Robust Control Design

Model-based control designs require system models

- System models are approximation of actual physical plants
- There are generally model uncertainties (mismatches)

Question:

- Is the designed controller robust against model uncertainties?
  - If the designed controller can tolerate the model mismatch, the controller is called “robust”
    - The controlled system’s performance does not degrade significantly in the presence of model mismatch using robust controller

Some robust control techniques:

- Linear quadratic Gaussian (LQG) control
- Loop transfer recovery (LTR)
- H_2 and H_∞ control

Linear Quadratic Gaussian (LQG) Control

System model:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\xi \xi(t) \\ y(t) = Cx(t) + \theta(t) \end{cases}$$

where $\xi(t)$ and $\theta(t)$ are zero-mean Gaussian random processes (noise) with symmetric covariance matrices $Q_f = E\{\xi(t)\xi^T(t)\} \geq 0$ and $R_f = E\{\theta(t)\theta^T(t)\} > 0$, and that $\xi(t)$ and $\theta(t)$ are mutually independent: $E\{\xi(t)\theta^T(t)\} = 0$

Performance index of optimal control:

$$J = E\left\{\int_0^\infty [z^T(t)Q\, z(t) + u^T(t)R\, u(t)]dt\right\}$$

where $z(t) = Mx(t)$, $Q = Q^T \geq 0$, and $\mathrm{R} = R^T > 0$

The LQG problem is divided into:

1. LQ optimal state feedback control
2. State estimation with disturbance

LQG with Kalman Filter

Optimal state estimator (Kalman filter)

System model:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\xi \xi(t) \\ y(t) = Cx(t) + \theta(t) \end{cases}$$

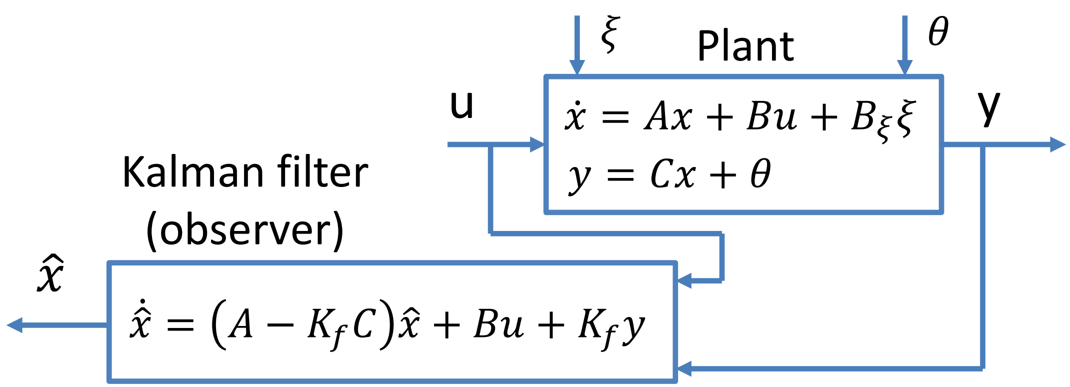
$\xi(t)$ and $\theta(t)$ are zero-mean noise with covariance matrices $Q_f = Q_f{}^T \geq 0$ and $R_f = R_f{}^T > 0$

Performance index:

$$J = \lim_{t\to\infty} E\{\tilde{x}(t)^T C^T C \tilde{x}(t)\}, \quad \tilde{x} = x - \hat{x}\,, \quad \hat{x} \text{ = estimate of } x$$

Kalman filter:

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases}$$



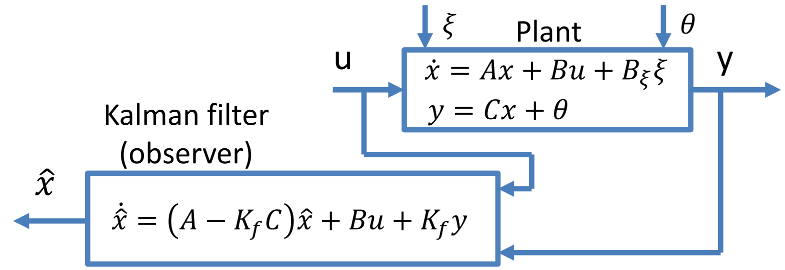
Kalman gain:

$$K_f = P_f C^T R_f{}^{-1}$$

where $P_f = P_f{}^T \geq 0$ is the solution of the filter algebraic Riccati equation (FARE):

$$P_f A^T + AP_f + B_\xi Q_f B_\xi{}^T - P_f C^T R_f{}^{-1} C P_f = 0$$

Optimal state estimator (Kalman filter)



Matlab function: kalman(),

Syntax: $[G_k, K_f, P_f]$=kalman(G, Q_f, R_f)

where $G = [A, \tilde{B}, C, \tilde{D}]$ is the extended state-space model of the system with $\tilde{B} = [B, B_\xi]$, $\tilde{D} = [D, D]$, G_k is the state-space model of the Kalman filter, and K_f and P_f are the Kalman gain and the solution to FARE

Example: For $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\xi \xi(t) \\ y(t) = Cx(t) + \theta(t) \end{cases}$, $A = \begin{bmatrix} -0.02 & 0.005 & 2.4 & -32 \\ -0.14 & 0.44 & -1.3 & -30 \\ 0 & 0.018 & -1.6 & 1.2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0.14 \\ 0.36 \\ 0.35 \\ 0 \end{bmatrix}$, $B_\xi = \begin{bmatrix} -0.12 \\ -0.86 \\ 0.009 \\ 0 \end{bmatrix}$,

$C = [0 \quad 1 \quad 0 \quad 0]$, $Q_f = 10^{-3}$, and $R_f = 10^{-7}$, design a Kalman filter.

Matlab code:

```
clear all, clc,
A=[-0.02,0.005,2.4,-32;-0.14,0.44,-1.3,-30;
0,0.018,-1.6,1.2;0,0,1,0]; B=[0.14;0.36;0.35;0];
Bz=[-0.12;-0.86;0.009;0]; C=[0,1,0,0]; D=[0];
G=ss(A,[B,Bz],C,[D,D]); Qf=1e-3; Rf=1e-7;
[Gk,Kf,Pf]=kalman(G,Qf,Rf),
```

Solution:

$$K_f = \begin{bmatrix} 215.33 \\ 87.371 \\ -2.5369 \\ -3.5741 \end{bmatrix}$$

$$P_f = \begin{bmatrix} 0.0044 & 2.1533 \times 10^{-5} & -3.6456 \times 10^{-5} & -7.7729 \times 10^{-5} \\ 2.1533 \times 10^{-5} & 8.7371 \times 10^{-6} & -2.5369 \times 10^{-7} & -3.5741 \times 10^{-7} \\ -3.6456 \times 10^{-5} & -2.5369 \times 10^{-7} & 3.0037 \times 10^{-7} & 6.3871 \times 10^{-7} \\ -7.7729 \times 10^{-5} & -3.5741 \times 10^{-7} & 6.3871 \times 10^{-7} & 1.3623 \times 10^{-6} \end{bmatrix}$$

Separation Principle for LQG Design

LQG Control = Optimal observer + Optimal state feedback

- The optimal state estimator and optimal control designs are solved separately, based on the "separation principle"
  - Design the Kalman filter first, and then viewing the estimated states as if they were the actual states, design the optimal state feedback control (LQR)
    - Requirements: (A, B) is controllable/stabilizable and (A, Q) is observable/detectable

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K_f(y - C\hat{x}) & \quad \textit{Kalman filter} \\ u = -K_c\hat{x} & \quad \textit{State feedback} \end{cases}$$

where the controller gains (row vector K_c and column vector K_f) are found as:

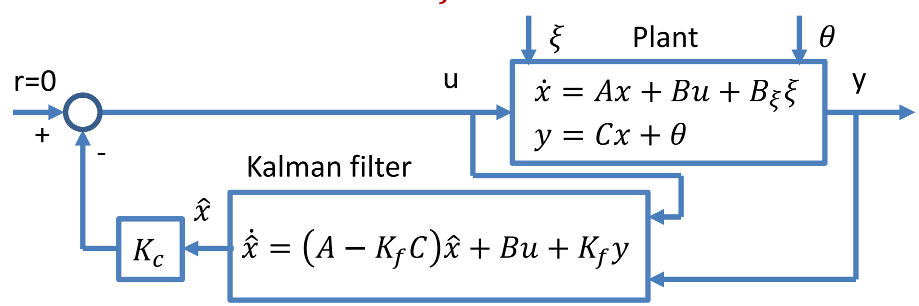
$$K_c = R^{-1}B^TP_c$$

$$A^TP_c + P_cA + M^TQM - P_cBR^{-1}B^TP_c = 0$$

and, by duality:

$$K_f = P_fC^TR_f^{-1}$$

$$P_fA^T + AP_f + B_\xi Q_f B_\xi^T - P_fC^TR_f^{-1}CP_f = 0$$



Observer-Based LQG Controller

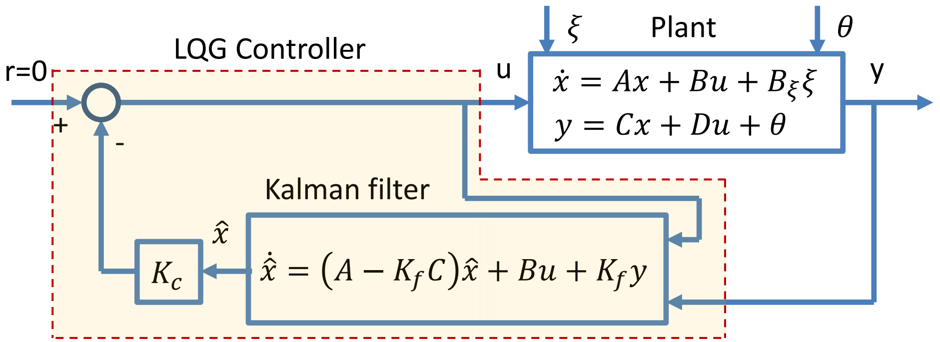
Plant model: $\begin{cases} \dot{x} = Ax + Bu + B_\xi \xi \\ y = Cx + Du + \theta \end{cases}$

Performance index:

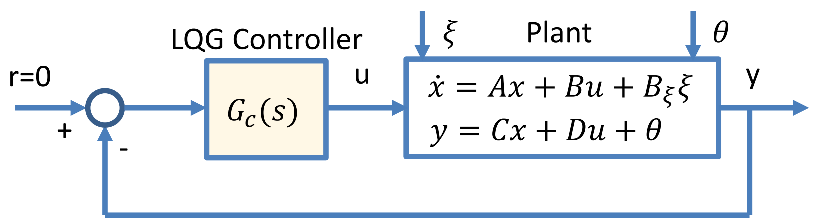
$$J = \lim_{t_f \to \infty} E\left\{ \int_0^{t_f} \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{bmatrix} Q & N_c \\ N_c^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt \right\}, \qquad \text{Normally } N_c = 0$$

LQG controller:

$$\begin{cases} \dot{\hat{x}} = \overbrace{(A - BK_c - K_f C + K_f D K_c)}^{\bar{A}} \hat{x} + \overbrace{(K_f)}^{\bar{B}} y \\ y = \overbrace{(K_c)}^{\bar{C}} \hat{x} + \overbrace{(0)}^{\bar{D}} y \end{cases}$$



or $\quad G_c(s) = \left[\begin{array}{c|c} A - K_f C - BK_c + K_f D k_c & K_f \\ K_c & 0 \end{array}\right]$

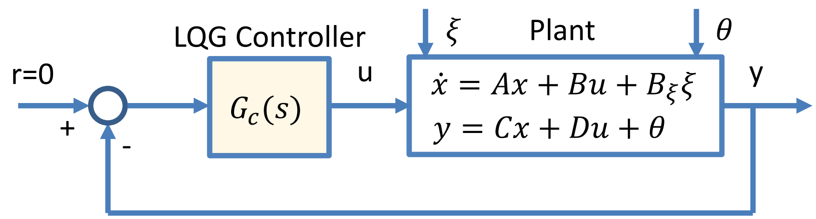


Equivalent LQG transfer function:

$$G_c(s) = K_c \left(sI - A + K_f C + BK_c - K_f D k_c \right)^{-1} K_f$$

LQG Control Design in Matlab

Matlab function: lqg(),



Syntax: Gc=-lqg(G,W,V)

or: $[A_f, B_f, C_f, D_f]$=lqg(A,B,C,D,W,V)

where (A_f, B_f, C_f, D_f) is the state-space model of the LQG controller $G_c(s)$, $W = \begin{bmatrix} Q & N_c \\ N_c^T & R \end{bmatrix}$, and $V = \begin{bmatrix} B_\xi Q_f B_\xi^T & N_f \\ N_f^T & R_f \end{bmatrix}$.

Typically, N_c and N_f are assumed to be zero

Example: For $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\xi \xi(t) \\ y(t) = Cx(t) + Du(t) + \theta(t) \end{cases}$, $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5000 & -100/3 & 500 & 100/3 \\ 0 & -1 & 0 & 1 \\ 0 & 100/3 & -4 & -60 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 25/3 \\ 0 \\ -1 \end{bmatrix}$, $B_\xi = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$,

$C = [0 \quad 0 \quad 1 \quad 0]$, $D = [0]$, $Q_f = 7 \times 10^{-4}$, $R_f = 10^{-8}$, [0], $Q = diag(5000{,}0{,}50000{,}1)$, and $R = 0.001$,

design the LQG control.

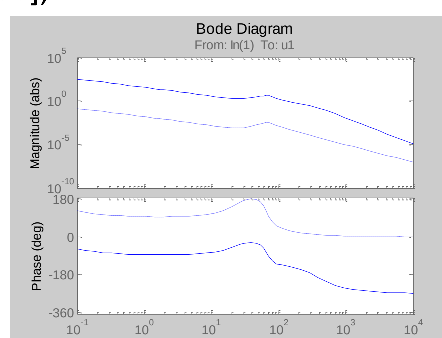
LQG controller:

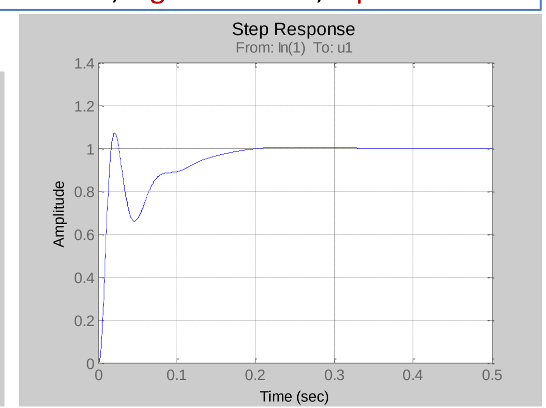
$$G_c(s) = \frac{-1231049.0702(s+40.47)(s^2+105.5s+5000)}{(s^2+39.17s+868.2)(s^2+493.9s+1.234\times10^5)}$$

Gm=4.3728, Pm=43.0423, wgc=323.2224, wpc=125.1672

Matlab code:

```
close all, clear all, clc,
A=[0,1,0,0;-5000,-100/3,500,100/3;0,-1,0,1;0,100/3,-4,-60];
B=[0;25/3;0;-1]; Bz=[-1;0;0;0]; C=[0,0,1,0]; D=[0];
Q=diag([5000,0,50000,1]); R=0.001; Qf=7e-4; Rf=1e-8;
V=[Bz*Qf*Bz',zeros(4,1);zeros(1,4),Rf];
W=[Q,zeros(4,1);zeros(1,4),R]; sys=ss(A,B,C,D);
G_cont=-lqg(sys,W,V); Gc=zpk(G_cont), pole(Gc)
figure(1), step(feedback(Gc*sys,1),0.5),
figure(2), bode(sys,':',Gc*sys,'-',{0.1,10000}),
[Gm,Pm,wgc,wpc]=margin(sys*Gc),
```



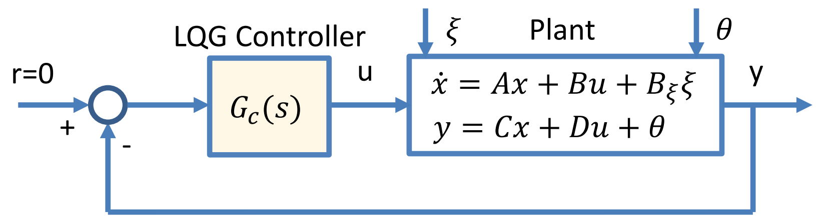


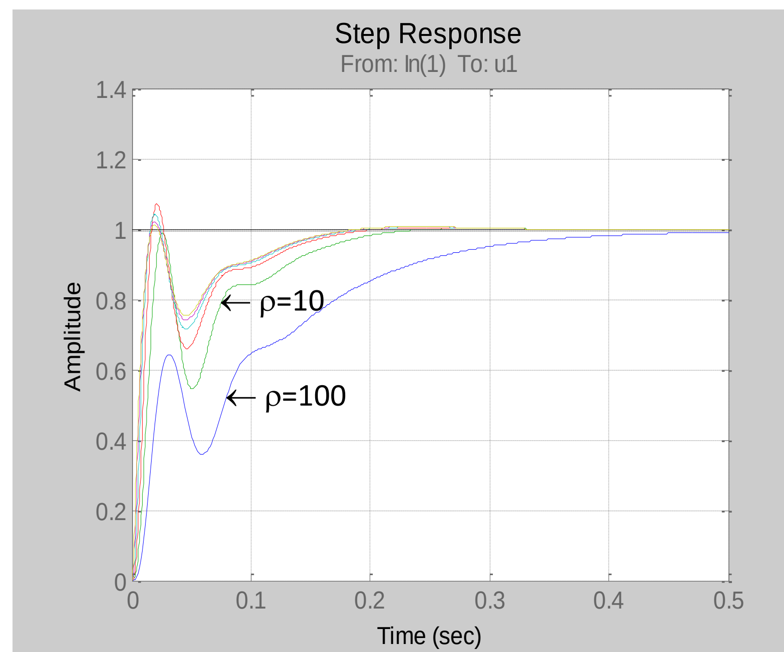
LQG Control Design in Matlab

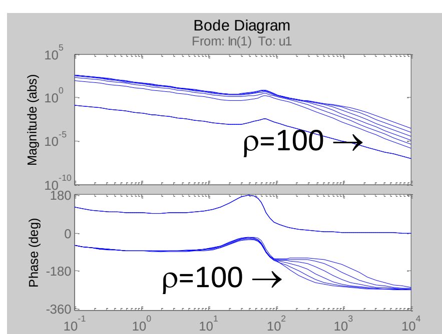
Example: For $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\xi \xi(t) \\ y(t) = Cx(t) + Du(t) + \theta(t) \end{cases}$, $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5000 & -100/3 & 500 & 100/3 \\ 0 & -1 & 0 & 1 \\ 0 & 100/3 & -4 & -60 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 25/3 \\ 0 \\ -1 \end{bmatrix}$, $B_\xi = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$,
$C = [0 \quad 0 \quad 1 \quad 0]$, $D = [0]$, $Q_f = 7 \times 10^{-4}$, $R_f = 10^{-8}$, [0], $Q = diag(5000{,}0{,}50000{,}1)$, and various weights $R = \{100, 10, 1, 0.1, 0.01{,}0.001\}$, design the LQG controls and compare.

Matlab code:

```
close all, clear all, clc,
A=[0,1,0,0;-5000,-100/3,500,100/3;0,-1,0,1;0,100/3,-4,-60];
B=[0;25/3;0;-1]; Bz=[-1;0;0;0]; C=[0,0,1,0]; D=[0];
Q=diag([5000,0,50000,1]); R=0.001; Qf=7e-4; Rf=1e-8;
V=[Bz*Qf*Bz',zeros(4,1);zeros(1,4),Rf]; sys=ss(A,B,C,D);
for rho=[100,10,1,0.1,0.01,0.001],
  rR=rho*R; W=[Q,zeros(4,1);zeros(1,4),rR]; Gc=-lqg(sys,W,V);
  figure(1), step(feedback(Gc*sys,1),0.5), hold on,
  figure(2), bode(sys,':',Gc*sys,'-',{0.1,10000}), hold on,
end
```







LQG Control with Loop Transfer Recovery

- LQG controllers, unlike LQR, may have very small stability margins
  - Small disturbances may drive the system unstable
  - Open-loop transfer function with LQR: $G_{ol_{LQR}}(s) = K_c(sI - A)^{-1}B$
  - Open-loop transfer function with LQG: $G_{ol_{LQG}}(s) = K_c\left(sI - A + BK_c + K_f C\right)^{-1} K_f C(sI - A)^{-1}B$
  - Loop transfer recovery technique can be used to reduce the difference between these open-loop TFs
- Basic idea:
  - Let $\bar{Q}_f = Q_f + qBB^T$. Then, if the plant is minimum-phase, as $q \to \infty$, the open-loop TF with LQG control approaches the open-loop TF with LQR control
    - $\lim_{q\to\infty} K_c\left(sI - A + BK_c + K_f C\right)^{-1} K_f C(sI - A)^{-1}B = K_c(sI - A)^{-1}B$
- LQG/LTR control design procedure
  1. Design an optimal LQR control with the specified weighting matrices Q and R
  2. Set $\bar{Q}_f = Q_f + qBB^T$. Increase the value of q only so much that the return difference of the compensated system approaches: $-K_c(j\omega I - A)^{-1}B$, (check Bode or Nyquist plots)
     - With the selected q the FARE is changed to: $\frac{P_f A^T}{q} + \frac{AP_f}{q} + \frac{BQ_f B^T}{q} + BR_f B^T - \frac{P_f C^T {R_f}^{-1} C P_f}{q} = 0$
     - The Kalman filter gain, as $q \to \infty$, becomes: $K_f = \sqrt{q}\, BR_f^{-1/2}$
     - Stability margin of the closed-loop system increases, as $q \to \infty$

LQG/LTR Control Design Techniques

LTR designs at the system's input and output

1. The LTR design can be performed at the input of the plant model, by replacing Q_f with $\bar{Q}_f = Q_f + qBB^T$ in the FARE and letting $q \to \infty$
   - The open-loop TF with LQG control approaches the open-loop TF with LQR control
   $$\lim_{q\to\infty} K_c\left(sI - A + BK_c + K_f C\right)^{-1} K_f C(sI - A)^{-1}B = K_c(sI - A)^{-1}B$$

2. The LTR design can be performed at the output of the plant model, by replacing Q with $\bar{Q} = Q + qC^T C$ in the ARE and letting $q \to \infty$
   - The open-loop TF with LQG control approaches the open-loop TF of the Kalman filter
   $$\lim_{q\to\infty} K_c\left(sI - A + BK_c + K_f C\right)^{-1} K_f C(sI - A)^{-1}B = C(sI - A)^{-1}K_f$$

- Good loop recovery is possible even for nonminimum-phase plants if the RHP zero is located sufficiently outside the loop passband

LQG/LTR Control Design in Matlab

Matlab functions: ltru(), ltry(), (in Robust Control Toolbox)

Syntax: Gc=ltru(G,Kc,Qf,Rf,q,ω)

or [Af, Bf, Cf, Df] = ltru(A,B,C,D,Kc,Qf,Rf,q,ω)

Gc=ltry(G,Kf,Q,R,q,ω)

or [Af, Bf, Cf, Df] = ltry(A,B,C,D,Kf,Q,R,q,ω)

- q is a vector of increasingly large numbers
- ω Is the frequency range for Nyquist plot

Example: For system $G(s) = \frac{-(948.12s^3+30235s^2+56482s+1215.3)}{s^6+64.554s^5+1167s^4+3728.6s^3-4595.4s^2+1102s+708.1}$, find the LQG/LTR control for various values of fictitious noise q.

Matlab code:

```
close all, clear all, clc,
num=-[948.12, 30325, 56482, 1215.3];
den=[1, 64.554, 1167, 3728.6, -5495.4, 1102, 708.1];
G=ss(tf(num,den)), Qf=1e-4; Rf=1e-5;
A=G.a; B=G.b; C=G.c; D=G.d;
Q=C'*C; R=1; Kc=lqr(A,B,Q,R)
q=[1,1e4,1e6,1e8,1e10,1e12,1e14];
w=logspace(-2,2);
Gc=ltru(G,Kc,Qf,Rf,q,w);
Gc=zpk(Gc)
figure, step(feedback(Gc*G,1),10)
```

LQG/LTR control for $q = 10^{14}$:

$$G_c(s) = \frac{-22026697072.6516(s+30.22)(s+29.71)(s+6.673)(s+1.315)(s+0.01261)}{(s+1.4445\times10^4)(s+30)(s+1.963)(s+0.0218)(s^2+1.4444\times10^4 s+2.087\times10^8)}$$