CBE507 LECTURE IV
Multivariable and Optimal Control

Professor Dae Ryook Yang

Fall 2013

Dept. of Chemical and Biological Engineering
Korea University

Decoupling

- **Handling MIMO processes**
  - MIMO process can be converted into SISO process.
    - Neglect some features to get SISO model
    - Cannot be done always
  - Decouple the control gain matrix K and estimator gain L.
    - Depending on the importance, neglect some gains.
    - Simpler
    - Performance degradation
    - Examples

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -\begin{bmatrix} K_{11} & K_{12} & 0 & 0 \\ 0 & 0 & K_{23} & K_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x}_c(k+1) \\ \mathbf{x}_s(k+1) \end{bmatrix} = -\begin{bmatrix} \mathbf{\Phi}_{cc} & \mathbf{\Phi}_{cs} \\ \mathbf{\Phi}_{sc} & \mathbf{\Phi}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{x}_c(k) \\ \mathbf{x}_s(k) \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma}_c \\ \mathbf{\Gamma}_s \end{bmatrix} u(k) \Rightarrow \begin{matrix} \overline{\mathbf{x}}_c(k+1) = \mathbf{\Phi}_{cc}\overline{\mathbf{x}}_c(k) + \mathbf{\Phi}_{cs}\overline{\mathbf{x}}_s(k) + \mathbf{\Gamma}_c u(k) + \mathbf{L}_c(y_c - \overline{y}_c) \\ \overline{\mathbf{x}}_s(k+1) = \mathbf{\Phi}_{sc}\overline{\mathbf{x}}_c(k) + \mathbf{\Phi}_{ss}\overline{\mathbf{x}}_s(k) + \mathbf{\Gamma}_s u(k) + \mathbf{L}_s(y_s - \overline{y}_s) \end{matrix}$$

Time-Varying Optimal Control

- **Cost function**
  - **A discrete plant:** $\mathbf{x}(k+1) = \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k)$

$$\min_{\mathbf{u}(k)} J = \frac{1}{2}\sum_{k=0}^{N}[\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)]$$

  - **Q_1 and Q_2 are nonnegative symmetric weighting matrix**
  - **Plant model works as constraints.**
- **Lagrange multiplier:** $\lambda(k)$

$$\min_{\mathbf{u}(k),\mathbf{x}(k),\boldsymbol{\lambda}(k)} J = \sum_{k=0}^{N}[\frac{1}{2}\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \frac{1}{2}\mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k) + \boldsymbol{\lambda}^T(k+1)(-\mathbf{x}(k+1) + \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k))]$$

  - **minimization**

$$\frac{\partial J}{\partial \mathbf{u}(k)} = \mathbf{u}^T(k)\mathbf{Q}_2 + \boldsymbol{\lambda}^T(k+1)\mathbf{\Gamma} = 0 \qquad \text{(control equations)}$$

$$\frac{\partial J}{\partial \boldsymbol{\lambda}(k+1)} = -\mathbf{x}(k+1) + \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) = 0 \qquad \text{(state equations)}$$

$$\frac{\partial J}{\partial \mathbf{x}(k)} = \mathbf{x}^T(k)\mathbf{Q}_1 - \boldsymbol{\lambda}^T(k) + \boldsymbol{\lambda}^T(k+1)\mathbf{\Phi} = 0 \qquad \text{(adjoint equations)}$$

- **Control law:** $\mathbf{u}(k) = -\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\boldsymbol{\lambda}(k+1)$
- **Lagrange multiplier update:**

$$\boldsymbol{\lambda}(k) = \mathbf{\Phi}^T\boldsymbol{\lambda}(k+1) + \mathbf{Q}_1\mathbf{x}(k) \Rightarrow \boldsymbol{\lambda}(k+1) = \mathbf{\Phi}^{-T}\boldsymbol{\lambda}(k) - \mathbf{\Phi}^{-T}\mathbf{Q}_1\mathbf{x}(k)$$

- **Optimal control problem (Two-point boundary-value problem)**
  - **x**(0) **and u**(0) **are known, but** λ(0) **is unknown.**
  - **Since u(*N*) has no effect on x(*N*), λ(*N*+1)=0.**

$$\mathbf{x}(k) = \mathbf{\Phi}\mathbf{x}(k-1) + \mathbf{\Gamma}\mathbf{u}(k-1)$$

$$\boldsymbol{\lambda}(k+1) = \mathbf{\Phi}^{-T}\boldsymbol{\lambda}(k) - \mathbf{\Phi}^{-T}\mathbf{Q}_1\mathbf{x}(k)$$

$$\mathbf{u}(k) = -\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\boldsymbol{\lambda}(k+1)$$

Boundary Conditions

$$\boldsymbol{\lambda}(N) = \mathbf{Q}_1\mathbf{x}(N)$$

$$\mathbf{x}(0) = \mathbf{x}_0$$

  - **If *N* is decided, u(*k*) will be obtained by solving above two-point boundary-value problem. (Not easy)**
  - **The obtained solution, u(*k*) is the optimal control policy.**

- **Sweep method (by Bryson and Ho, 1975)**
  - **Assume $\boldsymbol{\lambda}(k) = \mathbf{S}(k)\mathbf{x}(k)$.**

$$\mathbf{Q}_2\mathbf{u}(k) = -\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\mathbf{x}(k+1) = -\boldsymbol{\Gamma}^T\mathbf{S}(k+1)(\boldsymbol{\Phi}\mathbf{x}(k)+\boldsymbol{\Gamma}\mathbf{u}(k))$$

$$\Rightarrow \mathbf{u}(k) = -(\mathbf{Q}_2+\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Phi}\mathbf{x}(k) = -\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Phi}\mathbf{x}(k)$$

$$\text{where } \mathbf{R} = \mathbf{Q}_2+\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Gamma}$$

  - **Solution of $\mathbf{S}(k)$**

$$\boldsymbol{\lambda}(k) = \boldsymbol{\Phi}^T\boldsymbol{\lambda}(k+1)+\mathbf{Q}_1\mathbf{x}(k) \Rightarrow \mathbf{S}(k)\mathbf{x}(k) = \boldsymbol{\Phi}^T\mathbf{S}(k+1)\mathbf{x}(k+1)+\mathbf{Q}_1\mathbf{x}(k)$$

$$\Rightarrow \mathbf{S}(k)\mathbf{x}(k) = \boldsymbol{\Phi}^T\mathbf{S}(k+1)(\boldsymbol{\Phi}\mathbf{x}(k)-\boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Phi}\mathbf{x}(k))+\mathbf{Q}_1\mathbf{x}(k)$$

$$\Rightarrow [\mathbf{S}(k)-\boldsymbol{\Phi}^T\mathbf{S}(k+1)\boldsymbol{\Phi}+\boldsymbol{\Phi}^T\mathbf{S}(k+1)\boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\mathbf{S}(k+1)\boldsymbol{\Phi}-\mathbf{Q}_1]\mathbf{x}(k) = 0$$

- **Discrete Riccati equation**

$$\mathbf{S}(k) = \boldsymbol{\Phi}^T[\mathbf{S}(k+1)-\mathbf{S}(k+1)\boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\mathbf{S}(k+1)]\boldsymbol{\Phi}+\mathbf{Q}_1$$

  - **Single boundary condition: $\mathbf{S}(N)=\mathbf{Q}_1$.**
  - **The recursive equation must be solved backward.**

– **Optimal time-varying feedback gain, $\mathbf{K}(k)$**

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k)$$

where $\mathbf{K}(k) = [\mathbf{Q}_2 + \mathbf{\Gamma}^T \mathbf{S}(k+1)\mathbf{\Gamma}]^{-1}\mathbf{\Gamma}^T \mathbf{S}(k+1)\mathbf{\Phi}$

- **The optimal gain, $\mathbf{K}(k)$, changes at each time but can be pre-computed if N is known.**
- **It is independent of $\mathbf{x}(0)$.**

– **Optimal cost function value**

$$J = \frac{1}{2}\sum_{k=0}^{N}[\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k) - \boldsymbol{\lambda}^T(k+1)\mathbf{x}(k+1) + (\boldsymbol{\lambda}^T(k) - \mathbf{Q}_1)\mathbf{x}(k) - \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)]$$

$$= \frac{1}{2}\sum_{k=0}^{N}[\boldsymbol{\lambda}^T(k)\mathbf{x}(k) - \boldsymbol{\lambda}^T(k+1)\mathbf{x}(k+1)]$$

$$= \frac{1}{2}\boldsymbol{\lambda}^T(0)\mathbf{x}(0) - \frac{1}{2}\boldsymbol{\lambda}^T(N+1)\mathbf{x}(N+1) = \frac{1}{2}\boldsymbol{\lambda}^T(0)\mathbf{x}(0) = \frac{1}{2}\mathbf{x}^T(0)\mathbf{S}(0)\mathbf{x}(0)$$

LQR Steady-State Optimal Control

- **Linear Quadratic Regulator (LQR)**
  - Infinite time problem of regulation case
  - LQR applies to linear systems with quadratic cost function.
  - Algebraic Riccati Equation (ARE)

$$\mathbf{S}_\infty = \mathbf{\Phi}^T[\mathbf{S}_\infty - \mathbf{S}_\infty \mathbf{\Gamma} \mathbf{R}^{-1} \mathbf{\Gamma}^T \mathbf{S}_\infty] \mathbf{\Phi} + \mathbf{Q}_1$$

    - ARE has two solutions and the right solution should be positive definite. (J=$\mathbf{x}^T(0)\mathbf{S}(0)\mathbf{x}(0)$ is positive)
    - Numerical solution should be seek except very few cases.
  - Hamilton's equations or Euler-Lagrange equations

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) = \mathbf{\Phi}\mathbf{x}(k) - \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T \boldsymbol{\lambda}(k+1)$$

$$\boldsymbol{\lambda}(k+1) = \mathbf{\Phi}^{-T}\boldsymbol{\lambda}(k) - \mathbf{\Phi}^{-T}\mathbf{Q}_1\mathbf{x}(k)$$

$$\Rightarrow \begin{bmatrix} \mathbf{x}(k+1) \\ \boldsymbol{\lambda}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T}\mathbf{Q}_1 & -\mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T} \\ -\mathbf{\Phi}^{-T}\mathbf{Q}_1 & \mathbf{\Phi}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \boldsymbol{\lambda}(k) \end{bmatrix}$$

: System dynamics

Hamiltonian matrix, $\mathbf{H}_c$

– **Hamiltonian matrix has 2*n* eigenvalues. (*n* stable + *n* unstable)**

- **Using z-transform**

$$
\begin{aligned}
z\mathbf{X}(z) &= \mathbf{\Phi}\mathbf{X}(z) + \mathbf{\Gamma}\mathbf{U}(z) \\
\mathbf{U}(z) &= -z\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\mathbf{\Lambda}(z) \\
\mathbf{\Lambda}(z) &= \mathbf{Q}_1\mathbf{X}(z) + z\mathbf{\Phi}^T\mathbf{\Lambda}(z)
\end{aligned}
\quad \Rightarrow \begin{bmatrix} z\mathbf{I}-\mathbf{\Phi} & \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T \\ -\mathbf{Q}_1 & z^{-1}\mathbf{I}-\mathbf{\Phi}^T \end{bmatrix} \begin{bmatrix} \mathbf{X}(z) \\ z\mathbf{\Lambda}(z) \end{bmatrix} = \mathbf{0}
$$

- **Characteristic equation**

$$
\det\begin{bmatrix} z\mathbf{I}-\mathbf{\Phi} & \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T \\ -\mathbf{Q}_1 & z^{-1}\mathbf{I}-\mathbf{\Phi}^T \end{bmatrix} = \det\begin{bmatrix} z\mathbf{I}-\mathbf{\Phi} & \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T \\ \mathbf{0} & z^{-1}\mathbf{I}-\mathbf{\Phi}^T + \mathbf{Q}_1(z\mathbf{I}-\mathbf{\Phi})^{-1}\mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T \end{bmatrix} = \mathbf{0}
$$

$$
\Rightarrow \det(z\mathbf{I}-\mathbf{\Phi})\det((z^{-1}\mathbf{I}-\mathbf{\Phi}^T)[\mathbf{I}+(z^{-1}\mathbf{I}-\mathbf{\Phi}^T)^{-1}\mathbf{Q}_1(z\mathbf{I}-\mathbf{\Phi})^{-1}\mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T]) = 0
$$

$$
\Rightarrow \det(z\mathbf{I}-\mathbf{\Phi})\det(z^{-1}\mathbf{I}-\mathbf{\Phi}^T)\det(\mathbf{I}+(z^{-1}\mathbf{I}-\mathbf{\Phi}^T)^{-1}\mathbf{Q}_1(z\mathbf{I}-\mathbf{\Phi})^{-1}\mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T) = 0
$$

  – $\det(z\mathbf{I}-\Phi)=\alpha(z)$ **is the plant characteristics and** $\det(z^{-1}\mathbf{I}-\Phi)=\alpha(z^{-1})$.

  – **Called "Reciprocal Root properties**

- **The system dynamics using u(*k*)=–K$_\infty$x(*k*) will have *n* stable poles.**

- **Eigenvalue Decomposition of Hamiltonian matrix**
  - Assume that the Hamiltonian matrix, $\mathbf{H}_c$, is diagonalizable.

$$\mathbf{H}_c^* = \mathbf{W}^{-1}\mathbf{H}_c\mathbf{W} = \begin{bmatrix} \mathbf{E}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix}$$

  - Eigenvectors of $\mathbf{H}_c$ (transformation matrix): $\mathbf{W} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{\Lambda}_I & \mathbf{\Lambda}_O \end{bmatrix}$

$$\begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix} = \mathbf{W}^{-1}\begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{W}\begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{\Lambda}_I & \mathbf{\Lambda}_O \end{bmatrix}\begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\lambda}^* \end{bmatrix}$$

  - Solution

$$\begin{bmatrix} \mathbf{x}^*(N) \\ \boldsymbol{\lambda}^*(N) \end{bmatrix} = \begin{bmatrix} \mathbf{E}^{-N} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^{N} \end{bmatrix}\begin{bmatrix} \mathbf{x}^*(0) \\ \boldsymbol{\lambda}^*(0) \end{bmatrix}$$

    - Since $\mathbf{x}^*$ goes to zero as $N\rightarrow\infty$, $\boldsymbol{\lambda}^*(0)$ should be zero.

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{X}_I\mathbf{x}^*(k) = \mathbf{X}_I\mathbf{E}^{-k}\mathbf{x}^*(0) \\ \boldsymbol{\lambda}(k) &= \mathbf{\Lambda}_I\mathbf{x}^*(k) = \mathbf{\Lambda}_I\mathbf{E}^{-k}\mathbf{x}^*(0) \end{aligned} \Rightarrow \begin{aligned} \mathbf{x}^*(0) &= \mathbf{E}^{k}\mathbf{X}_I^{-1}\mathbf{x}(k) \\ \boldsymbol{\lambda}(k) &= \mathbf{\Lambda}_I\mathbf{X}_I^{-1}\mathbf{x}(k) = \mathbf{S}_\infty\mathbf{x}(k) \end{aligned}$$

$$\mathbf{u}(k) = -\mathbf{K}_\infty\mathbf{x}(k) \quad \text{where } \mathbf{K}_\infty = (\mathbf{Q}_2 + \mathbf{\Gamma}^T\mathbf{S}_\infty\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{S}_\infty\mathbf{\Phi}$$

- **Cost Equivalent**
  - **The cost will be dependent on the sampling time.**
  - **If the cost equivalent is used, the dependency can be reduced.**

$$\min_{\mathbf{u}(k)} J = \frac{1}{2}\sum_{k=0}^{N}[\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k)+\mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)] \Leftrightarrow \min_{\mathbf{u}(k)} J_c = \frac{1}{2}\int_0^{N\Delta t}[\mathbf{x}^T\mathbf{Q}_{c1}\mathbf{x}+\mathbf{u}^T\mathbf{Q}_{c2}\mathbf{u}]d\tau$$

$$J_c = \frac{1}{2}\sum_{k=0}^{N-1}\int_{k\Delta t}^{(k+1)\Delta t}[\mathbf{x}^T\mathbf{Q}_{c1}\mathbf{x}+\mathbf{u}^T\mathbf{Q}_{c2}\mathbf{u}]d\tau = \frac{1}{2}\sum_{k=0}^{N-1}\begin{bmatrix}\mathbf{x}^T(k) & \mathbf{u}^T(k)\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{11} & \mathbf{Q}_{12}\\ \mathbf{Q}_{21} & \mathbf{Q}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{u}(k)\end{bmatrix}$$

where $$\begin{bmatrix}\mathbf{Q}_{11} & \mathbf{Q}_{12}\\ \mathbf{Q}_{21} & \mathbf{Q}_{22}\end{bmatrix} = \int_0^{\Delta t}\begin{bmatrix}\mathbf{\Phi}^T(\tau) & \mathbf{0}\\ \mathbf{\Gamma}^T(\tau) & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{c1} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_{c2}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}(\tau) & \mathbf{\Gamma}(\tau)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}d\tau$$

- **Van Loan (1978)**

$$\begin{bmatrix}\mathbf{Q}_{11} & \mathbf{Q}_{12}\\ \mathbf{Q}_{21} & \mathbf{Q}_{22}\end{bmatrix} = \mathbf{\Phi}_{22}^T\mathbf{\Phi}_{12} \quad \text{where } \mathbf{\Phi}_{12} = \begin{bmatrix}\mathbf{Q}_{c1} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_{c2}\end{bmatrix}, \text{ and } \mathbf{\Phi}_{22} = \begin{bmatrix}\mathbf{\Phi} & \mathbf{\Gamma}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}$$

- **Computation of the continuous cost from discrete samples of the states and control is useful for comparing digital controllers of a system with different sample rates.**

Optimal Estimation

- **Least square estimation**
  - Linear static process: y=Hx+v (v: measurement error)
  - Least squares solution

$$J=\frac{1}{2}\mathbf{v}^T\mathbf{v}=\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^T(\mathbf{y}-\mathbf{H}\mathbf{x})\Rightarrow\frac{\partial J}{\partial \mathbf{x}}=(\mathbf{y}-\mathbf{H}\mathbf{x})^T(-\mathbf{H})$$

$$\Rightarrow \mathbf{H}^T\mathbf{y}=\mathbf{H}^T\mathbf{H}\mathbf{x}\Rightarrow\hat{\mathbf{x}}=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$$

- Difference between the estimate and the actual value

$$\hat{\mathbf{x}}-\mathbf{x}=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T(\mathbf{H}\mathbf{x}+\mathbf{v})-\mathbf{x}=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{v}$$

- If v has zero mean, the error has zero mean. (Unbiased estimate)
- Covariance of the estimate error

$$\mathbf{P}=E\{(\hat{\mathbf{x}}-\mathbf{x})(\hat{\mathbf{x}}-\mathbf{x})^T\}=E\{(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{v}\mathbf{v}^T\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\}$$

$$=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T E\{\mathbf{v}\mathbf{v}^T\}\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}$$

  - If v are uncorrelated with one another, and all the element of v have the same uncertainty,

$$E\{\mathbf{v}\mathbf{v}^T\}=\mathbf{R}=\sigma^2\mathbf{I}\quad\Rightarrow\quad\mathbf{P}=(\mathbf{H}^T\mathbf{H})^{-1}\sigma^2$$

- **Weighted least squares**

$$J = \frac{1}{2}\mathbf{v}^T\mathbf{W}\mathbf{v} = \frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^T\mathbf{W}(\mathbf{y}-\mathbf{H}\mathbf{x}) \Rightarrow \frac{\partial J}{\partial \mathbf{x}} = (\mathbf{y}-\mathbf{H}\mathbf{x})^T\mathbf{W}(-\mathbf{H})$$

$$\Rightarrow \mathbf{H}^T\mathbf{W}\mathbf{y} = \mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{x} \Rightarrow \hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}$$

- **Covariance of the estimate error**

$$\mathbf{P} = E\{(\hat{\mathbf{x}}-\mathbf{x})(\hat{\mathbf{x}}-\mathbf{x})^T\} = E\{(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{v}\mathbf{v}^T\mathbf{W}\mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\}$$

$$= (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}E\{\mathbf{v}\mathbf{v}^T\}\mathbf{W}\mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}$$

- **Best linear unbiased estimate**
  - **A logical choice for W is to let it be inversely proportional to R.**
  - **Need to have a priori mean square error ($\mathbf{W}=\mathbf{R}^{-1}$)**

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}$$

  - **Covariance**

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

- Recursive least squares
  - **Problem (subscript *o*: old data, *n*: newly acquired data)**

$$\begin{bmatrix}\mathbf{y}_o\\ \mathbf{y}_n\end{bmatrix}=\begin{bmatrix}\mathbf{H}_o\\ \mathbf{H}_n\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{v}_o\\ \mathbf{v}_n\end{bmatrix}$$

  - **Best estimate of x: $\hat{\mathbf{x}}$**

$$\begin{bmatrix}\mathbf{H}_o\\ \mathbf{H}_n\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_o^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}_n^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{H}_o\\ \mathbf{H}_n\end{bmatrix}\hat{\mathbf{x}}=\begin{bmatrix}\mathbf{H}_o\\ \mathbf{H}_n\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_o^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}_n^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{y}_o\\ \mathbf{y}_n\end{bmatrix}$$

  - **Best estimate based on only old data**

$$\hat{\mathbf{x}}_n=\hat{\mathbf{x}}_o+\delta\hat{\mathbf{x}}$$

$$[\mathbf{H}_o^T\mathbf{R}_o^{-1}\mathbf{H}_o]\hat{\mathbf{x}}_o=\mathbf{H}_o^T\mathbf{R}_o^{-1}\mathbf{y}_o \qquad \mathbf{P}_o=(\mathbf{H}_o^T\mathbf{R}_o^{-1}\mathbf{H}_o)^{-1}$$

  - **Correction using new data**

$$[\mathbf{H}_n^T\mathbf{R}_n^{-1}\mathbf{H}_n]\hat{\mathbf{x}}_o+[\mathbf{H}_o^T\mathbf{R}_o^{-1}\mathbf{H}_o+\mathbf{H}_n^T\mathbf{R}_n^{-1}\mathbf{H}_n]\delta\hat{\mathbf{x}}=\mathbf{H}_n^T\mathbf{R}_n^{-1}\mathbf{y}_n$$

$$\delta\hat{\mathbf{x}}=[\mathbf{H}_o^T\mathbf{R}_o^{-1}\mathbf{H}_o+\mathbf{H}_n^T\mathbf{R}_n^{-1}\mathbf{H}_n]^{-1}\mathbf{H}_n^T\mathbf{R}_n^{-1}(\mathbf{y}_n-\mathbf{H}_n\hat{\mathbf{x}}_o)$$

$$\delta\hat{\mathbf{x}}=\mathbf{P}_n\mathbf{H}_n^T\mathbf{R}_n^{-1}(\mathbf{y}_n-\mathbf{H}_n\hat{\mathbf{x}}_o) \qquad \mathbf{P}_n=(\mathbf{P}_o^{-1}+\mathbf{H}_n^T\mathbf{R}_n^{-1}\mathbf{H}_n)^{-1}$$

- **Kalman filter**
  - **Plant:** $\mathbf{x}(k+1) = \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) + \mathbf{\Gamma}_1 \mathbf{w}(k); \quad \mathbf{y}(k) = \mathbf{Hx}(k) + \mathbf{v}(k)$
  - **Process and measurement noises: $\mathbf{w}(k)$ and $\mathbf{v}(k)$**
    - **Zero mean white noise**

$$E\{\mathbf{w}(k)\} = E\{\mathbf{v}(k)\} = \mathbf{0}$$

$$E\{\mathbf{w}(i)\mathbf{w}^T(j)\} = E\{\mathbf{v}(i)\mathbf{v}^T(j)\} = \mathbf{0} \quad (\text{if } i \neq j)$$

$$E\{\mathbf{w}(k)\mathbf{w}^T(k)\} = \mathbf{R_w}, \quad E\{\mathbf{v}(k)\mathbf{v}^T(k)\} = \mathbf{R_v}$$

  - **Optimal estimation ($\mathbf{M}=\mathbf{P}_o$, $\mathbf{P}(k)=\mathbf{P}_n$, $\mathbf{H}=\mathbf{H}_n$, $\mathbf{R}_v=\mathbf{R}_n$)**

$$\hat{\mathbf{x}}(k) = \overline{\mathbf{x}}(k) + \mathbf{L}(k)(\mathbf{y}(k) - \mathbf{H}\overline{\mathbf{x}}(k))$$

where $\mathbf{L}(k) = \mathbf{P}(k)\mathbf{H}^T(k)\mathbf{R_v}^{-1}$

$$\mathbf{P}(k) = [\mathbf{M}^{-1} + \mathbf{H}^T\mathbf{R_v}^{-1}\mathbf{H}]^{-1}$$

    - **Using matrix inversion lemma**

$$\mathbf{P}(k) = \mathbf{M}(k) - \mathbf{M}(k)\mathbf{H}^T(\mathbf{HM}(k)\mathbf{H}^T + \mathbf{R_v})^{-1}\mathbf{HM}(k)$$

where $\mathbf{M}(k)$ is the covariance of the state estimate before measurement.

- Covariance update

$$\bar{\mathbf{x}}(k) = \mathbf{\Phi}\hat{\mathbf{x}}(k-1) + \mathbf{\Gamma}\mathbf{u}(k-1)$$

$$\mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1) = \mathbf{\Phi}(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) + \mathbf{\Gamma}_1\mathbf{w}(k)$$

$$\mathbf{M}(k+1) = E\{(\mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1))(\mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1))^T\}$$

$$= E\{\mathbf{\Phi}(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\mathbf{\Phi}^T + \mathbf{\Gamma}_1\mathbf{w}(k)\mathbf{w}^T(k)\mathbf{\Gamma}_1^T\}$$

$$\mathbf{P}(k) = E\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\}, \quad \mathbf{R}_\mathbf{w} = E\{\mathbf{w}(k)\mathbf{w}^T(k)\}$$

$$\mathbf{M}(k+1) = \mathbf{\Phi}\mathbf{P}(k)\mathbf{\Phi}^T + \mathbf{\Gamma}_1\mathbf{R}_\mathbf{w}\mathbf{\Gamma}_1^T$$

- Kalman filter equations
  - Measurement update

$$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{P}(k)\mathbf{H}^T(k)\mathbf{R}_\mathbf{v}^{-1}(\mathbf{y}(k) - \mathbf{H}\bar{\mathbf{x}}(k))$$

$$\mathbf{P}(k) = \mathbf{M}(k) - \mathbf{M}(k)\mathbf{H}^T(\mathbf{H}\mathbf{M}(k)\mathbf{H}^T + \mathbf{R}_\mathbf{v})^{-1}\mathbf{H}\mathbf{M}(k)$$

  - Time update

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k)$$

$$\mathbf{M}(k+1) = \mathbf{\Phi}\mathbf{P}(k)\mathbf{\Phi}^T + \mathbf{\Gamma}_1\mathbf{R}_\mathbf{w}\mathbf{\Gamma}_1^T$$

  - The initial condition for state and covariance should be known.

- **Tuning parameters**
  - **Measurement noise covariance, R_v, is based on sensor accuracy.**
    - » **High R_v makes the estimate to rely less on the measurements. Thus, the measurement errors would not be reflected on the estimate too much.**
    - » **Low R_v makes the estimate to rely more on the measurements. Thus, the measurement errors changes the estimate rapidly.**
  - **Process noise covariance, R_w, is based on process nature.**
    - » **White noise assumption is a mathematical artifice for simplification.**
    - » **R_w is crudely accounting for unknown disturbances or model error.**

- **Noise matrices and discrete equivalents**

$$\mathbf{R_w} = E\{\mathbf{w}(k)\mathbf{w}^T(k)\}, \quad \mathbf{R_v} = E\{\mathbf{v}(k)\mathbf{v}^T(k)\}$$

$$E\{\mathbf{w}(\eta)\mathbf{w}^T(\tau)\} = \mathbf{R}_{\mathbf{w}psd}\delta(\eta-\tau), \quad E\{\mathbf{v}(\eta)\mathbf{v}^T(\tau)\} = \mathbf{R}_{\mathbf{v}psd}\delta(\eta-\tau)$$

  - **When ΔT is very small compared to the system time constant (τ_c),**

$$\mathbf{R_w} \cong \mathbf{R}_{\mathbf{w}psd}/\Delta T, \quad \mathbf{R_v} = \mathbf{R}_{\mathbf{v}psd}/\Delta T$$

$$\mathbf{R}_{\mathbf{w}psd} \cong 2\tau_c E\{w^2(t)\}, \quad \mathbf{R}_{\mathbf{v}psd} = 2\tau_c E\{v^2(t)\}$$

– **Linear Quadratic Gaussian (LQG) problem**

- **Estimator gain will reach steady state eventually.**
- **Substantial simplification is possible if constant gain is adopted.**
- **Assumption: noise has a Gaussian distribution**
- **Comparison with LQR: Dual of LQG**

$$\begin{array}{c} \mathbf{M}(k) = \mathbf{S}(k) - \mathbf{S}(k)\mathbf{\Gamma}[\mathbf{Q}_2 + \mathbf{\Gamma}^T\mathbf{S}(k)\mathbf{\Gamma}]^{-1}\mathbf{\Gamma}^T\mathbf{S}(k) \\ \mathbf{S}(k) = \mathbf{\Phi}^T\mathbf{M}(k+1)\mathbf{\Phi} + \mathbf{Q}_1 \end{array} \Leftrightarrow \begin{array}{c} \mathbf{P}(k) = \mathbf{M}(k) - \mathbf{M}(k)\mathbf{H}^T(\mathbf{H}\mathbf{M}(k)\mathbf{H}^T + \mathbf{R}_\mathbf{v})^{-1}\mathbf{H}\mathbf{M}(k) \\ \mathbf{M}(k+1) = \mathbf{\Phi}\mathbf{P}(k)\mathbf{\Phi}^T + \mathbf{\Gamma}_1\mathbf{R}_\mathbf{w}\mathbf{\Gamma}_1^T \end{array}$$

$$\mathbf{H}_c = \begin{bmatrix} \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T}\mathbf{Q}_1 & -\mathbf{\Gamma}\mathbf{Q}_2^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T} \\ -\mathbf{\Phi}^{-T}\mathbf{Q}_1 & \mathbf{\Phi}^{-T} \end{bmatrix} \Leftrightarrow \mathbf{H}_e = \begin{bmatrix} \mathbf{\Phi}^T + \mathbf{H}^T\mathbf{R}_\mathbf{v}\mathbf{H}\mathbf{\Gamma}\mathbf{\Phi}^{-1}\mathbf{\Gamma}_1\mathbf{R}_\mathbf{w}\mathbf{\Gamma}_1^T & -\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}\mathbf{\Phi}^{-1} \\ -\mathbf{\Phi}^{-1}\mathbf{\Gamma}_1\mathbf{R}_\mathbf{w}\mathbf{\Gamma}_1^T & \mathbf{\Phi}^{-1} \end{bmatrix}$$

- **Steady-state Kalman filter gain**

$$\mathbf{S}_\infty = \mathbf{\Lambda}_I\mathbf{X}_I^{-1} \Leftrightarrow \mathbf{M}_\infty = \mathbf{\Lambda}_I\mathbf{X}_I^{-1}$$

$$\mathbf{K}_\infty = (\mathbf{Q}_2 + \mathbf{\Gamma}^T\mathbf{S}_\infty\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{S}_\infty\mathbf{\Phi} \Leftrightarrow \mathbf{L}_\infty = \mathbf{M}_\infty\mathbf{H}^T(\mathbf{H}\mathbf{M}_\infty\mathbf{H}^T + \mathbf{R}_\mathbf{v})^{-1}$$

where [X_I; Λ_I] are the eigenvectors of H_c associated with its stable eigenvalues.

– **Assumption of Gaussian noise is not necessary, but with this assumption, the LQG become maximum likelihood estimate.**

Implementation Issues

- **Selection of weighting matrices Q_1 and Q_2**
  - **The states enter the cost via the important outputs**

$$J = \frac{1}{2}\sum_{k=0}^{N}[\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)] \Rightarrow J = \frac{1}{2}\sum_{k=0}^{N}[\rho\mathbf{x}^T(k)\mathbf{H}^T\bar{\mathbf{Q}}_1\mathbf{H}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)]$$

where $\bar{\mathbf{Q}}_1$ and $\mathbf{Q}_2$ are diagonal matrices.

    - **The ρ is a tuning parameter deciding the relative importance between errors and input movements.**
  - **Bryson's rule**
    - $y_{i,\max}$ **is the maximum deviation of the output** y_i**, and** $u_{i,\max}$ **is the maximum value for the input** u_i**.**

$$\bar{\mathbf{Q}}_{1,ii} = 1/y_{i,\max}^2 \text{ and } \mathbf{Q}_{2,ii} = 1/u_{i,\max}^2$$

- **Pincer Procedure**
  - **If all the poles are inside a circle of radius $1/\alpha$ ($\alpha \geq 1$), every transient in the closed loop will decay at least as faster as $1/\alpha^k$.**

$$J_\alpha = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)]\alpha^{2k}$$

$$J_\alpha = \frac{1}{2}\sum_{k=0}^{\infty}[(\alpha^k\mathbf{x})^T\mathbf{Q}_1(\alpha^k\mathbf{x}) + (\alpha^k\mathbf{u})^T\mathbf{Q}_2(\alpha^k\mathbf{u})] = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{z}^T\mathbf{Q}_1\mathbf{z} + \mathbf{v}^T(k)\mathbf{Q}_2\mathbf{v}]\alpha^{2k}$$

where $\mathbf{z}(k)=\alpha^k\mathbf{x}(k)$, $\mathbf{v}(k)=\alpha^k\mathbf{v}(k)$.

  - **The state equation**

$$\alpha^{k+1}\mathbf{x}(k+1) = \alpha^{k+1}(\mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k)) \Rightarrow \mathbf{z}(k+1) = \alpha\mathbf{\Phi}(\alpha^k\mathbf{x}(k)) + \alpha\mathbf{\Gamma}(\alpha^k\mathbf{u}(k))$$

$$\Rightarrow \mathbf{z}(k+1) = \alpha\mathbf{\Phi z}(k) + \alpha\mathbf{\Gamma v}(k)$$

  - **State feedback control (LQR)**
    - **Find the feedback gain for system ($\alpha\Phi$, $\alpha\Gamma$)**

$$\mathbf{v} = -\mathbf{Kz} \Rightarrow \alpha^k\mathbf{u}(k) = -\mathbf{K}(\alpha^k\mathbf{x}(k)) \Rightarrow \mathbf{u}(k) = -\mathbf{Kx}(k)$$

    - **Choice of α:** $\mathbf{x}(t_s/\Delta T) \approx \mathbf{x}(0)(1/\alpha)^k \leq 0.01\mathbf{x}(0) \Rightarrow \alpha > 100^{1/k} = 100^{\Delta T/t_s}$