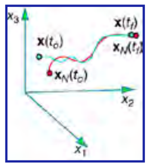
Neighboring-Optimal Control via Linear-Quadratic Feedback

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

- Linearization of nonlinear dynamic models
  - Nominal trajectory
  - Perturbations about the nominal trajectory
  - Linear, time-invariant dynamic models
  - Examples
- Linear, time-varying feedback control
- Discrete-time and sampled-data linear dynamic models
- Dynamic programming approach to optimal sampled-data control *(supplement)*



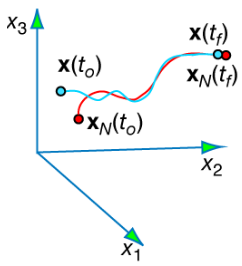
Neighboring Trajectories

- Nominal (or reference) trajectory and control history

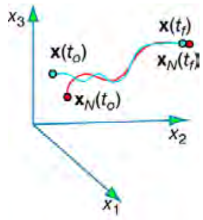
$$\left\{\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t)\right\} \quad \text{for } t \text{ in } [t_o, t_f]$$

$\mathbf{x}$: *dynamic state*
$\mathbf{u}$: *control input*
$\mathbf{w}$: *disturbance input*

- Trajectory perturbed by
  - Small initial condition variation
  - Small control variation
  - Small disturbance variation



$$\left\{\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)\right\} \quad \text{for } t \text{ in } [t_o, t_f]$$
$$= \left\{\mathbf{x}_N(t) + \Delta\mathbf{x}(t), \mathbf{u}_N(t) + \Delta\mathbf{u}(t), \mathbf{w}_N(t) + \Delta\mathbf{w}(t)\right\}$$



Both Paths Satisfy the Same Dynamic Equations

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t)], \quad \mathbf{x}_N(t_o) \text{ given}$$
$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)], \quad \mathbf{x}(t_o) \text{ given}$$

- **Neighboring-trajectory *dynamic model is the same* as the nominal dynamic model**

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t), t]$$
$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}_N(t) + \Delta\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}_N(t) + \Delta\mathbf{x}(t), \mathbf{u}_N(t) + \Delta\mathbf{u}(t), \mathbf{w}_N(t) + \Delta\mathbf{w}(t), t]$$

$$\Delta\mathbf{x}(t_o) = \mathbf{x}(t_o) - \mathbf{x}_N(t_o)$$
$$\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_N(t)$$
$$\Delta\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_N(t)$$

$$\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_N(t)$$
$$\Delta\mathbf{w}(t) = \mathbf{w}(t) - \mathbf{w}_N(t)$$

Approximate Neighboring Trajectory as a Linear Perturbation to the Nominal Trajectory

- **Nominal nonlinear dynamic equation plus linear perturbation equation**

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}_N(t) + \Delta\dot{\mathbf{x}}(t) \approx$$
$$\mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t), t] + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Delta\mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Delta\mathbf{u}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\Delta\mathbf{w}(t),$$
$$\mathbf{x}(t_o) = \mathbf{x}_N(t_o) + \Delta\mathbf{x}(t_o) \text{ given}$$

Linearized Equation Approximates Perturbation Dynamics

- Solve for the nominal and perturbation trajectories separately

$$\boxed{\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t),\mathbf{u}_N(t),\mathbf{w}_N(t),t], \quad \mathbf{x}_N(t_o) \text{ given}}$$

$$\boxed{\Delta\dot{\mathbf{x}}(t) \approx \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t)\Delta\mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(t)\Delta\mathbf{u}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(t)\Delta\mathbf{w}(t), \quad \Delta\mathbf{x}(t_o) \text{ given}}$$

- Jacobian matrices of the linear model are evaluated along the nominal trajectory

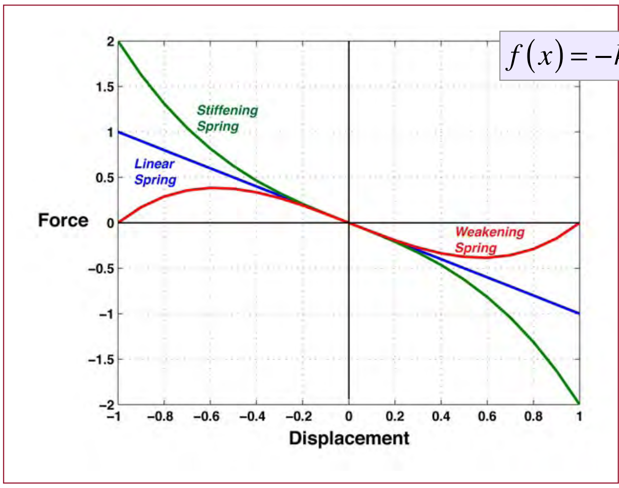
$$\boxed{\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \triangleq \mathbf{F}(t) \quad ; \quad \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \triangleq \mathbf{G}(t) \quad ; \quad \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \triangleq \mathbf{L}(t)}$$

$$\boxed{\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t) + \mathbf{L}(t)\Delta\mathbf{w}(t), \quad \Delta\mathbf{x}(t_o) \text{ given}}$$

Linearization Examples

Cubic Springs

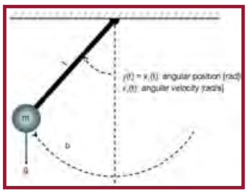
Force is a nonlinear function of deflection



$$f(x) = -k_1 x \pm k_2 x^3$$

Example: Expand $g \sin\theta$ in a power series

$$moment(\theta) \approx -k_1\theta + k_2\theta^3$$



Stiffening Cubic Spring Example

2nd-order nonlinear dynamic model

$$\dot{x}_1(t) = f_1[\mathbf{x}(t)] = x_2(t)$$
$$\dot{x}_2(t) = f_2[\mathbf{x}(t)] = -10x_1(t) - 10x_1^{\,3}(t) - x_2(t)$$

Integrate equations to produce nominal path

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \rightarrow \int_0^{t_f} \begin{bmatrix} f_{1_N}[\mathbf{x}(t)] \\ f_{2_N}[\mathbf{x}(t)] \end{bmatrix} dt \rightarrow \begin{bmatrix} x_{1_N}(t) \\ x_{2_N}(t) \end{bmatrix} \quad in \quad [0, t_f]$$

Evaluate partial derivatives of the Jacobian matrices

$$\partial f_1/\partial x_1 = 0; \quad \partial f_1/\partial x_2 = 1$$
$$\partial f_2/\partial x_1 = -10 - 30x_{1_N}^2(t); \quad \partial f_2/\partial x_2 = -1$$

$$\partial f_1/\partial u = 0; \quad \partial f_1/\partial w = 0$$
$$\partial f_2/\partial u = 0; \quad \partial f_2/\partial w = 0$$

Nominal and Perturbation Dynamic Equations

Nonlinear Equation

$$\dot{x}_{1_N}(t) = x_{2_N}(t)$$
$$\dot{x}_{2_N}(t) = -10x_{1_N}(t) - 10x_{1_N}{}^3(t) - x_{2_N}(t)$$

Local Linearization of Nonlinear Model

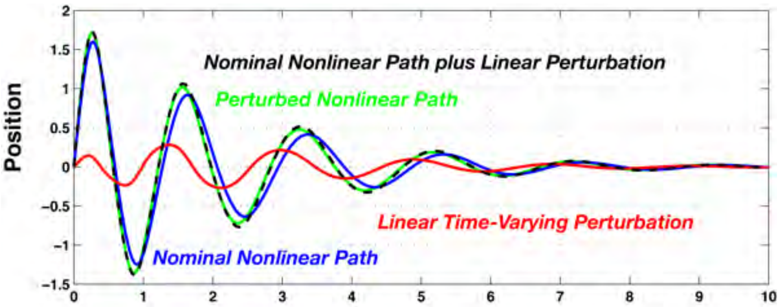
$$\begin{bmatrix} \Delta\dot{x}_1(t) \\ \Delta\dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\left(10 + 30x_{1_N}^2(t)\right) & -1 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix}$$

Initial Conditions for Nonlinear and Linear Models

$$\begin{bmatrix} x_{1_N}(0) \\ x_{2_N}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \end{bmatrix}; \qquad \begin{bmatrix} \Delta x_1(0) \\ \Delta x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

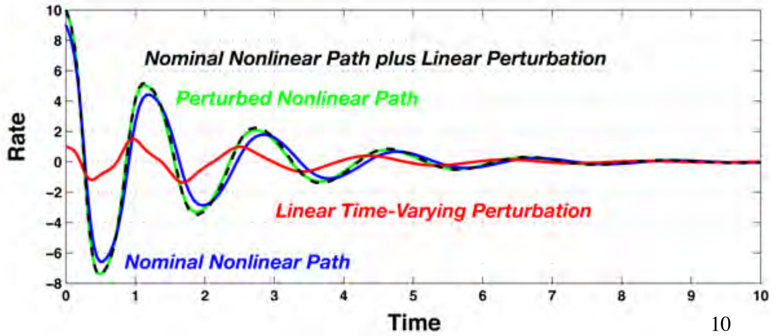
Comparison of Approximate and Exact Solutions

$\mathbf{x}_N(t)$
$\Delta\mathbf{x}(t)$
$\mathbf{x}_N(t) + \Delta\mathbf{x}(t)$
$\mathbf{x}(t)$



$\dot{\mathbf{x}}_N(t)$
$\Delta\dot{\mathbf{x}}(t)$
$\dot{\mathbf{x}}_N(t) + \Delta\dot{\mathbf{x}}(t)$
$\dot{\mathbf{x}}(t)$

$x_{2_N}(0) = 9$
$\Delta x_2(0) = 1$
$x_{2_N}(t) + \Delta x_2(t) = 10$
$x_2(t) = 10$



Euler-Lagrange Equations for Minimizing Variational Cost Function

Expand Optimal Control Function

Expand optimized cost function to second degree

$$J\left\{\left[\mathbf{x}^*(t_o)+\Delta\mathbf{x}(t_o)\right],\left[\mathbf{x}^*(t_f)+\Delta\mathbf{x}(t_f)\right]\right\} \simeq$$
$$J^*\left[\mathbf{x}^*(t_o),\mathbf{x}^*(t_f)\right]+\cancel{\Delta J\left[\Delta\mathbf{x}(t_o),\Delta\mathbf{x}(t_f)\right]}+\Delta^2 J\left[\Delta\mathbf{x}(t_o),\Delta\mathbf{x}(t_f)\right]$$

$$= J^*\left[\mathbf{x}^*(t_o),\mathbf{x}^*(t_f)\right]+\Delta^2 J\left[\Delta\mathbf{x}(t_o),\Delta\mathbf{x}(t_f)\right]$$

because **First Variation**, $\Delta J\left[\Delta\mathbf{x}(t_o),\Delta\mathbf{x}(t_f)\right]=0$

Nominal optimized cost, plus nonlinear dynamic constraint

$$J^*\left[\mathbf{x}^*(t_o),\mathbf{x}^*(t_f)\right]=\phi\left[\mathbf{x}^*(t_f)\right]+\int_{t_o}^{t_f} L\left[\mathbf{x}^*(t),\mathbf{u}^*(t)\right]dt$$

subject to nonlinear dynamic equation

$$\dot{\mathbf{x}}^*(t)=\mathbf{f}\left[\mathbf{x}^*(t),\mathbf{u}^*(t)\right],\ \mathbf{x}(t_o)=\mathbf{x}_o$$

2^{nd} Variation of the Cost Function

Objective: Given optimal nominal solution, minimize 2^{nd}-variational cost subject to linear dynamic constraint

$$\min_{\Delta\mathbf{u}} \Delta^2 J = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\phi_{\mathbf{xx}}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}L_{\mathbf{xx}}(t) & L_{\mathbf{xu}}(t)\\ L_{\mathbf{ux}}(t) & L_{\mathbf{uu}}(t)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

subject to perturbation dynamics

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t),\ \Delta\mathbf{x}(t_o) = \Delta\mathbf{x}_o$$

Cost weighting matrices expressed as

$$\mathbf{P}(t_f) \triangleq \phi_{\mathbf{xx}}(t_f) = \frac{\partial^2\phi}{\partial\mathbf{x}^2}(t_f)$$

$$\begin{bmatrix}\mathbf{Q}(t) & \mathbf{M}(t)\\ \mathbf{M}^T(t) & \mathbf{R}(t)\end{bmatrix} \triangleq \begin{bmatrix}L_{\mathbf{xx}}(t) & L_{\mathbf{xu}}(t)\\ L_{\mathbf{ux}}(t) & L_{\mathbf{uu}}(t)\end{bmatrix}$$

$$\dim\left[\mathbf{P}(t_f)\right] = \dim\left[\mathbf{Q}(t)\right] = n\times n$$
$$\dim\left[\mathbf{R}(t)\right] = m\times m$$
$$\dim\left[\mathbf{M}(t)\right] = n\times m$$

2^{nd} Variational Hamiltonian

Variational cost function

$$\Delta^2 J = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q}(t) & \mathbf{M}(t)\\ \mathbf{M}^T(t) & \mathbf{R}(t)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

$$= \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\left[\Delta\mathbf{x}^T(t)\mathbf{Q}(t)\Delta\mathbf{x}(t) + 2\Delta\mathbf{x}^T(t)\mathbf{M}(t)\Delta\mathbf{u}(t) + \Delta\mathbf{u}^T(t)\mathbf{R}(t)\Delta\mathbf{u}(t)\right]dt\right\}$$

Variational Lagrangian plus adjoined dynamic constraint

$$H\left[\Delta\mathbf{x}(t),\Delta\mathbf{u}(t),\Delta\boldsymbol{\lambda}(t)\right] = L\left[\Delta\mathbf{x}(t),\Delta\mathbf{u}(t)\right] + \Delta\boldsymbol{\lambda}^T(t)\mathbf{f}\left[\Delta\mathbf{x}(t),\Delta\mathbf{u}(t)\right]$$
$$= \frac{1}{2}\left[\Delta\mathbf{x}^T(t)\mathbf{Q}(t)\Delta\mathbf{x}(t) + 2\Delta\mathbf{x}^T(t)\mathbf{M}(t)\Delta\mathbf{u}(t) + \Delta\mathbf{u}^T(t)\mathbf{R}(t)\Delta\mathbf{u}(t)\right]$$
$$+ \Delta\boldsymbol{\lambda}^T(t)\left[\mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t)\right]$$

2nd Variational Euler-Lagrange Equations

Terminal condition, solution for adjoint vector, and optimality condition

$$\boxed{\Delta\boldsymbol{\lambda}(t_f) = \phi_{\mathbf{xx}}(t_f)\Delta\mathbf{x}(t_f) = \mathbf{P}(t_f)\Delta\mathbf{x}(t_f)}$$

$$\boxed{\Delta\dot{\boldsymbol{\lambda}}(t) = -\left\{\frac{\partial H\left[\Delta\mathbf{x}(t),\Delta\mathbf{u}(t),\Delta\boldsymbol{\lambda}(t)\right]}{\partial\Delta\mathbf{x}}\right\}^T = -\mathbf{Q}(t)\Delta\mathbf{x}(t) - \mathbf{M}(t)\Delta\mathbf{u}(t) - \mathbf{F}^T(t)\Delta\boldsymbol{\lambda}(t)}$$

$$\boxed{\left\{\frac{\partial H\left[\Delta\mathbf{x}(t),\Delta\mathbf{u}(t),\Delta\boldsymbol{\lambda}(t)\right]}{\partial\Delta\mathbf{u}}\right\}^T = \mathbf{M}^T(t)\Delta\mathbf{x}(t) + \mathbf{R}(t)\Delta\mathbf{u}(t) - \mathbf{G}^T(t)\Delta\boldsymbol{\lambda}(t) = \mathbf{0}}$$

Two-Point Boundary-Value Problem

State Equation

$$\boxed{\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t)}$$

$$\boxed{\Delta\mathbf{x}(t_o) = \Delta\mathbf{x}_o}$$

Adjoint Vector Equation

$$\boxed{\Delta\dot{\boldsymbol{\lambda}}(t) = -\mathbf{Q}(t)\Delta\mathbf{x}(t) - \mathbf{M}(t)\Delta\mathbf{u}(t) - \mathbf{F}^T(t)\Delta\boldsymbol{\lambda}(t)}$$

$$\boxed{\Delta\boldsymbol{\lambda}(t_f) = \mathbf{P}(t_f)\Delta\mathbf{x}(t_f)}$$

Use Control Law to Solve the Two-Point Boundary-Value Problem

From $H_u = 0$

$$\boxed{\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}(t)\left[\mathbf{M}^T(t)\Delta\mathbf{x}(t) + \mathbf{G}^T(t)\Delta\boldsymbol{\lambda}(t)\right]}$$

Control law that feeds back <u>state and adjoint vectors</u>

Substitute for control in system and adjoint equations

$$\begin{bmatrix} \Delta\dot{\mathbf{x}}(t) \\ \Delta\dot{\boldsymbol{\lambda}}(t) \end{bmatrix} = \begin{Bmatrix} \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] & -\mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t) \\ \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] & -\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \end{Bmatrix} \begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\boldsymbol{\lambda}(t) \end{bmatrix}$$

Adjoint relationship at end point

$$\begin{bmatrix} \Delta\mathbf{x}(t_o) \\ \Delta\boldsymbol{\lambda}(t_f) \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{x}_o \\ \mathbf{P}_f\Delta\mathbf{x}_f \end{bmatrix}$$

Perturbation state vector
Perturbation adjoint vector

Use Control Law to Solve the Two-Point Boundary-Value Problem

Assume the adjoint relationship between state and control applies over the entire interval

$$\boxed{\Delta\boldsymbol{\lambda}(t) = \mathbf{P}(t)\Delta\mathbf{x}(t)}$$

Control law feeds back <u>state alone</u>

$$\begin{aligned} \Delta\mathbf{u}(t) &= -\mathbf{R}^{-1}(t)\left[\mathbf{M}^T(t)\Delta\mathbf{x}(t) + \mathbf{G}^T(t)\mathbf{P}(t)\Delta\mathbf{x}(t)\right] \\ &= -\mathbf{R}^{-1}(t)\left[\mathbf{M}^T(t) + \mathbf{G}^T(t)\mathbf{P}(t)\right]\Delta\mathbf{x}(t) \\ &\triangleq -\mathbf{C}(t)\Delta\mathbf{x}(t) \end{aligned}$$

$\dim(\mathbf{C}) = m \times n$

Linear-Quadratic (LQ) Optimal Control Gain Matrix

$$\boxed{\Delta\mathbf{u}(t) = -\mathbf{C}(t)\Delta\mathbf{x}(t)}$$

Optimal feedback gain matrix

$$\boxed{\mathbf{C}(t) \quad = \mathbf{R}^{-1}(t)\left[\mathbf{G}^T(t)\mathbf{P}(t) + \mathbf{M}^T(t)\right]}$$

- Properties of feedback gain matrix
  - Full state feedback *(m x n)*
  - Time-varying matrix
- R, G, and M given
  - Control weighting matrix, R
  - State-control weighting matrix, M
  - Control effect matrix, G
- P(*t*) remains to be determined

Solution for the Adjoining Matrix, P(*t*)

Time-derivative of adjoint vector

$$\boxed{\Delta\dot{\boldsymbol{\lambda}}(t) = \dot{\mathbf{P}}(t)\Delta\mathbf{x}(t) + \mathbf{P}(t)\Delta\dot{\mathbf{x}}(t)}$$

Rearrange

$$\boxed{\dot{\mathbf{P}}(t)\Delta\mathbf{x}(t) = \Delta\dot{\boldsymbol{\lambda}}(t) - \mathbf{P}(t)\Delta\dot{\mathbf{x}}(t)}$$

Recall coupled state/adjoint equation

$$\begin{bmatrix} \Delta\dot{\mathbf{x}}(t) \\ \Delta\dot{\boldsymbol{\lambda}}(t) \end{bmatrix} = \left\{ \begin{matrix} \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] & -\mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t) \\ \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] & -\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \end{matrix} \right\} \begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\boldsymbol{\lambda}(t) \end{bmatrix}$$

Substitute in adjoint matrix equation

$$\boxed{\begin{gathered} \dot{\mathbf{P}}(t)\Delta\mathbf{x}(t) = \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]\Delta\mathbf{x}(t) - \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \Delta\boldsymbol{\lambda}(t) \\ -\mathbf{P}(t)\left\{\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]\Delta\mathbf{x}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\Delta\boldsymbol{\lambda}(t)\right\} \end{gathered}}$$

Solution for the Adjoining Matrix, P(*t*)

Substitute for adjoint vector

$$\boxed{\Delta\boldsymbol{\lambda}(t) = \mathbf{P}(t)\Delta\mathbf{x}(t)}$$

$$\begin{aligned}
\dot{\mathbf{P}}(t)\Delta\mathbf{x}(t) &= \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]\Delta\mathbf{x}(t) \\
&-\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \mathbf{P}(t)\Delta\mathbf{x}(t) \\
&-\mathbf{P}(t)\left\{\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]\Delta\mathbf{x}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)\Delta\mathbf{x}(t)\right\}
\end{aligned}$$

... and eliminate state vector

Matrix Riccati Equation for P(*t*)

The result is a nonlinear, ordinary differential equation for P(*t*), with terminal boundary conditions

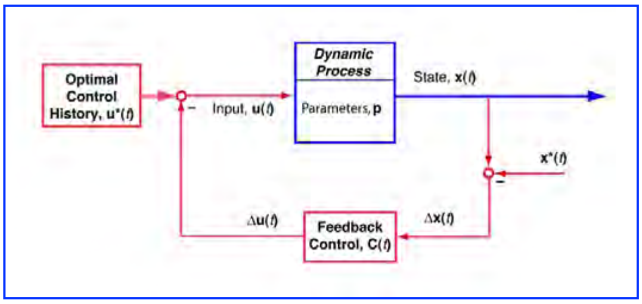
$$\begin{aligned}
\dot{\mathbf{P}}(t) &= \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] - \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \mathbf{P}(t) \\
&-\mathbf{P}(t)\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right] + \mathbf{P}(t)\mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t) \\
\mathbf{P}(t_f) &= \phi_{\mathbf{xx}}(t_f)
\end{aligned}$$

Time-varying or time-invariant?

Characteristics of the Adjoining (Riccati) Matrix, **P**(t)

- **P**(t_f) is symmetric, n x n, and typically positive semi-definite
- Matrix Riccati equation is symmetric
- Therefore, **P**(t) is symmetric and positive semi-definite throughout
- Once **P**(t) has been determined, optimal feedback control gain matrix, **C**(t) can be calculated

Neighboring-Optimal (LQ) Feedback Control Law



Full state is fed back to all available controls

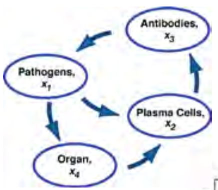
$$\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}(t)\left[\mathbf{M}^T(t) + \mathbf{G}^T(t)\mathbf{P}(t)\right]\Delta\mathbf{x}(t) = -\mathbf{C}(t)\Delta\mathbf{x}(t)$$

Nominal control history plus feedback correction

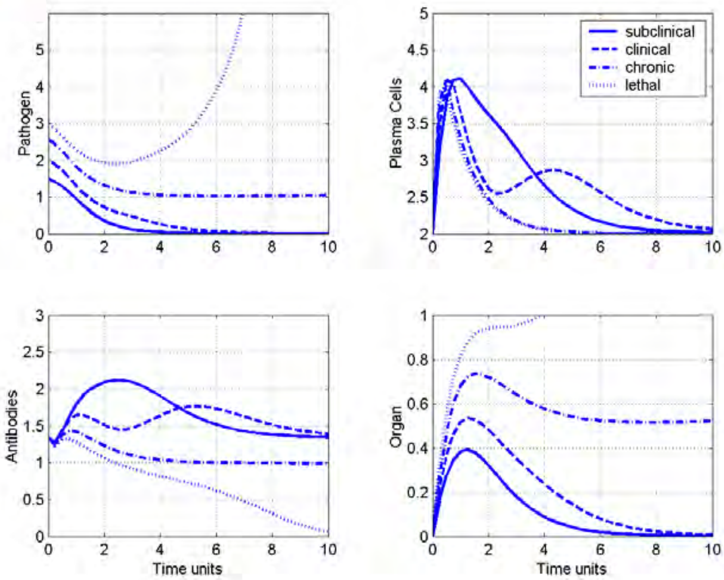
$$\mathbf{u}(t) = \mathbf{u}^*(t) - \mathbf{C}(t)\Delta\mathbf{x}(t) = \mathbf{u}^*(t) - \mathbf{C}(t)\left[\mathbf{x}(t) - \mathbf{x}^*(t)\right]$$

Example of Neighboring-Optimal Control: Improved Infection Treatment via Feedback

25



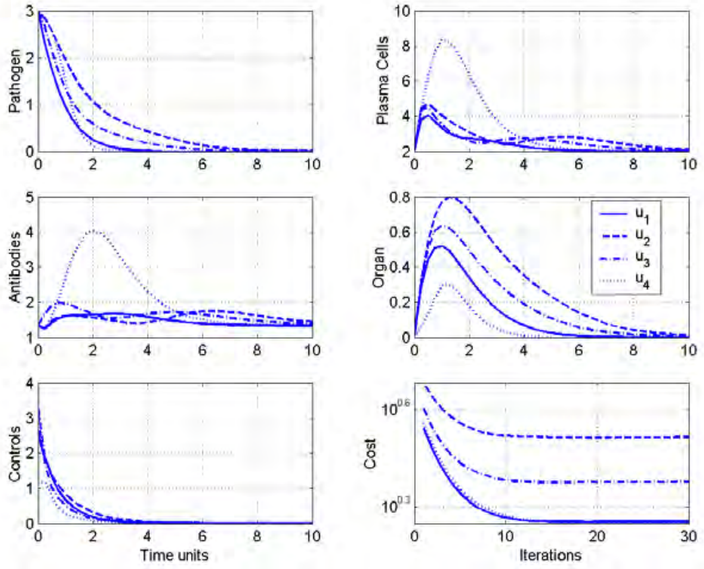
Natural Response to Pathogen Assault (No Therapy)



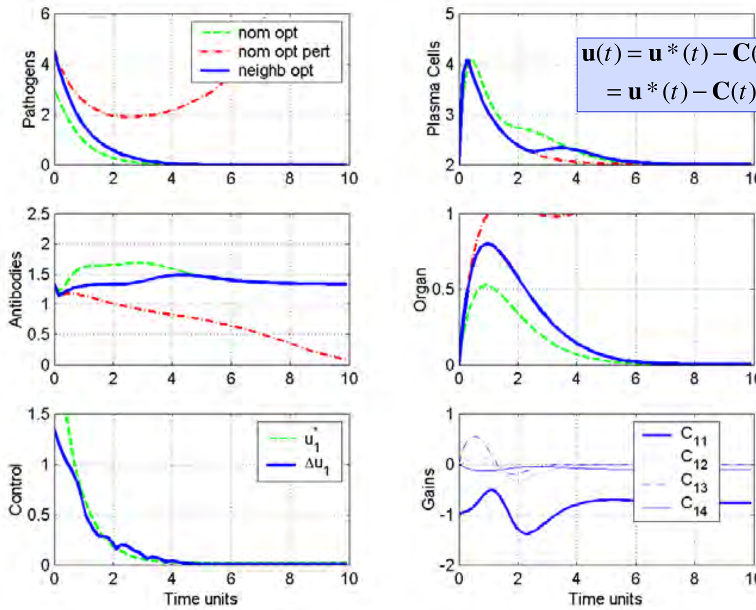
26

Tradeoff Between Rate of Killing Pathogen, Preservation of Organ Health, and Drug Use

Optimal Open-Loop Control



50% Increased Initial Infection and Scalar Neighboring-Optimal Control (*u*1)



$$\mathbf{u}(t) = \mathbf{u}^*(t) - \mathbf{C}(t)\Delta\mathbf{x}(t)$$
$$= \mathbf{u}^*(t) - \mathbf{C}(t)[\mathbf{x}(t) - \mathbf{x}^*(t)]$$

Linear-Quadratic Control of Time-Invariant Systems

Time-Varying System with Linear-Quadratic (LQ) Feedback Control

Continuous-time linear dynamic system

$$\boxed{\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t)}$$

LQ optimal control law

$$\boxed{\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}(t)\left[\mathbf{M}^T(t) + \mathbf{G}^T(t)\mathbf{P}(t)\right]\Delta\mathbf{x}(t) \triangleq -\mathbf{C}(t)\Delta\mathbf{x}(t)}$$

Linear dynamic system with LQ feedback control

$$\boxed{\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t)}$$

$$\boxed{\begin{aligned} &= \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\left[-\mathbf{C}(t)\Delta\mathbf{x}(t)\right] \\ &= \left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{C}(t)\right]\Delta\mathbf{x}(t) \end{aligned}}$$

Time-Invariant Linear System with Linear-Quadratic (LQ) Feedback Control

LTI dynamic system

$$\boxed{\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t)}$$

Time-invariant cost function

$$\boxed{\Delta^2 J = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{M}\\ \mathbf{M}^T & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}}$$

Riccati ordinary differential equation

$$\dot{\mathbf{P}}(t) = \left[-\mathbf{Q} + \mathbf{M}\mathbf{R}^{-1}\mathbf{M}^T\right] - \left[\mathbf{F} - \mathbf{G}\mathbf{R}^{-1}\mathbf{M}^T\right]^T\mathbf{P}(t) - \mathbf{P}(t)\left[\mathbf{F} - \mathbf{G}\mathbf{R}^{-1}\mathbf{M}^T\right]$$
$$+\mathbf{P}(t)\mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}(t) \quad , \quad \mathbf{P}(t_f) = \phi_{\mathbf{xx}}(t_f)$$

Linear, Time-Invariant (LTI) System with *Time-Varying* LQ Feedback Control

Control gain matrix varies over time

$$\boxed{\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}\left[\mathbf{M}^T + \mathbf{G}^T\mathbf{P}(t)\right]\Delta\mathbf{x}(t) \triangleq -\mathbf{C}(t)\Delta\mathbf{x}(t)}$$

Linear dynamic system with <u>time-varying</u> LQ feedback control

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\left[-\mathbf{C}(t)\Delta\mathbf{x}(t)\right]$$
$$= \left[\mathbf{F} - \mathbf{G}\mathbf{C}(t)\right]\Delta\mathbf{x}(t) = \mathbf{F}_{closed-loop}(t)\Delta\mathbf{x}(t)$$

Example: LQ Optimal Control of a First-Order System

$$\Delta^2 J = \frac{1}{2} p_f \Delta x^2(t_f) + \frac{1}{2}\int_{t_o}^{t_f} \left(q\Delta x^2 + (0)\Delta x \Delta u + r\Delta u^2 \right) dt$$

$$\Delta \dot{x} = f\Delta x + g\Delta u$$

$$\dot{p}(t) = -q - 2fp(t) + \frac{g^2 p^2(t)}{r}$$

$$p(t_f) = p_f$$

$$\Delta u = -r^{-1}\left[gp(t)\right]\Delta x(t) = -\frac{gp(t)}{r}\Delta x$$

Example: LQ Optimal Control of a *Stable* First-Order System

$$f = -1; \quad g = 1$$

$$\Delta \dot{x} = -\Delta x + \Delta u; \quad x(0) = 1$$

$$q = r = 1$$

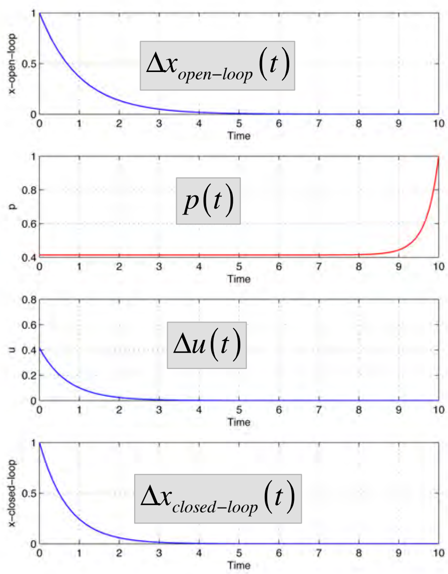
$$\dot{p}(t) = -1 + 2p(t) + p^2(t)$$

$$p(t_f) = 1$$

$$\text{Control gain} = p(t)$$

$$\Delta u = -p(t)\Delta x$$

$$\Delta \dot{x} = -\left[1 + p(t)\right]\Delta x$$



Example: LQ Optimal Control of an *Unstable* First-Order System

$f = 1; \quad g = 1$

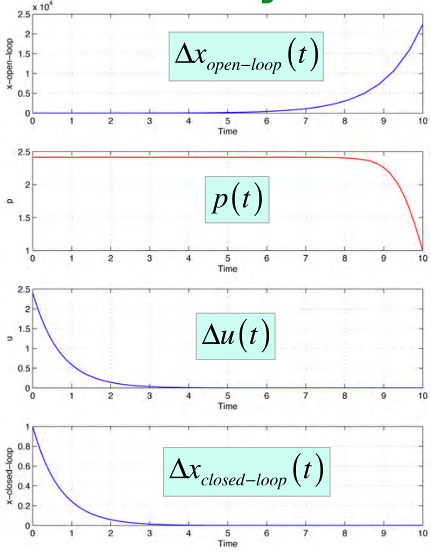
$$\Delta\dot{x} = \Delta x + \Delta u; \quad x(0) = 1$$

$$\dot{p}(t) = -1 - 2p(t) + p^2(t)$$
$$p(t_f) = 1$$

Control gain $= p(t)$

$$\Delta u = -p(t)\Delta x$$

$$\Delta\dot{x} = [1 - p(t)]\Delta x$$

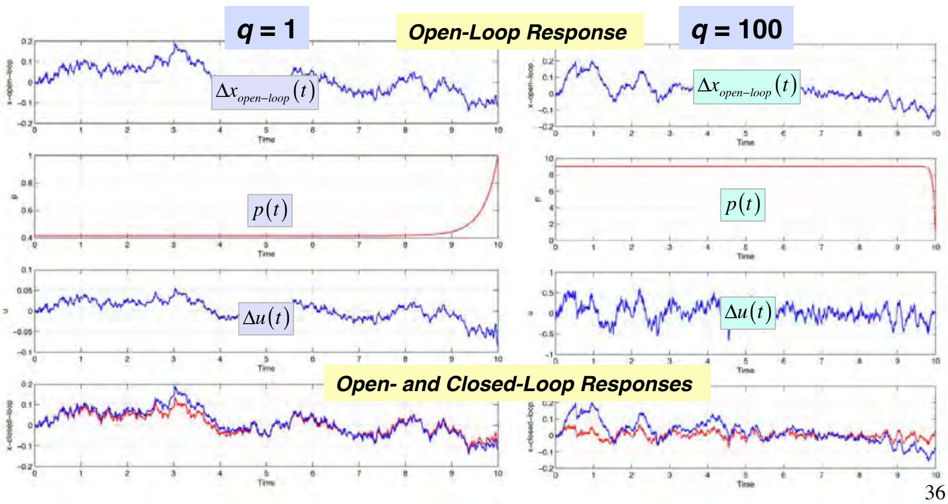


Example: LQ Optimal Control, *Stable* First-Order System, "White-Noise" Disturbance

$$\Delta\dot{x} = -\Delta x + \Delta u + \Delta w; \quad x(0) = 0$$

$$\Delta u = -p(t)\Delta x \qquad \Delta\dot{x} = -[1 + p(t)]\Delta x$$

$$\dot{p}(t) = -q + 2p(t) + p^2(t); \quad q = 1 \text{ or } 100$$
$$p(t_f) = 1$$



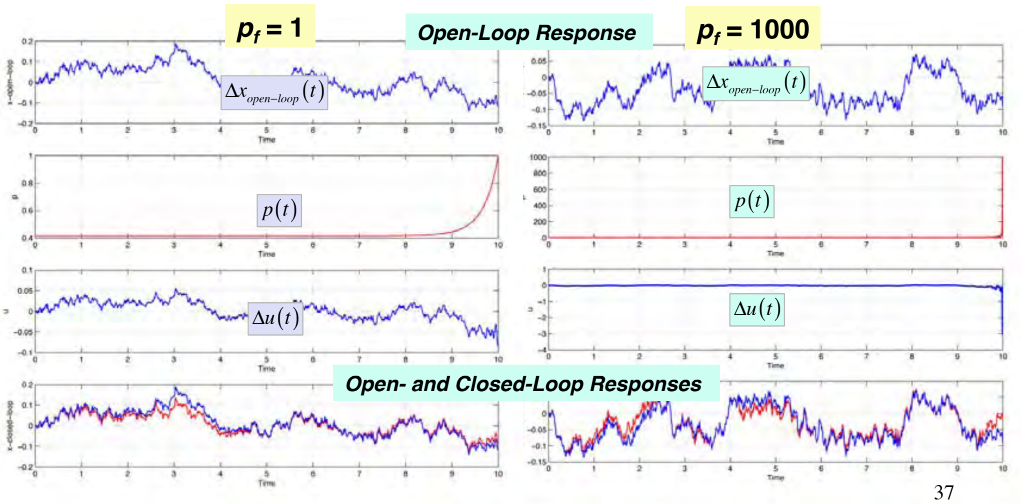
Example: LQ Optimal Control, *Stable* First-Order System, "White-Noise" Disturbance

$$\Delta\dot{x} = -\Delta x + \Delta u + \Delta w; \quad x(0) = 0$$

$$\Delta u = -p(t)\Delta x \qquad \Delta\dot{x} = -\left[1 + p(t)\right]\Delta x$$

$$\dot{p}(t) = -1 + 2p(t) + p^2(t)$$

$$p(t_f) = 1 \text{ or } 1000$$



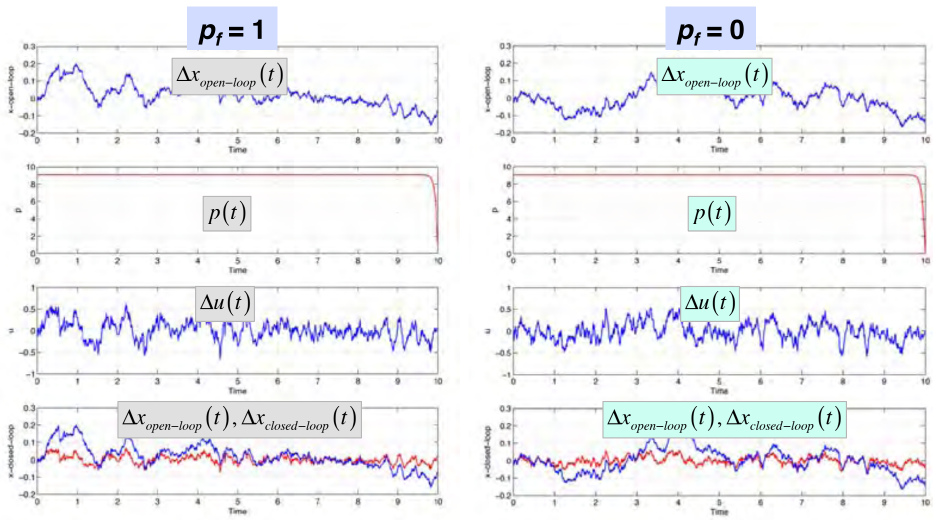
Example: LQ Optimal Control, *Stable* First-Order System, "White-Noise" Disturbance

$$\Delta\dot{x} = -\Delta x + \Delta u + \Delta w; \quad x(0) = 0$$

$$\Delta u = -p(t)\Delta x \qquad \Delta\dot{x} = -\left[1 + p(t)\right]\Delta x$$

$$\dot{p}(t) = -100 + 2p(t) + p^2(t)$$

$$p(t_f) = 1 \text{ or } 0$$



Discrete-Time and Sampled-Data Systems

Continuous-Time LTI System Model

Continuous-time ("analog") model is based on an ordinary differential equation

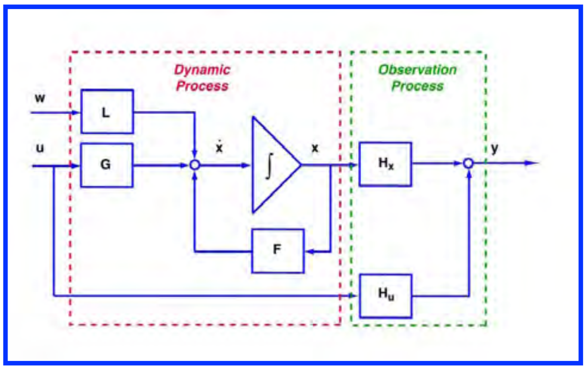
$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t) + \mathbf{L}\Delta\mathbf{w}(t)$$

$$\Delta\mathbf{x}(t_o)\ given$$

$$\Delta\mathbf{y}(t) = \mathbf{H_x}\Delta\mathbf{x}(t) + \mathbf{H_u}\Delta\mathbf{u}(t) + \mathbf{H_w}\Delta\mathbf{w}(t)$$

Dynamic Process

Observation Process



Discrete-Time LTI System Model

Discrete-time ("digital") model is based on an ordinary difference equation

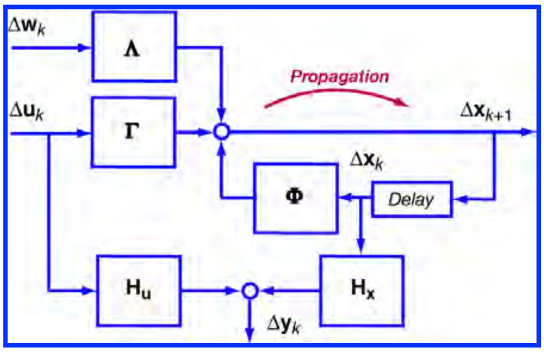
$$\Delta\mathbf{x}(t_{k+1}) = \mathbf{\Phi}\Delta\mathbf{x}(t_k) + \mathbf{\Gamma}\Delta\mathbf{u}(t_k) + \mathbf{\Lambda}\Delta\mathbf{w}(t_k)$$

$$\Delta\mathbf{x}(t_o)\ given$$

$$\Delta\mathbf{y}(t_k) = \mathbf{H_x}\Delta\mathbf{x}(t_k) + \mathbf{H_u}\Delta\mathbf{u}(t_k) + \mathbf{H_w}\Delta\mathbf{w}(t_k)$$

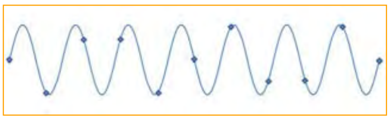
Dynamic Process

Observation Process

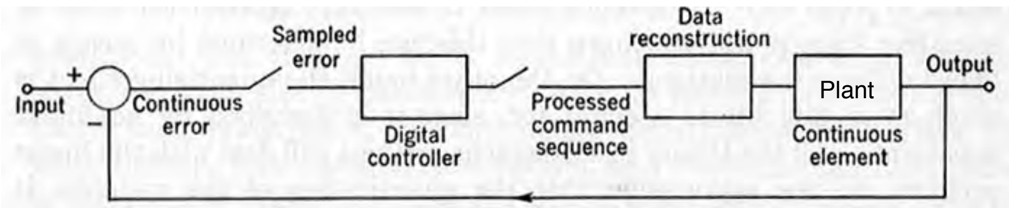


Digital Control Systems Use Sampled Data

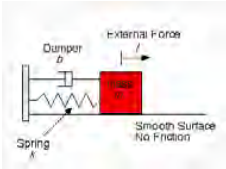
- **Periodic sequence**



$$\Delta x_k = \Delta x\left(t_k\right) = \Delta x\left(k\Delta t\right)$$



- **Sampler is an analog-to-digital (A/D) converter**
- **Reconstructor is a digital-to-analog (D/A) converter**



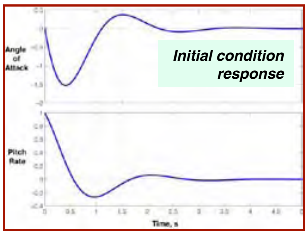
System Response to Inputs and Initial Conditions

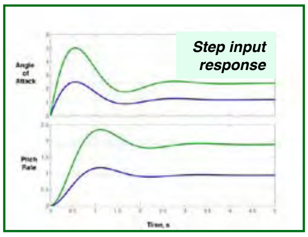
Solution of a linear dynamic model

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t) + \mathbf{L}(t)\Delta\mathbf{w}(t), \quad \Delta\mathbf{x}(t_o)\ given$$

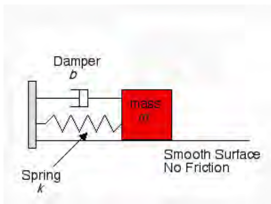
$$\Delta\mathbf{x}(t) = \Delta\mathbf{x}(t_o) + \int_{t_o}^{t} \left[\boxed{\mathbf{F}(\tau)\Delta\mathbf{x}(\tau)} + \boxed{\mathbf{G}(\tau)\Delta\mathbf{u}(\tau) + \mathbf{L}(\tau)\Delta\mathbf{w}(\tau)} \right] d\tau$$

- **... has two parts**
  - **Unforced (homogeneous) response to initial conditions**
  - **Forced response to control and disturbance inputs**





Unforced Response to Initial Conditions



Neglecting forcing functions

$$\Delta\mathbf{x}(t) = \Delta\mathbf{x}(t_o) + \int_{t_o}^{t} \left[\mathbf{F}(\tau)\Delta\mathbf{x}(\tau) \right] d\tau = \mathbf{\Phi}\left(t, t_o\right)\Delta\mathbf{x}(t_o)$$

For a linear, time-varying (LTV) system, the state transition matrix propagates the state from t_o to t by a single multiplication

For a linear, time-invariant (LTI) system

$$\Delta\mathbf{x}(t) = \Delta\mathbf{x}(t_o) + \int_{t_o}^{t} \left[\mathbf{F}\Delta\mathbf{x}(\tau) \right] d\tau$$
$$= e^{\mathbf{F}(t - t_o)}\Delta\mathbf{x}(t_o) = \mathbf{\Phi}\left(t - t_o\right)\Delta\mathbf{x}(t_o)$$

State Transition Matrix is the Matrix Exponential

$$e^{\mathbf{F}(t-t_o)} = \textit{Matrix Exponential}$$

$$= \mathbf{I} + \mathbf{F}(t-t_o) + \frac{1}{2!}\left[\mathbf{F}(t-t_o)\right]^2 + \frac{1}{3!}\left[\mathbf{F}(t-t_o)\right]^3 + \ldots$$

$$= \mathbf{\Phi}(t-t_o) = \textit{State Transition Matrix}$$

See pages 79-84 of ***Optimal Control and Estimation*** for a description of how the State Transition Matrix is calculated for an **LTV system**, i.e., if **F** is a function of time, **F**(t)

Initial-Condition Response via State Transition

Propagation of $\Delta\mathbf{x}(t_k)$ in LTI system

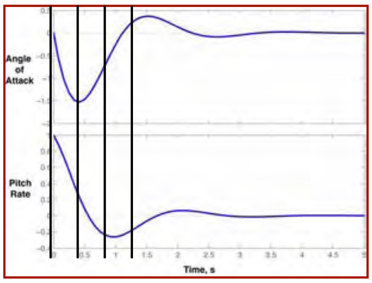
$$\Delta\mathbf{x}(t_1) = \mathbf{\Phi}(t_1 - t_o)\Delta\mathbf{x}(t_o)$$
$$\Delta\mathbf{x}(t_2) = \mathbf{\Phi}(t_2 - t_1)\Delta\mathbf{x}(t_1)$$
$$\Delta\mathbf{x}(t_3) = \mathbf{\Phi}(t_3 - t_2)\Delta\mathbf{x}(t_2)$$
$$\ldots$$

State transition matrix is constant if

$$(t_k - t_{k-1}) = \delta t = \text{ constant}$$

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{F}(\delta t) + \frac{1}{2!}\left[\mathbf{F}(\delta t)\right]^2 + \frac{1}{3!}\left[\mathbf{F}(\delta t)\right]^3 + \ldots$$

$$\Delta\mathbf{x}(t_1) = \mathbf{\Phi}(\delta t)\Delta\mathbf{x}(t_o) = \mathbf{\Phi}\Delta\mathbf{x}(t_o)$$
$$\Delta\mathbf{x}(t_2) = \mathbf{\Phi}\Delta\mathbf{x}(t_1) = \mathbf{\Phi}^2\Delta\mathbf{x}(t_o)$$
$$\Delta\mathbf{x}(t_3) = \mathbf{\Phi}\Delta\mathbf{x}(t_2) = \mathbf{\Phi}^3\Delta\mathbf{x}(t_o)$$
$$\ldots$$

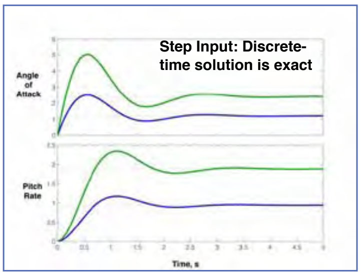


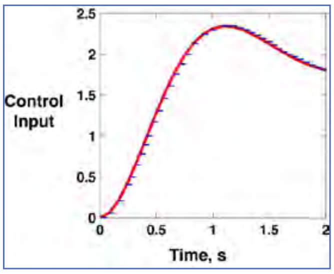
Response to Inputs

Solution of the LTI model with *piecewise-constant forcing functions*

$$\Delta\mathbf{x}(t_k) = \Delta\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \left[\mathbf{F}\Delta\mathbf{x}(\tau) + \mathbf{G}\Delta\mathbf{u}(\tau) + \mathbf{L}\Delta\mathbf{w}(\tau)\right] d\tau$$

$$\begin{aligned}\Delta\mathbf{x}(t_k) &= \boldsymbol{\Phi}(\delta t)\Delta\mathbf{x}(t_{k-1}) + \boldsymbol{\Phi}(\delta t)\int_{t_{k-1}}^{t_k}\left[e^{-\mathbf{F}(\tau - t_{k-1})}\right]d\tau\left[\mathbf{G}\Delta\mathbf{u}(t_{k-1}) + \mathbf{L}\Delta\mathbf{w}(t_{k-1})\right] \\ &= \boldsymbol{\Phi}\Delta\mathbf{x}(t_{k-1}) + \boldsymbol{\Gamma}\Delta\mathbf{u}(t_{k-1}) + \boldsymbol{\Lambda}\Delta\mathbf{w}(t_{k-1})\end{aligned}$$





Discrete-Time LTI System
Response to Step Input

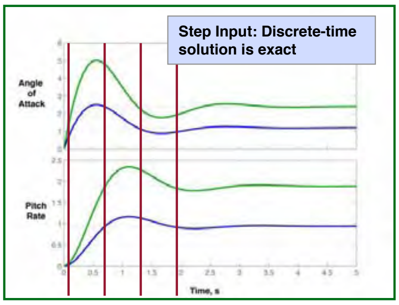
Propagation of $\Delta\mathbf{x}(t_k)$ with constant $\boldsymbol{\Phi}$, $\boldsymbol{\Gamma}$, and $\boldsymbol{\Lambda}$

$$\begin{aligned}\Delta\mathbf{x}(t_1) &= \boldsymbol{\Phi}\Delta\mathbf{x}(t_o) + \boldsymbol{\Gamma}\Delta\mathbf{u}(t_o) + \boldsymbol{\Lambda}\Delta\mathbf{w}(t_o) \\ \Delta\mathbf{x}(t_2) &= \boldsymbol{\Phi}\Delta\mathbf{x}(t_1) + \boldsymbol{\Gamma}\Delta\mathbf{u}(t_1) + \boldsymbol{\Lambda}\Delta\mathbf{w}(t_1) \\ \Delta\mathbf{x}(t_3) &= \boldsymbol{\Phi}\Delta\mathbf{x}(t_2) + \boldsymbol{\Gamma}\Delta\mathbf{u}(t_2) + \boldsymbol{\Lambda}\Delta\mathbf{w}(t_2) \\ &\ddots\end{aligned}$$

$$\boldsymbol{\Phi} = e^{\mathbf{F}\delta t}$$

$$\boldsymbol{\Gamma} = \left(e^{\mathbf{F}\delta t} - \mathbf{I}\right)\mathbf{F}^{-1}\mathbf{G}$$

$$\boldsymbol{\Lambda} = \left(e^{\mathbf{F}\delta t} - \mathbf{I}\right)\mathbf{F}^{-1}\mathbf{L}$$



Relationship Between Continuous-Time and Discrete-Time LTI Models

$$\boxed{\Phi = \mathbf{I} + \mathbf{F}(\delta t) + \frac{1}{2!}\left[\mathbf{F}(\delta t)\right]^2 + \frac{1}{3!}\left[\mathbf{F}(\delta t)\right]^3 + \ldots}$$

$$\Gamma = \left(e^{\mathbf{F}\delta t} - \mathbf{I}\right)\mathbf{F}^{-1}\mathbf{G} = \left(\mathbf{I} + \frac{1}{2!}\left[\mathbf{F}(\delta t)\right] + \frac{1}{3!}\left[\mathbf{F}(\delta t)\right]^2 + \ldots\right)\mathbf{G}\delta t$$

$$\Lambda = \left(e^{\mathbf{F}\delta t} - \mathbf{I}\right)\mathbf{F}^{-1}\mathbf{L} = \left(\mathbf{I} + \frac{1}{2!}\left[\mathbf{F}(\delta t)\right] + \frac{1}{3!}\left[\mathbf{F}(\delta t)\right]^2 + \ldots\right)\mathbf{L}\delta t$$

As time interval becomes very small, discrete-time model approaches continuous-time model

$$\Phi \xrightarrow[\delta t \to 0]{} \left(\mathbf{I} + \mathbf{F}\delta t\right)$$

$$\Gamma \xrightarrow[\delta t \to 0]{} \mathbf{G}\delta t$$

$$\Lambda \xrightarrow[\delta t \to 0]{} \mathbf{L}\delta t$$

Example: Equivalent Continuous-Time and Discrete-Time System Matrices

Continuous-time ("analog") system

$$\mathbf{F} = \begin{bmatrix} -1.2794 & -7.9856 \\ 1 & -1.2709 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -9.069 \\ 0 \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} -7.9856 \\ -1.2709 \end{bmatrix}$$

Discrete-time ("digital") system

$\delta t = 0.1s$

$$\Phi = \begin{bmatrix} 0.845 & -0.6936 \\ 0.0869 & 0.8457 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} -0.8404 \\ -0.0414 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} -0.6936 \\ -0.1543 \end{bmatrix}$$

$\delta t = 0.5s$

$$\Phi = \begin{bmatrix} 0.0823 & -1.4751 \\ 0.1847 & 0.0839 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} -2.4923 \\ -0.6429 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} -1.4751 \\ -0.9161 \end{bmatrix}$$

Time interval has a large effect on the discrete-time matrices

Sampled-Data Cost Function

<u>Sampled-Data Cost Function</u>: a Discrete-Time Cost Function that accounts for system response between sampling instants

$$\min_{\Delta\mathbf{u}(t)} \Delta^2 J = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{M}\\ \mathbf{M}^T & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

Sum integrals over short time intervals, (t_k, t_{k+1})

$$\min_{\Delta\mathbf{u}(t)} \Delta^2 J = \frac{1}{2}\Delta\mathbf{x}_{k_f}^T\mathbf{P}_{k_f}\Delta\mathbf{x}_{k_f} + \frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\int_{t_k}^{t_{k+1}}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{M}\\ \mathbf{M}^T & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

Minimize subject to sampled-data dynamic constraint

$$\Delta\mathbf{x}(t_{k+1}) = \mathbf{\Phi}(\delta t)\Delta\mathbf{x}(t_k) + \mathbf{\Gamma}(\delta t)\Delta\mathbf{u}(t_k)$$

Integrand of Sampled-Data Cost Function

Use dynamic equation ...

$$\begin{aligned}\Delta\mathbf{x}(t) &= \mathbf{\Phi}(t,t_k)\Delta\mathbf{x}(t_k) + \mathbf{\Gamma}(t,t_k)\Delta\mathbf{u}(t_k)\\ &\triangleq \mathbf{\Phi}(t,t_k)\Delta\mathbf{x}_k + \mathbf{\Gamma}(t,t_k)\Delta\mathbf{u}_k\end{aligned}$$

...to express the <u>integrand</u> in the sampling interval, (t_k, t_{k+1})

$$\frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\int_{t_k}^{t_{k+1}}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{M}\\ \mathbf{M}^T & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

Integrand of Sampled-Data Cost Function

$$\frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\int_{t_k}^{t_{k+1}}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{M}\\ \mathbf{M}^T & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

Bring state and control out of integral

Assume control is constant in sampling interval

$$=\frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\begin{bmatrix}\Delta\mathbf{x}_k^T & \Delta\mathbf{u}_k^T\end{bmatrix}\int_{t_k}^{t_{k+1}}\begin{bmatrix}\boldsymbol{\Phi}^T(t,t_k)\mathbf{Q}\boldsymbol{\Phi}(t,t_k) & \boldsymbol{\Phi}^T(t,t_k)\left[\mathbf{Q}\boldsymbol{\Gamma}(t,t_k)+\mathbf{M}\right]\\ \left[\mathbf{Q}\boldsymbol{\Gamma}(t,t_k)+\mathbf{M}\right]^T\boldsymbol{\Phi}(t,t_k) & \left[\mathbf{R}+\boldsymbol{\Gamma}^T(t,t_k)\mathbf{M}+\mathbf{M}^T\boldsymbol{\Gamma}(t,t_k)+\boldsymbol{\Gamma}^T(t,t_k)\mathbf{Q}\boldsymbol{\Gamma}(t,t_k)\right]\end{bmatrix}_k dt\begin{bmatrix}\Delta\mathbf{x}_k\\ \Delta\mathbf{u}_k\end{bmatrix}\right\}$$

Integration has been replaced by summation

$$=\frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\begin{bmatrix}\Delta\mathbf{x}_k{}^T & \Delta\mathbf{u}_k{}^T\end{bmatrix}\begin{bmatrix}\hat{\mathbf{Q}} & \hat{\mathbf{M}}\\ \hat{\mathbf{M}}^T & \hat{\mathbf{R}}\end{bmatrix}_k\begin{bmatrix}\Delta\mathbf{x}_k\\ \Delta\mathbf{u}_k\end{bmatrix}\right\}$$

Berman, Gran, *J. Aircraft,* 1974

Sampled-Data Cost Function Weighting Matrices

Assume $\mathbf{Q}$, $\mathbf{M}$, and $\mathbf{R}$ are **constant** in the integration interval

$\boldsymbol{\Phi}(t,t_k)$ and $\boldsymbol{\Gamma}(t,t_k)$ **vary** in the integration interval

$$\hat{\mathbf{Q}}=\int_{t_k}^{t_{k+1}}\boldsymbol{\Phi}^T(t,t_k)\mathbf{Q}\boldsymbol{\Phi}(t,t_k)\,dt$$

$$\hat{\mathbf{M}}=\int_{t_k}^{t_{k+1}}\boldsymbol{\Phi}^T(t,t_k)\left[\mathbf{Q}\boldsymbol{\Gamma}(t,t_k)+\mathbf{M}\right]dt$$

$$\hat{\mathbf{R}}=\int_{t_k}^{t_{k+1}}\left[\mathbf{R}+\boldsymbol{\Gamma}^T(t,t_k)\mathbf{M}+\mathbf{M}^T\boldsymbol{\Gamma}(t,t_k)+\boldsymbol{\Gamma}^T(t,t_k)\mathbf{Q}\boldsymbol{\Gamma}(t,t_k)\right]dt$$

The integrand accounts for continuous-time variations of the LTI system between sampling instants ("Inter-sample ripple")

Evaluating Sampled-Data Weighting Matrices

$\boldsymbol{\Phi}(t,t_k)$ and $\boldsymbol{\Gamma}(t,t_k)$ **vary** in the integration interval

Break interval into smaller intervals, and approximate as sum of short rectangular integration steps

$$\hat{\mathbf{Q}} = \int_0^{\Delta t} \boldsymbol{\Phi}^T(t,0)\mathbf{Q}\boldsymbol{\Phi}(t,0)\,dt$$

$$\simeq \sum_{k=1}^{100}\left[\boldsymbol{\Phi}^T(t_{k-1},0)\mathbf{Q}\boldsymbol{\Phi}(t_{k-1},0)\delta t\right], \quad \delta t = \Delta t/100, \quad t_k = k\delta t$$

$$\simeq \sum_{k=1}^{100}\left[e^{\mathbf{F}^T t_{k-1}}\mathbf{Q}e^{\mathbf{F}t_{k-1}}\delta t\right]$$

Evaluating Sampled-Data Weighting Matrices

$$\boldsymbol{\Gamma} = \left(e^{\mathbf{F}\delta t} - \mathbf{I}\right)\mathbf{F}^{-1}\mathbf{G} = \left(\mathbf{I} + \frac{1}{2!}\left[\mathbf{F}(\delta t)\right] + \frac{1}{3!}\left[\mathbf{F}(\delta t)\right]^2 + \ldots\right)\mathbf{G}\delta t$$

$\hat{\mathbf{Q}}, \hat{\mathbf{M}}$, and $\hat{\mathbf{R}}$ evaluated just once for LTI system

$$\hat{\mathbf{M}} = \int_0^{\Delta t} \boldsymbol{\Phi}^T(t,0)\left[\mathbf{Q}\boldsymbol{\Gamma}(t,0) + \mathbf{M}\right]dt$$

$$\simeq \sum_{k=1}^{100}\left\{\left[e^{\mathbf{F}^T t_{k-1}}\mathbf{Q}\left(\mathbf{I} + \frac{1}{2!}\left[\mathbf{F}t_{k-1}\right] + \frac{1}{3!}\left[\mathbf{F}t_{k-1}\right]^2 + \ldots\right)\mathbf{G}t_{k-1}\right] + \mathbf{M}\right\}\delta t$$

$$\hat{\mathbf{R}} \simeq \sum_{k=1}^{100}\left[\mathbf{R} + \boldsymbol{\Gamma}^T(t_{k-1})\mathbf{M} + \mathbf{M}^T\boldsymbol{\Gamma}(t_{k-1}) + \boldsymbol{\Gamma}^T(t_{k-1})\mathbf{Q}\boldsymbol{\Gamma}(t_{k-1})\right]\delta t$$

Sampled-Data Cost Function Weighting Always Includes State-Control Weighting

$$\hat{\mathbf{M}} = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}^T(t,t_k)\left[\mathbf{Q}\mathbf{\Gamma}(t,t_k) + \mathbf{M}\right]dt$$

$$= \int_{t_k}^{t_{k+1}} \mathbf{\Phi}^T(t,t_k)\mathbf{Q}\mathbf{\Gamma}(t,t_k)dt \quad \text{even if } \mathbf{M} = \mathbf{0}$$

Sampled-Data Lagrangian

$$L_k = \frac{1}{2}\left[\Delta\mathbf{x}_k^T\hat{\mathbf{Q}}\Delta\mathbf{x}_k + 2\Delta\mathbf{x}_k^T\hat{\mathbf{M}}\Delta\mathbf{u}_k + \Delta\mathbf{u}_k^T\hat{\mathbf{R}}\Delta\mathbf{u}_k\right]$$

Dynamic Programming Approach to Sampled-Data Optimal Control

Discrete-Time Hamilton-Jacobi-Bellman Equation

Value Function at t_o

$$V(t_o) = \varphi_{k_f} + \sum_{k=0}^{k_f - 1} L_k$$

Discrete HJB equation

$$V_k^* = -\min_{\Delta \mathbf{u}_k} \left\{ L_k + V_{k+1}^* \right\}$$

$$= -\min_{\Delta \mathbf{u}_k} H_k, \quad V^*\left[\Delta \mathbf{x}_{k_f}^*\right] = given$$

subject to

$$\Delta \mathbf{x}_{k+1} = \mathbf{\Phi} \Delta \mathbf{x}_k + \mathbf{\Gamma} \Delta \mathbf{u}_k$$

- Begin at terminal point with optimal value function
- Working backward, add minimum value function increment *(Bellman's Principle of Optimality)*

… optimal policy … whatever the initial state and initial decision …remaining decisions must constitute an optimal policy with regard to the current state

Sampled-Data Cost Function Contains Terminal and Summation Costs

Integral cost has been replaced by a summation cost
Terminal cost is the same

$$\min_{\Delta u_k} J_{sampled}$$

$$= \min_{\Delta u_k} \left\{ \frac{1}{2} \Delta \mathbf{x}_{k_f}^T \mathbf{P}_{k_f} \Delta \mathbf{x}_{k_f} + \frac{1}{2} \sum_{k=0}^{k_f - 1} \left\{ \begin{bmatrix} \Delta \mathbf{x}_k^T & \Delta \mathbf{u}_k^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{Q}} & \hat{\mathbf{M}} \\ \hat{\mathbf{M}}^T & \hat{\mathbf{R}} \end{bmatrix}_k \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{u}_k \end{bmatrix} \right\} \right\}$$

subject to

$$\Delta \mathbf{x}_{k+1} = \mathbf{\Phi} \Delta \mathbf{x}_k + \mathbf{\Gamma} \Delta \mathbf{u}_k$$

Dynamic Programming Approach to Sampled-Data LQ Control

Quadratic Value Function at t_o

$$V(t_o)=\frac{1}{2}\Delta\mathbf{x}_{k_f}^T\mathbf{P}_{k_f}\Delta\mathbf{x}_{k_f}+\frac{1}{2}\sum_{k=0}^{k_f-1}\left\{\begin{bmatrix}\Delta\mathbf{x}_k^T & \Delta\mathbf{u}_k^T\end{bmatrix}\begin{bmatrix}\hat{\mathbf{Q}} & \hat{\mathbf{M}}\\ \hat{\mathbf{M}}^T & \hat{\mathbf{R}}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}_k\\ \Delta\mathbf{u}_k\end{bmatrix}\right\}$$

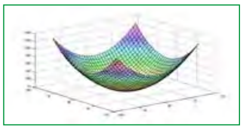
Discrete HJB equation

$$V_k{}^*=-\min_{\Delta\mathbf{u}_k}\left\{\frac{1}{2}\left[\Delta\mathbf{x}_k{}^{*T}\hat{\mathbf{Q}}\Delta\mathbf{x}_k{}^*+2\Delta\mathbf{x}_k{}^{*T}\hat{\mathbf{M}}\Delta\mathbf{u}_k+\Delta\mathbf{u}_k{}^T\hat{\mathbf{R}}\Delta\mathbf{u}_k\right]+V_{k+1}{}^*\right\}$$

$$=-\min_{\Delta\mathbf{u}_k}H_k,\quad V^*\left[\Delta\mathbf{x}_{k_f}{}^*\right]=\Delta\mathbf{x}_{k_f}{}^{*T}\mathbf{P}_{k_f}\Delta\mathbf{x}_{k_f}{}^{*T}$$

$$\text{subject to } \Delta\mathbf{x}_{k+1}=\boldsymbol{\Phi}\Delta\mathbf{x}_k+\boldsymbol{\Gamma}\Delta\mathbf{u}_k$$

Optimality Condition



Assume value function takes a quadratic form

$$V_k=\frac{1}{2}\Delta\mathbf{x}_k^T\mathbf{P}_k\Delta\mathbf{x}_k;\quad V_{k+1}=\frac{1}{2}\Delta\mathbf{x}_{k+1}{}^T\mathbf{P}_{k+1}\Delta\mathbf{x}_{k+1}$$

Optimality condition

$$\frac{\partial H_k}{\partial\Delta\mathbf{u}_k}=\left[\Delta\mathbf{x}_k^T\hat{\mathbf{M}}+\Delta\mathbf{u}_k^T\hat{\mathbf{R}}\right]+\frac{\partial V_{k+1}}{\partial\Delta\mathbf{u}_k}=\mathbf{0}$$

where

$$\frac{\partial V_{k+1}}{\partial\Delta\mathbf{u}_k}=\frac{\partial\left[\frac{1}{2}\Delta\mathbf{x}_{k+1}{}^T\mathbf{P}_{k+1}\Delta\mathbf{x}_{k+1}\right]}{\partial\Delta\mathbf{u}_k}=\left[\boldsymbol{\Phi}\Delta\mathbf{x}_k+\boldsymbol{\Gamma}\Delta\mathbf{u}_k\right]^T\mathbf{P}_{k+1}\boldsymbol{\Gamma}$$

hence

$$\left[\Delta\mathbf{x}_k^T\hat{\mathbf{M}}+\Delta\mathbf{u}_k^T\hat{\mathbf{R}}\right]+\left[\Delta\mathbf{x}_k^T\boldsymbol{\Phi}^T+\Delta\mathbf{u}_k^T\boldsymbol{\Gamma}^T\right]\mathbf{P}_{k+1}\boldsymbol{\Gamma}=\mathbf{0}$$

Minimizing Value of Control

$$\frac{\partial H_k}{\partial \Delta \mathbf{u}_k} = \Delta \mathbf{x}_k{}^T \left[\hat{\mathbf{M}} + \boldsymbol{\Phi}^T \mathbf{P}_{k+1} \boldsymbol{\Gamma} \right] + \Delta \mathbf{u}_k{}^T \left[\hat{\mathbf{R}} + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Gamma} \right] = \mathbf{0}$$

$$\Delta \mathbf{u}_k = -\left[\hat{\mathbf{R}} + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Gamma} \right]^{-1} \left[\hat{\mathbf{M}}^T + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Phi} \right] \Delta \mathbf{x}_k \triangleq -\mathbf{C}_k \Delta \mathbf{x}_k$$

Must find $\mathbf{P}_k$ in $\left(0, k_f\right)$

Use definitions of V^* and $\Delta \mathbf{u}$ in HJB equation

Solution for $\mathbf{P}_k$

$$V_k = \frac{1}{2} \Delta \mathbf{x}_k^T \mathbf{P}_k \Delta \mathbf{x}_k; \quad V_{k+1} = \frac{1}{2} \Delta \mathbf{x}_{k+1}{}^T \mathbf{P}_{k+1} \Delta \mathbf{x}_{k+1}$$

$$\Delta \mathbf{u}_k = -\left[\hat{\mathbf{R}} + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Gamma} \right]^{-1} \left[\hat{\mathbf{M}}^T + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Phi} \right] \Delta \mathbf{x}_k \triangleq -\mathbf{C}_k \Delta \mathbf{x}_k$$

Substitute for V_k, V_{k+1}, and $\Delta \mathbf{u}_k$ in discrete-time HJB equation

$$\frac{1}{2} \Delta \mathbf{x}_k{}^T \mathbf{P}_k \Delta \mathbf{x}_k =$$
$$-\min_{\Delta \mathbf{u}_k} \left\{ \frac{1}{2} \left[\Delta \mathbf{x}_k{}^{*T} \hat{\mathbf{Q}} \Delta \mathbf{x}_k{}^* + 2 \Delta \mathbf{x}_k{}^{*T} \hat{\mathbf{M}} \left(-\mathbf{C}_k \Delta \mathbf{x}_k \right) + \left(-\mathbf{C}_k \Delta \mathbf{x}_k \right)^T \hat{\mathbf{R}} \left(-\mathbf{C}_k \Delta \mathbf{x}_k \right) + \Delta \mathbf{x}_{k+1}{}^T \mathbf{P}_{k+1} \Delta \mathbf{x}_{k+1} \right] \right\}$$

Rearrange and cancel $\Delta \mathbf{x}_k$ on both sides of the equation to yield the discrete-time Riccati equation

$$\mathbf{P}_k = \hat{\mathbf{Q}} + \boldsymbol{\Phi}^T \mathbf{P}_{k+1} \boldsymbol{\Phi} - \left[\hat{\mathbf{M}}^T + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Phi} \right]^T \left[\hat{\mathbf{R}} + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Gamma} \right]^{-1} \left[\hat{\mathbf{M}}^T + \boldsymbol{\Gamma}^T \mathbf{P}_{k+1} \boldsymbol{\Phi} \right]$$

$\mathbf{P}_{k_f}$ given

Discrete-Time System with Linear-Quadratic Feedback Control

Dynamic System

$$\Delta\mathbf{x}_{k+1} = \mathbf{\Phi}\Delta\mathbf{x}_k + \mathbf{\Gamma}\Delta\mathbf{u}_k$$

Control law

$$\Delta\mathbf{u}_k = -\left[\hat{\mathbf{R}} + \mathbf{\Gamma}^T\mathbf{P}_{k+1}\mathbf{\Gamma}\right]^{-1}\left[\hat{\mathbf{M}}^T + \mathbf{\Gamma}^T\mathbf{P}_{k+1}\mathbf{\Phi}\right]\Delta\mathbf{x}_k \triangleq -\mathbf{C}_k\Delta\mathbf{x}_k$$

Dynamic system with LQ feedback control

$$\begin{aligned}\Delta\mathbf{x}_{k+1} &= \mathbf{\Phi}\Delta\mathbf{x}_k + \mathbf{\Gamma}\Delta\mathbf{u}_k \\ &= \mathbf{\Phi}\Delta\mathbf{x}_k + \mathbf{\Gamma}\left(-\mathbf{C}_k\Delta\mathbf{x}\right)_k \\ &= \left(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{C}_k\right)\Delta\mathbf{x}_k\end{aligned}$$

Example: 1st-Order System with LQ Feedback Control

1st-order discrete-time dynamic system

$$\Delta x_{k+1} = \phi\Delta x_k + \gamma\Delta u_k$$

LQ optimal control law

$$\Delta u_k = -\frac{m + \phi\gamma p_{k+1}}{r + \gamma^2 p_{k+1}}\Delta x_k \triangleq -c_k\Delta x_k$$

$$p_k = q + \phi^2 p_{k+1} - \frac{\left(m + \phi\gamma p_{k+1}\right)^2}{r + \gamma^2 p_{k+1}}, \quad p_{k_f} \text{ given}$$

Dynamic system with LQ feedback control

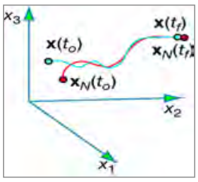
$$\begin{aligned}\Delta x_{k+1} &= \phi\Delta x_k + \gamma\Delta u_k \\ &= \phi\Delta x_k + \gamma\left(-c_k\Delta x\right)_k \\ &= \left(\phi - \gamma c_k\right)\Delta x_k\end{aligned}$$

Next Time:
Dynamic System Stability

Reading

OCE: Section 2.5

SUPPLEMENTAL MATERIAL



Example: Separate Solutions for Nominal and Perturbation Trajectories

Original nonlinear equation describes nominal dynamics

$$\dot{\mathbf{x}}_N = \begin{bmatrix}\dot{x}_{1_N}\\ \dot{x}_{2_N}\\ \dot{x}_{3_N}\end{bmatrix} = \begin{bmatrix} x_{2_N} + dw_{1_N} \\ a_2\left(x_{3_N}-x_{2_N}\right) + a_1\left(x_{3_N}-x_{1_N}\right)^2 + b_1 u_{1_N} \quad + b_2 u_{2_N} \\ c_2 x_{3_N}^3 + c_1\left(x_{1_N}+x_{2_N}\right) + b_3 x_{1_N} u_{1_N}\end{bmatrix}, \quad \begin{bmatrix}x_{1_N}\\ x_{2_N}\\ x_{3_N}\end{bmatrix} \text{ given}$$

Linear, time-varying equation describes perturbation dynamics

$$\begin{bmatrix}\Delta\dot{x}_1\\ \Delta\dot{x}_2\\ \Delta\dot{x}_3\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2a_1\left(x_{3_N}-x_{1_N}\right) & -a_2 & a_2 + 2a_1\left(x_{3_N}-x_{1_N}\right) \\ \left(c_1 + b_3 u_{1_N}\right) & c_1 & 3c_2 x_{3_N}{}^2 \end{bmatrix}\begin{bmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\end{bmatrix}$$
$$+\begin{bmatrix} 0 & 0 \\ b_1 & b_2 \\ b_3 x_{1_N} & 0\end{bmatrix}\begin{bmatrix}\Delta u_1\\ \Delta u_2\end{bmatrix} + \begin{bmatrix} d \\ 0 \\ 0\end{bmatrix}\Delta w_1 \,; \quad \begin{bmatrix}\Delta x_1(t_o)\\ \Delta x_2(t_o)\\ \Delta x_3(t_o)\end{bmatrix} \text{ given}$$

Multivariable LQ Optimal Control with Cross Weighting, M, = 0

No state/control coupling in cost function

$$\Delta^2 J = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f) + \frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix}\Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t)\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}(t)\\ \Delta\mathbf{u}(t)\end{bmatrix}dt\right\}$$

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t)$$

$$\dot{\mathbf{P}}(t) = [-\mathbf{Q}] - [\mathbf{F}]^T\mathbf{P}(t) - \mathbf{P}(t)[\mathbf{F}] + \mathbf{P}(t)\mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}(t)$$
$$\mathbf{P}\left(t_f\right) = \mathbf{P}_f$$

$$\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}\left[\mathbf{G}^T\mathbf{P}(t)\right]\Delta\mathbf{x}(t)$$

First-Order LQ Example Code

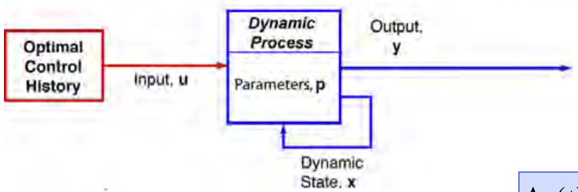
```
%    First-Order LQ Example
%    Rob Stengel
%    2/23/2011

     clear
     global tp p q tw w

     xo = 0; to = 0;      tf = 10;
     tspanx  =    [to tf];
     tw      =    [0:0.01:10];
     for k   =    1:length(tw)
         w(k)     =    randn(1);
     end
     [tx,x]  =    ode15s('First',tspanx,xo);
     pf      =    0;      q           =    100;
     tspanp  =    [tf to];
     [tp,p]  =    ode15s('FirstRiccati',tspanp,pf);
     [tc,xc] =    ode15s('FirstCL',tspanx,xo);
     u       =    interp1(tp,p,tc).*xc;
     figure
     subplot(4,1,1)
     plot(tx,x),grid,xlabel('Time'),ylabel('x-open-loop')
     subplot(4,1,2)
     plot(tp,p,'r'),grid,xlabel('Time'),ylabel('p')
     subplot(4,1,3)
     plot(tc,u),grid, xlabel('Time'),ylabel('u')
     subplot(4,1,4)
     plot(tc,xc,'r',tx,x,'b'),grid,xlabel('Time'),ylabel('x- closed-loop')
```

```
function xdot    =    First(t,x)
            global tp p tw w
            wt       =    interp1(tw,w,t,'nearest');
            xdot     =    -x + wt;

function pdot    =    FirstRiccati(t,p)
            global q
            pdot     =    -q + 2*p + p^2;

function xdot    =    FirstCL(tc,xc);
            global tp p tw w
            wt       =    interp1(tw,w,tc,'nearest');
            pt       =    interp1(tp,p,tc);
            xdot     =    -(1 + pt)*xc + wt;
```


Nominal- and Neighboring-Optimal Control of the Dynamic Model

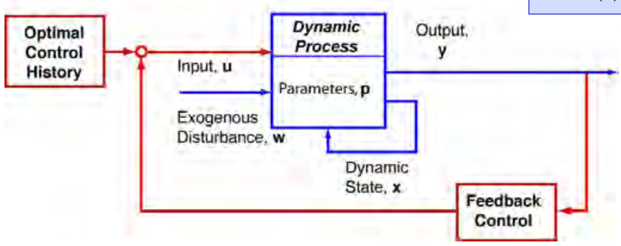
Nominal, Open-Loop Optimal Control

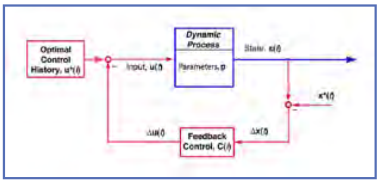
$$\mathbf{u}(t) = \mathbf{u}_{opt}(t)$$
$$\mathbf{y} = \mathbf{I}\mathbf{x}$$



Neighboring-Optimal Control

$$\Delta\mathbf{u}(t) = -\mathbf{C}(t)\left[\mathbf{x}(t) - \mathbf{x}_{opt}(t)\right]$$
$$\mathbf{u}(t) = \mathbf{u}_{opt}(t) + \Delta\mathbf{u}(t)$$





Nonlinear System with Neighboring-Optimal Feedback Control

Nonlinear dynamic system

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t)\right]$$

Neighboring-optimal control law

$$\mathbf{u}(t) = \mathbf{u}^*(t) - \mathbf{C}(t)\Delta\mathbf{x}(t) = \mathbf{u}^*(t) - \mathbf{C}(t)\left[\mathbf{x}(t) - \mathbf{x}^*(t)\right]$$

Nonlinear dynamic system with neighboring-optimal feedback control

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left\{\mathbf{x}(t), \left[\mathbf{u}^*(t) - \mathbf{C}(t)\left[\mathbf{x}(t) - \mathbf{x}^*(t)\right]\right]\right\}$$

Development of Neighboring-Optimal Therapy

- **Expand dynamic equation to first degree**

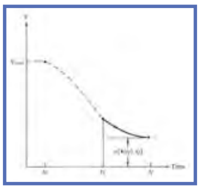
$$\mathbf{x}(t) = \mathbf{x}^*(t) + \Delta\mathbf{x}(t)$$
$$\mathbf{u}(t) = \mathbf{u}^*(t) + \Delta\mathbf{u}(t)$$

- **Compute nominal optimal control history using original nonlinear dynamic model**
- **Compute optimal perturbation control using locally linearized dynamic model**
- **Sum the two for neighboring-optimal control of the dynamic system**

$$\mathbf{u}(t) = \mathbf{u}_{opt}(t)$$

$$\Delta\mathbf{u}(t) = -\mathbf{C}(t)\left[\mathbf{x}(t) - \mathbf{x}_{opt}(t)\right]$$
$$\mathbf{u}(t) = \mathbf{u}_{opt}(t) + \Delta\mathbf{u}(t)$$

Continuous-Time LQ Optimization via Dynamic Programming



Dynamic Programming Approach to Continuous-Time LQ Control

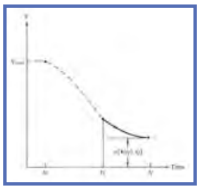
Value Function at t_o

$$V(t_o) = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)_f\Delta\mathbf{x}(t_f)$$
$$+\frac{1}{2}\left\{\int_{t_o}^{t_f}\begin{bmatrix} \Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t) \end{bmatrix}\begin{bmatrix} \mathbf{Q}(t) & \mathbf{M}(t) \\ \mathbf{M}^T(t) & \mathbf{R}(t) \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\mathbf{u}(t) \end{bmatrix}dt\right\}$$

Value Function at t_1

$$V(t_1) = \frac{1}{2}\Delta\mathbf{x}^T(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}(t_f)$$
$$-\frac{1}{2}\left\{\int_{tf}^{t_1}\begin{bmatrix} \Delta\mathbf{x}^T(t) & \Delta\mathbf{u}^T(t) \end{bmatrix}\begin{bmatrix} \mathbf{Q}(t) & \mathbf{M}(t) \\ \mathbf{M}^T(t) & \mathbf{R}(t) \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x}(t) \\ \Delta\mathbf{u}(t) \end{bmatrix}dt\right\}$$

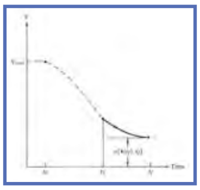
Dynamic Programming Approach to LQ Control



Time Derivative of Value Function

$$\frac{\partial V^*\left[\Delta \mathbf{x}^*(t_1)\right]}{\partial t} = -\min_{\Delta \mathbf{u}} \left\{ \begin{array}{c} \frac{1}{2}\left[\Delta\mathbf{x}^{*T}(t_1)\mathbf{Q}(t_1)\Delta\mathbf{x}^*(t_1) + 2\Delta\mathbf{x}^{*T}(t_1)\mathbf{M}(t_1)\Delta\mathbf{u}(t_1) + \Delta\mathbf{u}^T(t_1)\mathbf{R}(t_1)\Delta\mathbf{u}(t_1)\right] \\ + \frac{\partial V^*\left[\Delta\mathbf{x}^*(t_1)\right]}{\partial \Delta\mathbf{x}}\left[\mathbf{F}(t_1)\Delta\mathbf{x}^*(t_1)+\mathbf{G}(t_1)\Delta\mathbf{u}(t_1)\right] \end{array} \right\}$$

Dynamic Programming Approach to LQ Control



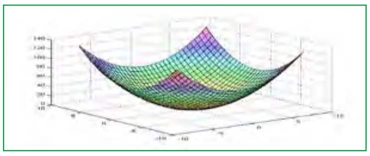
Hamiltonian

$$H\left[\Delta\mathbf{x}^*, \Delta\mathbf{u}, \frac{\partial V^*}{\partial \Delta\mathbf{x}}\right] \triangleq \frac{1}{2}\left[\Delta\mathbf{x}^{*T}\mathbf{Q}\Delta\mathbf{x}^* + 2\Delta\mathbf{x}^{*T}\mathbf{M}\Delta\mathbf{u} + \Delta\mathbf{u}^T\mathbf{R}\Delta\mathbf{u}\right] + \frac{\partial V^*\left[\Delta\mathbf{x}^*\right]}{\partial \Delta\mathbf{x}}\left[\mathbf{F}\Delta\mathbf{x}^* + \mathbf{G}\Delta\mathbf{u}\right]$$

HJB Equation

$$\frac{\partial V^*\left[\Delta\mathbf{x}^*\right]}{\partial t} = -\min_{\Delta\mathbf{u}} H\left[\Delta\mathbf{x}^*, \Delta\mathbf{u}, \frac{\partial V^*}{\partial \Delta\mathbf{x}}\right],$$

$$V^*\left[\Delta\mathbf{x}(t_f)\right] = \Delta\mathbf{x}^{*T}(t_f)\mathbf{P}(t_f)\Delta\mathbf{x}^{*T}(t_f)$$



Plausible Form for the Value Function

Quadratic Function of State Perturbation

$$V^*\left[\Delta\mathbf{x}^*(t)\right] = \Delta\mathbf{x}^{*T}(t)\mathbf{P}(t)\Delta\mathbf{x}^{*T}(t)$$

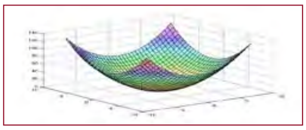
Time Derivative of the Value Function

$$\frac{\partial V^*}{\partial t} = -\frac{1}{2}\left[\Delta\mathbf{x}^{*T}(t_1)\dot{\mathbf{P}}(t_1)\Delta\mathbf{x}^*(t_1)\right]$$

Gradient of the Value Function with respect to the state

$$\frac{\partial V^*}{\partial \Delta\mathbf{x}} = \Delta\mathbf{x}^{*T}(t)\mathbf{P}(t)$$

Optimal Control Law and HJB Equation



Optimal control law

$$\frac{\partial H}{\partial \mathbf{u}} = \Delta\mathbf{x}^T\mathbf{M} + \Delta\mathbf{u}^T\mathbf{R} + \Delta\mathbf{x}^T\mathbf{P}\mathbf{G} = \mathbf{0}$$

$$\Delta\mathbf{u}(t) = -\mathbf{R}^{-1}\left(\mathbf{G}^T\mathbf{P} + \mathbf{M}^T\right)\Delta\mathbf{x}(t)$$

Incorporate Value Function Model in HJB equation

$$\Delta\mathbf{x}^T\dot{\mathbf{P}}\Delta\mathbf{x} =$$

$$\Delta\mathbf{x}^T\left\{\left[-\mathbf{Q}+\mathbf{M}\mathbf{R}^{-1}\mathbf{M}^T\right] - \left[\mathbf{F}-\mathbf{G}\mathbf{R}^{-1}\mathbf{M}^T\right]^T\mathbf{P} - \mathbf{P}\left[\mathbf{F}-\mathbf{G}\mathbf{R}^{-1}\mathbf{M}^T\right] + \mathbf{P}\mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}\right\}\Delta\mathbf{x}$$

$\Delta\mathbf{x}(t)$ can be cancelled on left and right

Matrix Riccati Equation

$$\dot{\mathbf{P}}(t) = \left[-\mathbf{Q}(t) + \mathbf{M}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]$$
$$-\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]^T \mathbf{P}(t)$$
$$-\mathbf{P}(t)\left[\mathbf{F}(t) - \mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{M}^T(t)\right]$$
$$+\mathbf{P}(t)\mathbf{G}(t)\mathbf{R}^{-1}(t)\mathbf{G}^T(t)\mathbf{P}(t)$$
$$\mathbf{P}(t_f) = \phi_{\mathbf{xx}}(t_f)$$