Linear Quadratic (LQ) optimal control

”Principle of optimality” or Dynamic Programming (Bellman 1957) is one way to approach the problem. Variational calculus is another one.

Books:

- Kirk (1998), “Optimal Control Theory”
- Lewis and Syrmos (1995), ”Optimal Control”
- Bryson and Ho (1975), “Applied Optimal Control: Optimization, Estimation, and Control”
- Athans and Falb (1966), “Optimal Control: An Introduction To The Theory And Its Applications “

Optimization from control viewpoint

Static Optimization

Lagrange multipliers

Numerical methods

Shooting methods, TPBVPs

Dynamic Programming

Weights

Calculus of Variations

Euler-Lagrange equations

The *brachistochrone* problem

Pontryagin's maximum principle

Hamiltonian

Bang-Bang control

Linear Quadratic (LQ)-optimal control

Linear matrix inequalities (LMI)

Hamilton-Jacobi-Bellman equation

The Maximum (Minimum) Principle

- Pontryagin + co-workers, 1962
- Classical ”Calculus of Variations”
- Calculus of variations in optimal control problems
- A special case of the maximum principle

- Maximum principle (nonlinear system, restrictions in state and input variables, possibly nonlinear cost, minimum time problems, minimum fuel problems etc.)
- Mathematically involved

Note: Min J = -Max (-J) always

Concepts

$$\dot{x}(t) = f(x(t), u(t), t), \quad x(t_0) = x_0$$ Process

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

- States and co-states (adjoint states)
- Hamiltonian function
- State equations for states and co-states
- Conditions for the Hamiltonian
- Boundary conditions
- Two-point boundary value problems

Principle of Optimality

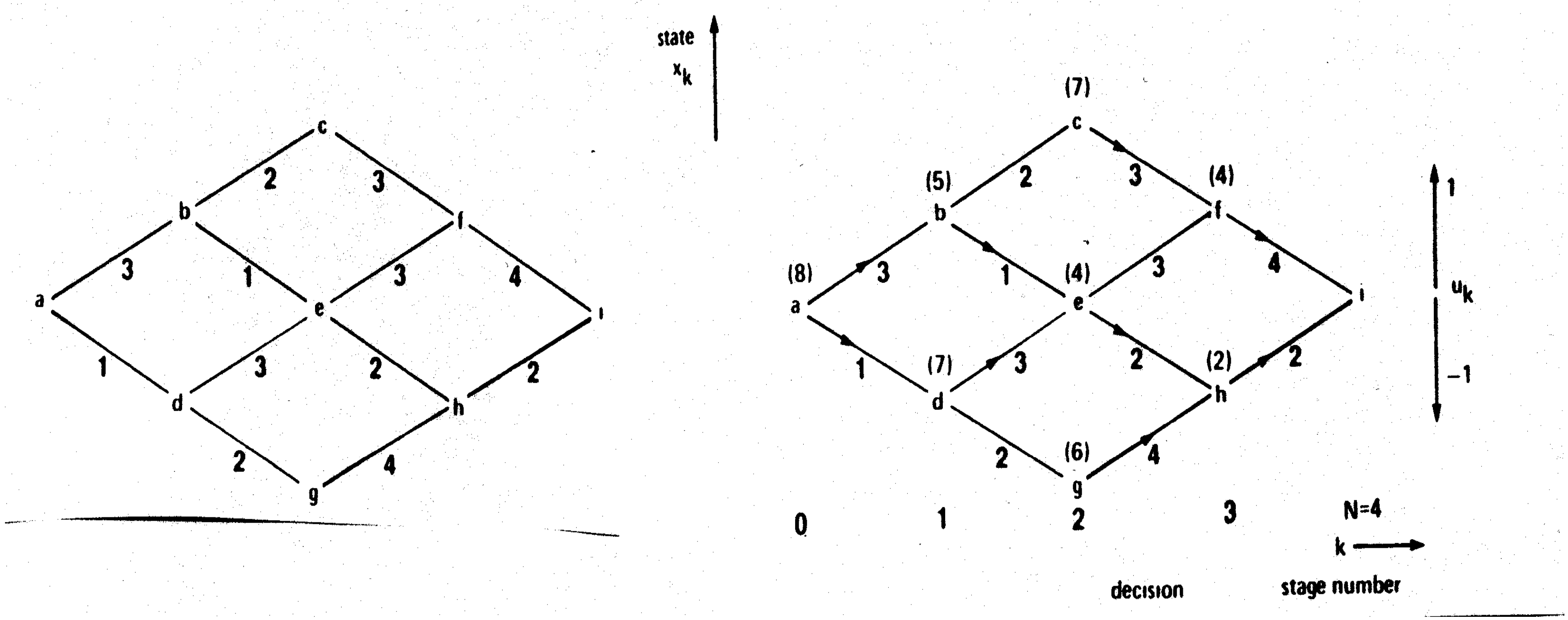
(Bellman 1957)

“An optimal policy has the property that no matter what the previous decision (i.e. controls) have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.”

By applying this principle the number of candidates for the optimal solution can be reduced.

Calculations ”backwards in time”.

Ex. Routing problem



Discrete-time optimization problem

$$x_{k+1} = f^k(x_k, u_k)$$

Process

$$J_i(x_i) = \phi(N, x_N) + \sum_{k=i}^{N-1} L^k(x_k, u_k)$$

Criterion to be minimized

Use the principle of optimality. Let the optimal control be calculated from time k+1 to N for all states x at time k+1 and consider what happens

$$x_{k+1} = f^k(x_k, u_k)$$

$$J_i(x_i) = \phi(N, x_N) + \sum_{k=i}^{N-1} L^k(x_k, u_k)$$

Problem

$$L^k(x_k, u_k) + J^*_{k+1}(x_{k+1})$$

$$J^*_k(x_k) = \min\left[L^k(x_k, u_k) + J^*_{k+1}(x_{k+1})\right]$$

Determination of the solution by the principle of optimality

Find u_k such that the expression is minimized; optimal cost at time *k.*

Solution of the discrete-time LQ-problem by using dynamic programming

$$x_{k+1} = Ax_k + Bu_k$$ Process

$$J = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=i}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right)$$ Criterion

$$(S_N \geq 0, \quad Q \geq 0, \quad R > 0)$$ symmetric

x_i given $\quad x_N$ free

Find u_k^* in inteval $[i, N]$ minimizing the criterion

$$J_N^* = \frac{1}{2} x_N^T S_N x_N, \quad k = N$$

Cost from the end state

$$J_{N-1} = \frac{1}{2} x_{N-1}^T Q x_{N-1} + \frac{1}{2} u_{N-1}^T R u_{N-1} + \frac{1}{2} x_N^T S_N x_N$$

Backwards in time to time instant N-1

$$J_{N-1} = \frac{1}{2} x_{N-1}^T Q x_{N-1} + \frac{1}{2} u_{N-1}^T R u_{N-1} + \frac{1}{2} \left(A x_{N-1} + B u_{N-1} \right)^T S_N \left(A x_{N-1} + B u_{N-1} \right)$$

$$0 = \frac{\partial J_{N-1}}{\partial u_{N-1}} = R u_{N-1} + B^T S_N \left(A x_{N-1} + B u_{N-1} \right)$$

Minimize

$$u_{N-1}^* = -\left(B^T S_N B + R \right)^{-1} B^T S_N A x_{N-1}$$

The solution can be presented in the form

$$u^*_{N-1} = -K_{N-1}x_{N-1}, \quad K_{N-1} \triangleq \left(B^T S_N B + R\right)^{-1} B^T S_N A$$

By substituting into J_{N-1} gives the optimal cost

$$J^*_{N-1} = \frac{1}{2} x^T_{N-1}\left[\left(A - BK_{N-1}\right)^T S_N \left(A - BK_{N-1}\right) + K^T_{N-1} R K_{N-1} + Q\right] x_{N-1}$$

Define

$$S_{N-1} \triangleq \left(A - BK_{N-1}\right)^T S_N \left(A - BK_{N-1}\right) + K^T_{N-1} R K_{N-1} + Q$$

$$J^*_{N-1} = \frac{1}{2} x^T_{N-1} S_{N-1} x_{N-1}$$

Backwards to the time instant k = N-2

$$J_{N-2} = \frac{1}{2} x_{N-2}^T Q x_{N-2} + \frac{1}{2} u_{N-2}^T R u_{N-2} + \frac{1}{2} x_{N-1}^T S_{N-1} x_{N-1}$$

Now determine u_{N-2}^* , but the equations have the same form as above. We obtain the general solution

$$K_k = \left(B^T S_{k+1} B + R\right)^{-1} B^T S_{k+1} A$$

$$u_k^* = -K_k x_k$$

$$S_k = (A - BK_k)^T S_{k+1} (A - BK_k) + K_k^T R K_k + Q$$

$$J_k^* = \frac{1}{2} x_k^T S_k x_k$$

Continuous-time case: The Hamilton-Jacobi-Bellman equation

$$\dot{x}(t) = f(x(t), u(t), t)$$ System

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(\tau), u(\tau), \tau) d\tau$$ Criterion

Consider the problem as a part of the larger problem

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

Let us try to minimize this for all admissible $x(t)$ and for all $t \le t_f$

The minimum cost function is then

$$J^*(x(t),t) = \min_{\underbrace{u(\tau)}_{t \le \tau \le t_f}} \{ \int_t^{t_f} g(x(\tau),u(\tau),\tau)d\tau + h(x(t_f),t_f) \}$$

By dividing the optimization interval to two parts we obtain

$$J^*(x(t),t) = \min_{\underbrace{u(\tau)}_{t \le \tau \le t_f}} \{ \int_t^{t+\Delta t} g d\tau + \int_{t+\Delta t}^{t_f} g d\tau + h(x(t_f),t_f) \} \qquad \Delta t \text{ small}$$

Use the principle of optimality to get

$$J^*(x(t),t) = \min_{\underbrace{u(\tau)}_{t \le \tau \le t_f}} \{ \int_t^{t+\Delta t} g d\tau + J^*(x(t+\Delta t),t+\Delta t) \}$$

Expand $J^*\left(x(t+\Delta t), t+\Delta t\right)$ as a Taylor series about the point $(x(t), t)$ gives

$$J^*(x(t),t) \approx \underbrace{\min}_{\substack{u(\tau) \\ \underbrace{}_{t \le \tau \le t_f}}} \{ \int_t^{t+\Delta t} g\, d\tau + J^*(x(t),t) + \left[\frac{\partial J^*}{\partial t}(x(t),t) \right] \Delta t$$

$$+ \left[\frac{\partial J^*}{\partial x}(x(t),t) \right] [x(t+\Delta t) - x(t)] \}$$

and for small Δt

$$J^*(x(t),t) \approx \underbrace{\min}_{u(t)} \{ g(x(t),u(t),t)\Delta t + J^*(x(t),t)$$

$$+ J_t^*(x(t),t)\Delta t + J_x^*(x(t),t)\left[f(x(t),u(t),t) \right] \Delta t$$

Minimization (terms that do not depend on *u*)

$$0 \approx J_t^*(x(t),t)\Delta t + \underbrace{\min}_{u(t)} \{g(x(t),u(t),t)\Delta t + J_x^*(x(t),t)\left[f(x(t),u(t),t)\right]\Delta t\}$$

Dividing by Δt and letting $\Delta t \to 0$ gives

$$0 = J_t^*(x(t),t) + \underbrace{\min}_{u(t)} \{g(x(t),u(t),t) + J_x^*(x(t),t)\left[f(x(t),u(t),t)\right]\}$$

Setting $t = t_f$ the boundary condition is found

$$J^*(x(t_f),t_f) = h(x(t_f),t_f)$$

Define the Hamiltonian as

$$H(x(t),u(t),J_x^*,t) = g(x(t),u(t),t) + J_x^*(x(t),t)\left[f(x(t),u(t),t)\right]$$

and

$$H(x(t),u^*(x(t),J_x^*,t),J_x^*,t) = \underbrace{\min}_{u(t)} H(x(t),u(t),J_x^*,t)$$

since the minimizing control depends on x, J_x^* and t.

The H-J-B equation can be written in the form

$$0 = J_t^*(x(t),t) + H(x(t),u^*(x(t),J_x^*,t),J_x^*,t)$$

Example: $\dot{x}(t) = x(t) + u(t)$

$$\text{Min } J = \frac{1}{4}x^2(T) + \int_0^T \frac{1}{4}u^2(t)\,dt$$ (T fixed)

$$H(x,u,J_x^*,t) = \frac{1}{4}u^2 + J_x^*(x+u)$$

Necessary condition for optimality $\frac{\partial H}{\partial u} = \frac{1}{2}u + J_x^* = 0$

Note: $\frac{\partial^2 H}{\partial u^2} = \frac{1}{2} > 0$ implying this is a minimum (because of linear system with quadratic criterion)

$$u^*(t) = -2J_x^*(x(t),t)$$

Substitute into H-J-B

$$0 = J_t^* + \frac{1}{4}\left[-2J_x^*\right]^2 + J_x^* x - 2\left[J_x^*\right]^2$$

$$= J_t^* - \left[J_x^*\right]^2 + \left[J_x^*\right]x$$

Boundary condition

$$J^*(x(T),T) = \frac{1}{4}x^2(T)$$

Next, guess a solution form (for LQ problems this may work)

$$J^*(x(t),t) = \frac{1}{2}K(t)x^2(t) \Rightarrow J_x^*(x(t),t) = K(t)x(t)$$

This is the *Riccati transformation*

$$u^*(t) = -2K(t)x(t)$$

Setting $K(T)$ = ½ fulfils the boundary condition.

Now $J_t^*(x(t),t) = \frac{1}{2}\dot{K}(t)x^2(t)$ and the H-J-B gives

$$0 = \frac{1}{2}\dot{K}(t)x^2(t) - K^2(t)x^2(t) + K(t)x^2(t)$$

That must be satisfied for all $x(t)$

$$\frac{1}{2}\dot{K}(t) - K^2(t) + K(t) = 0 \Rightarrow K(t) = \frac{e^{T-t}}{e^{T-t} + e^{-(T-t)}}$$

$$\Rightarrow u^*(t) = -2J_x^*(x(t),t) = -2K(t)x(t)$$

The solution is in the form of a state feedback control law.

Linear Regulator Problems

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

LQ problem, *Q* positive semi-definite, *R* positive definite

$$J = \frac{1}{2}x^T(t_f)Hx(t_f) + \int_{t_0}^{t_f} \frac{1}{2}\left[x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\right]dt$$

Form the Hamiltonian

$$H(x(t),u(t),J_x^*,t) = \frac{1}{2}x^T(t)Q(t)x(t) + \frac{1}{2}u^T(t)R(t)u(t) + J_x^*(x(t),t) \cdot \left[A(t)x(t) + B(t)u(t)\right]$$

and the necessary condition for optimality

$$\frac{\partial H}{\partial u}(x(t),u(t),J_x^*,t) = u^T(t)R(t) + J_x^*(x(t),t)B(t) = 0$$

Note that since $\frac{\partial^2 H}{\partial u^2} = R(t)$ is positive definite and *H* is a quadratic form in *u*, the optimum is global.

$$u^*(t) = -R^{-1}(t)B^T(t)J_x^{*T}(x(t),t)$$

$$\Rightarrow H(x(t),u^*(t),J_x^*,t) = \frac{1}{2}x^T Qx + \frac{1}{2}J_x^* BR^{-1}B^T J_x^{*T}$$
$$+ J_x^* Ax - J_x^* BR^{-1}B^T J_x^{*T}$$
$$= \frac{1}{2}x^T Qx - \frac{1}{2}J_x^* BR^{-1}B^T J_x^{*T} + J_x^* Ax$$

H-J-B: $0 = J_t^* + \frac{1}{2}x^T Qx - \frac{1}{2}J_x^* BR^{-1}B^T J_x^{*T} + J_x^* Ax$

Boundary condition $\quad J^*(x(t_f),t_f) = \frac{1}{2}x^T(t_f)Hx(t_f)$

Guess a solution of the form

$$J^*(x(t),t) = \frac{1}{2}x^T(t)K(t)x(t)$$

K symmetric, positive definitive matrix

and substitute into H-J-B

$$0 = \frac{1}{2}x^T\dot{K}x + \frac{1}{2}x^TQx - \frac{1}{2}x^TKBR^{-1}B^TKx + x^TKAx$$

But

$$x^TKAx = x^T(KA + (KA)^T - (KA)^T)x = x^T\left(KA + A^TK\right)x - x^TKAx$$

$$\Rightarrow x^TKAx = \frac{1}{2}\left(x^TKAx\right) + \frac{1}{2}\left(x^TA^TKx\right)$$

so that

$$0 = \frac{1}{2}x^T\dot{K}x + \frac{1}{2}x^TQx - \frac{1}{2}x^TKBR^{-1}B^TKx + \frac{1}{2}x^TKAx + \frac{1}{2}x^TA^TKx$$

This equation must hold for all $x(t)$, so that

$$0 = \dot{K}(t) + Q(t) - K(t)B(t)R^{-1}(t)B^{T}(t)K(t) + K(t)A(t) + A^{T}(t)K(t)$$

with the boundary condition

$$K(t_f) = H$$

This is of course the well-known Riccati equation with a boundary condition.

The optimal control becomes

$$u^*(t) = -R^{-1}(t)B^{T}(t)K(t)x(t)$$

It can be proven that the condition of optimality (in H-J-B) is not only *necessary*, but also *sufficient*.

To introduce co-states, take $p^T(t) = J_x^*(x(t), t)$

Results when control is unbounded and all signals differentiable

$$\dot{x}(t) = f(x(t), u(t), t), \quad x(t_0) = x_0$$

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

Take co-states (adjoint states) $p_j(t)$ and define the Hamiltonian

$$H(x, u, p, t) = g(x, u, t) + p^T f(x, u, t)$$

The necessary conditions for the solution x^*, u^* are

$$
\begin{cases}
\dot{p}^T(t) = -\dfrac{\partial H}{\partial x}(x^*, u^*, p^*, t) \\
\dfrac{\partial H}{\partial u}(x^*, u^*, p^*, t) = 0 \\
\dot{x}^* = f(x^*, u^*, p^*, t) = \left(\dfrac{\partial H}{\partial p}\right)^T
\end{cases}
$$

Boundary conditions:

1. $x(t_0) = x_0$

2. Free final state $p(t_f) = 0$
   Fixed final state $x(t_f) = x_f$
   Final state has the cost $h(x(t_f), t_f)$: $p(t_f) = \dfrac{\partial h}{\partial x}(t_f)$

Example

Minimize the performance measure $J(u) = \int_{t_0}^{t_f} \frac{1}{2} u^2(t) dt$

For the system $\dot{x}_1(t) = x_2(t)$

$$\dot{x}_2(t) = -x_2(t) + u(t)$$

- Control is unbounded, final state is required to lie in x(2)=[5 2]’

Solution

- Form the Hamiltonian

$$H(x(t), u(t), p(t)) = \frac{1}{2} u^2(t) + p_1(t) x_2(t) - p_2(t) x_2(t) + p_2(t) u(t)$$

Calculate the necessary conditions for optimality

$$\dot{p}_1^*(t) = -\frac{\partial H}{\partial x_1} = 0$$

$$\dot{p}_2^*(t) = -\frac{\partial H}{\partial x_2} = -p_1^*(t) + p_2^*(t)$$

$$\frac{\partial H}{\partial u} = u^*(t) + p_2^*(t) = 0$$

Solve the optimal control and substitute into state equations and solve the equations

$$x_1{}^*(t) = c_1 + c_2[1 - e^{-t}] + c_3[-t - \frac{1}{2}e^{-t} + \frac{1}{2}e^{t}] + c_4[1 - \frac{1}{2}e^{-t} - \frac{1}{2}e^{t}]$$

$$x_2{}^*(t) = c_2 e^{-t} + c_3[-1 + \frac{1}{2}e^{-t} + \frac{1}{2}e^{t}] + c_4[\frac{1}{2}e^{-t} - \frac{1}{2}e^{t}]$$

$$p_1{}^*(t) = c_3$$

$$p_2{}^*(t) = c_3[1 - e^{t}] + c_4 e^{t}$$

Solve the system parameters with respect to the boundary conditions

- $x(0)=[0\ 0]'$
- $x(2)=[5\ 2]'$

$$x_1^*(t)=7,289t-6,103+6,696e^{-t}-0,593e^{t}$$

$$x_2^*(t)=7,289-6,696e^{-t}-0,593e^{t}$$

Note that analytical solutions are seldom possible. Specifically, Two-Point-Boundary-Value-Problems (TPBVP) often occur (initial value for state and final value of co-state are known. "Shooting algorithms" can be used to find approximative solutions.

Summary:

Discrete-time case (this is relatively easy to derive starting from the Principle of Optimality (Dynamic Programming). See Lecture 8 of the course ELEC-E8101 Digital and Optimal Control).

$$x_{k+1} = A_k x_k + B_k u_k , \quad k > i$$

$$J_i = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=i}^{N-1} \left(x_k^T Q_k x_k + u_k^T R_k u_k \right)$$

$$S_N \geq 0, \quad Q_k \geq 0, \quad R_k > 0$$

Solution :

$$S_k = (A - BK_k)^T S_{k+1} (A - BK_k) + K_k^T R K_k + Q$$

$$K_k = \left(B_k^T S_{k+1} B_k + R_k\right)^{-1} B_k^T S_{k+1} A_k, \quad k < N$$

$$u_k = -K_k x_k, \quad k < N$$

$$J_i^* = \frac{1}{2} x_i^T S_i x_i$$

The Riccati equation can also be written in the form

$$S_k = A_k^T \left[S_{k+1} - S_{k+1} B_k \left(B_k^T S_{k+1} B_k + R_k \right)^{-1} B_k^T S_{k+1} \right] A_k + Q_k, \; k < N, \; S_N \text{ given}$$

Continuous-time case:

$$\dot{x} = Ax + Bu, \quad t \geq t_0$$

$$J(t_0) = \frac{1}{2} x^T(t_f) S(t_f) x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left(x^T Q x + u^T R u \right) dt$$

$$S(t_f) \geq 0, \quad Q \geq 0, \quad R > 0$$

Note. The matrices can also be time-varying, *A* = *A(t)* etc. like previously in the discrete case.

Riccati equation

$$-\dot{S}(t) = A^T S + SA - SBR^{-1}B^T S + Q, \quad t \le t_f,$$

boundary condition $S(t_f)$

$$K = R^{-1}B^T S$$

$$u = -Kx$$

$$J^*(t_0) = \frac{1}{2}x^T(t_0)S(t_0)x(t_0)$$

But what about the *servo problem*. How to get rid of the steady-state error?

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

The optimal control, when reference r is connected

$$u = -Lx + r$$

leads to the closed-loop system

$$\dot{x} = (A - BL)x + Br$$

The corresponding transfer function is

$$Y(s) = C\left[(sI - (A - BL))^{-1}\right]B\; R(s)$$

but the static gain

$$-C(A - BL)^{-1}B$$

is not necessarily one. If the reference is a known constant, a suitable (static) precompensator can be used, which makes the gain from r to z one.

But what if r varies? Solution: add integration to the system (controller), which removes the error.

How to add Integration?

Take a new state variable

$$x_{n+1}$$

such that

$$\dot{x}_{n+1} = r - y = r - Cx$$

An *augmented* state-space realization is obtained

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$$

Apply the state feeback to this

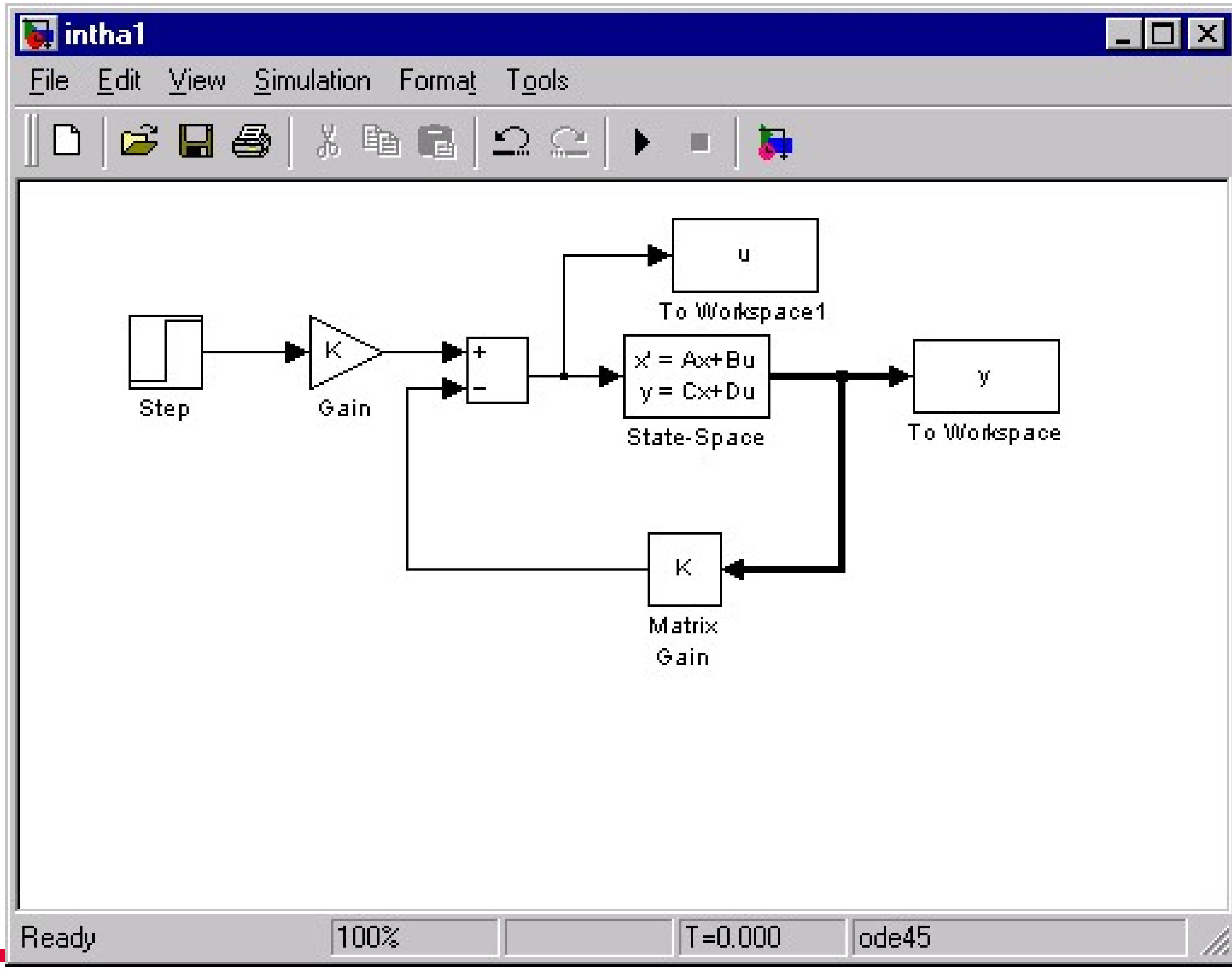
$$u = -\begin{bmatrix} L & l_{n+1} \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + r \qquad l_{n+1} \text{ is scalar}$$

The closed loop system is then

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A - BL & -Bl_{n+1} \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + \begin{bmatrix} B \\ 1 \end{bmatrix} r$$

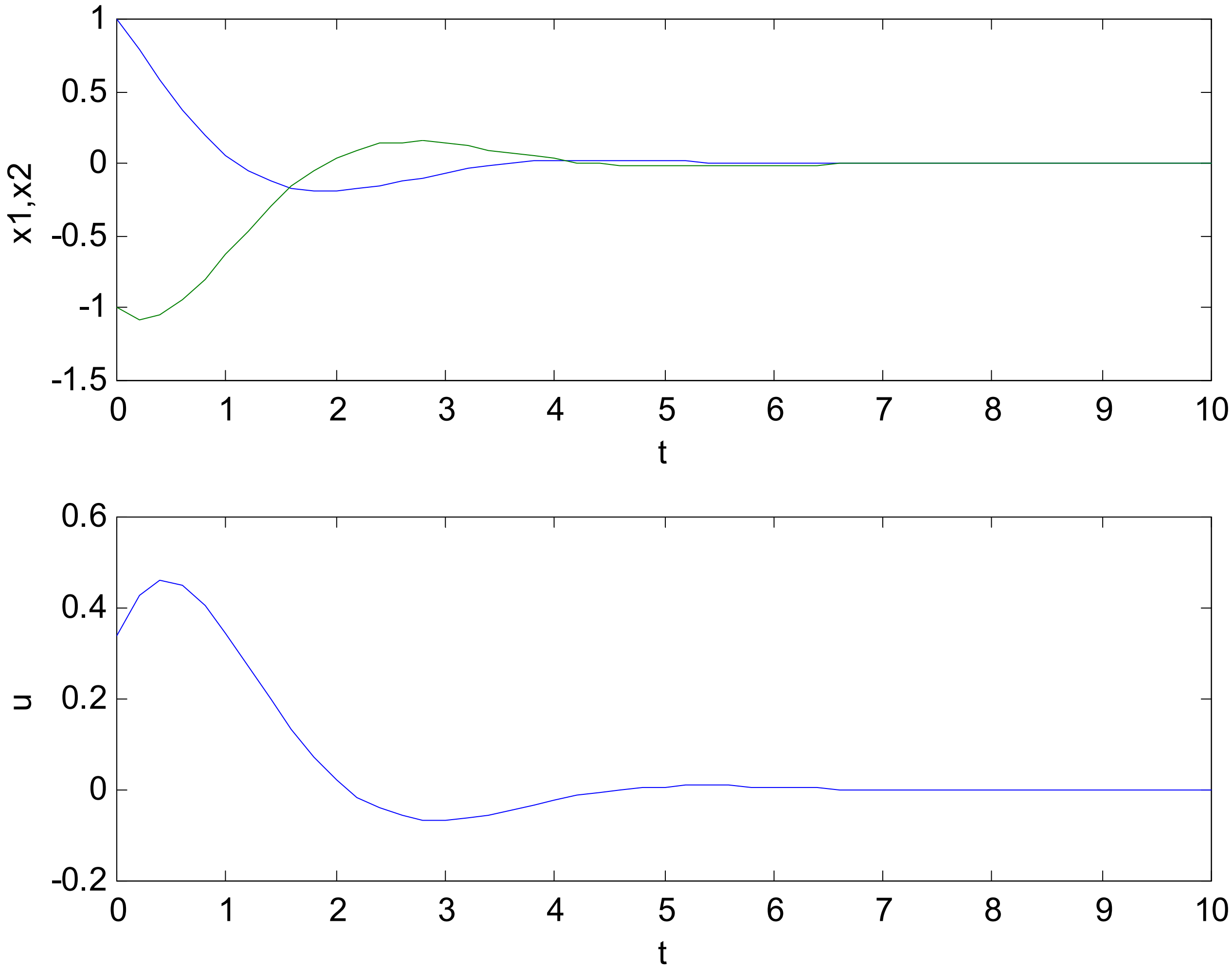
When the state moves to a constant value, the component $\dot{x}_{n+1}$ moves to the origin; then the output follows the reference. Note that this is a suboptimal solution.

Example.



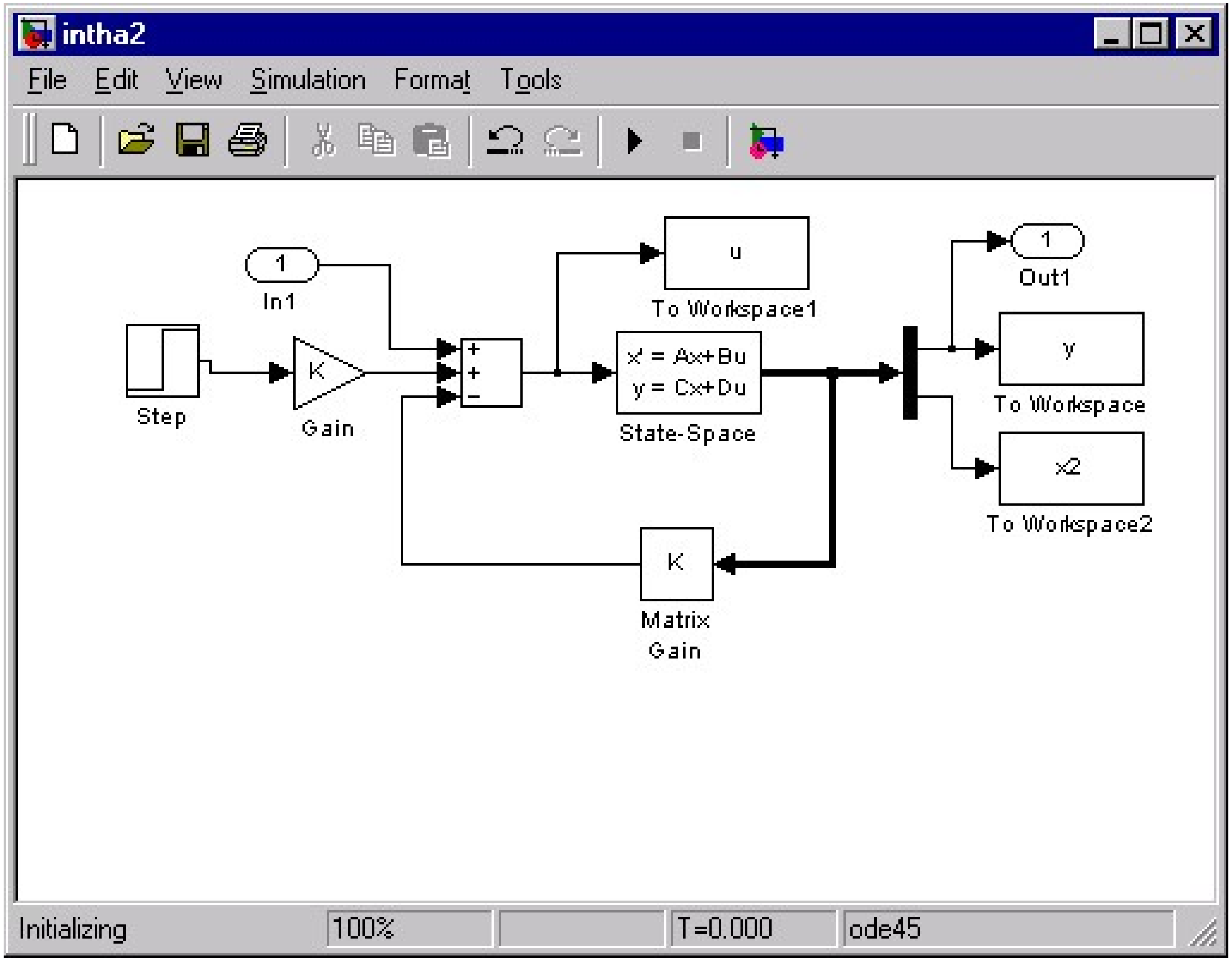
intha1
File Edit View Simulation Format Tools
u
To Workspace1
Step
K
Gain
+
−
x' = Ax+Bu
y = Cx+Du
State-Space
y
To Workspace
K
Matrix
Gain
Ready
100%
T=0.000
ode45

```
Q=[1 0;0 1];
R=1;
[L,S,E]=lqr(A,B,Q,R);
L = 0.2361    0.5723
S = 1.5158    0.2361
    0.2361    0.5723
E =-0.7862 + 1.2720i
   -0.7862 - 1.2720i
```



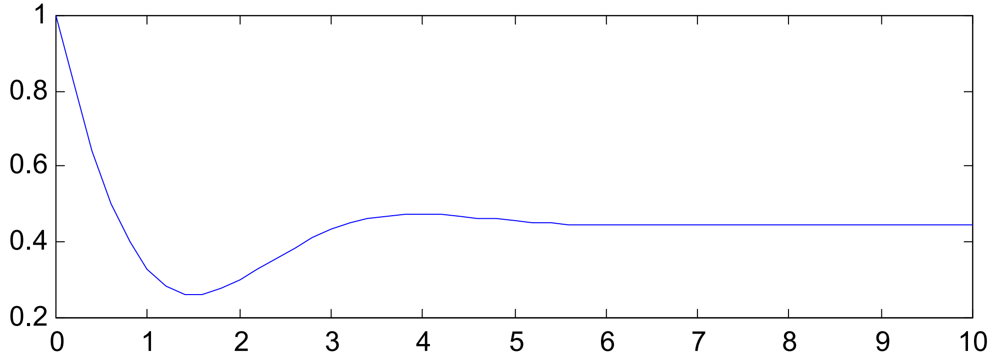
1
0.5
0
-0.5
-1
-1.5
x1,x2
0 1 2 3 4 5 6 7 8 9 10
t
0.6
0.4
0.2
0
-0.2
u
0 1 2 3 4 5 6 7 8 9 10
t

Reference is constant; calculate the static gain

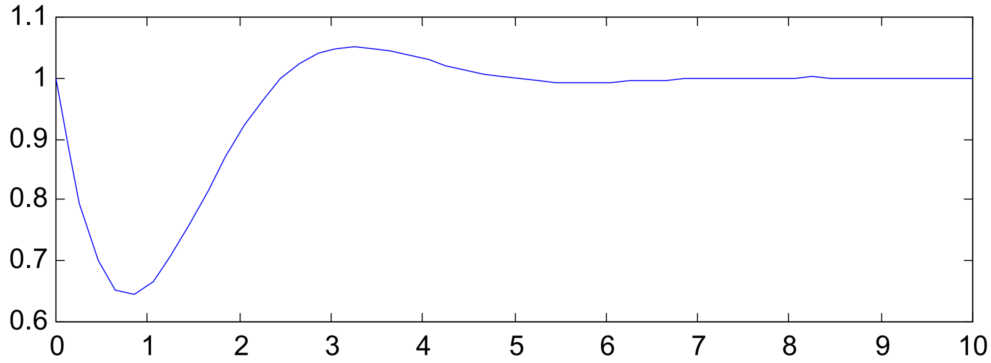


[A1,B1,C1,D1]=linmod('intha2')

K=1/dcgain(A1,B1,C1,D1)

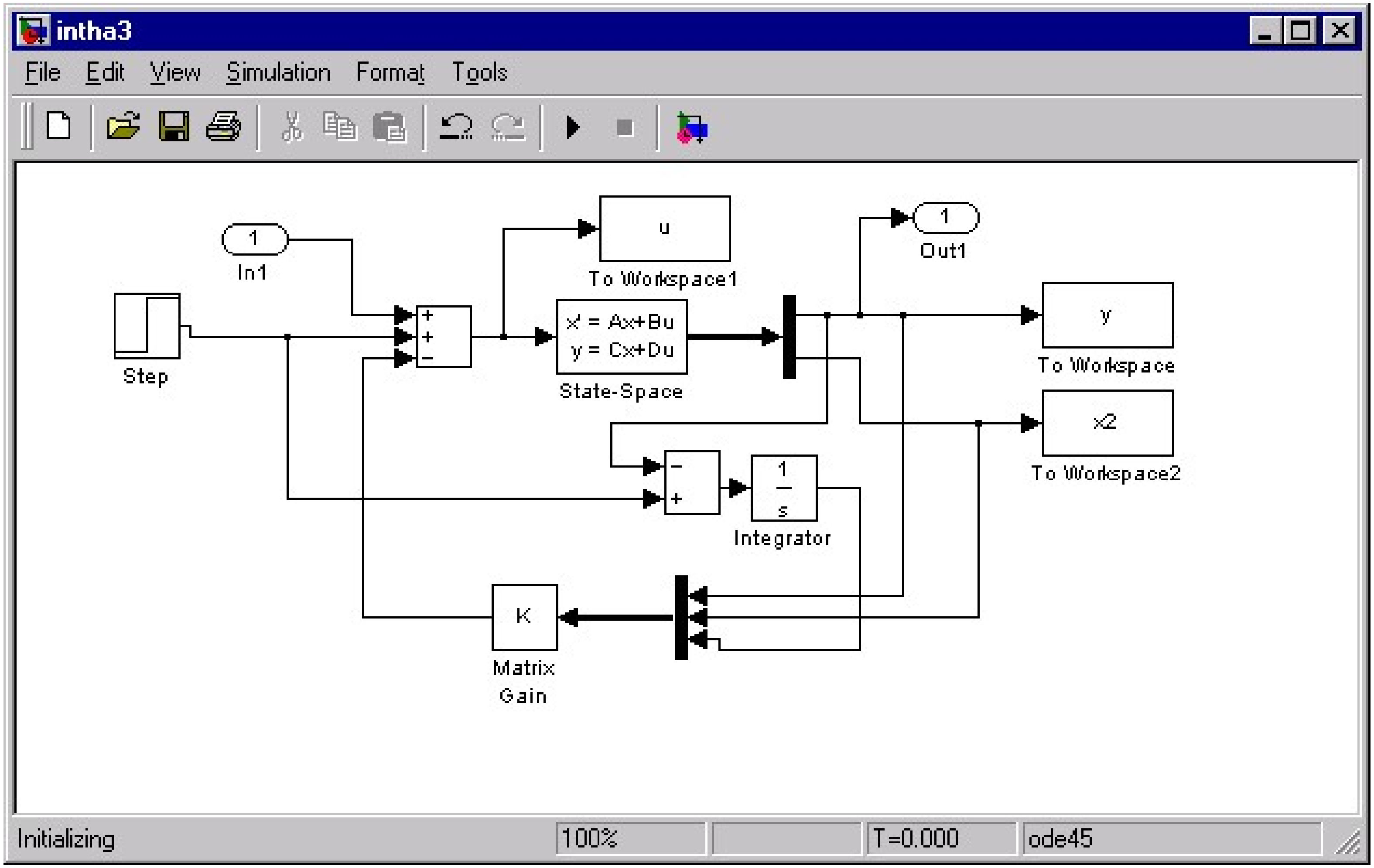


without pre-compensator

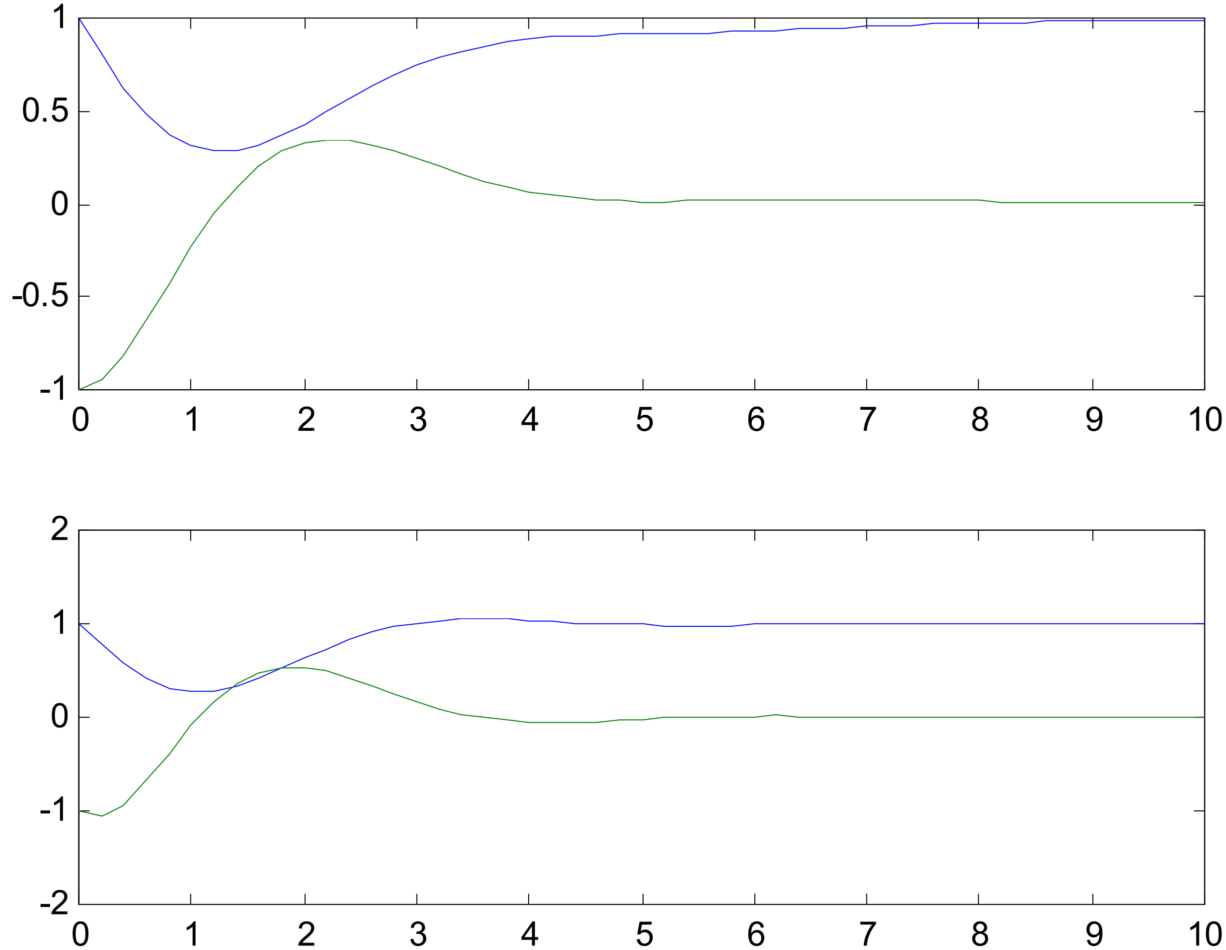


with pre-compensator

Adding an integrator



```
C2=[1 0];
A2=[A zeros(2,1);-C2 0];
B2=[B;0];
Q2=eye(3);
R2=1;
[L,S,E]=lqr(A2,B2,Q2,R2);
```



In the lower figure the component x_3 has been given more weight in the criterion.