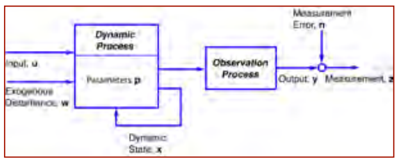
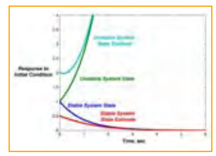
Linear-Optimal Estimation

Robert Stengel
Optimal Control and Estimation MAE 546
Princeton University, 2015





Linear-optimal Gaussian estimator for discrete-time system (Kalman filter)

2nd-order example

Alternative forms of the Kalman filter equations

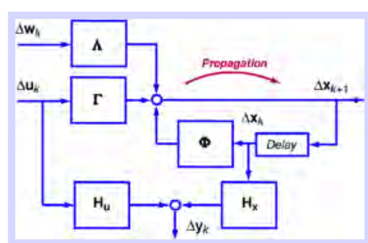
𝒥𝓁𝓁 conditioning

Correlated inputs and measurement noise

Time-correlated measurement noise

http://www.princeton.edu/~stengel/MAE546.html
http://www.princeton.edu/~stengel/OptConEst.html

Uncertain Linear, Time-Varying (LTV) Dynamic Model



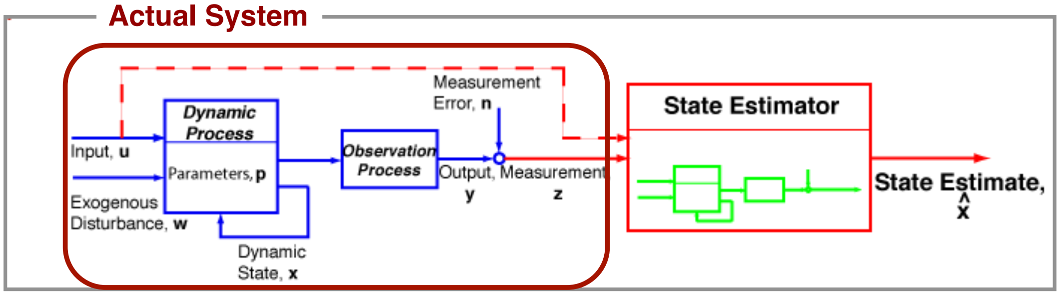
- Discrete-time LTV model with known coefficients

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1} + \mathbf{\Lambda}_{k-1}\mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k$$

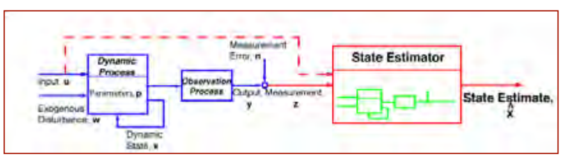
$$\dim(\mathbf{w}_k) = s \times 1$$
$$\dim(\mathbf{z}_k) = r \times 1$$

- Initial condition and disturbance inputs are not known precisely
- Measurement of state is transformed and is subject to error

State Estimation

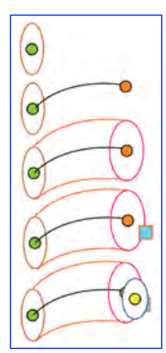


- **Goals**
  - **Minimize effects of measurement error on knowledge of the state**
  - **Reconstruct full state from reduced measurement set ($r \le n$)**
  - **Average redundant measurements ($r \ge n$) to estimate the full state**
- **Method**
  - **Provide optimal balance between measurements and estimates based on the dynamic model alone**



Linear-Optimal State Estimation

- **Kalman filter is the optimal estimator for discrete-time linear systems with Gaussian uncertainty**
- **It has five equations**
  1) **State estimate extrapolation**
  2) **Covariance estimate extrapolation**
  3) **Filter gain computation**
  4) **State estimate update**
  5) **Covariance estimate "update"**



- **Notation**

$$\hat{\mathbf{x}}_k(-): \text{ Estimate at } k^{th} \text{ instant } \textbf{before} \text{ measurement update}$$
$$\hat{\mathbf{x}}_k(+): \text{ Estimate at } k^{th} \text{ instant } \textbf{after} \text{ measurement update}$$

Equations of the Kalman Filter

1) State estimate extrapolation (or propagation)

$$\boxed{\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\,\hat{\mathbf{x}}_{k-1}(+) + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1}}$$

2) Covariance estimate extrapolation (or propagation)

$$\boxed{\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\,\mathbf{P}_{k-1}(+)\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}}$$

Equations of the Kalman Filter

3) Filter gain computation

$$\boxed{\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}}$$

4) State estimate update

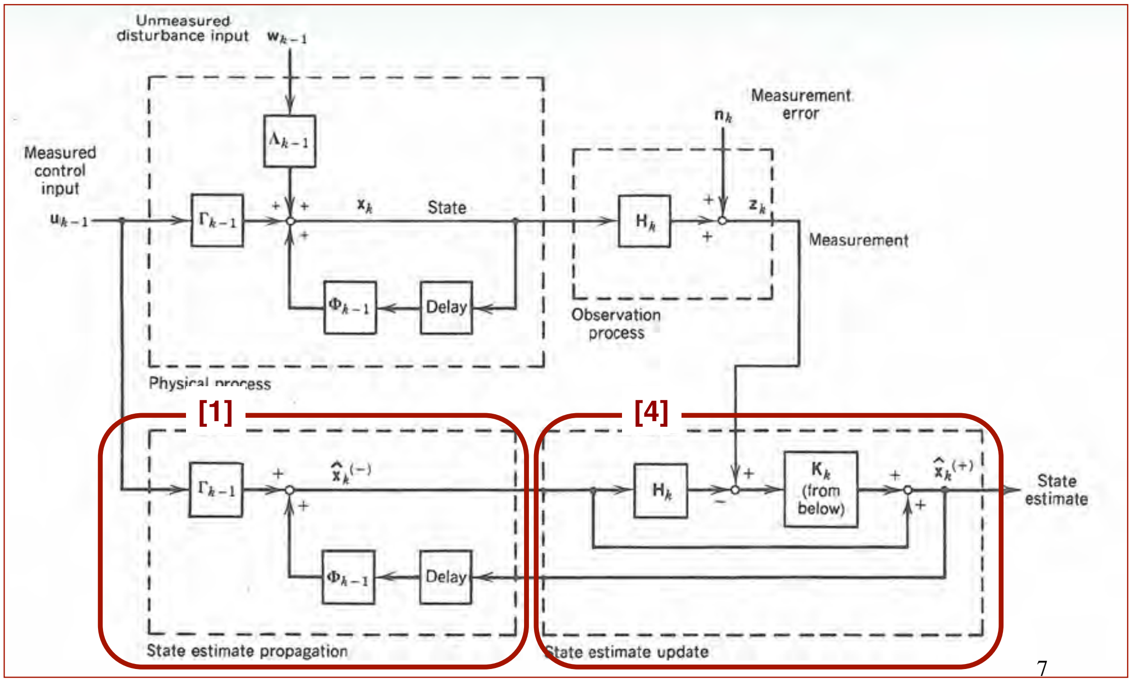
$$\boxed{\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)\right]}$$

5) Covariance estimate “update”

$$\boxed{\mathbf{P}_k(+) = \left[\mathbf{P}_k^{-1}(-) + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right]^{-1}}$$

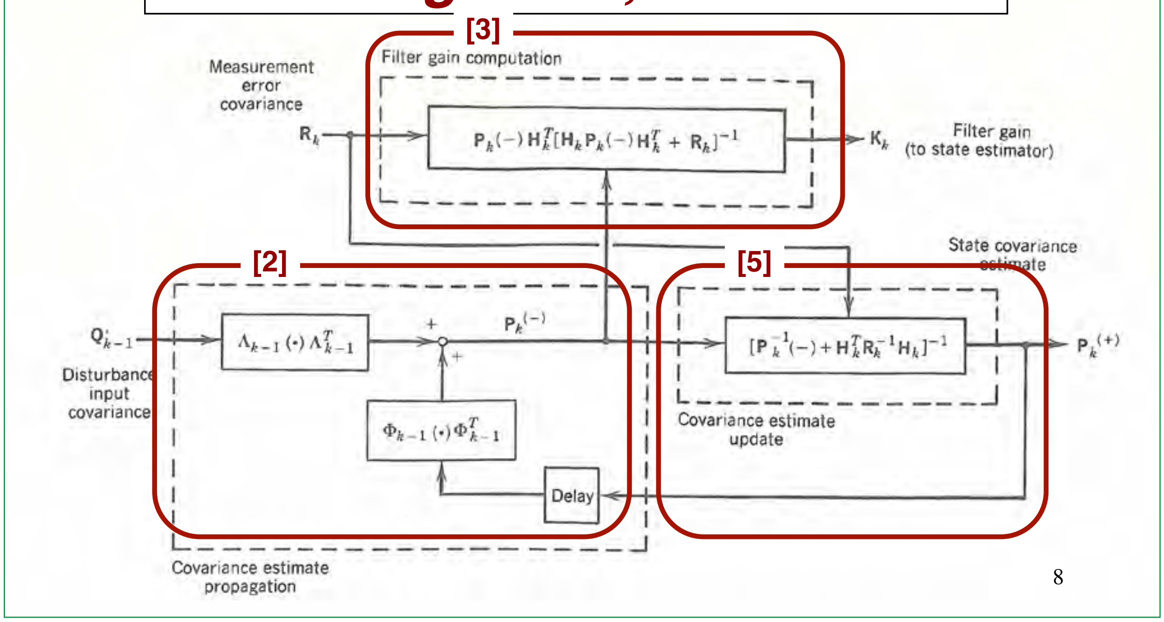
Diagram of State Estimate

Fig. 4.3-2, OCE



Covariance Estimate and Filter Gain Matrix

Fig. 4.3-3, OCE



Alternative Expressions for $\mathbf{K}_k$

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$$

$\mathbf{I}_r = \mathbf{R}_k^{-1}\mathbf{R}_k$

$$= \mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{R}_k\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$$

Matrix Identity

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1} + \mathbf{I}_r\right]^{-1}$$

$$\mathbf{K}_k\left[\mathbf{I}_r + \mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}\right] = \mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}$$

Multiply by inverse

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}$$

Subtract from both sides

$$\mathbf{K}_k = \left(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1}$$

Collect terms

Alternative Expressions for $\mathbf{K}_k$ and $\mathbf{P}_k(+)$

- **From matrix inversion lemma**

$$\mathbf{P}_k(+) = \left[\mathbf{P}_k^{-1}(-) + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right]^{-1}$$

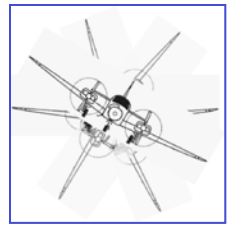
$$= \mathbf{P}_k(-) - \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}\mathbf{H}_k\mathbf{P}_k(-)$$

$$= \left(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k(-)$$

- **This covariance update does not inherently preserve symmetry**
- **With $\mathbf{P}_k(+)$ known, estimation gain is**

$$\mathbf{K}_k = \left(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k(-)\mathbf{H}_k^T\mathbf{R}_k^{-1} = \mathbf{P}_k(+)\mathbf{H}_k^T\mathbf{R}_k^{-1}$$

Second-Order Example of Kalman Filter



- **Rolling motion of an airplane, continuous-time**

$$\begin{bmatrix} \Delta\dot{p} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \Delta p \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} L_{\delta A} \\ 0 \end{bmatrix}\Delta\delta A + \begin{bmatrix} L_p \\ 0 \end{bmatrix}\Delta p_w$$

$$\begin{bmatrix} \Delta p \\ \Delta\phi \end{bmatrix} = \begin{bmatrix} \text{Roll rate, rad/s} \\ \text{Roll angle, rad} \end{bmatrix}$$

$\Delta\delta A$ = Aileron deflection, rad

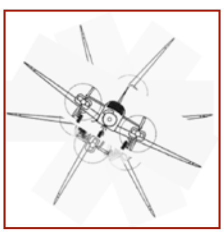
Δp_w = Turbulence disturbance, rad/s

- **Rolling motion of an airplane, discrete-time**

$$\begin{bmatrix} \Delta p_k \\ \Delta\phi_k \end{bmatrix} = \begin{bmatrix} e^{L_pT} & 0 \\ \dfrac{\left(e^{L_pT}-1\right)}{L_p} & 1 \end{bmatrix}\begin{bmatrix} \Delta p_{k-1} \\ \Delta\phi_{k-1} \end{bmatrix} + \begin{bmatrix} \sim L_{\delta A}T \\ 0 \end{bmatrix}\Delta\delta A_{k-1} + \begin{bmatrix} \sim L_pT \\ 0 \end{bmatrix}\Delta p_{w_{k-1}}$$

$$= \begin{bmatrix} \varphi_{11} & 0 \\ \varphi_{21} & 1 \end{bmatrix}\begin{bmatrix} \Delta p_{k-1} \\ \Delta\phi_{k-1} \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ 0 \end{bmatrix}\Delta\delta A_{k-1} + \begin{bmatrix} \lambda_1 \\ 0 \end{bmatrix}\Delta p_{w_{k-1}}$$

T = sampling interval, s



Second-Order Example of Kalman Filter

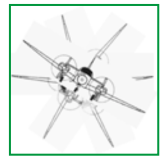
Rate and Angle Measurement

$$\begin{bmatrix} \Delta p_M \\ \Delta\phi_M \end{bmatrix}_k = \begin{bmatrix} \Delta p + \Delta n_p \\ \Delta\phi + \Delta n_\phi \end{bmatrix}_k = \mathbf{I}\Delta\mathbf{x}_k + \Delta\mathbf{n}_k$$

1) State Estimate Extrapolation

$$\begin{bmatrix} \Delta\hat{p}_k(-) \\ \Delta\hat{\phi}_k(-) \end{bmatrix} = \begin{bmatrix} \varphi_{11} & 0 \\ \varphi_{21} & 1 \end{bmatrix}\begin{bmatrix} \Delta\hat{p}_{k-1}(+) \\ \Delta\hat{\phi}_{k-1}(+) \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ 0 \end{bmatrix}\Delta\delta A_{k-1}$$

Second-Order Example of Kalman Filter



2) Covariance Extrapolation

$$\begin{bmatrix} p_{11}(-) & p_{12}(-) \\ p_{21}(-) & p_{22}(-) \end{bmatrix}_k = \begin{bmatrix} \varphi_{11} & 0 \\ \varphi_{21} & 1 \end{bmatrix} \begin{bmatrix} p_{11}(+) & p_{12}(+) \\ p_{21}(+) & p_{22}(+) \end{bmatrix}_{k-1} \begin{bmatrix} \varphi_{11} & \varphi_{21} \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{Q}_{k-1} \approx \begin{bmatrix} L_p \\ 0 \end{bmatrix} Q'_C \begin{bmatrix} L_p & 0 \end{bmatrix} T = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & 0 \end{bmatrix}$$

3) Gain Computation

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}_k = \begin{bmatrix} p_{11}(-) & p_{12}(-) \\ p_{21}(-) & p_{22}(-) \end{bmatrix}_k \left\{ \begin{bmatrix} p_{11}(-) & p_{12}(-) \\ p_{21}(-) & p_{22}(-) \end{bmatrix}_k + \begin{bmatrix} \sigma_{p_M}^2 & 0 \\ 0 & \sigma_{\phi_M}^2 \end{bmatrix}_k \right\}^{-1}$$

$$\mathbf{R}_k \delta_{jk} = \begin{bmatrix} \sigma_{p_M}^2 & 0 \\ 0 & \sigma_{\phi_M}^2 \end{bmatrix}_k$$

Second-Order Example of Kalman Filter

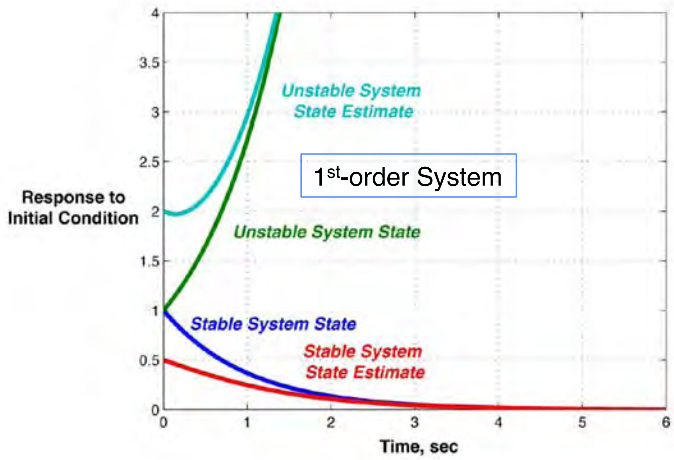
4) State Estimate Update

$$\begin{bmatrix} \Delta\hat{p}_k(+) \\ \Delta\hat{\phi}_k(+) \end{bmatrix} = \begin{bmatrix} \Delta\hat{p}_k(-) \\ \Delta\hat{\phi}_k(-) \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}_k \left\{ \begin{bmatrix} \Delta p_{M_k} \\ \Delta\phi_{M_k} \end{bmatrix} - \begin{bmatrix} \Delta\hat{p}_k(-) \\ \Delta\hat{\phi}_k(-) \end{bmatrix} \right\}$$

5) Covariance “Update”

$$\begin{bmatrix} p_{11}(+) & p_{12}(+) \\ p_{21}(+) & p_{22}(+) \end{bmatrix}_k = \left\{ \begin{bmatrix} p_{11}(-) & p_{12}(-) \\ p_{21}(-) & p_{22}(-) \end{bmatrix}_k + \begin{bmatrix} \dfrac{1}{\sigma_{p_M}^2} & 0 \\ 0 & \dfrac{1}{\sigma_{\phi_M}^2} \end{bmatrix}_k \right\}^{-1}$$

Example of Stability of Kalman Filter Estimate



Estimate is stable because it converges to actual state

Example: Propagating a Scalar Probability Density Function

Scalar LTI system with zero-mean Gaussian random input

$$x_{k+1} = bx_k + \sqrt{1-b^2}u_k + \sqrt{1-b^2}w_k\,, \quad x_0 \ given$$

Propagation of the mean value

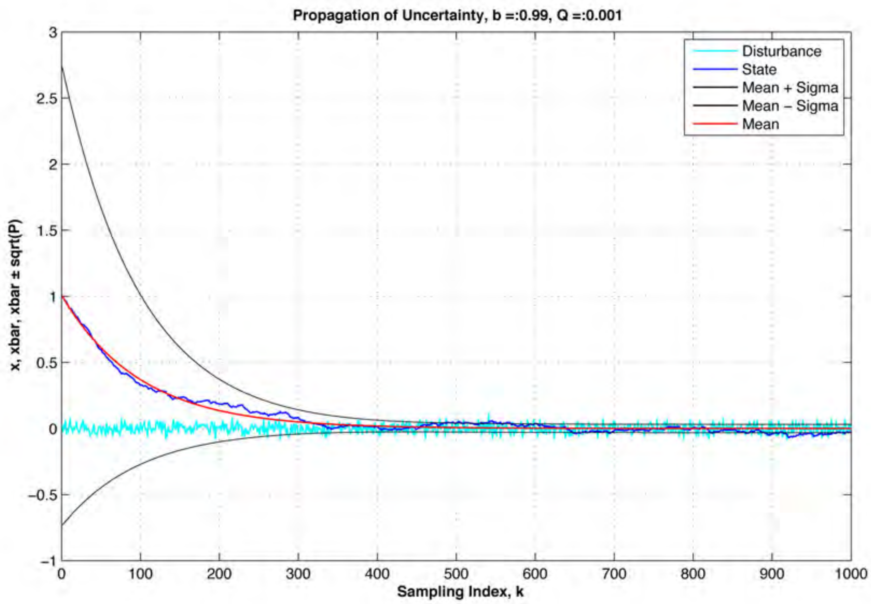
$$\bar{x}_{i+1} = b\bar{x}_i + \sqrt{1-b^2}\bar{u}_i\,, \quad \bar{x}_0 \ given$$

Propagation of the variance

$$P_{k+1} = b^2 P_k + \left(1-b^2\right)Q_k\,, \quad P_0 \ given$$

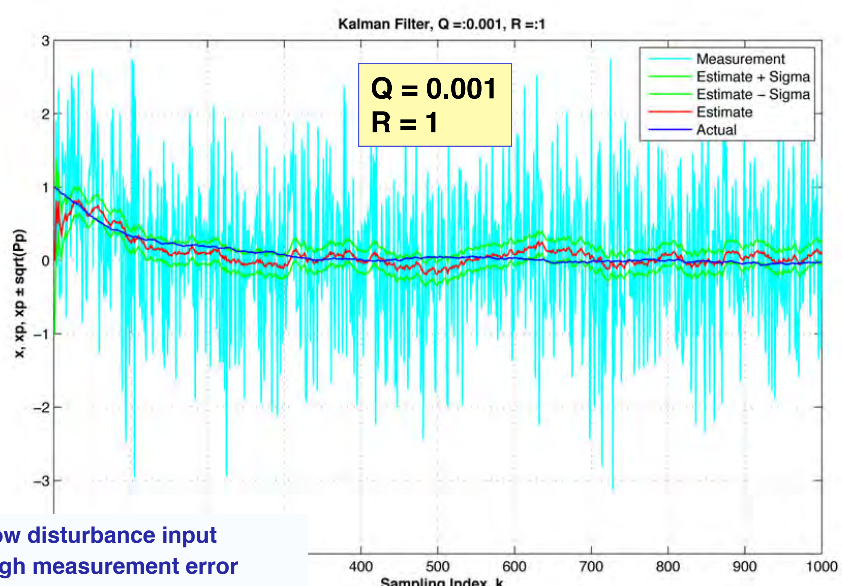
$$Q_k = E\left(w_k^2\right)$$

Example: System Response to Small Disturbance



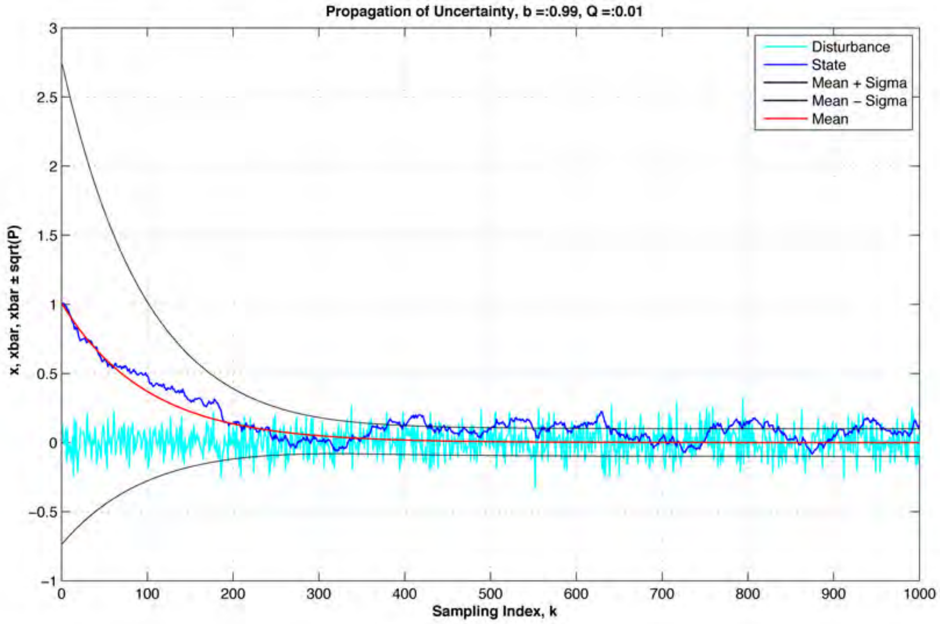
Large estimate of initial uncertainty

Example of Kalman Filter Estimation

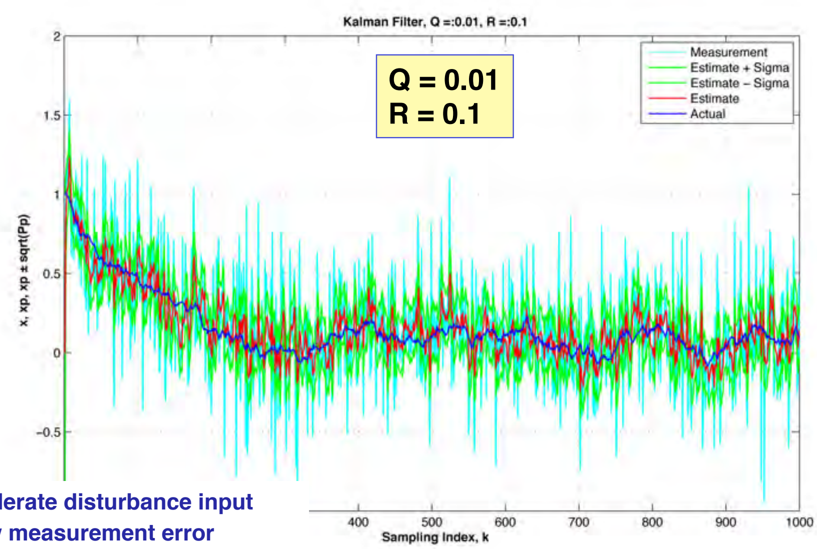


- Low disturbance input
- High measurement error
- *State estimate emphasizes system model*

Example: System Response to Larger Disturbance



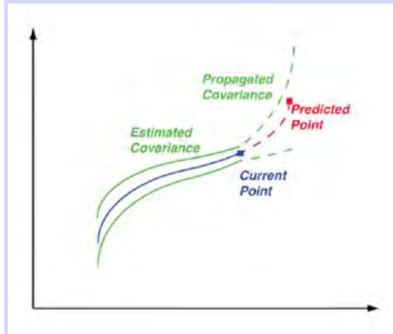
Example of Kalman Filter Estimation



- Moderate disturbance input
- Low measurement error
- *State estimate emphasizes measurement*

Linear-Optimal Predictor

t_k : Current time, sec
t_K : Future time, sec



- State estimate extrapolation (or propagation) from last estimate

$$\boxed{\hat{\mathbf{x}}_K = \mathbf{\Phi}\left(t_K - t_k\right)\hat{\mathbf{x}}_k(+) + \mathbf{\Gamma}\left(t_K - t_k\right)\mathbf{u}_k}$$

- Covariance estimate extrapolation (or propagation)

$$\boxed{\mathbf{P}_K = \mathbf{\Phi}\left(t_K - t_k\right)\mathbf{P}_k(+)\mathbf{\Phi}^T\left(t_K - t_k\right) + \mathbf{Q}_k\left(t_K - t_k\right)}$$

- Predictor analogous to Kalman filter without measurement

Alternative Forms of the Kalman Filter

Simplifying the Gain Calculation

Filter gain computation for *r* measurements

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$$

r x r inversion required

For *r* = 1

$$\mathbf{K}_k = \frac{\mathbf{P}_k(-)\mathbf{H}_k^T}{\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + r_k}$$

scalar division

Consider One Measurement at Each Sampling Interval

- With *r* measurements and diagonal **R**, consider just one measurement at a time

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \cdots \\ z_r \end{bmatrix}; \quad \mathbf{R} = \begin{bmatrix} r_{11} & 0 & \cdots & 0 \\ 0 & r_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & r_{rr} \end{bmatrix}; \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \cdots \\ \mathbf{H}_r \end{bmatrix}$$

$$\begin{aligned} \mathbf{z}_{k+1} &= z_1 \\ \mathbf{z}_{k+2} &= z_2 \\ &\cdots \\ \mathbf{z}_{k+r} &= z_r \end{aligned}$$

Scalar Update

$$\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}(+) + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1}$$
$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_{i_k}\left[\mathbf{z}_{i_k} - \mathbf{H}_{i_k}\hat{\mathbf{x}}_k(-)\right]$$

$$\mathbf{K}_{i_k} = \frac{\mathbf{P}_k(-)\mathbf{H}_{i_k}^T}{\mathbf{H}_{i_k}\mathbf{P}_k(-)\mathbf{H}_{i_k}^T + r_{i_k}}$$

- Cycle through all measurements varying **H**, and repeat cycle
- Wasteful, as it does not use all available information

Sequential Processing of Measurements at Each Sampling Interval

- **With *r* measurements, form an inner loop of calculations, processing one sample at a time**

$$\mathbf{z}_{1_k} = z_1 \qquad \mathbf{z}_{2_k} = z_2 \qquad \cdots \qquad \mathbf{z}_{r_k} = z_r$$

for $i = 1, r$

$$\mathbf{K}_{i_k} = \mathbf{P}_{i-1_k}(+)\mathbf{H}_{i_k}^T \Big/ \left[\mathbf{H}_{i_k}\mathbf{P}_{i-1_k}(+)\mathbf{H}_{i_k}^T + r_{i_k}\right]$$

$$\mathbf{P}_{i_k}(+) = \left(\mathbf{I}_n - \mathbf{K}_{i_k}\mathbf{H}_{i_k}\right)\mathbf{P}_{i-1_k}(+), \quad \mathbf{P}_{0_k}(+) = \mathbf{P}_k(-)$$

$$\hat{\mathbf{x}}_{i_k}(+) = \hat{\mathbf{x}}_{i-1_k}(+) + \mathbf{K}_{i_k}\left[\mathbf{z}_{i_k} - \mathbf{H}_{i_k}\hat{\mathbf{x}}_{i-1_k}(+)\right]$$

end

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_{r_k}(+)$$

$$\mathbf{P}_k(+) = \mathbf{P}_{r_k}(+)$$

$$\mathbf{H}_{1_k} = \mathbf{H}_1 \qquad \mathbf{H}_{2_k} = \mathbf{H}_2 \qquad \cdots \qquad \mathbf{H}_{r_k} = \mathbf{H}_r$$

Joseph Form of the Filter ("Stabilized Kalman Filter")

Guaranteed to retain positive-definiteness and symmetry

State update is

$$\hat{\mathbf{x}}_k(+) = \left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]\hat{\mathbf{x}}_k(-) + \mathbf{K}_k\mathbf{z}_k$$

Pre- and post-update measurement errors

$$\boldsymbol{\varepsilon}_k(-) = \mathbf{x}_k - \hat{\mathbf{x}}_k(-); \quad \boldsymbol{\varepsilon}_k(+) = \mathbf{x}_k - \hat{\mathbf{x}}_k(+)$$

Measurement error is updated by

$$\boldsymbol{\varepsilon}_k(+) = \left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]\boldsymbol{\varepsilon}_k(-) + \mathbf{K}_k\mathbf{n}_k$$

Joseph Form of the Filter
("Stabilized Kalman Filter")

$$\boxed{\boldsymbol{\varepsilon}_k(+) = \left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]\boldsymbol{\varepsilon}_k(-) + \mathbf{K}_k\mathbf{n}_k}$$

Definitions

$$\boxed{E\left(\boldsymbol{\varepsilon}_k\boldsymbol{\varepsilon}_k^T\right) = \mathbf{P}_k; \quad E\left(\mathbf{n}_k\mathbf{n}_k^T\right) = \mathbf{R}_k; \quad E\left(\boldsymbol{\varepsilon}_k\mathbf{n}_k^T\right) = \mathbf{0}}$$

Then, covariance update is the outer product of the expected error

$$\boxed{\mathbf{P}_k(+) = \left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]\mathbf{P}_k(-)\left[\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k\right]^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T}$$

- **Update equation is symmetric**
- **Equation updates covariance whether or not K is optimal**
  - **Evaluate error covariance of a sub-optimal filter**
  - **Design and evaluate a low-order filter for a high-order system**
- **Does not require an (n x n) inversion**

Information Matrix Form of the Kalman Filter

- **Filter based on the inverse of P**

$\mathbf{P}$: State error covariance (small is good)

$\mathscr{I} = \mathbf{P}^{-1}$: Information matrix (large is good)

- **Information filter equations replace large inversions by small inversions**
- ***See OCE***

Ill Conditioning of the Filter Computation

- **Calculations may be inaccurate if system contains**
  - very fast and slow modes
  - very noisy and near-perfect measurements
  - very large and small disturbance inputs
- **Condition number of a matrix, P, is the ratio of singular values**

$$k(\mathbf{P}) = \left[\frac{\lambda_{\max}\left(\mathbf{P}^T\mathbf{P}\right)}{\lambda_{\min}\left(\mathbf{P}^T\mathbf{P}\right)}\right]^{1/2} = \frac{\sigma_{\max}(\mathbf{P})}{\sigma_{\min}(\mathbf{P})} = \frac{\bar{\sigma}(\mathbf{P})}{\underline{\sigma}(\mathbf{P})}$$

Solutions to *Ill* Conditioning

- **Double, triple, ..., precision arithmetic, or**
- **Formulate the equations to solve for the square root of P**
- ***Define***

$$\mathbf{P} \triangleq \mathbf{S}\mathbf{S}^T \text{ or } \mathbf{S}^T\mathbf{S}$$

$$\text{then } k(\mathbf{P}) = k\left(\mathbf{S}\mathbf{S}^T\right) = k^2(\mathbf{S}), \quad \text{which is order } 10^{x}$$

$$k(\mathbf{S}) = \sqrt{k(\mathbf{P})}, \quad \text{which is order } 10^{x/2}$$

"U-D" Formulation of the Kalman Filter

Factorization of P

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T \quad \sim \left(\mathbf{U}\mathbf{D}^{1/2}\right)\left(\mathbf{U}\mathbf{D}^{1/2}\right)^T \quad \sim \mathbf{S}\mathbf{S}^T$$

where

$$\mathbf{U}: \quad \text{Unit upper triangular matrix:} \begin{bmatrix} 1 & \ddots & \ddots \\ 0 & 1 & \ddots \\ 0 & 0 & 1 \end{bmatrix}$$

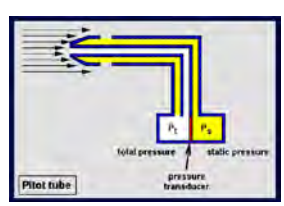
$$\mathbf{D}: \quad \text{Diagonal matrix:} \begin{bmatrix} \ddots & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \ddots \end{bmatrix}$$

- **No square roots in the factorization, but formulation has square-root conditioning**
- **Covariance update uses <u>sequential processing</u>**
- **Algorithm originally expressed in pseudo-code** (Bierman and Thornton, 1977)
- ***See OCE (pp. 357-360) for equations and pseudo-code***

Kalman Filters for More Complex Systems*

- **Correlated Disturbance Input and Measurement Error**
  - **Measurements may be corrupted by the same processes that force the system (e.g., turbulence)**
  - **Consider estimation-error dynamics, as in Joseph form**
  - **Stepwise minimization of estimation cost function *w.r.t.* filter gain matrix**
  - ***See OCE***

- **Time-Correlated ("Colored") Measurement Error**
  - **Augment system with measurement error dynamics**
  - **Use measurement differencing**
  - **Two-step state and covariance estimates**
  - ***See OCE***



**** Arthur Bryson and students***

Next Time:
Kalman-Bucy Filters for Continuous-Time Systems

Supplemental Material

Descriptions of Random Variables

$$E(\mathbf{x}_0) = \hat{\mathbf{x}}_o \,; \quad E\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_o)(\mathbf{x}_0 - \hat{\mathbf{x}}_o)^T\right] = \mathbf{P}_0$$

$$E(\mathbf{w}_k) = \mathbf{0}; \; E\left(\mathbf{w}_j \mathbf{w}_k^T\right) = \mathbf{Q}'_k \delta_{jk}$$

$$E[\mathbf{u}_k] = \mathbf{u}_k; \quad E\left\{[\mathbf{u}_k - \bar{\mathbf{u}}_k][\mathbf{u}_k - \bar{\mathbf{u}}_k]^T\right\} = \mathbf{0}$$

$$E(\mathbf{n}_k) = \mathbf{0}; \; E\left(\mathbf{n}_j \mathbf{n}_k^T\right) = \mathbf{R}_k \delta_{jk}$$

$$E\left(\mathbf{w}_j \mathbf{n}_k^T\right) = \mathbf{0}$$

Program for Example of Kalman Filter Estimate Error Stability

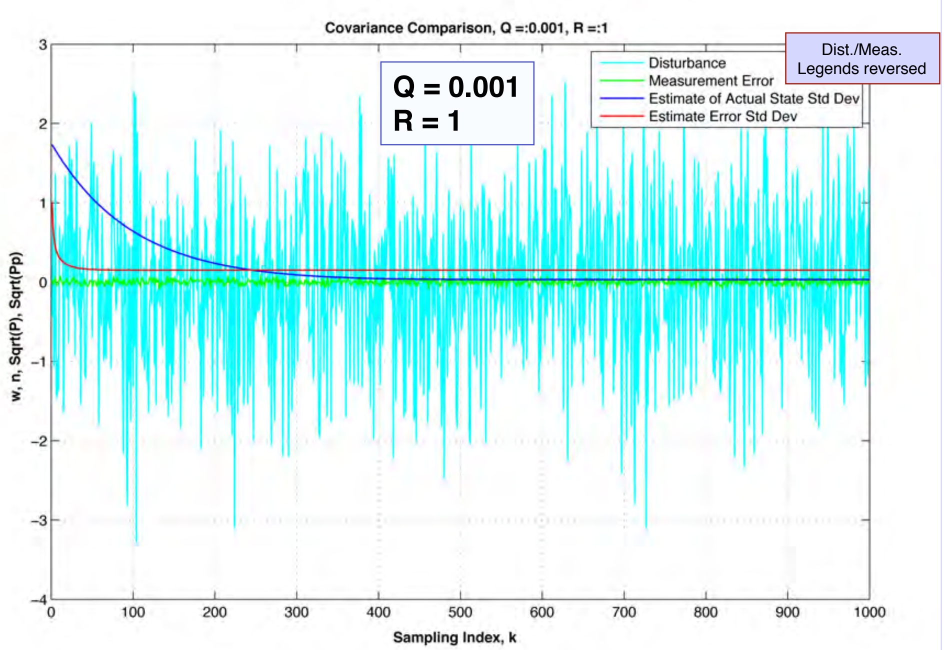
```
%   Kalman-Bucy Filter Estimate Stability
%   Copyright by Robert Stengel.  All rights reserved.
%   4/11/2010

    clear

%   First-Order Stable and Unstable Systems

    StableSys   =   ss(-1,1,1,0);
    UnstableSys =   ss(1,1,1,0);

    [Kstable,Lstable,Pstable]    = kalman(StableSys,1,1,0);
    [Kunstable,Lunstable,Punstable]    = kalman(UnstableSys,1,1,0);

    StableSysEst    =   ss((-1-Lstable),1,1,0);
    UnstableSysEst  =   ss((1-Lunstable),1,1,0);

%   Response

    t           =   [0:0.01:6];
    [y1,t1,x1]  =   initial(StableSys,1,t);
    [y2,t2,x2]  =   initial(UnstableSys,1,t);
    [y3,t3,x3]  =   initial(StableSysEst,0.5,t);
    [y4,t4,x4]  =   initial(UnstableSysEst,-1,t);

    figure
    y3          =   interp1(t3,y3,t1);
    y4          =   interp1(t4,y4,t2);
    plot(t1,y1,t2,y2,t1,(y1-y3),t2,(y2-y4)),grid, axis([0 6 0 4]) ax
```

Program for Covariance Propagation and Kalman Filter Estimatation

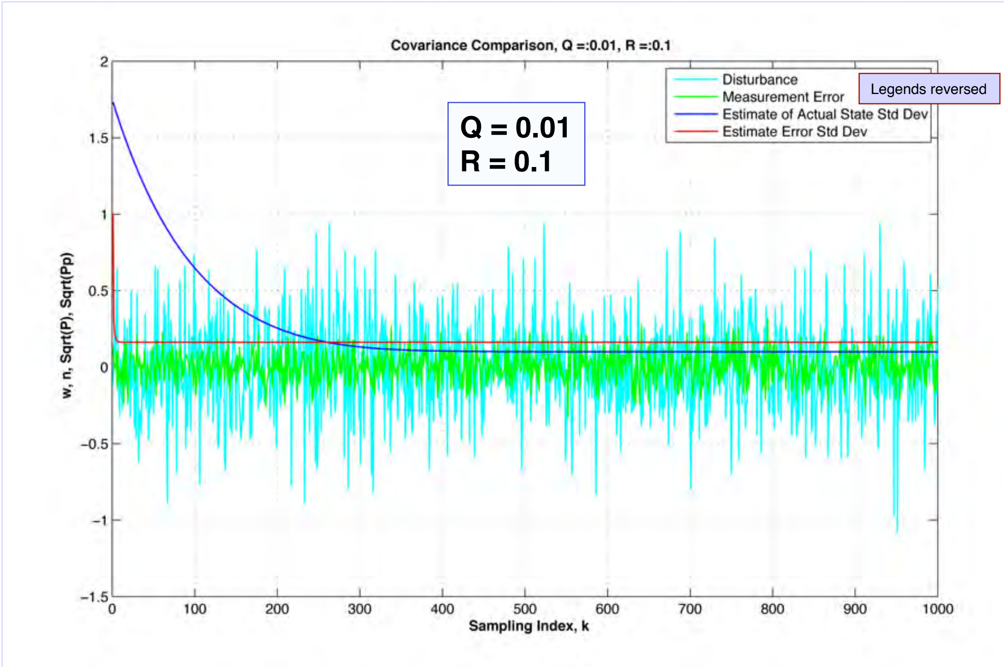
```
%   Scalar Propagation of Mean and Variance
%   Kalman Filter
%   Copyright by Robert Stengel.  All rights reserved.
%   4/12/2011

    clear
    '==========================='
    'Propagation of Uncertainty'
    date

    b       =   0.99
    b2      =   b*b
    bb      =   1 - b^2
    sqrb    =   sqrt(bb)
    w       =   [];
    x       =   [];
    xbar    =   [];


    x(1)    =   1
    xbar(1) =   x(1)
    P(1)    =   3
    Q       =   0.01
    u       =   0


    for  k  =   1:999
        w(k)        =   sqrt(Q)*randn(1);
        x(k+1)      =   b*x(k) + sqrb*u + sqrb*w(k);
        xbar(k+1)   =   b*xbar(k) + sqrb*u;
        P(k+1)      =   b2*P(k) + bb*Q;
    end

    k       =   [1:1000];
    w(1000) =    sqrt(Q)*randn(1);

    figure
    plot(k,w,'c',k,x,'b',k,(xbar + sqrt(P)),'k',k,(xbar -
sqrt(P)),'k',k,xbar,'r'),grid
    title(['Propagation of Uncertainty, b =:', num2str(b), ', Q
=:',num2str(Q)]), xlabel('Sampling Index, k'), ylabel('x, xbar, xbar
± sqrt(P)')
    legend('Disturbance','State','Mean + Sigma','Mean -
Sigma','Mean')
```

```
%   Kalman Filter
    '============='
    'Kalman Filter'

    xm(1)   =   0
    xp(1)   =   0
    Pm(1)   =   1
    Pp(1)   =   1
    z(1)    =   xm(1)
    Q       =   Q
    R       =   0.1
    u       =   0

    for  k  =   1:999
        n(k)    =   sqrt(R)*randn(1);
        xm(k+1) =   b*xp(k) + sqrb*u;
        Pm(k+1) =   b2*Pp(k) + Q;
        K       =   Pm(k+1) / (Pm(k+1) + R);
        z(k+1)  =   x(k+1) + n(k);
        xp(k+1) =   xm(k+1) + K*(z(k+1) - xm(k+1));
        Pp(k+1) =   1 / ((1/Pm(k+1)) + 1/R);
    end

    k       =   [1:1000];
    n(1000) =   sqrt(R)*randn(1);
    figure
    plot(k,z,'c',k,(xp + sqrt(Pp)),'g',k,(xp -
sqrt(Pp)),'g',k,xp,'r',k,x,'b'),grid
    title(['Kalman Filter, Q =:',num2str(Q),', R =:',num2str(R)]),
xlabel('Sampling Index, k'), ylabel('x, xp, xp ± sqrt(Pp)')
    legend('Measurement','Estimate + Sigma','Estimate -
Sigma','Estimate','Actual')

%   Covariance Comparison

    figure
    SP  =   sqrt(P);
    SPp =   sqrt(Pp);
    plot(k,n,'c',k,w,'g',k,SP,'b',k,SPp,'r'), grid
    title(['Covariance Comparison, Q =:',num2str(Q),', R =:'
num2str(R)]), xlabel('Sampling Index, k'),ylabel('w, n, Sqrt(P),
Sqrt(Pp)')
    legend('Measurement Error','Disturbance','Estimate of Actual
State Std Dev','Estimate Error Std Dev')
```

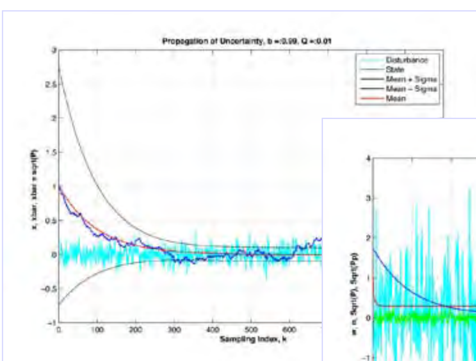

Example of Kalman Filter Estimation

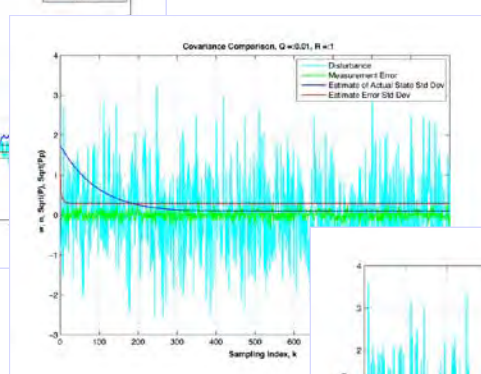


Example of Kalman Filter Estimation

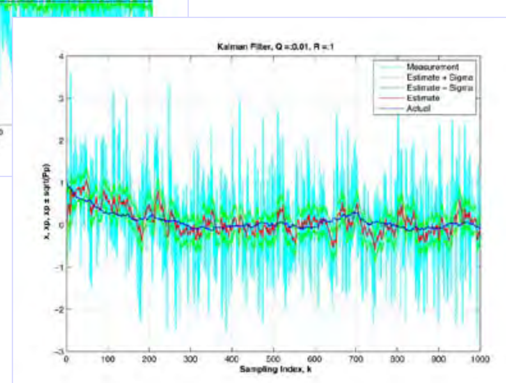


Example of Kalman Filter Estimation





Q = 0.01
R = 1



- Moderate disturbance input
- High measurement error