16.323 Principles of Optimal Control
Spring 2008

16.323 Lecture 4

HJB Equation

- DP in continuous time
- HJB Equation
- Continuous LQR

Factoids: for symmetric R

$$\frac{\partial \mathbf{u}^T R \mathbf{u}}{\partial \mathbf{u}} = 2\mathbf{u}^T R$$

$$\frac{\partial R \mathbf{u}}{\partial \mathbf{u}} = R$$

DP in Continuous Time

- Have analyzed a couple of approximate solutions to the classic control problem of minimizing:

$$\min J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\, dt$$

subject to

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{x}(t_0) &= \text{given} \\ \mathbf{m}(\mathbf{x}(t_f), t_f) &= 0 \text{ set of terminal conditions} \\ \mathbf{u}(t) &\in \mathcal{U} \text{ set of possible constraints} \end{aligned}$$

- Previous approaches discretized in time, state, and control actions
  - Useful for implementation on a computer, but now want to consider the exact solution in continuous time
  - Result will be a nonlinear partial differential equation called the **Hamilton-Jacobi-Bellman** equation (**HJB**) – a key result.

- First step: consider cost over the interval $[t, t_f]$, where $t \leq t_f$ of any control sequence $\mathbf{u}(\tau)$, $t \leq \tau \leq t_f$

$$J(\mathbf{x}(t), t, \mathbf{u}(\tau)) = h(\mathbf{x}(t_f), t_f) + \int_{t}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau$$

  - Clearly the goal is to pick $\mathbf{u}(\tau)$, $t \leq \tau \leq t_f$ to minimize this cost.

$$J^\star(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau)\in\mathcal{U} \\ t\leq\tau\leq t_f}} J(\mathbf{x}(t), t, \mathbf{u}(\tau))$$

- Approach:
  - Split time interval $[t, t_f]$ into $[t, t+\Delta t]$ and $[t+\Delta t, t_f]$, and are specifically interested in the case where $\Delta t \to 0$
  - Identify the optimal cost-to-go $J^\star(\mathbf{x}(t+\Delta t), t+\Delta t)$
  - Determine the "stage cost" in time $[t, t+\Delta t]$
  - Combine above to find best strategy from time t.
  - Manipulate result into HJB equation.

- Split:

$$J^\star(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau)\in\mathcal{U} \\ t\le\tau\le t_f}} \left\{ h(\mathbf{x}(t_f), t_f) + \int_t^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau))\, d\tau \right\}$$

$$= \min_{\substack{\mathbf{u}(\tau)\in\mathcal{U} \\ t\le\tau\le t_f}} \left\{ h(\mathbf{x}(t_f), t_f) + \int_t^{t+\Delta t} g(\mathbf{x}, \mathbf{u}, \tau)\, d\tau + \int_{t+\Delta t}^{t_f} g(\mathbf{x}, \mathbf{u}, \tau)\, d\tau \right\}$$

- Implicit here that at time $t+\Delta t$, the system will be at state $\mathbf{x}(t+\Delta t)$.
  - But from the **principle of optimality**, we can write that the optimal cost-to-go from this state is:

$$J^\star(\mathbf{x}(t+\Delta t), t+\Delta t)$$

- Thus can rewrite the cost calculation as:

$$J^\star(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau)\in\mathcal{U} \\ t\le\tau\le t+\Delta t}} \left\{ \int_t^{t+\Delta t} g(\mathbf{x}, \mathbf{u}, \tau)\, d\tau + J^\star(\mathbf{x}(t+\Delta t), t+\Delta t) \right\}$$

- Assuming $J^\star(\mathbf{x}(t+\Delta t), t+\Delta t)$ has bounded second derivatives in both arguments, can expand this cost as a Taylor series about $\mathbf{x}(t), t$

$$\begin{aligned} J^\star(\mathbf{x}(t+\Delta t), t+\Delta t) &\approx J^\star(\mathbf{x}(t), t) + \left[\frac{\partial J^\star}{\partial t}(\mathbf{x}(t), t)\right]\Delta t \\ &+ \left[\frac{\partial J^\star}{\partial \mathbf{x}}(\mathbf{x}(t), t)\right](\mathbf{x}(t+\Delta t) - \mathbf{x}(t)) \end{aligned}$$

  – Which for small Δt can be compactly written as:

$$\begin{aligned} J^\star(\mathbf{x}(t+\Delta t), t+\Delta t) &\approx J^\star(\mathbf{x}(t), t) + J_t^\star(\mathbf{x}(t), t)\Delta t \\ &+ J_{\mathbf{x}}^\star(\mathbf{x}(t), t)\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t \end{aligned}$$

- Substitute this into the cost calculation with a small Δt to get

$$\begin{aligned} J^\star(\mathbf{x}(t), t) &= \min_{\mathbf{u}(t)\in\mathcal{U}} \{g(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t + J^\star(\mathbf{x}(t), t) \\ &+ J_t^\star(\mathbf{x}(t), t)\Delta t + J_{\mathbf{x}}^\star(\mathbf{x}(t), t)\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t\} \end{aligned}$$

- Extract the terms that are independent of $\mathbf{u}(t)$ and cancel

$$0 = J_t^\star(\mathbf{x}(t), t) + \min_{\mathbf{u}(t)\in\mathcal{U}} \{g(\mathbf{x}(t), \mathbf{u}(t), t) + J_{\mathbf{x}}^\star(\mathbf{x}(t), t)\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)\}$$

  – This is a partial differential equation in $J^\star(\mathbf{x}(t), t)$ that is solved backwards in time with an initial condition that

$$J^\star(\mathbf{x}(t_f), t_f) = h(\mathbf{x}(t_f))$$

  for $\mathbf{x}(t_f)$ and t_f combinations that satisfy $m(\mathbf{x}(t_f), t_f) = 0$

HJB Equation

- For simplicity, define the **Hamiltonian**

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, J^\star_{\mathbf{x}}, t) = g(\mathbf{x}(t), \mathbf{u}(t), t) + J^\star_{\mathbf{x}}(\mathbf{x}(t), t)\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)$$

then the **HJB equation** is

$$-J^\star_t(\mathbf{x}(t), t) = \min_{\mathbf{u}(t)\in\mathcal{U}} \mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), J^\star_{\mathbf{x}}(\mathbf{x}(t), t), t)$$

  - A very powerful result, that is both a **necessary and sufficient** condition for optimality
  - But one that is hard to solve/use in general.

- Some references on numerical solution methods:
  - M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 282, no. 2, pp. 487–502, 1984.
  - M. Bardi and I. Capuzzo-Dolcetta (1997), "Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations," *Systems & Control: Foundations & Applications,* Birkhauser, Boston.
- Can use it to directly solve the continuous LQR problem

HJB Simple Example

- Consider the system with dynamics

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}$$

for which $A + A^T = 0$ and $\|\mathbf{u}\| \leq 1$, and the cost function

$$J = \int_0^{t_f} dt = t_f$$

- Then the Hamiltonian is

$$\mathcal{H} = 1 + J^\star_{\mathbf{x}}(A\mathbf{x} + \mathbf{u})$$

and the constrained minimization of $\mathcal{H}$ with respect to $\mathbf{u}$ gives

$$\mathbf{u}^\star = -(J^\star_{\mathbf{x}})^T / \|J^\star_{\mathbf{x}}\|$$

- Thus the HJB equation is:

$$-J^\star_t = 1 + J^\star_{\mathbf{x}}(A\mathbf{x}) - \|J^\star_{\mathbf{x}}\|$$

- As a candidate solution, take $J^\star(\mathbf{x}) = \mathbf{x}^T\mathbf{x}/\|\mathbf{x}\| = \|\mathbf{x}\|$, which is not an explicit function of t, so

$$J^\star_{\mathbf{x}} = \frac{\mathbf{x}^T}{\|\mathbf{x}\|} \quad \text{and} \quad J^\star_t = 0$$

which gives:

$$\begin{aligned} 0 &= 1 + \frac{\mathbf{x}^T}{\|\mathbf{x}\|}(A\mathbf{x}) - \frac{\|\mathbf{x}\|}{\|\mathbf{x}\|} \\ &= \frac{1}{\|\mathbf{x}\|}(\mathbf{x}^T A\mathbf{x}) \\ &= \frac{1}{\|\mathbf{x}\|}\frac{1}{2}\mathbf{x}^T(A + A^T)\mathbf{x} = 0 \end{aligned}$$

so that the HJB is satisfied and the optimal control is:

$$\mathbf{u}^\star = -\frac{\mathbf{x}}{\|\mathbf{x}\|}$$

Continuous LQR

- Specialize to a linear system model and a quadratic cost function

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t)$$

$$J = \frac{1}{2}\mathbf{x}(t_f)^T H \mathbf{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left\{ \mathbf{x}(t)^T R_{\mathrm{xx}}(t)\mathbf{x}(t) + \mathbf{u}(t)^T R_{\mathrm{uu}}(t)\mathbf{u}(t) \right\} dt$$

  – Assume that t_f fixed and there are no bounds on $\mathbf{u}$,
  – Assume $H, R_{\mathrm{xx}}(t) \geq 0$ and $R_{\mathrm{uu}}(t) > 0$, then

$$\begin{aligned} \mathcal{H}(\mathbf{x}, \mathbf{u}, J^\star_{\mathbf{x}}, t) &= \frac{1}{2}\left[\mathbf{x}(t)^T R_{\mathrm{xx}}(t)\mathbf{x}(t) + \mathbf{u}(t)^T R_{\mathrm{uu}}(t)\mathbf{u}(t) \right] \\ &\quad + J^\star_{\mathbf{x}}(\mathbf{x}(t), t)\left[A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \right] \end{aligned}$$

- Now need to find the minimum of $\mathcal{H}$ with respect to $\mathbf{u}$, which will occur at a stationary point that we can find using (no constraints)

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = \mathbf{u}(t)^T R_{\mathrm{uu}}(t) + J^\star_{\mathbf{x}}(\mathbf{x}(t), t)B(t) = 0$$

  – Which gives the **optimal control law:**

$$\mathbf{u}^\star(t) = -R_{\mathrm{uu}}^{-1}(t)B(t)^T J^\star_{\mathbf{x}}(\mathbf{x}(t), t)^T$$

  – Since

$$\frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}^2} = R_{\mathrm{uu}}(t) > 0$$

  then this defines a global minimum.

- Given this control law, can rewrite the Hamiltonian as:

$$\mathcal{H}(\mathbf{x}, \mathbf{u}^\star, J_{\mathbf{x}}^\star, t) =$$
$$\frac{1}{2}\left[\mathbf{x}(t)^T R_{\rm xx}(t)\mathbf{x}(t) + J_{\mathbf{x}}^\star(\mathbf{x}(t),t)B(t)R_{\rm uu}^{-1}(t)R_{\rm uu}(t)R_{\rm uu}^{-1}(t)B(t)^T J_{\mathbf{x}}^\star(\mathbf{x}(t),t)^T\right]$$
$$+J_{\mathbf{x}}^\star(\mathbf{x}(t),t)\left[A(t)\mathbf{x}(t) - B(t)R_{\rm uu}^{-1}(t)B(t)^T J_{\mathbf{x}}^\star(\mathbf{x}(t),t)^T\right]$$

$$= \frac{1}{2}\mathbf{x}(t)^T R_{\rm xx}(t)\mathbf{x}(t) + J_{\mathbf{x}}^\star(\mathbf{x}(t),t)A(t)\mathbf{x}(t)$$
$$-\frac{1}{2}J_{\mathbf{x}}^\star(\mathbf{x}(t),t)B(t)R_{\rm uu}^{-1}(t)B(t)^T J_{\mathbf{x}}^\star(\mathbf{x}(t),t)^T$$

- Might be difficult to see where this is heading, but note that the boundary condition for this PDE is:

$$J^\star(\mathbf{x}(t_f), t_f) = \frac{1}{2}\mathbf{x}^T(t_f)H\mathbf{x}(t_f)$$

  – So a candidate solution to investigate is to maintain a quadratic form for this cost for all time t. So could assume that

$$J^\star(\mathbf{x}(t), t) = \frac{1}{2}\mathbf{x}^T(t)P(t)\mathbf{x}(t), \qquad P(t) = P^T(t)$$

  and see what conditions we must impose on $P(t)$. [6]

  – Note that in this case, $J^\star$ is a function of the variables $\mathbf{x}$ and t[7]

$$\frac{\partial J^\star}{\partial \mathbf{x}} = \mathbf{x}^T(t)P(t)$$

$$\frac{\partial J^\star}{\partial t} = \frac{1}{2}\mathbf{x}^T(t)\dot{P}(t)\mathbf{x}(t)$$

- To use HJB equation need to evaluate:

$$-J_t^\star(\mathbf{x}(t), t) = \min_{\mathbf{u}(t)\in\mathcal{U}} \mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), J_{\mathbf{x}}^\star, t)$$

---

[6] See AM, pg. 21 on how to avoid having to make this assumption.

[7] Partial derivatives taken wrt one variable assuming the other is fixed. Note that there are 2 independent variables in this problem x and t. x is time-varying, but it is not a function of t.

- Substitute candidate solution into HJB:

$$
\begin{aligned}
-\frac{1}{2}\mathbf{x}(t)^T\dot{P}(t)\mathbf{x}(t) &= \frac{1}{2}\mathbf{x}(t)^T R_{\mathrm{xx}}(t)\mathbf{x}(t) + \mathbf{x}^T P(t)A(t)\mathbf{x}(t) \\
&\quad -\frac{1}{2}\mathbf{x}^T(t)P(t)B(t)R_{\mathrm{uu}}^{-1}(t)B(t)^T P(t)\mathbf{x}(t) \\
&= \frac{1}{2}\mathbf{x}(t)^T R_{\mathrm{xx}}(t)\mathbf{x}(t) + \frac{1}{2}\mathbf{x}^T(t)\{P(t)A(t) + A(t)^T P(t)\}\mathbf{x}(t) \\
&\quad -\frac{1}{2}\mathbf{x}^T(t)P(t)B(t)R_{\mathrm{uu}}^{-1}(t)B(t)^T P(t)\mathbf{x}(t)
\end{aligned}
$$

which must be true for all $\mathbf{x}(t)$, so we require that $P(t)$ solve

$$
\begin{aligned}
-\dot{P}(t) &= P(t)A(t) + A(t)^T P(t) + R_{\mathrm{xx}}(t) - P(t)B(t)R_{\mathrm{uu}}^{-1}(t)B(t)^T P(t) \\
P(t_f) &= H
\end{aligned}
$$

- If $P(t)$ solves this **Differential Riccati Equation**, then the HJB equation is satisfied by the candidate $J^\star(\mathbf{x}(t), t)$ and the resulting control is **optimal**.

- Key thing about this $J^\star$ solution is that, since $J^\star_{\mathbf{x}} = \mathbf{x}^T(t)P(t)$, then

$$
\begin{aligned}
\mathbf{u}^\star(t) &= -R_{\mathrm{uu}}^{-1}(t)B(t)^T J^\star_{\mathbf{x}}(\mathbf{x}(t), t)^T \\
&= -R_{\mathrm{uu}}^{-1}(t)B(t)^T P(t)\mathbf{x}(t)
\end{aligned}
$$

  - Thus **optimal feedback control is a linear state feedback** with gain

$$
F(t) = R_{\mathrm{uu}}^{-1}(t)B(t)^T P(t) \Rightarrow \mathbf{u}(t) = -F(t)\mathbf{x}(t)
$$

    ◇ Can be solved for ahead of time.

- As before, can evaluate the performance of some arbitrary time-varying feedback gain $\mathbf{u} = -G(t)\mathbf{x}(t)$, and the result is that

$$J_G = \frac{1}{2}\mathbf{x}^T S(t)\mathbf{x}$$

where $S(t)$ solves

$$\begin{aligned} -\dot{S}(t) &= \{A(t) - B(t)G(t)\}^T S(t) + S(t)\{A(t) - B(t)G(t)\} \\ &\quad + R_{\text{xx}}(t) + G(t)^T R_{\text{uu}}(t)G(t) \\ S(t_f) &= H \end{aligned}$$

  – Since this must be true for arbitrary G, then would expect that this reduces to Riccati Equation if $G(t) \equiv R_{\text{uu}}^{-1}(t)B^T(t)S(t)$

- If we assume LTI dynamics and let $t_f \to \infty$, then at any finite time t, would expect the Differential Riccati Equation to settle down to a steady state value (if it exists) which is the solution of

$$PA + A^T P + R_{\text{xx}} - PBR_{\text{uu}}^{-1}B^T P = 0$$

  – Called the **(Control) Algebraic Riccati Equation (CARE)**
  – Typically assume that $R_{\text{xx}} = C_z^T R_{\text{zz}} C_z$, $R_{\text{zz}} > 0$ associated with performance output variable $\mathbf{z}(t) = C_z\mathbf{x}(t)$

LQR Observations

- With terminal penalty, $H = 0$, the solution to the differential Riccati Equation (DRE) approaches a constant iff the system has no poles that are unstable, uncontrollable[8], and observable[9] by $\mathbf{z}(t)$
  - If a constant steady state solution to the DRE exists, then it is a positive semi-definite, symmetric solution of the CARE.

- If $[A, B, C_z]$ is both stabilizable and detectable (i.e. all modes are stable or seen in the cost function), then:
  - Independent of $H \geq 0$, the steady state solution P_{ss} of the DRE approaches the **unique** PSD symmetric solution of the CARE.

- If a steady state solution exists P_{ss} to the DRE, then the closed-loop system using the static form of the feedback

$$\mathbf{u}(t) = -R_{\rm uu}^{-1} B^T P_{ss} \mathbf{x}(t) = -F_{ss} \mathbf{x}(t)$$

  is **asymptotically stable** if and only if the system $[A, B, C_z]$ is stabilizable and detectable.
  - This steady state control minimizes the infinite horizon cost function $\lim_{t_f \to \infty} J$ for all $H \geq 0$

- The solution P_{ss} is **positive definite** if and only if the system $[A, B, C_z]$ is stabilizable and completely observable.

- See Kwakernaak and Sivan, page 237, Section 3.4.3.

---

[8] 16.31 Notes on Controllability

[9] 16.31 Notes on Observability

Scalar LQR Example

- A scalar system with dynamics $\dot{x} = ax + bu$ and with cost ($R_{\rm xx} > 0$ and $R_{\rm uu} > 0$)

$$J = \int_0^\infty (R_{\rm xx} x^2(t) + R_{\rm uu} u^2(t))\ dt$$

- This simple system represents one of the few cases for which the differential Riccati equation can be solved analytically:

$$P(\tau) = \frac{(aP_{t_f} + R_{\rm xx})\sinh(\beta\tau) + \beta P_{t_f}\cosh(\beta\tau)}{(b^2 P_{t_f}/R_{\rm uu} - a)\sinh(\beta\tau) + \beta\cosh(\beta\tau)}$$

where $\tau = t_f - t$, $\beta = \sqrt{a^2 + b^2(R_{\rm xx}/R_{\rm uu})}$.

  – Note that for given a and b, ratio $R_{\rm xx}/R_{\rm uu}$ determines the time constant of the transient in $P(t)$ (determined by β).

- The steady-state P solves the CARE:

$$2aP_{ss} + R_{\rm xx} - P_{ss}^2 b^2/R_{\rm uu} = 0$$

which gives (take positive one) that

$$P_{ss} = \frac{a + \sqrt{a^2 + b^2 R_{\rm xx}/R_{\rm uu}}}{b^2/R_{\rm uu}} = \frac{a+\beta}{b^2/R_{\rm uu}} = \frac{a+\beta}{b^2/R_{\rm uu}}\left(\frac{-a+\beta}{-a+\beta}\right) > 0$$

- With $P_{t_f} = 0$, the solution of the differential equation reduces to:

$$P(\tau) = \frac{R_{\rm xx}\sinh(\beta\tau)}{(-a)\sinh(\beta\tau) + \beta\cosh(\beta\tau)}$$

where as $\tau \to t_f (\to \infty)$ then $\sinh(\beta\tau) \to \cosh(\beta\tau) \to e^{\beta\tau}/2$, so

$$P(\tau) = \frac{R_{\rm xx}\sinh(\beta\tau)}{(-a)\sinh(\beta\tau) + \beta\cosh(\beta\tau)} \to \frac{R_{\rm xx}}{(-a)+\beta} = P_{ss}$$

- Then the steady state feedback controller is $u(t) = -Kx(t)$ where

$$K_{ss} = R_{\rm uu}^{-1} b P_{ss} = \frac{a + \sqrt{a^2 + b^2 R_{\rm xx}/R_{\rm uu}}}{b}$$

- The closed-loop dynamics are

$$\begin{aligned} \dot{x} &= (a - bK_{ss})x = A_{cl}x(t) \\ &= \left(a - \frac{b}{b}(a + \sqrt{a^2 + b^2 R_{\rm xx}/R_{\rm uu}})\right) x \\ &= -\sqrt{a^2 + b^2 R_{\rm xx}/R_{\rm uu}}\; x \end{aligned}$$

which are clearly stable.

- As $R_{\rm xx}/R_{\rm uu} \to \infty$, $A_{cl} \approx -|b|\sqrt{R_{\rm xx}/R_{\rm uu}}$
  – **Cheap control** problem
  – Note that smaller $R_{\rm uu}$ leads to much faster response.

- As $R_{\rm xx}/R_{\rm uu} \to 0$, $K \approx (a + |a|)/b$
  – **Expensive control** problem
  – If $a < 0$ (open-loop stable), $K \approx 0$ and $A_{cl} = a - bK \approx a$
  – If $a > 0$ (OL unstable), $K \approx 2a/b$ and $A_{cl} = a - bK \approx -a$

- Note that in the expensive control case, the controller tries to do as little as possible, but it must stabilize the unstable open-loop system.
  – Observation: optimal definition of "as little as possible" is to put the closed-loop pole at the reflection of the open-loop pole about the imaginary axis.

Numerical P Integration

- To numerically integrate solution of P, note that we can use standard Matlab integration tools if we can rewrite the DRE in vector form.
  – Define a vec operator so that

$$\texttt{vec}(P) = \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{1n} \\ P_{22} \\ P_{23} \\ \vdots \\ P_{nn} \end{bmatrix} \equiv y$$

  – The $\texttt{unvec}(y)$ operation is the straightforward
  – Can now write the DRE as differential equation in the variable y

- Note that with $\tau = t_f - t$, then $d\tau = -dt$,
  – $t = t_f$ corresponds to $\tau = 0$, $t = 0$ corresponds to $\tau = t_f$
  – Can do the integration forward in time variable $\tau : 0 \rightarrow t_f$

- Then define a Matlab function as

```
doty = function(y);
global A B Rxx Ruu %
P=unvec(y); %
% assumes that P derivative wrt to tau (so no negative)
dot P = (P*A + A^T*P+Rxx-P*B*Ruu^{-1}*B^T*P);%
doty = vec(dotP); %
return
```

  – Which is integrated from $\tau = 0$ with initial condition H
  – Code uses a more crude form of integration

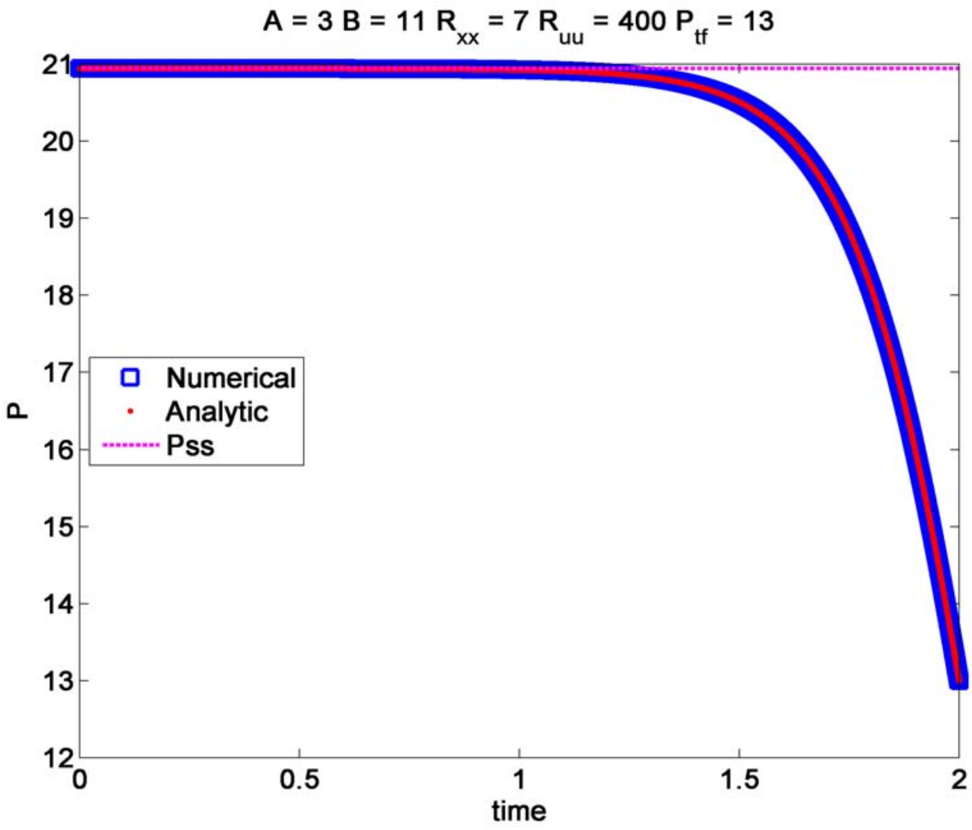


Figure 4.1: Comparison of numerical and analytical P

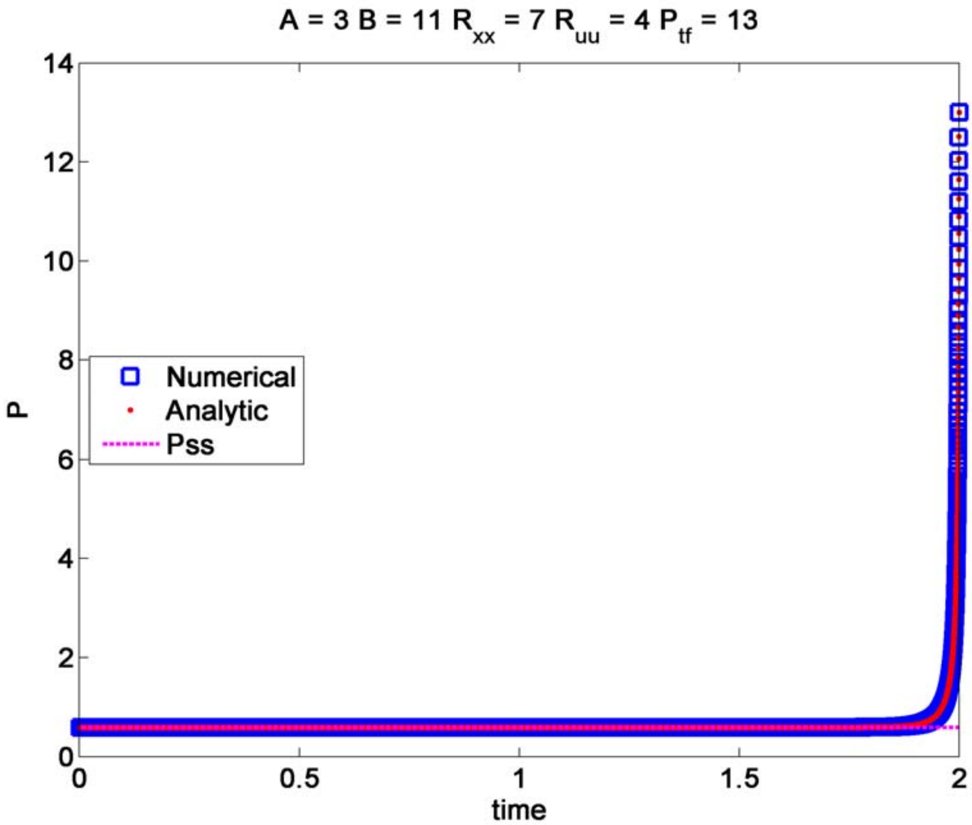
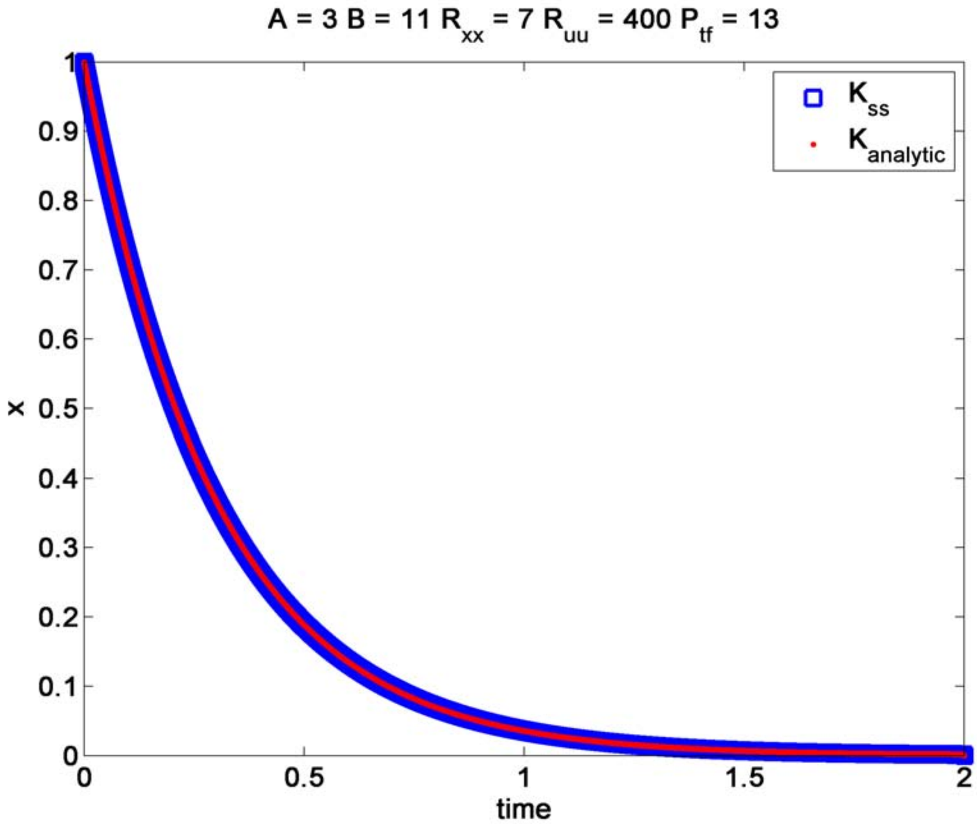


Figure 4.2: Comparison showing response with much larger R_{xx}/R_{uu}



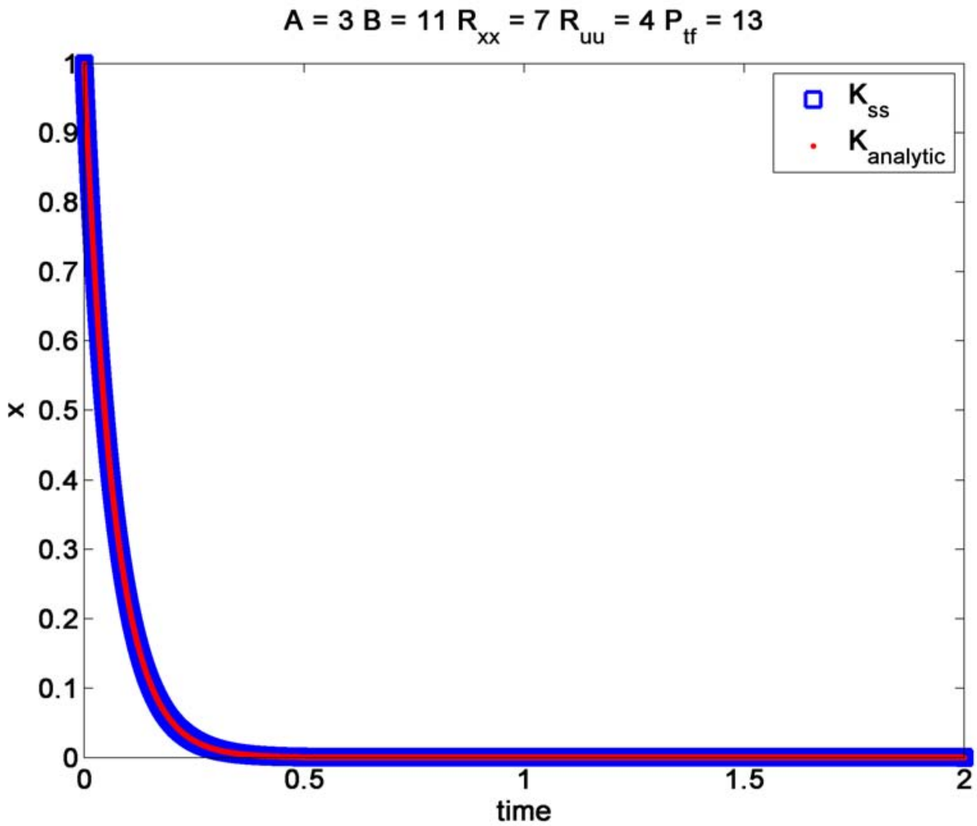


Figure 4.3: State response with high and low R_{uu}. State response with time-varying gain almost indistinguishable – highly dynamic part of x response ends before significant variation in P.

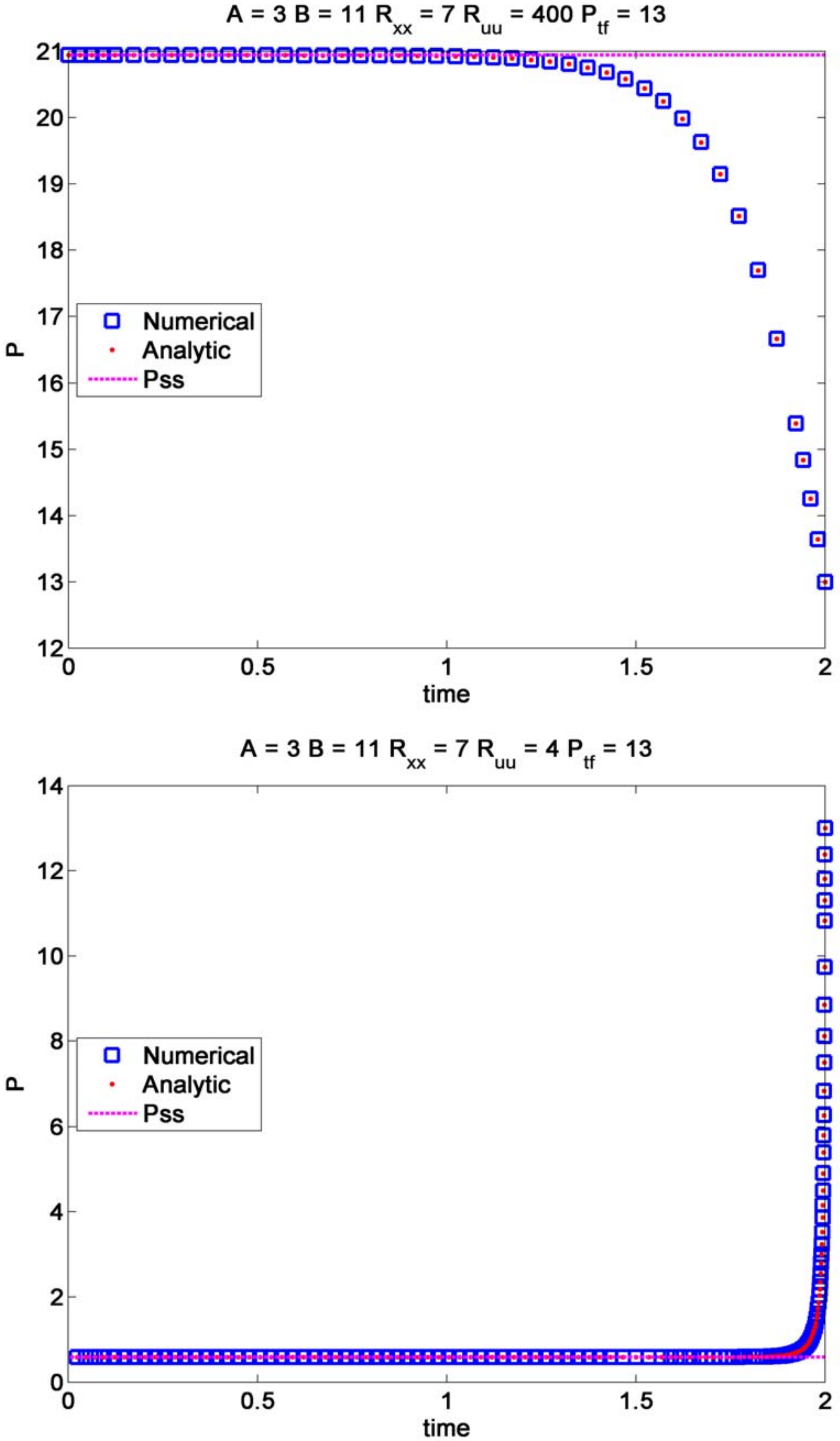


Figure 4.4: Comparison of numerical and analytical P using a better integration scheme

Numerical Calculation of P

```
% Simple LQR example showing time varying P and gains
% 16.323 Spring 2008
% Jonathan How
% reg2.m
clear all;close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')
global A B Rxx Ruu

A=3;B=11;Rxx=7;Ptf=13;tf=2;dt=.0001;
Ruu=20^2;
Ruu=2^2;

% integrate the P backwards (crude form)
time=[0:dt:tf];
P=zeros(1,length(time));K=zeros(1,length(time));Pcurr=Ptf;
for kk=0:length(time)-1
  P(length(time)-kk)=Pcurr;
  K(length(time)-kk)=inv(Ruu)*B'*Pcurr;
  Pdot=-Pcurr*A-A'*Pcurr-Rxx+Pcurr*B*inv(Ruu)*B'*Pcurr;
  Pcurr=Pcurr-dt*Pdot;
end

options=odeset('RelTol',1e-6,'AbsTol',1e-6)
[tau,y]=ode45(@doty,[0 tf],vec(Ptf));
Tnum=[];Pnum=[];Fnum=[];
for i=1:length(tau)
    Tnum(length(tau)-i+1)=tf-tau(i);
    temp=unvec(y(i,:));
    Pnum(length(tau)-i+1,:,:)=temp;
    Fnum(length(tau)-i+1,:)=-inv(Ruu)*B'*temp;
end

% get the SS result from LQR
[klqr,Plqr]=lqr(A,B,Rxx,Ruu);

% Analytical pred
beta=sqrt(A^2+Rxx/Ruu*B^2);
t=tf-time;
Pan=((A*Ptf+Rxx)*sinh(beta*t)+beta*Ptf*cosh(beta*t))./...
    ((B^2*Ptf/Ruu-A)*sinh(beta*t)+beta*cosh(beta*t));
Pan2=((A*Ptf+Rxx)*sinh(beta*(tf-Tnum))+beta*Ptf*cosh(beta*(tf-Tnum)))./...
    ((B^2*Ptf/Ruu-A)*sinh(beta*(tf-Tnum))+beta*cosh(beta*(tf-Tnum)));

figure(1);clf
plot(time,P,'bs',time,Pan,'r.',[0 tf],[1 1]*Plqr,'m--')
title(['A = ',num2str(A),' B = ',num2str(B),' R_{xx} = ',num2str(Rxx),...
    ' R_{uu} = ',num2str(Ruu),' P_{tf} = ',num2str(Ptf)])
legend('Numerical','Analytic','Pss','Location','West')
xlabel('time');ylabel('P')
if Ruu > 10
    print -r300 -dpng reg2_1.png;
else
    print -r300 -dpng reg2_2.png;
end

figure(3);clf
plot(Tnum,Pnum,'bs',Tnum,Pan2,'r.',[0 tf],[1 1]*Plqr,'m--')
title(['A = ',num2str(A),' B = ',num2str(B),' R_{xx} = ',num2str(Rxx),...
    ' R_{uu} = ',num2str(Ruu),' P_{tf} = ',num2str(Ptf)])
legend('Numerical','Analytic','Pss','Location','West')
xlabel('time');ylabel('P')
if Ruu > 10
    print -r300 -dpng reg2_13.png;
else
    print -r300 -dpng reg2_23.png;
end
```

```
Pan2=inline('((A*Ptf+Rxx)*sinh(beta*t)+beta*Ptf*cosh(beta*t))/((B^2*Ptf/Ruu-A)*sinh(beta*t)+beta*cosh(beta*t))');
x1=zeros(1,length(time));x2=zeros(1,length(time));
xcurr1=[1]';xcurr2=[1]';
for kk=1:length(time)-1
  x1(:,kk)=xcurr1;  x2(:,kk)=xcurr2;
  xdot1=(A-B*Ruu^(-1)*B'*Pan2(A,B,Ptf,Ruu,Rxx,beta,tf-(kk-1)*dt))*x1(:,kk);
  xdot2=(A-B*klqr)*x2(:,kk);
  xcurr1=xcurr1+xdot1*dt;
  xcurr2=xcurr2+xdot2*dt;
end

figure(2);clf
plot(time,x2,'bs',time,x1,'r.');xlabel('time');ylabel('x')
title(['A = ',num2str(A),' B = ',num2str(B),' R_{xx} = ',num2str(Rxx),...
       ' R_{uu} = ',num2str(Ruu),' P_{tf} = ',num2str(Ptf)])
legend('K_{ss}','K_{analytic}','Location','NorthEast')
if Ruu > 10
    print -r300 -dpng reg2_11.png;
else
    print -r300 -dpng reg2_22.png;
end
```

```
function [doy]=doty(t,y);
global A B Rxx Ruu;
P=unvec(y);
dotP=P*A+A'*P+Rxx-P*B*Ruu^(-1)*B'*P;
doy=vec(dotP);
return
```

```
function y=vec(P);

y=[];
for ii=1:length(P);
    y=[y;P(ii,ii:end)'];
end

return
```

```
function P=unvec(y);

N=max(roots([1 1 -2*length(y)]));
P=[];kk=N;kk0=1;
for ii=1:N;
    P(ii,ii:N)=[y(kk0+[0:kk-1])]';
    kk0=kk0+kk;
    kk=kk-1;
end
P=(P+P')-diag(diag(P));
return
```

Finite Time LQR Example

- Simple system with $t_0 = 0$ and $t_f = 10$sec.

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$2J = x^T(10) \begin{bmatrix} 0 & 0 \\ 0 & h \end{bmatrix} x(10) + \int_0^{10} \left\{ x^T(t) \begin{bmatrix} q & 0 \\ 0 & 0 \end{bmatrix} x(t) + ru^2(t) \right\} dt$$

- Compute gains using both time-varying $P(t)$ and steady-state value.

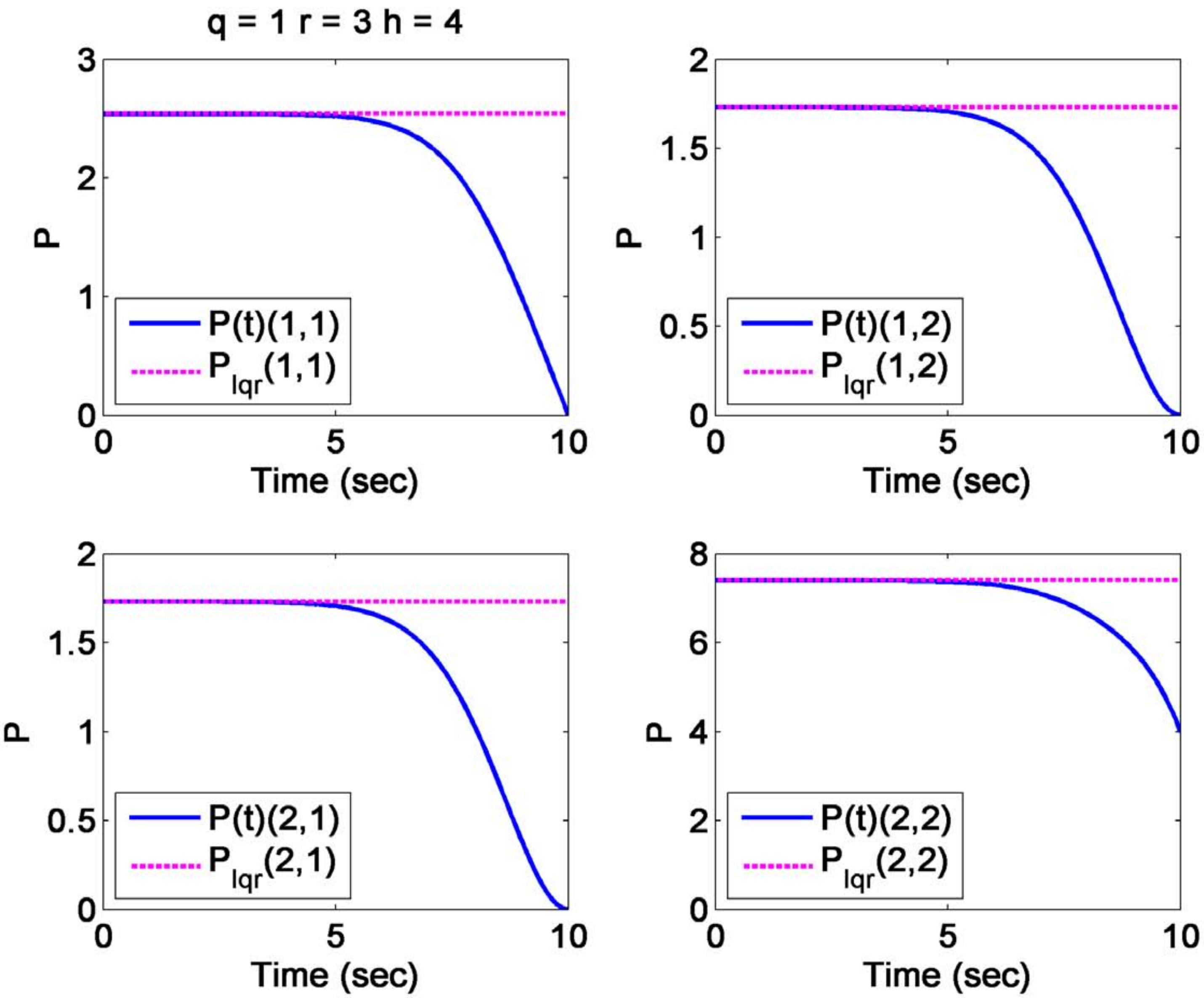


Figure 4.5: Set $q = 1$, $r = 3$, $h = 4$

- Find state solution $x(0) = [1 \;\; 1]^T$ using both sets of gains

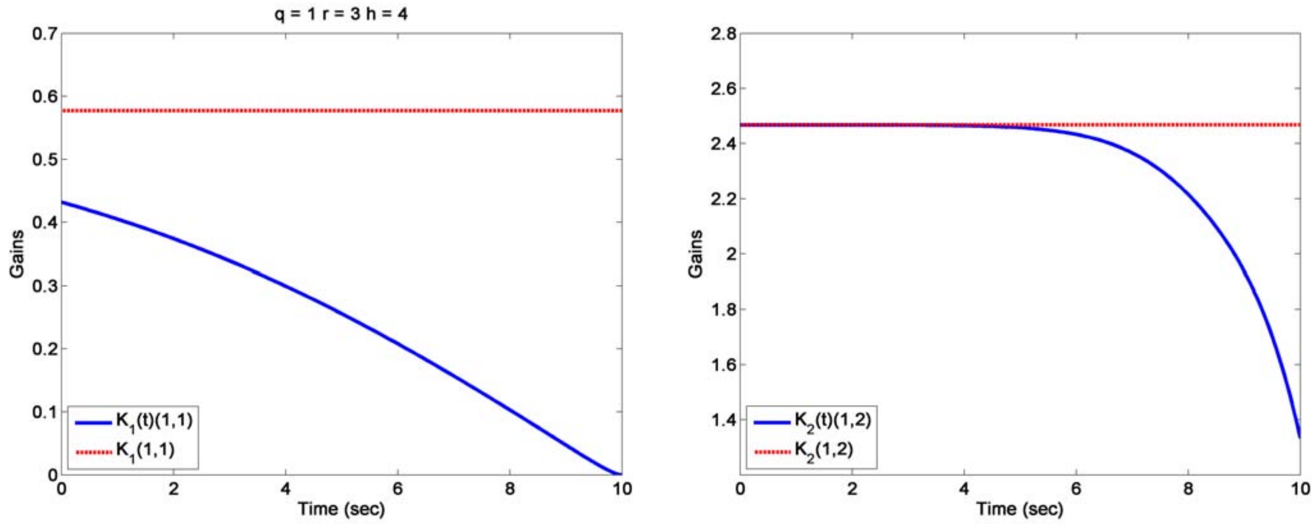


Figure 4.6: Time-varying and constant gains - $K_{lqr} = [0.5774 \; 2.4679]$

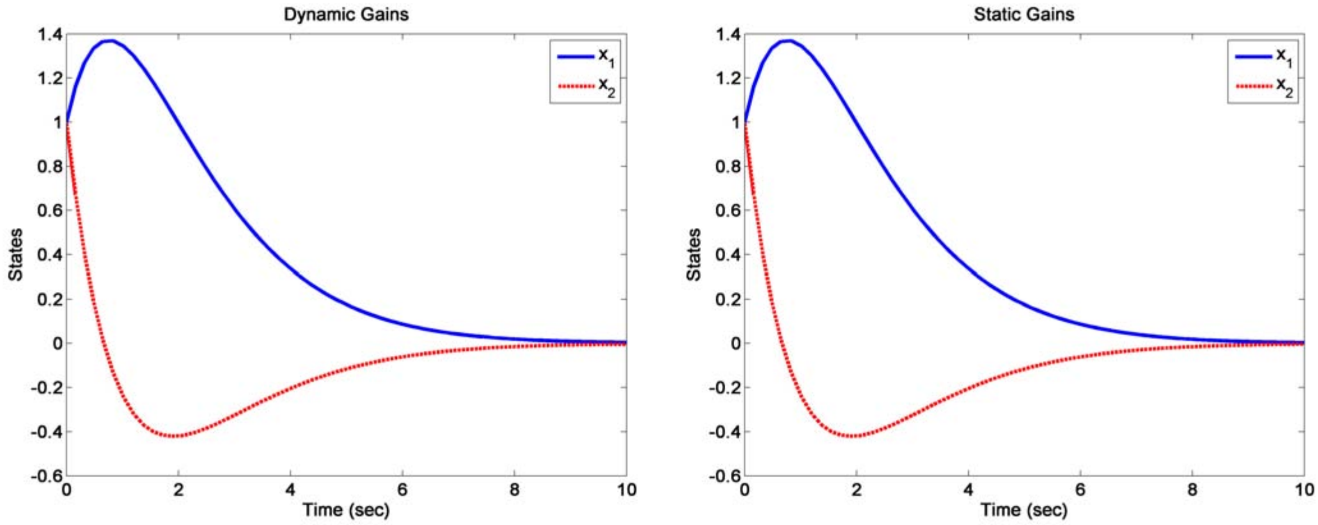


Figure 4.7: State response - Constant gain and time-varying gain almost indistinguishable because the transient dies out before the time at which the gains start to change – effectively a steady state problem.

- For most applications, the static gains are more than adequate - it is only when the terminal conditions are important in a short-time horizon problem that the time-varying gains should be used.
  - **Significant savings** in implementation complexity & computation.

Finite Time LQR Example

```
% Simple LQR example showing time varying P and gains
% 16.323 Spring 2008
% Jonathan How
% reg1.m
%
clear all;%close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')
global A B Rxx Ruu
jprint = 0;

h=4;q=1;r=3;
A=[0 1;0 1];B=[0 1]';tf=10;dt=.01;
Ptf=[0 0;0 h];Rxx=[q 0;0 0];Ruu=r;
Ptf=[0 0;0 1];Rxx=[q 0;0 100];Ruu=r;

% alternative calc of Ricc solution
H=[A -B*B'/r ; -Rxx -A'];
[V,D]=eig(H); % check order of eigenvalues
Psi11=V(1:2,1:2);
Psi21=V(3:4,1:2);
Ptest=Psi21*inv(Psi11);

if 0

% integrate the P backwards (crude)
time=[0:dt:tf];
P=zeros(2,2,length(time));
K=zeros(1,2,length(time));
Pcurr=Ptf;
for kk=0:length(time)-1
  P(:,:,length(time)-kk)=Pcurr;
  K(:,:,length(time)-kk)=inv(Ruu)*B'*Pcurr;
  Pdot=-Pcurr*A-A'*Pcurr-Rxx+Pcurr*B*inv(Ruu)*B'*Pcurr;
  Pcurr=Pcurr-dt*Pdot;
end

else
% integrate forwards (ODE)
options=odeset('RelTol',1e-6,'AbsTol',1e-6)
[tau,y]=ode45(@doty,[0 tf],vec(Ptf),options);
Tnum=[];Pnum=[];Fnum=[];
for i=1:length(tau)
    time(length(tau)-i+1)=tf-tau(i);
    temp=unvec(y(i,:));
    P(:,:,length(tau)-i+1)=temp;
    K(:,:,length(tau)-i+1)=inv(Ruu)*B'*temp;
end

end % if 0

% get the SS result from LQR
[klqr,Plqr]=lqr(A,B,Rxx,Ruu);

x1=zeros(2,1,length(time));
x2=zeros(2,1,length(time));
xcurr1=[1 1]';
xcurr2=[1 1]';
for kk=1:length(time)-1
  dt=time(kk+1)-time(kk);
  x1(:,:,kk)=xcurr1;
  x2(:,:,kk)=xcurr2;
  xdot1=(A-B*K(:,:,kk))*x1(:,:,kk);
  xdot2=(A-B*klqr)*x2(:,:,kk);
  xcurr1=xcurr1+xdot1*dt;
  xcurr2=xcurr2+xdot2*dt;
end
```

```
x1(:,:,length(time))=xcurr1;
x2(:,:,length(time))=xcurr2;

figure(5);clf
subplot(221)
plot(time,squeeze(K(1,1,:)),[0 10],[1 1]*klqr(1),'m--','LineWidth',2)
legend('K_1(t)','K_1')
xlabel('Time (sec)');ylabel('Gains')
title(['q = ',num2str(1),' r = ',num2str(r),' h = ',num2str(h)])
subplot(222)
plot(time,squeeze(K(1,2,:)),[0 10],[1 1]*klqr(2),'m--','LineWidth',2)
legend('K_2(t)','K_2')
xlabel('Time (sec)');ylabel('Gains')
subplot(223)
plot(time,squeeze(x1(1,1,:)),time,squeeze(x1(2,1,:)),'m--','LineWidth',2),
legend('x_1','x_2')
xlabel('Time (sec)');ylabel('States');title('Dynamic Gains')
subplot(224)
plot(time,squeeze(x2(1,1,:)),time,squeeze(x2(2,1,:)),'m--','LineWidth',2),
legend('x_1','x_2')
xlabel('Time (sec)');ylabel('States');title('Static Gains')

figure(6);clf
subplot(221)
plot(time,squeeze(P(1,1,:)),[0 10],[1 1]*Plqr(1,1),'m--','LineWidth',2)
legend('P(t)(1,1)','P_{lqr}(1,1)','Location','SouthWest')
xlabel('Time (sec)');ylabel('P')
title(['q = ',num2str(1),' r = ',num2str(r),' h = ',num2str(h)])
subplot(222)
plot(time,squeeze(P(1,2,:)),[0 10],[1 1]*Plqr(1,2),'m--','LineWidth',2)
legend('P(t)(1,2)','P_{lqr}(1,2)','Location','SouthWest')
xlabel('Time (sec)');ylabel('P')
subplot(223)
plot(time,squeeze(P(2,1,:)),[0 10],[1 1]*squeeze(Plqr(2,1)),'m--','LineWidth',2),
legend('P(t)(2,1)','P_{lqr}(2,1)','Location','SouthWest')
xlabel('Time (sec)');ylabel('P')
subplot(224)
plot(time,squeeze(P(2,2,:)),[0 10],[1 1]*squeeze(Plqr(2,2)),'m--','LineWidth',2),
legend('P(t)(2,2)','P_{lqr}(2,2)','Location','SouthWest')
xlabel('Time (sec)');ylabel('P')
axis([0 10 0 8])
if jprint;    print -dpng -r300 reg1_6.png
end

figure(1);clf
plot(time,squeeze(K(1,1,:)),[0 10],[1 1]*klqr(1),'r--','LineWidth',3)
legend('K_1(t)(1,1)','K_1(1,1)','Location','SouthWest')
xlabel('Time (sec)');ylabel('Gains')
title(['q = ',num2str(1),' r = ',num2str(r),' h = ',num2str(h)])
print -dpng -r300 reg1_1.png
figure(2);clf
plot(time,squeeze(K(1,2,:)),[0 10],[1 1]*klqr(2),'r--','LineWidth',3)
legend('K_2(t)(1,2)','K_2(1,2)','Location','SouthWest')
xlabel('Time (sec)');ylabel('Gains')
if jprint;    print -dpng -r300 reg1_2.png
end

figure(3);clf
plot(time,squeeze(x1(1,1,:)),time,squeeze(x1(2,1,:)),'r--','LineWidth',3),
legend('x_1','x_2')
xlabel('Time (sec)');ylabel('States');title('Dynamic Gains')
if jprint;    print -dpng -r300 reg1_3.png
end

figure(4);clf
plot(time,squeeze(x2(1,1,:)),time,squeeze(x2(2,1,:)),'r--','LineWidth',3),
legend('x_1','x_2')
xlabel('Time (sec)');ylabel('States');title('Static Gains');
if jprint;    print -dpng -r300 reg1_4.png
end
```

Weighting Matrix Selection

- A good rule of thumb when selecting the weighting matrices $R_{\rm xx}$ and $R_{\rm uu}$ is to normalize the signals:

$$R_{\rm xx} = \begin{bmatrix} \frac{\alpha_1^2}{(x_1)^2_{\rm max}} & & & \\ & \frac{\alpha_2^2}{(x_2)^2_{\rm max}} & & \\ & & \ddots & \\ & & & \frac{\alpha_n^2}{(x_n)^2_{\rm max}} \end{bmatrix}$$

$$R_{\rm uu} = \rho \begin{bmatrix} \frac{\beta_1^2}{(u_1)^2_{\rm max}} & & & \\ & \frac{\beta_2^2}{(u_2)^2_{\rm max}} & & \\ & & \ddots & \\ & & & \frac{\beta_m^2}{(u_m)^2_{\rm max}} \end{bmatrix}$$

- The $(x_i)_{\rm max}$ and $(u_i)_{\rm max}$ represent the largest desired response/control input for that component of the state/actuator signal.

- The $\sum_i \alpha_i^2 = 1$ and $\sum_i \beta_i^2 = 1$ are used to add an additional relative weighting on the various components of the state/control

- ρ is used as the last relative weighting between the control and state penalties $\Rightarrow$ gives us a relatively concrete way to discuss the relative size of $R_{\rm xx}$ and $R_{\rm uu}$ and their ratio $R_{\rm xx}/R_{\rm uu}$

- Note: to **directly** compare the continuous and discrete LQR, you must modify the weighting matrices for the discrete case, as outlined here using `lqrd`.