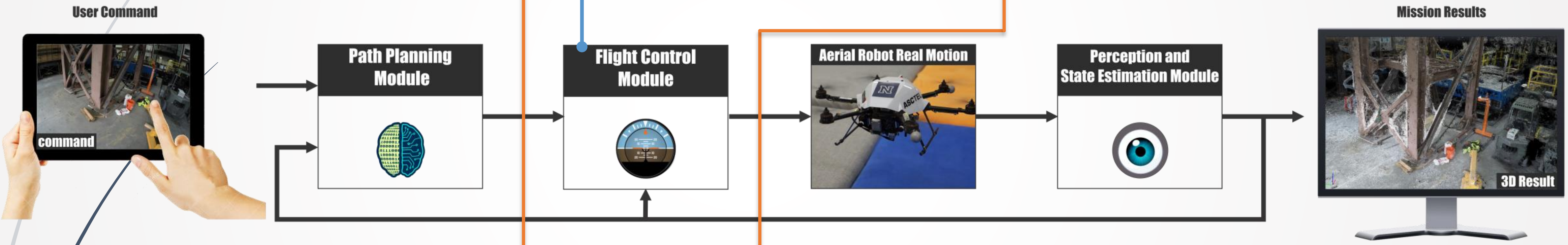
CS491/691: Introduction to Aerial Robotics

Topic: LQR Flight Control

Dr. Kostas Alexis (CSE)

The Aerial Robot Loop

- How to design the flight control of an aerial robot and program its autopilot.



Section 3 of our course

MAV Dynamics

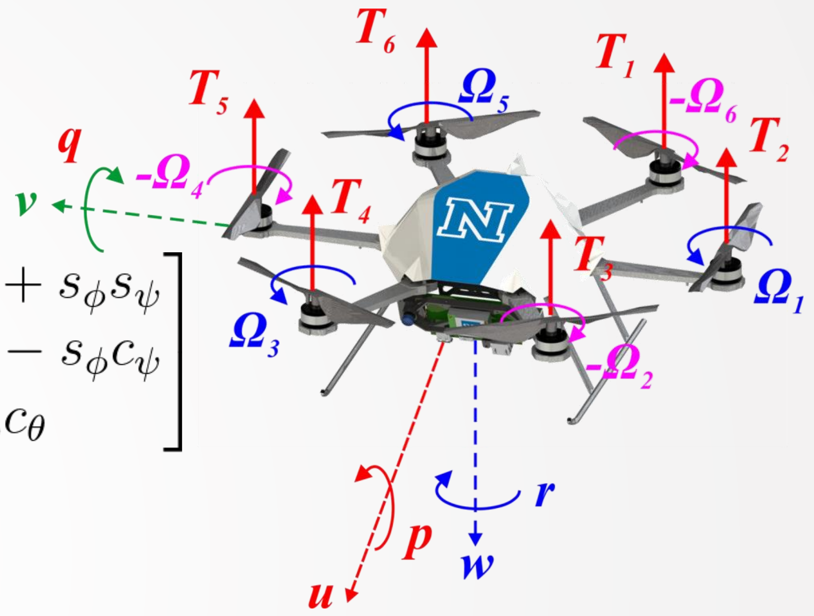
- To append the forces and moments we need to combine their formulation with

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = \mathcal{R}_b^v \begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad \mathcal{R}_b^v = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m}\begin{bmatrix}f_x\\ f_y\\ f_z\end{bmatrix}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} \end{bmatrix} \begin{bmatrix}p\\ q\\ r\end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} M_x \\ \frac{1}{J_y} M_y \\ \frac{1}{J_z} M_z \end{bmatrix}$$

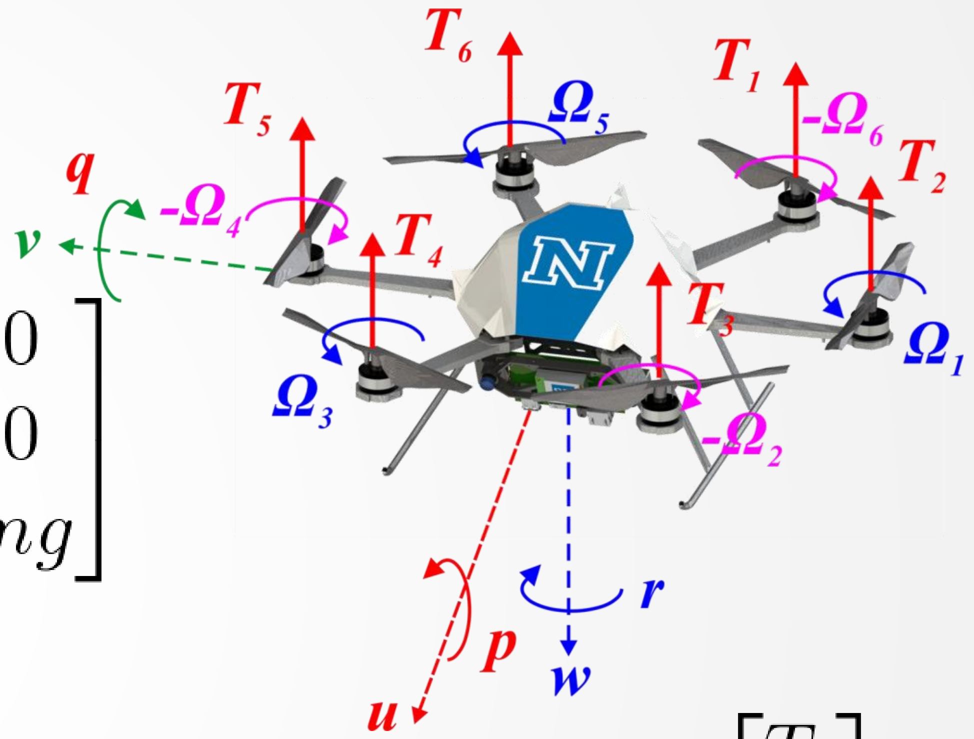


- ***Next step: append the MAV forces and moments***

MAV Dynamics

- MAV forces in the body frame:

$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$



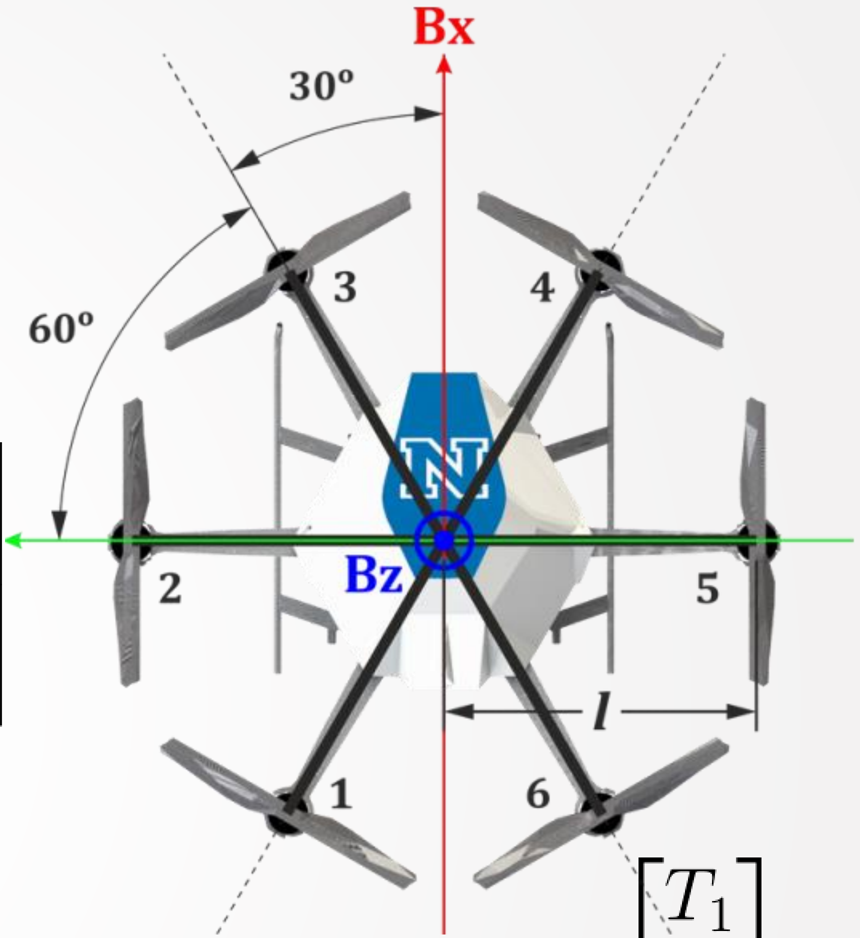
- Moments in the body frame:

$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & ls_{30} \\ -lc_{60} & 0 & lc_{60} & lc_{60} & 0 & -lc_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

MAV Dynamics

- MAV forces in the body frame:

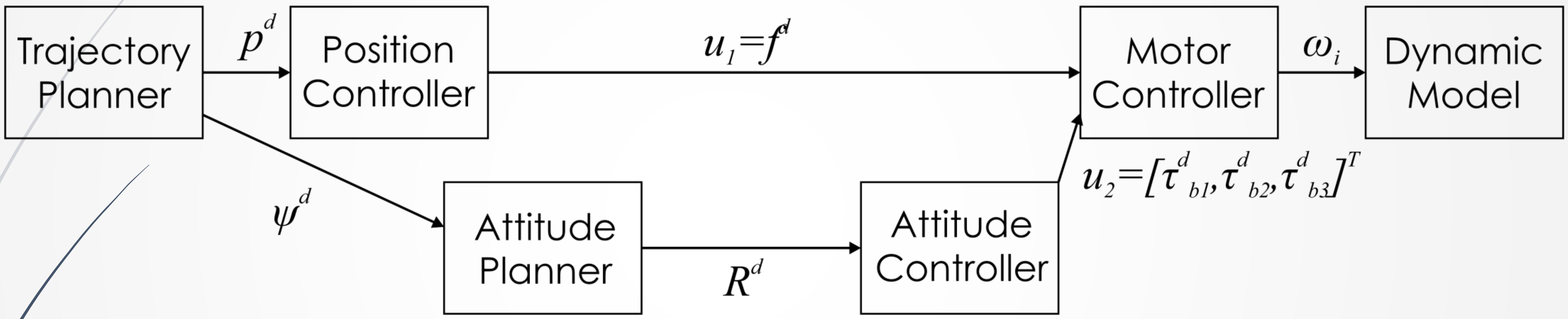
$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$



- Moments in the body frame:

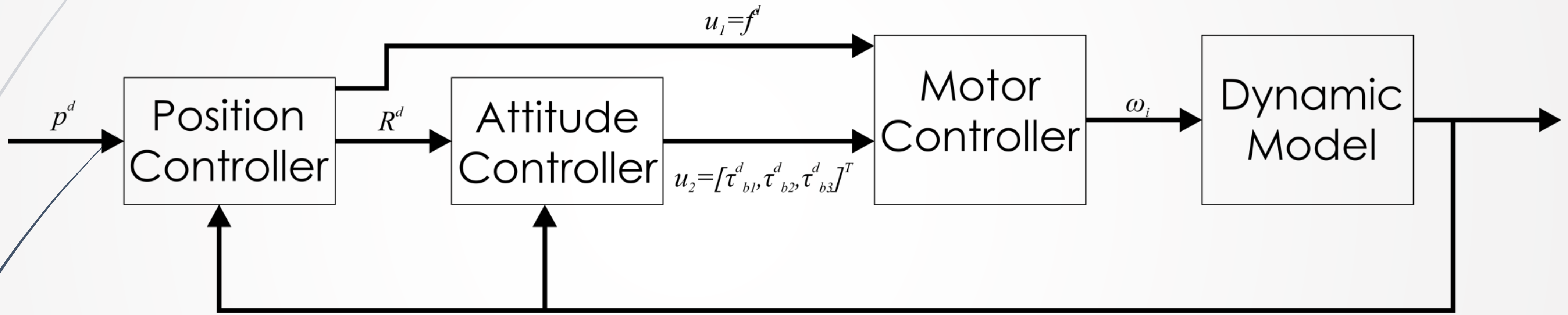
$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & ls_{30} \\ -lc_{60} & 0 & lc_{60} & lc_{60} & 0 & -lc_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

Control System Block Diagram



- There are simpler

Control System Block Diagram



- Simplified loop

Controlling a Multirotor along the x-axis

- Assume a single-axis multirotor.
  - The system has to coordinate its pitching motion and thrust to move to the desired point ahead of its axis.
  - Roll is considered to be zero, yaw is considered to be constant. No initial velocity. No motion is expressed in any other axis.
  - A system of only two degrees of freedom.

Controlling a Multirotor along the x-axis

- Simplified linear dynamics

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J_y \end{bmatrix} M_y$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -g \end{bmatrix} \theta$$

How does this system behave?

Optimal Model-Based Control

- Use model knowledge to design optimal control behaviors.
- The employed model must be simultaneously sufficiently accurate but also simple enough to enable efficient control computation.
- Optimal control can support linear and nonlinear systems as well as systems subject to state, output and input constraints. Further extensions (e.g. for hybrid systems) also exist.
- Established method for unconstrained linear systems regulation:
  - Linear Quadratic Regulator (LQR)
  - Generalization: Linear Quadratic Gaussian (LQG) control

Linear Quadratic Regulator

- Consider the system

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

- and suppose we want to design state feedback control $u=Fx$ to stabilize the system. The design of F is a trade-off between the transit response and the control effort. The optimal control approach to this design trade-off is to define the performance index (cost functional):

$$J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]\, dt$$

- and search for the control $u=Fx$ that minimizes this index. Q is an nxm symmetric positive semidefinite matrix and R is an mxm symmetric positive definite matrix.

Linear Quadratic Regulator

- The matrix $\boldsymbol{Q}$ can be written as $\boldsymbol{Q=M^TM}$, where $\boldsymbol{M}$ is a $\boldsymbol{p}$x$\boldsymbol{n}$ matrix, with $\boldsymbol{p\leq n}$. With this representation:

$$\boldsymbol{x^T Q x = x^T M^T M x = z^T z}$$

- Where $\boldsymbol{z=Mx}$ can be viewed as a controlled input.

- Optimal Control Problem: Find $\boldsymbol{u(t)=Fx(t)}$ to maximize $\boldsymbol{J}$ subject to the model:

$$\boldsymbol{\dot{x} = Ax + Bu}$$

- Since $\boldsymbol{J}$ is defined by an integral over $[\mathbf{0}, \infty)$, the first question we need to address is: Under what conditions will $\boldsymbol{J}$ exist and be finite?

Linear Quadratic Regulator

- Write J as:

$$J = \lim_{t_f \to \infty} \bar{J}(t_f)$$

$$\bar{J}(t_f) = \int_0^{t_f} \left[x^T(t) Q x(t) + u^T(t) R u(t) \right] dt$$

- $\bar{J}(t_f)$ is a monotonically increasing function of t_f. Hence, as $t_f \to \infty$, $\bar{J}(t_f)$ either converges to a finite limit or diverges to infinity.

- Under what conditions will $J = \lim_{t_f \to \infty} \bar{J}(t_f)$ be finite?

Linear Quadratic Regulator

- Recall that ***(A,B)*** is stabilizable if the uncontrollable eigenvalues of A, if any, have negative real parts.
- Notice that ***(A,B)*** is stabilizable if ***(A,B)*** is controllable or $R_e[\lambda(A)] < 0$
- Definition: ***(A,C)*** is detectable if the observable eigenvalues of A, if any, have negative real parts.
- Lemma 1: Suppose ***(A,B)*** is stabilizable, ***(A,M)*** is detectable, where $Q=M^TM$, and $u(t)=Fx(t)$. Then, J is finite for every $x(0) \in R^n$ if and only if:

$$R_e[\lambda(A + BF)] < 0$$

Linear Quadratic Regulator

- **Remarks:**
  - The need for ***(A,B)*** to be stabilizable is clear, for otherwise there would be no ***F*** such that:
  - To see why detectability of ***(A,M)*** is needed, consider:

$$\dot{x} = x + u, \quad J = \int_0^\infty u^2(t)\, dt$$

$$A = 1, \;\; B = 1, \;\; M = 0, \;\; R = 1$$

  - ***(A,B)*** is controllable, but ***(A,M)*** is not detectable

$$F = 0 \;\Rightarrow\; u(t) = 0 \;\Rightarrow\; J = 0$$

  - The control is clearly optimal and results in a finite ***J*** but it does not stabilize the system because $A + BF = A = 1$

Linear Quadratic Regulator

- **Lemma 2:** For any stabilizing control $u(t)=Fx(t)$, the cost given by:

$$J = x(0)^T W x(0)$$

- where W is a symmetric positive semidefinite matrix that satisfies the Lyapunov equation:

$$W(A+BF)+(A+BF)^T W + Q + F^T R F = 0$$

- Remark: The control $u(t)=Fx(t)$ is stabilizing if:

$$R_e[\lambda(A+BF)] < 0$$

Linear Quadratic Regulator

- **Theorem:** Consider the system: $\dot{x} = Ax + Bu$ and the performance index:

$$J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]\, dt$$

- where $Q = M^T M$, R is symmetric and positive definite, (A,B) is stabilizable, and (A,M) is detectable. The optimal control is:

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

- where P is the symmetric positive semidefinite solution of the Algebraic Riccati Equation (ARE):

$$0 = PA + A^T P + Q - PBR^{-1}B^T P$$

Linear Quadratic Regulator

- **Remarks:**
  - Since the control is stabilizing:

$$R_e[\lambda(A - BR^{-1}B^T P)] < 0$$

  - The control is optimal among all square integratable signals ***u(t)***, not just among ***u(t)=Fx(t)***
  - The Ricatti equation can have multiple solutions, but only one of them is positive semidefinite.

Linear Quadratic Regulator

- **Typical penalty matrices selection:**

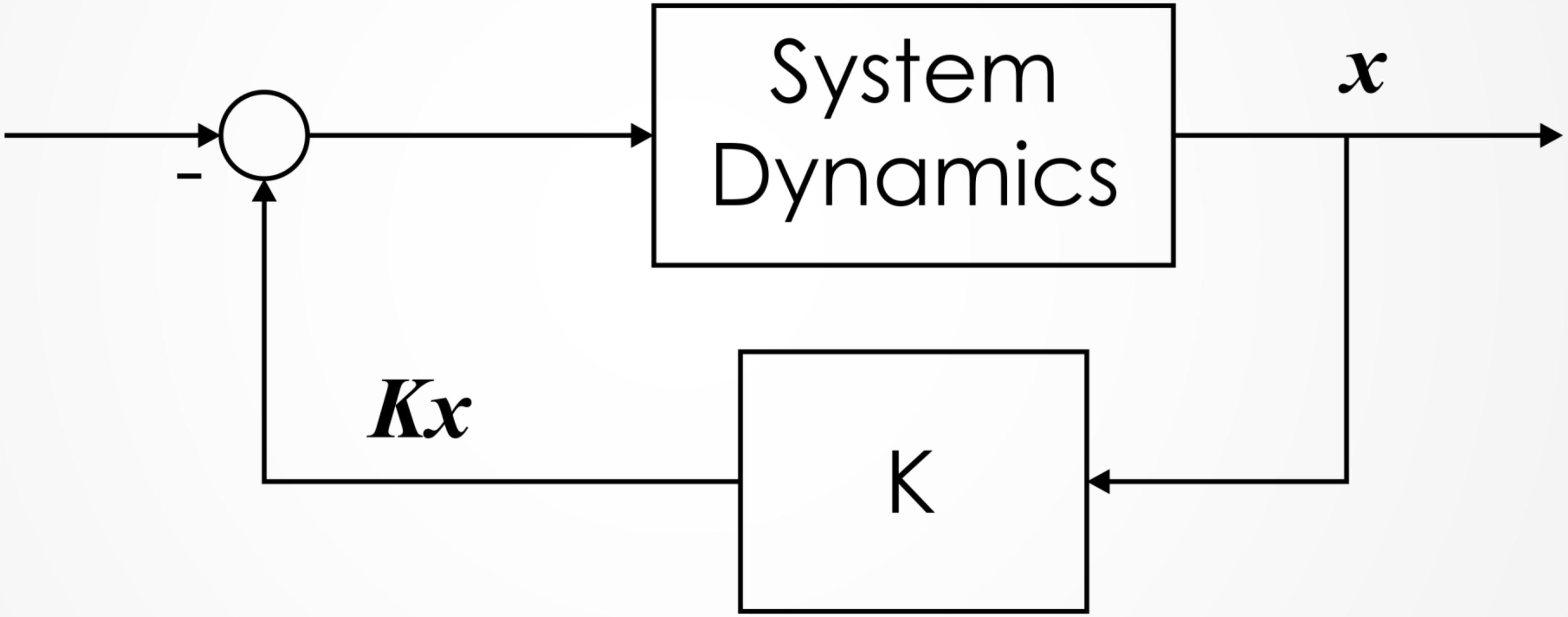
$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_n \end{bmatrix}, \quad R = \rho \begin{bmatrix} r_1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_m \end{bmatrix}$$

$$q_i = \frac{1}{t_{si}(x_{imax})^2}, \quad r_i = \frac{1}{(u_{imax})^2}, \quad \rho > 0$$

- t_{si} is the desired settling time of xi
- x_{imax} is a constraint on $|x_i|$
- u_{imax} is a constraint on $|u_i|$
- ρ is chosen to tradeoff regulation versus control effort

Linear Quadratic Regulator

- **LQR Loop (F=K)**



MATLAB Design Example

- Recall:

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 1/J_y\end{bmatrix} M_y$$

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ -g\end{bmatrix} \theta$$

MATLAB Design Example

```
%%  Simple Modeling and Control study
clear;
J_y = 1.2e-5;
g = 9.806; mass = 1.2;

%   Pitch Linear Model
A_p = [0 1; 0 0]; B_p = [0; 1/J_y];
C_p = eye(2); D_p = zeros(2,1);
ss_pitch = ss(A_p,B_p,C_p,D_p);

%   x Linear Model
A_x = [0 1; 0 0]; B_x = [0; -g];
C_x = eye(2); D_x = zeros(2,1);
ss_x = ss(A_x,B_x,C_x,D_x);

%   Observe the Step responses of the system
subplot(1,2,1); step(ss_pitch);
subplot(1,2,2); step(ss_x);
```

MATLAB Design Example

```
%%  System discretization
Ts_pitch = 0.005;
Ts_trans = 0.01;
ss_pitch_d = c2d(ss_pitch,Ts_pitch,'zoh');
ss_x_d = c2d(ss_x,Ts_trans,'zoh');
```

MATLAB Design Example

```
%%  Design the LQR Controller for Pitch

Q_pitch = diag([1000, 10]);
R_pitch = 1;
[K_pitch,S_pitch,e_pitch] = dlqr(ss_pitch_d.A, ss_pitch_d.B, Q_pitch, R_pitch);
```

```
[K,S,e] = dlqr(A,B,Q,R,N)
```

MATLAB Design Example

```
%%  Design the LQR Controller for Pitch

Q_pitch = diag([1000, 10]);
R_pitch = 1;
[K_pitch,S_pitch,e_pitch] = dlqr(ss_pitch_d.A, ss_pitch_d.B, Q_pitch, R_pitch);
```

```
%%  Design the LQR Controller for the translational X-Dynamics

Q_x = diag([100, 1]);
R_x = 1;
[K_x,S_x,e_x] = dlqr(ss_x_d.A, ss_x_d.B, Q_x, R_x)
```

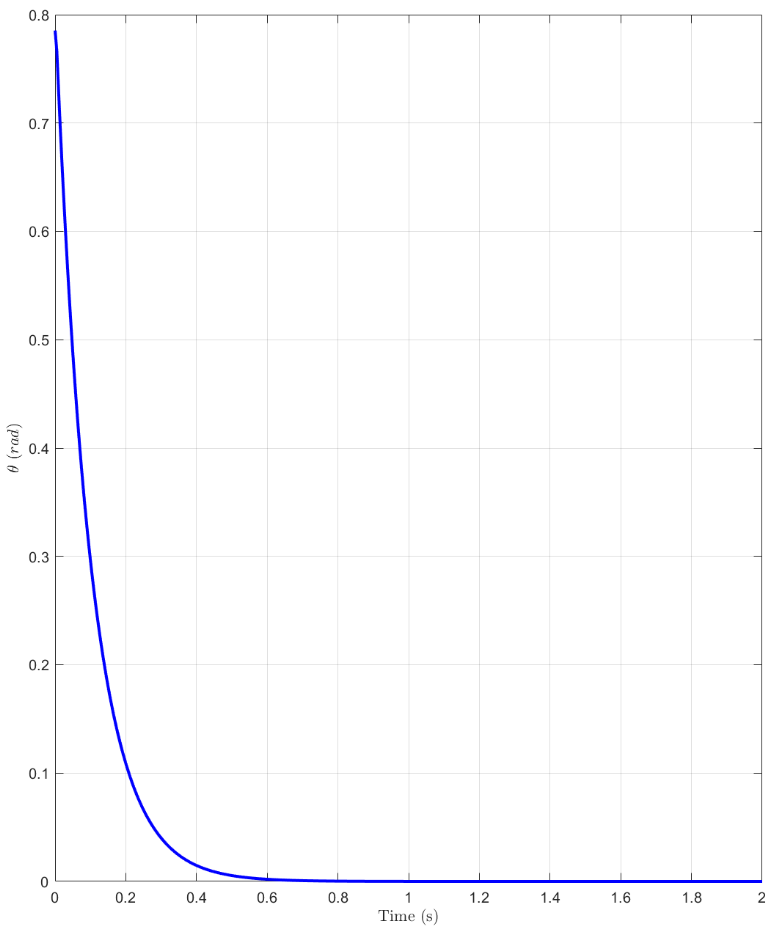
MATLAB Design Example

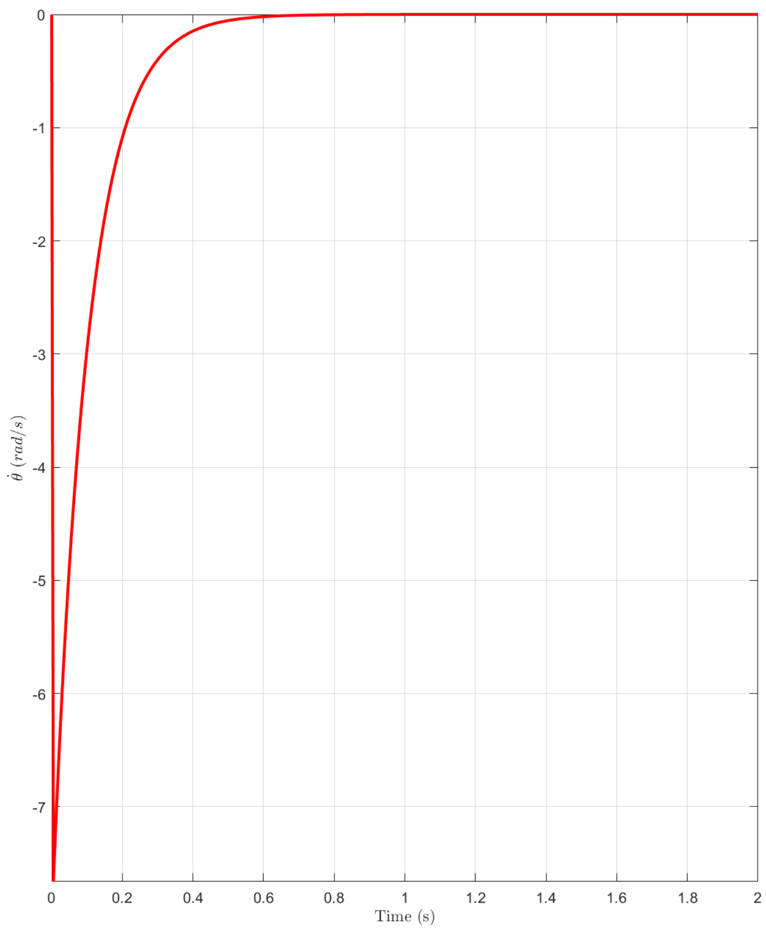
```
%%  Simulate the closed-loop response for pitch

ss_pitch_cl_d = feedback(ss_pitch_d,-K_pitch,1);
t_pitch = 0:0.005:2;
u_pitch = zeros(1,length(t_pitch));
x_pitch_0 = [pi/4,0];
x_pitch = lsim(ss_pitch_cl_d,u_pitch,t_pitch,x_pitch_0);

subplot(1,2,1); plot(t_pitch,x_pitch(:,1),'b','LineWidth',2);
xlabel('Time (s)','Interpreter','LaTex','FontSize',22); ylabel('$$\theta~(rad)$$','Interpreter','LaTex','FontSize',22); grid on;
subplot(1,2,2); plot(t_pitch,x_pitch(:,2),'r','LineWidth',2);
xlabel('Time (s)','Interpreter','LaTex','FontSize',22); ylabel('$$\dot \theta~(rad/s)$$','Interpreter','LaTex','FontSize',22); grid on;
axis tight;
```

MATLAB Design Example





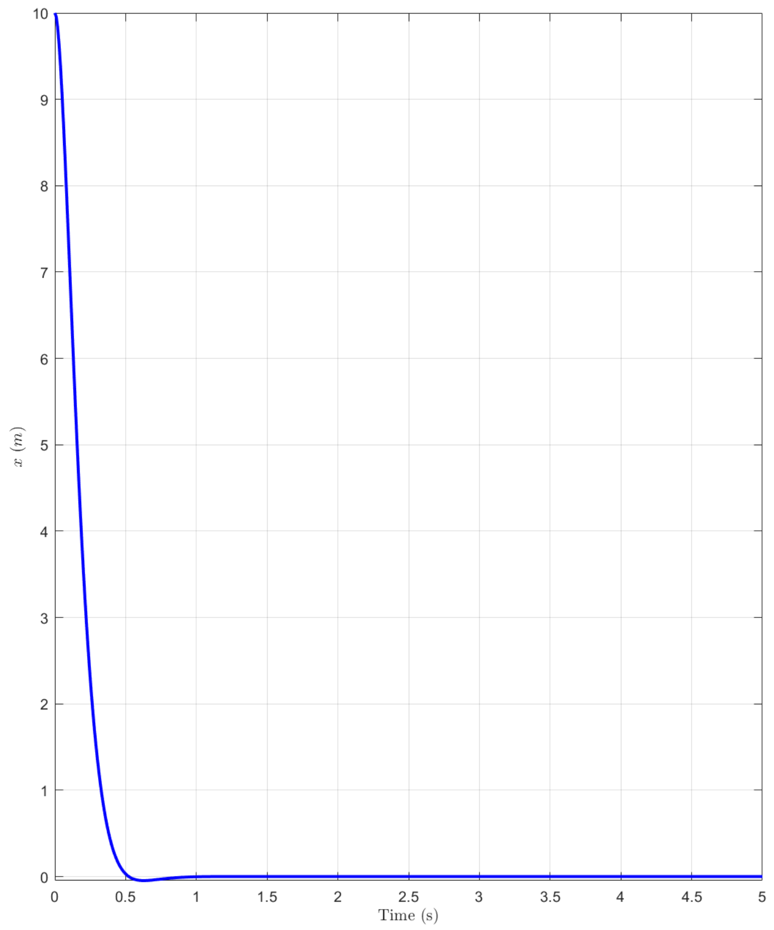
MATLAB Design Example

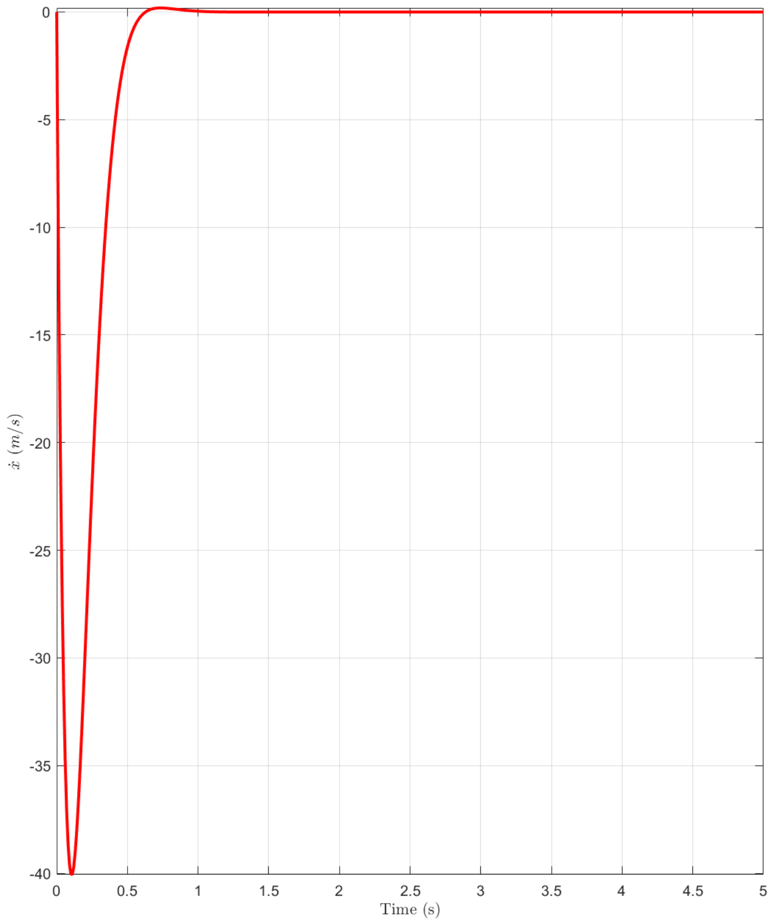
```
%%  Simulate the closed-loop response for pitch

ss_x_cl_d = feedback(ss_x_d,-K_x,1);
t_x = 0:0.01:5;
u_x = zeros(1,length(t_x));
x_x_0 = [10,0];
x_x = lsim(ss_x_cl_d,u_x,t_x,x_x_0);

subplot(1,2,1); plot(t_x,x_x(:,1),'b','LineWidth',2);
xlabel('Time (s)','Interpreter','LaTex','FontSize',22); ylabel('$$x~(m)$$','Interpreter','LaTex','FontSize',22); grid on;
axis tight
subplot(1,2,2); plot(t_x,x_x(:,2),'r','LineWidth',2);
xlabel('Time (s)','Interpreter','LaTex','FontSize',22); ylabel('$$\dot x~(m/s)$$','Interpreter','LaTex','FontSize',22); grid on;
axis tight
```

MATLAB Design Example





MATLAB Aircraft Pitch Example

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} u_{elev}$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix}$$

For a step reference of 0.2 radians, the design criteria are the following.

1. Overshoot less than 10%
2. Rise time less than 2 seconds
3. Settling time less than 10 seconds
4. Steady-state error less than 2%

MATLAB Aircraft Pitch Example

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} u_{elev}$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix}$$

For a step reference of 0.2 radians, the design criteria are the following.

1. Overshoot less than 10%
2. Rise time less than 2 seconds
3. Settling time less than 10 seconds
4. Steady-state error less than 2%

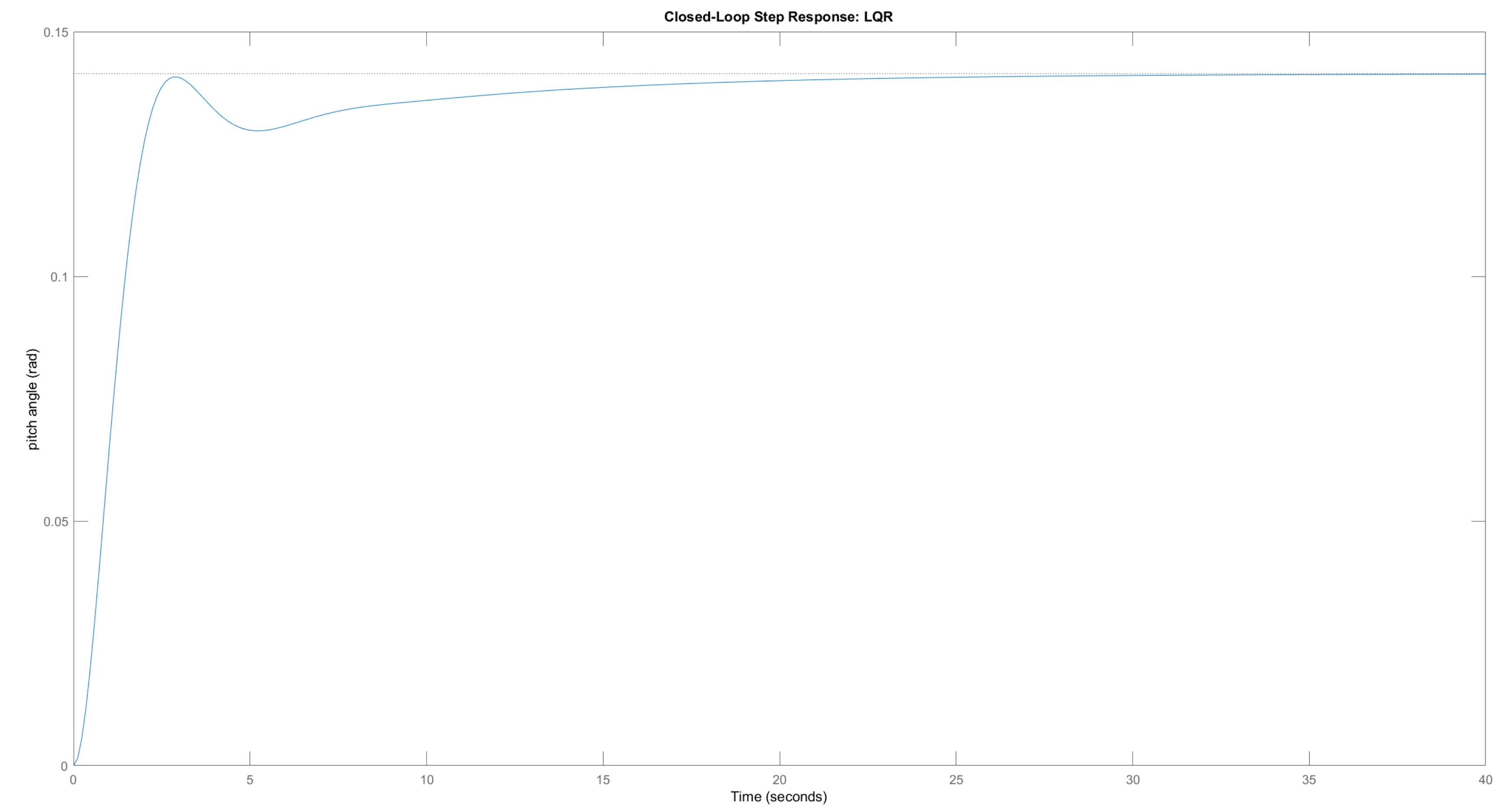
MATLAB Aircraft Pitch Example

```
%%  LQR Design

p = 2;
Q = p*C'*C;
R = 1;
[K] = lqr(A,B,Q,R);

sys_cl = ss(A-B*K, B, C, D);
step(0.2*sys_cl)
ylabel('pitch angle (rad)');
title('Closed-Loop Step Response: LQR');
```

MATLAB Aircraft Pitch Example

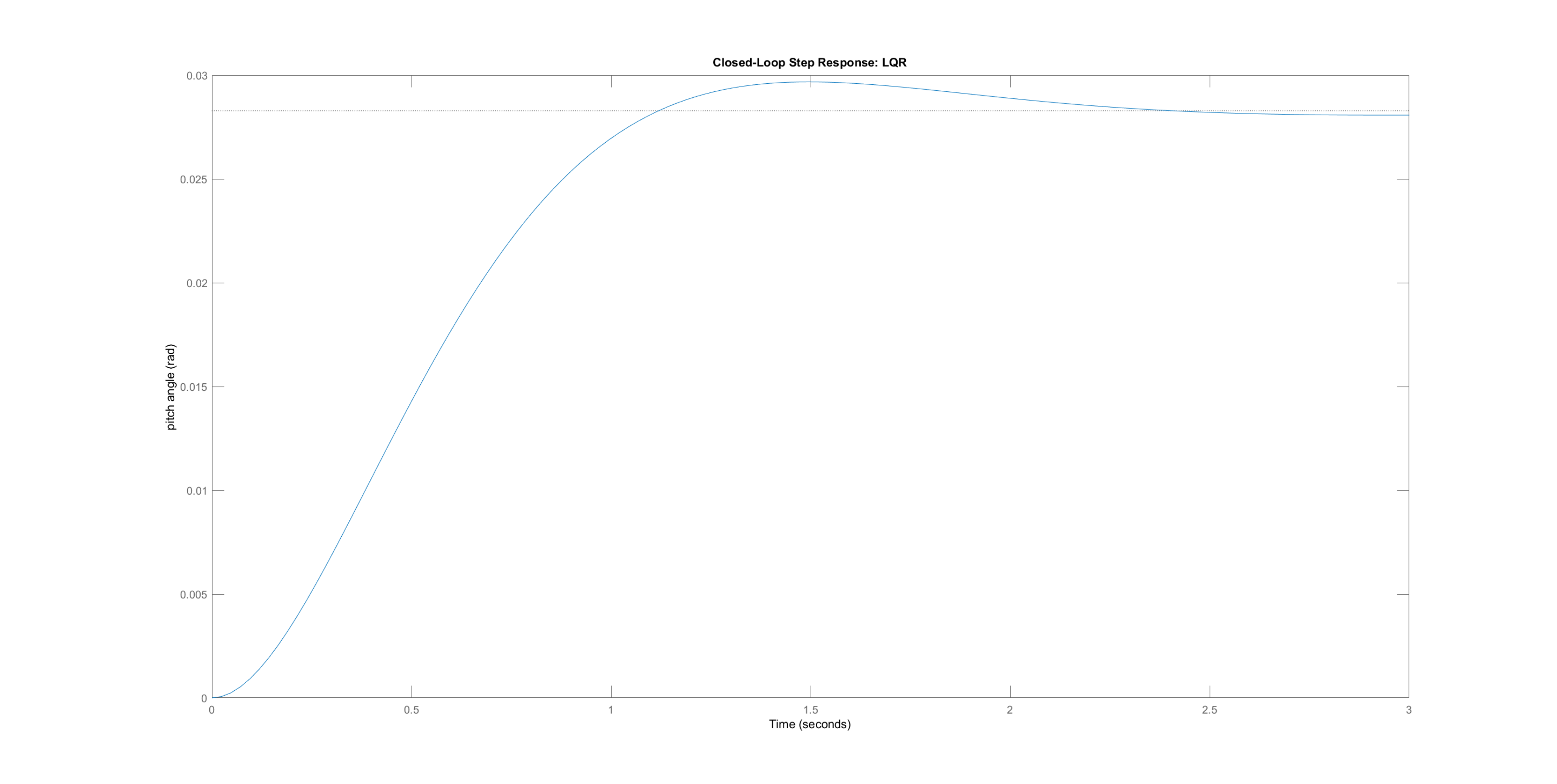


MATLAB Aircraft Pitch Example

```
%%  Closed-loop Simulation

p = 50;
Q = p*C'*C;
R = 1;
[K] = lqr(A,B,Q,R);
sys_cl = ss(A-B*K, B, C, D);
step(0.2*sys_cl)
ylabel('pitch angle (rad)');
title('Closed-Loop Step Response: LQR');
```

MATLAB Aircraft Pitch Example



MATLAB Aircraft Pitch Example

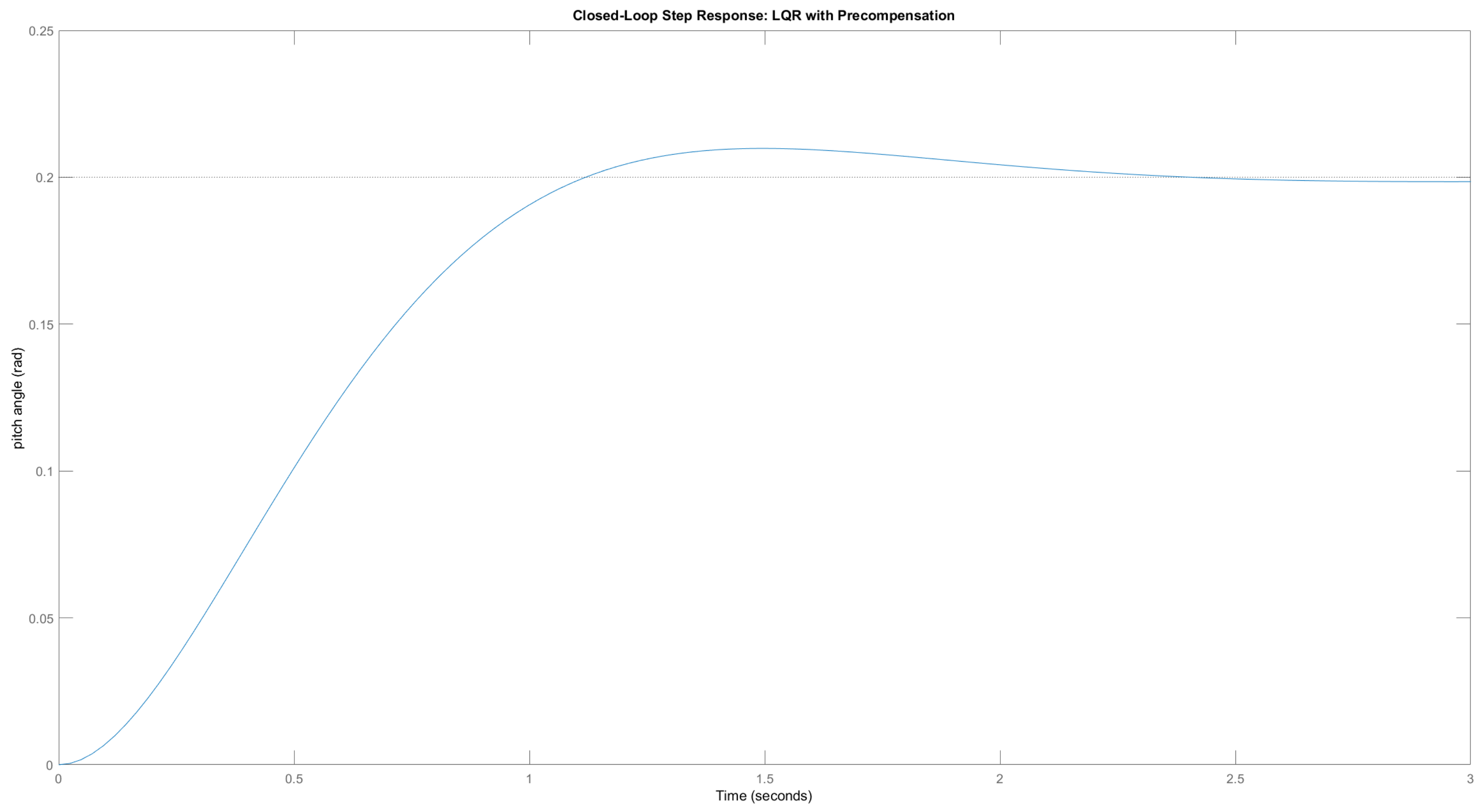
```
%%  Adding Precompensation

p = 50;
Q = p*C'*C;
R = 1;
[K] = lqr(A,B,Q,R);
Nbar = rscale(A,B,C,D,K);
```

MATLAB Aircraft Pitch Example

```
%%  Closed-loop Simulation with Precompensation
sys_cl = ss(A-B*K,B*Nbar,C,D);
step(0.2*sys_cl)
ylabel('pitch angle (rad)');
title('Closed-Loop Step Response: LQR with Precompensation');
```

MATLAB Aircraft Pitch Example



Find out more

- http://www.kostasalexis.com/pid-control.html
- http://www.kostasalexis.com/lqr-control.html
- http://www.kostasalexis.com/linear-model-predictive-control.html
- http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace

- http://www.kostasalexis.com/literature-and-links.html

Thank you!

Please ask your question!