta[n-1] + \omega\Delta t + N_\theta = F(\theta)$$

$$\hat{z}[n] = z[n-1] - \sum_{i \in \{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\frac{\Delta\theta_i^2}{2b_i^2}\right)\Delta t + N_z = G(z,\theta) \tag{3.2}$$

The components of the prediction Jacobian matrix are as follows.

$$\frac{\partial F}{\partial z} = 0, \tag{3.3}$$

$$\frac{\partial F}{\partial \theta} = \frac{\partial G}{\partial z} = 1, \tag{3.4}$$

$$\frac{\partial G}{\partial \theta} = \sum_{i \in \{P,Q,R,S,T\}} -\Delta \cdot a_i \left[1 - \frac{\Delta\theta_i^2}{b_i^2}\right] \exp\left(-\frac{\Delta\theta_i^2}{2b_i^2}\right) \tag{3.5}$$

The observation stage is linear. Hence doesn't need to calculate the Jacobian matrix.

$$C = [0\ 1]$$

The following MATLAB script shows the implementation of EKF.

```
% EKF

Theta = zeros(size(nECG));
z = zeros(size(nECG));

R = eye(2)*1;
Q = 1;
sigma = eye(2)*1;

C = [0 1];

for n = (2:N)

%%%%%%%%% Prediction %%%%%%%%%%%%%%%%%%%%%%%%%%%
    Theta(n) = mod(Theta(n-1) + omega*D_t,2*pi);
    z_dot = 0;
    for i = (1:5)
        D_theta = Theta(n)-theta_i(i);
        z_dot = z_dot -
           a_i(i)*D_theta*exp(-0.5*D_theta^2/b_i(i)^2);
    end
    z(n) = z(n-1) + D_t*z_dot;
    s = [Theta(n), z(n)]';

%%%%%%%%% Jacobian of state transition %%%%%%%%%

    G = eye(2);

    for i = (1:5)
        D_theta = Theta(n)-theta_i(i);
        G(2,1) = G(2,1) - D_t * a_i(i)*(1 - D_theta^2 /
           b_i(i)^2) * exp(-0.5*D_theta^2 / b_i(i)^2);
    end

%%%%%%%%% Error covariance of prediction %%%%%%%
    sigma = G*sigma*G' + R;

%%%%%%%%% Kalman Gain %%%%%%%%%%%%%%%%%%%%%%%%%%
    K = sigma*C'/(C*sigma*C' + Q);

%%%%%%%%% Correction using input signal %%%%%%%%
    s = s + K*(nECG(n) -C*s);
    Theta(n) = s(1);
    z(n) = s(2);
    sigma = (eye(2) - K*C)*sigma;
end
```

3.3 Results

Here also we can observe a similar effect of Process covariance on the filter output. When the process noise covariance matrix is small output tends to follow the prediction, which is the synthetic ECG signal. This is clearly visible in Figure 3.2 where R is set to be $10^{-6}I$ and Q is $0.027mV^2$. I is the identity matrix.

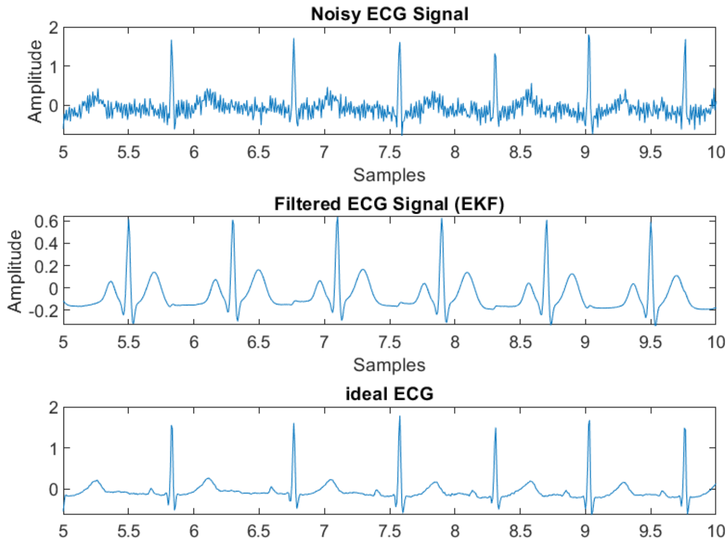

Figure 3.2: Filtered signal using EKF.(R $= 10^{-6}I$)

On the other hand, when the R is much higher the input signal will not be filtered at all. This is clearly visible in Figure 3.3 where R=I.

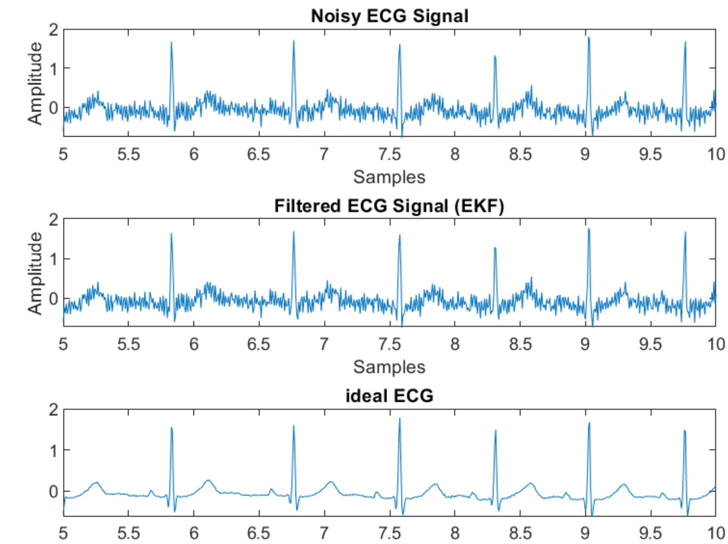

Figure 3.3: Filtered signal using EKF.(R $= I$)

Therefore by selecting a proper covsriance components we can obtain a better filtered signal. Figure 3.4 shows the filtered signal for R = [0.0588 0; 0 0.0425]. This values were obtained by comparing mean square errors. This signal has a mean square error of 0.0146.

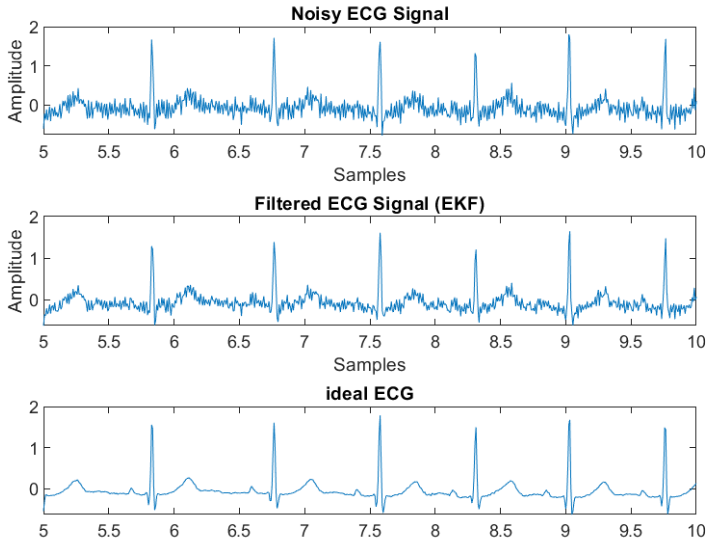

Figure 3.4: Filtered signal using EKF.(R = [0.0588 0; 0 0.0425])

4 Incoperating Phase Observation

When it comes to offline signal processing we can use non-causal signal processing techniques. Here we can further update the above EKF by incoperating phase of the ECG signal calculated from R-R interval.

4.1 Calculating the Phase

There are multiple algorithms to detect the R spikes in an ECG signal. Here I have used the Pan Tompkin algorithm [4]. After detecting R peaks we can map the R-R interval to a phase angle ϕ from 0 to 2π. Then remapped

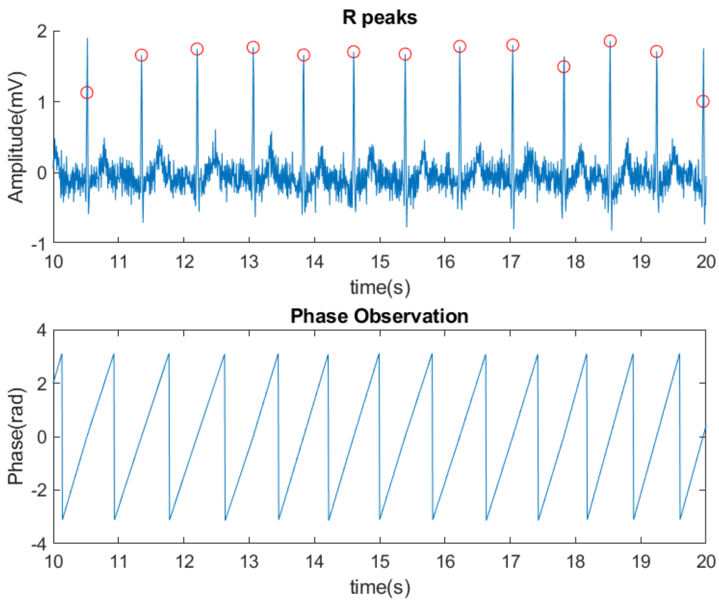

Figure 4.1: R peaks and Phase observation

4.2 Implementation

Since now we have two observations, the observation matrix will be changed to

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

And the observation noise is represented by a 2x2 covariance matrix **Q**.

The MATLAB implementation is as follows.

```
    Theta = zeros(size(nECG));
    z = zeros(size(nECG));

    R = [1, 0; 0, 1 ]*0.01;
    Q = [1, 0; 0, 1 ];
    sigma = eye(2)*1;

```

```
    C = eye(2);
    Theta(1) = -pi;
    for n = (2:N)

    %%%%%%%%% Prediction %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Theta(n) = mod(Theta(n-1) + omega*D_t+pi,2*pi)-pi;
        z_dot = 0;
        for i = (1:5)
            D_theta = Theta(n)-theta_i(i);
            z_dot = z_dot - a_i(i)*D_theta*exp(-0.5*D_theta^2/b_i(i)^2);
        end
        z(n) = z(n-1) + D_t*z_dot;
        s = [Theta(n), z(n)]';

    %%%%%%%%% Jacobian of state transition %%%%%%%%%%%%%%%%%%%%%%%

        G = eye(2);

        for i = (1:5)
            D_theta = Theta(n)-theta_i(i);
            G(2,1) = G(2,1) - D_t * a_i(i)*(1 - D_theta^2 / b_i(i)^2) * exp(-0.5*D_theta^2 / b_i(i)^2);
        end

    %%%%%%%%% Error covariance of prediction %%%%%%%%%%%%%%%%%%%%%
        sigma = G*sigma*G' + R;

    %%%%%%%%% Kalman Gain %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        K = sigma*C'/(C*sigma*C' + Q);

    %%%%%%%%% Correction using input signal %%%%%%%%%%%%%%%%%%%%%%
        s = s + K*(nECG(n) -C*s);
        Theta(n) = s(1);
        z(n) = s(2);
        sigma = (eye(2) - K*C)*sigma;
    end
```

4.3 Results

Figure 4.2 shows the filtered signal for the parameter choices, R $= 0.01I$ & Q $= I$.

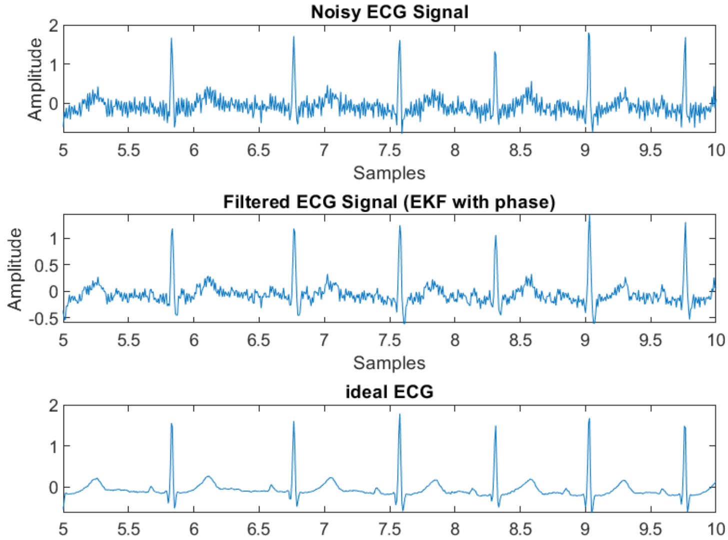

Figure 4.2: Filtered signal using EKF.(R = [0.0588 0; 0 0.0425])

These results can be improved by adjusting R and Q covariances.

5 Improvements and Conclusion

5.1 Improvements

In this implementation we have treated parameters a_i, b_i, θ_i and ω as fixed. But in reality these parameters varies from signal to signal. Even two beats in same ECG signal can have different values for these parameters. For an example ω varies with the heart rate. Therefore it's better to adjust these parameters accordingly.

One can further improve the filter by moddling these parameters as random processors rather than fixed values. Similar approach is taken in paper [2]. But these will result 17 random processors, drastically increasing the complexity.

5.2 Conclusion

This study evaluated the application of Kalman filters and Extended Kalman Filters (EKF) for denoising ECG signals. The scalar Kalman filter showcased effective denoising with proper parameter tuning, while the EKF excelled in capturing non-linear characteristics, particularly for synthetic ECG models. Incorporating phase observations further enhanced offline signal processing capabilities.

However, Kalman filters have notable limitations. Their reliance on accurate model assumptions and parameter tuning can pose challenges in practical scenarios, especially when system dynamics or noise characteristics deviate from these assumptions. For stationary signals, conventional filtering techniques, often outperform Kalman filters by offering simpler implementations. Additionally, the fixed parameters used in the EKF implementation may not adequately capture the variability inherent in real-world ECG signals.

Bibliography

[1] R. Sameni, M. Shamsollahi, and C. Jutten, "Filtering electrocardiogram signals using the extended kalman filter," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2006, pp. 5639–5642.

[2] R. Sameni, M. B. Shamsollahi, C. Jutten, and M. Babaie-Zade, "Filtering noisy ecg signals using the extended kalman filter based on a modified dynamic ecg model," in *Computers in Cardiology, 2005*. IEEE, 2005, pp. 1017–1020.

[3] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE transactions on biomedical engineering*, vol. 50, no. 3, pp. 289–294, 2003.

[4] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

MATLAB scripts:https://github.com/Tharusha-Sihan/KF-and-EKF-for-ECG-denoising