What to do when there are nonzero-mean noises?

- One of the assumptions made when deriving the linear KF is that both w_k and v_k are white.
  - One implication of this assumption is that we must have $\mathbb{E}[w_k] = \mathbb{E}[v_k] = 0$.
- You learned by example from Lesson 1.4.5 that the KF (especially its confidence bounds) fail when this assumption is violated.
- But, what are we to do if we must estimate the state of a system that is affected by noises that truly do have nonzero means?
- One solution is to estimate the noise biases and adjust the state estimate to compensate for these biases.
- The best method of which I am aware was published by Friedland, and I will summarize his approach in this lesson.[1]

[1] B. Friedland, "Treatment of Bias in Recursive Filtering," *IEEE Transactions on Automatic Control*, vol. AC-14, No. 4, Aug. 1969, pp. 359–367.

Defining a system model that includes biases

- We set up the problem by defining a single bias vector b_k that jointly contains the biases present in both noise sources.
- We re-write the system's state-space model as:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B^b b_k + \widetilde{w}_k \\ z_k &= Cx_k + Du_k + C^b b_k + \tilde{v}_k, \end{aligned}$$

  where $\widetilde{w}_k$ and $\tilde{v}_k$ are (zero-mean and) white and the bias is modeled only by b_k.
- The derivation assumes that $b_{k+1} = b_k$ (i.e., the bias is unchanging) but will adapt a time-varying estimate $\hat{b}_{k+1} \neq \hat{b}_k$, which we desire to converge to the true bias.

The solution approach

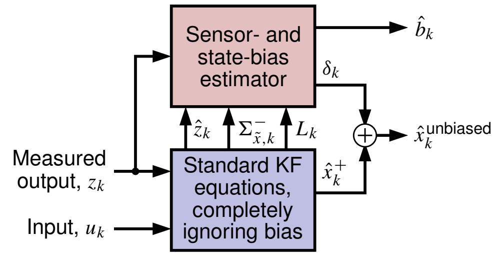
- We can write a set of Kalman-filter equations that augments the state estimate with the bias estimate into a single vector, and then derives a recursive update for this combined vector.
- Friedland, however, takes a clever approach that transforms the update for the combined vector into two separate updates:
  - One set of equations updates the state estimate $\hat{x}_k^+$;
  - The other set of equations updates the bias estimate $\hat{b}_k$.
- The beauty of this approach is that the equations that update $\hat{x}_k^+$ are exactly the same as the standard KF equations.
  - We know from Lesson 1.4.5 that the output of these equations will be biased, but
  - Friedland uses the output of the second set of equations that derives $\hat{b}_k$ to modify this $\hat{x}_k^+$ to produce a corrected state estimate.

Visualizing the solution approach

- The overall approach is visualized in the block diagram.
- A standard KF operates on u_k and z_k and produces $\hat{x}_k^+$.
- A separate set of equations operates on z_k and intermediate variables computed by the KF and produces $\hat{b}_k$ and δ_k.
- The unbiased estimate of the state is:

$$\hat{x}_k^{\text{unbiased}} = \hat{x}_k^+ + \delta_k.$$

- The method also computes confidence bounds on $\hat{b}_k$ (via $\Sigma_{\tilde{b},k}$) and produces corrected confidence bounds on $\hat{x}_k^{\text{unbiased}}$ (via $\Sigma_{\tilde{x},k}^+ + \Sigma_{\tilde{\delta},k}$).



The bias-filter equations

- I will not derive the equations (see the original paper), but I present them here for your reference.
- Friedland defines U_k and V_k such that:

$$\left[\Sigma_{\tilde{x},k}^-\right]^{\text{unbiased}} = \left[\Sigma_{\tilde{x},k}^-\right]^{\text{biased}} + U_k \Sigma_{\tilde{b},k} U_k^T$$
$$\left[\Sigma_{\tilde{x},k}^+\right]^{\text{unbiased}} = \left[\Sigma_{\tilde{x},k}^-\right]^{\text{biased}} + V_k \Sigma_{\tilde{b},k+1} V_k^T.$$

- Variable S_k is also introduced to simplify writing the bias-estimate gain matrix and updated covariance.

Step 1: Update U_k, connecting bias and predicted state
$$U_k = AV_k + B^b$$
Step 2: Compute S_k, connecting bias and measurement
$$S_k = CU_k + C^b$$
Step 3: Compute V_k, the conversion between bias and δ_k
$$V_k = U_k - L_k S_k$$
Step 4: Compute bias-estimate covariance
$$\Sigma_{\tilde{b},k} = \Sigma_{\tilde{b},k-1} - \Sigma_{\tilde{b},k-1} S_k^T \left(C\Sigma_{\tilde{x},k}^- C^T + \Sigma_{\tilde{v}} + S_k \Sigma_{\tilde{b},k-1} S_k^T \right)^{-1} S_k \Sigma_{\tilde{b},k-1}$$
Step 5: Compute bias gain matrix
$$L_k^b = \Sigma_{\tilde{b},k}(V_k^T C^T + (C^b)^T)\Sigma_{\tilde{v}}^{-1}$$
Step 6: Compute bias estimate and state correction
$$\hat{b}_k = (I - L_k^b S_k)\hat{b}_{k-1} + L_k^b(z_k - \hat{z}_k) \quad \text{and then} \quad \delta_k = V_k \hat{b}_k$$

- The table lists the six steps of the gain filter, executed once per measurement interval, utilizing $\hat{z}_k$, $\Sigma_{\tilde{x},k}^-$, and L_k from the main (biased) Kalman filter loop.

Implementing in Octave (initializing)

```
load simOutBias.mat Ad Bd Cd Dd SigmaV SigmaW dT t u x z

% Initialize biased state-estimating Kalman filter
[nx,nt] = size(x); [nz,~] = size(z);
xhat = zeros(nx,1); SigmaX = zeros(nx);
xhatstore = zeros(nx,nt); xboundstore = zeros(nx,nt);

% Initialize bias-filter variables; there are three biases
nb = 3;
Bb = [1 0 0; 0 1 0]; % First two bias states affect x
Cb = [0 0 1];        % Third bias state affects z
bhat = zeros(nb,1);  % Initialize bias-filter estimate of bias
SigmaB = diag([1e-5, 1e-5, 4e-5]); % Initialize bias-filter covariance
bhatstore = zeros(nb,nt); bboundstore = zeros(nb,nt);
V = zeros(nx,nb); % V(0) = Vx(0) in Friedland
U = zeros(nx,nb); % U(0) = Ux(0) in Friedland

% The actual bias used when making the biased dataset, used to plot results
wbias = [0.001; -0.0001];
vbias = -0.025;
```

Implementing in Octave (begin main loop)

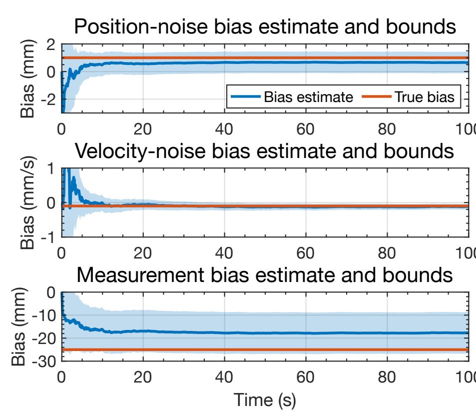
```
for k = 2:length(t)
  % Biased KF Step 1a: State estimate time update
  xhat = Ad*xhat + Bd*u(:,k-1); % use prior value of "u"

  % Biased KF Step 1b: Error covariance time update
  SigmaX = Ad*SigmaX*Ad' + SigmaW;

  % Biased KF Step 1c: Estimate system output
  zhat = Cd*xhat + Dd*u(k); % use present value of "u"

  % Biased KF Step 2a: Compute Kalman gain matrix
  L = SigmaX*Cd'/(Cd*SigmaX*Cd' + SigmaV);

    % Bias Step 1: Update U, connecting bias and state (prediction)
    U = Ad*V + Bb; % U is nx by nb

    % Bias Step 2: Compute S, connecting bias and measurement
    S = Cd*U + Cb; % S is nz by nb

    % Bias Step 3: Compute V, the conversion between bias and deltaX
    V = U - L*S;   % V is nx by nb [need L from KF]
```


Implementing in Octave (end main loop)

```
    % Bias Step 4: Compute bias estimate covariance
    SigmaB = SigmaB - (SigmaB*S'/(Cd*SigmaX*Cd' + SigmaV + ...
             S*SigmaB*S'))*S*SigmaB; % [need SigmaX from KF]

    % Bias Step 5: Compute bias gain
    Lb = SigmaB*(V'*Cd'+Cb')/SigmaV;

    % Bias Step 6: Bias estimate
    bhat = (eye(nb)-Lb*S)*bhat + Lb*(z(k)-zhat); % [need zhat from KF]

  % Biased KF Step 2b: State estimate measurement update
  xhat = xhat + L*(z(k) - zhat);

  % Biased KF Step 2c: Error covariance measurement update
  SigmaX = (eye(nx)-L*Cd)*SigmaX;

  % [Store information for evaluation/plotting purposes]
  xhatstore(:,k) = xhat + V*bhat;  bhatstore(:,k) = bhat;
  xboundstore(:,k) = 3*sqrt(diag(SigmaX) + diag(V*SigmaB*V'));
  bboundstore(:,k) = 3*sqrt(diag(SigmaB));
end
```

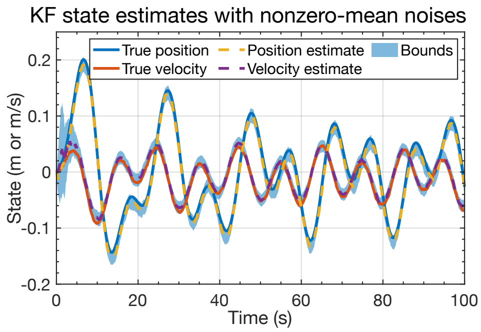

Bias-estimation results

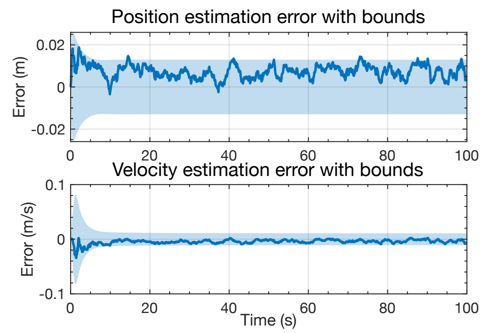
- Octave code for plotting results follows the same approach you have seen before, so I do not repeat it here.
- The figure shows the estimate $\hat{b}_k$ for the three biases in the example, along with their associated confidence bounds and the true biases.
- This example has a large bias on the measurement of position, and the estimator struggles somewhat to detangle the effects of bias in w_k and v_k on the measured position.
- However, all biases converge such that the confidence bounds encompass true bias.



State-estimation results

- What really matters is whether we can now estimate the true state well, even when there exist biased noises.
- The figures below show a large improvement compared with those in Lesson 1.4.5.
- Confidence bounds on position are still somewhat unreliable, but much better.





2.2.4: What do we do when the noises are nonzero-mean?

Summary

- You have learned that it is possible to modify the KF so that it works reasonably well when there exist nonzero noise biases.
- The Friedland approach runs a second filter in parallel with the standard KF equations to estimate the noise biases.
- Then, the composite output combines the biased state estimate and covariance from the standard KF with the variables of the bias estimator to compute an unbiased estimate of the state and its covariance.
- You saw that results are greatly improved compared with those in Lesson 1.4.5, which ran the standard KF equations and ignored the existence of bias.
- Still, even the corrected confidence bounds are imperfect.
  - If we somehow knew the bias exactly via some other means, then we should subtract its effect on the state and measurement directly.
  - "Don't estimate what you know already."