LQR
Linear Quadratic Regulator

A state space optimal control technique

Brett Shapiro

Summary

- LQR description
- LQR derivation
  - Double pendulum test mass control example
- LQR observer formulation
  - Quad pendulum damping observer example
- Augmenting the plant for more complexity
  - Example: Frequency domain LQR

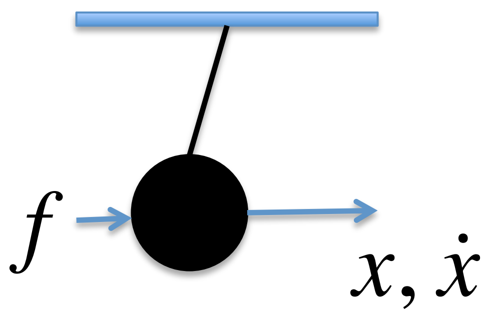
References

- Reference 1: Modern Control Engineering, 4^{th} Ed. Katsuhiko Ogata. 2002, Prentice Hall.
- Reference 2: Linear Optimal Control Systems. Huibert Kwakernaak, Raphael Sivan. 1972, Wiley-Interscience. Online access at http://www.ieeecss.org/publications/classic-books
- Reference 3: T1300301 – LQR in the frequency domain

What is it?

- An **automated** algorithm for finding **optimal** feedback for a linear system in state space form.

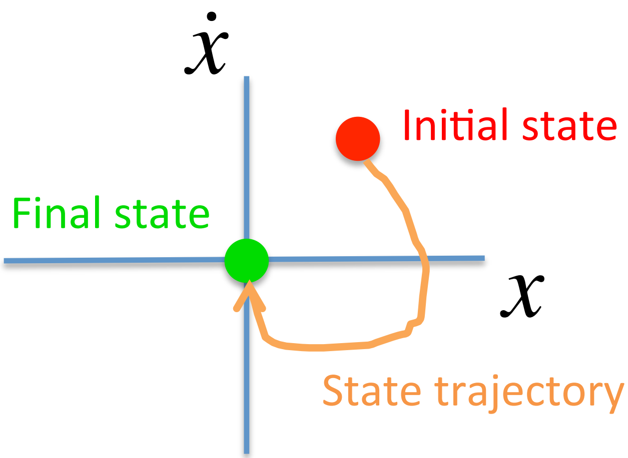
1 DOF pendulum example



f: actuation

x: displacement state

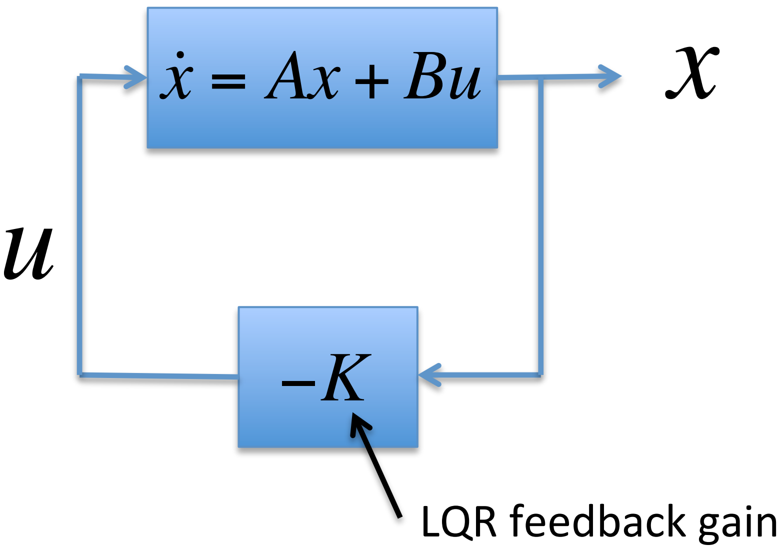
$\dot{x}$: velocity state



- LQR determines the feedback law to minimize the size of the state vector in the least time with the least control effort.
- Note, a state of [0,0] may be generalized to any arbitrary point.

System Description

System block diagram



Assumptions:

1) We have knowledge of **all states** x

2) The system is completely **controllable**.

This means that given infinite actuator range we can [ref 1, pgs 779-780, 829-832]

- drive the system from one arbitrary state to another in finite time

or, equivalently

- Set all closed loop poles to arbitrary values

Matrix dimensions for *n* states and *m* control inputs

$$x = \begin{bmatrix} x_1 \\ \ldots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ \ldots \\ u_m \end{bmatrix}$$

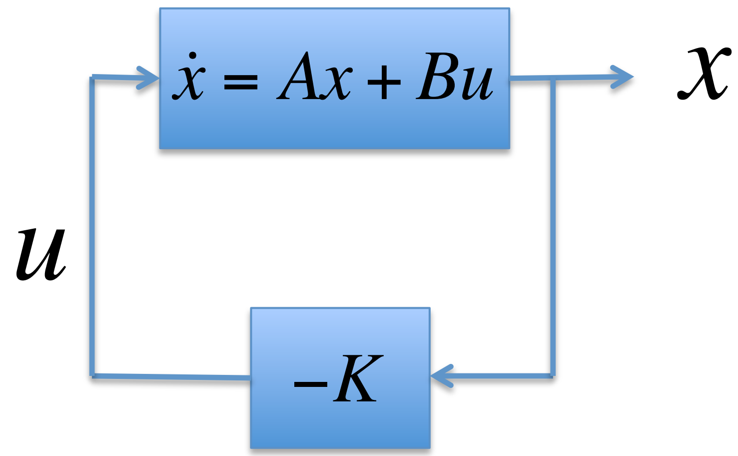
A is $n \times n$

B is $n \times m$

K is $m \times n$

Identifying the Goal - the Cost Function with competing interests

System block diagram



We want to make the size of x really small.

$$x^T x \to 0$$

But…also want to the size of u to be really small, to minimize noise, keep the actuators in range, avoid exciting higher order dynamics, etc.

$$u^T u \to 0$$

And, we want this to be true for all time. So we make the total cost:

$$J = \int_0^\infty \left(x^T x + u^T u \right) dt$$

But, we might care about the size of some states and actuators more than others. So we generalize the cost with some weighting matrices: Q and R

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt$$

Other LQR Cost Function Forms

Other forms you may see for the cost function

- 'Infinite time horizon'. The simplest, and most common form. The one we use here. Ref [1].

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt$$

- 'Finite time horizon'. More general form. Used in the most rigorous LQR derivations. Results in a time variant feedback law, i.e. *K = K(t)*. Ref [2].

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

- You may also see this form, which is used by Matlab and also frequency domain versions of LQR.

$$J = \int_0^\infty \left(x^T Q x + u^T R u + 2 x^T N u \right) dt$$

In all cases, the weighting matrices are assumed to have these properties

$$Q = Q^T \geq 0, \; R = R^T > 0, P = P^T > 0$$

Minimizing the Cost Function: Two Derivation Approaches

$$J = \int_0^{\infty} \left(x^T Q x + u^T R u \right) dt$$

Calculus of variations – the super gory rigorous method. See reference 2, Chap 3. Includes special cases, including stochastic noise.

Linear algebra – more concise. Get's the point across with some hand-waving. See Reference 1, pages 897-899.

Minimizing the Cost Function

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \qquad \dot{x} = Ax + Bu \qquad (1)$$

Plug in feedback relation

$$u = -Kx$$

$$J = \int_0^\infty \left(x^T Q x + x^T K^T R K x \right) dt \qquad \dot{x} = Ax - BKx \qquad (2)$$

Factor out x

$$J = \int_0^\infty x^T \left(Q + K^T R K \right) x dt \qquad \dot{x} = (A - BK)x \qquad (3)$$

Then make the following substitution where $P = P^T > 0$. The time derivative allows us to brings the system dynamics into the cost function.

$$x^T \left(Q + K^T R K \right) x = -\frac{d}{dt} \left(x^T P x \right) \qquad (4)$$

Minimizing the Cost Function

$$x^T\left(Q+K^TRK\right)x=-\dot{x}^TPx-x^TP\dot{x} \qquad (5)$$

Then substitute in the system dynamics $\dot{x}=(A-BK)x$

$$x^T\left(Q+K^TRK\right)x=-x^T\left[\left(A-BK\right)^TP+P\left(A-BK\right)\right]x \qquad (6)$$

Canceling the x's

$$Q+K^TRK=-\left[\left(A-BK\right)^TP+P\left(A-BK\right)\right] \qquad (7)$$

Splitting R into 'square roots' T, $R=T^TT$, and rearranging

$$A^TP+PA+\left[TK-(T^T)^{-1}B^TP\right]^T\left[TK-(T^T)^{-1}B^TP\right]-PBR^{-1}B^TP+Q=0 \qquad (8)$$

It can be shown that minimization of J with respect to K requires the minimization of

$$x\left[TK-(T^T)^{-1}B^TP\right]^T\left[TK-(T^T)^{-1}B^TP\right]x \qquad (9)$$

Minimizing the Cost Function

But, since

$$x\left[TK-(T^T)^{-1}B^TP\right]^T\left[TK-(T^T)^{-1}B^TP\right]x \geq 0 \qquad (10)$$

This minimization is solved when

$$TK=(T^T)^{-1}B^TP \qquad (11)$$

Which give us a relation between the feedback matrix K and P

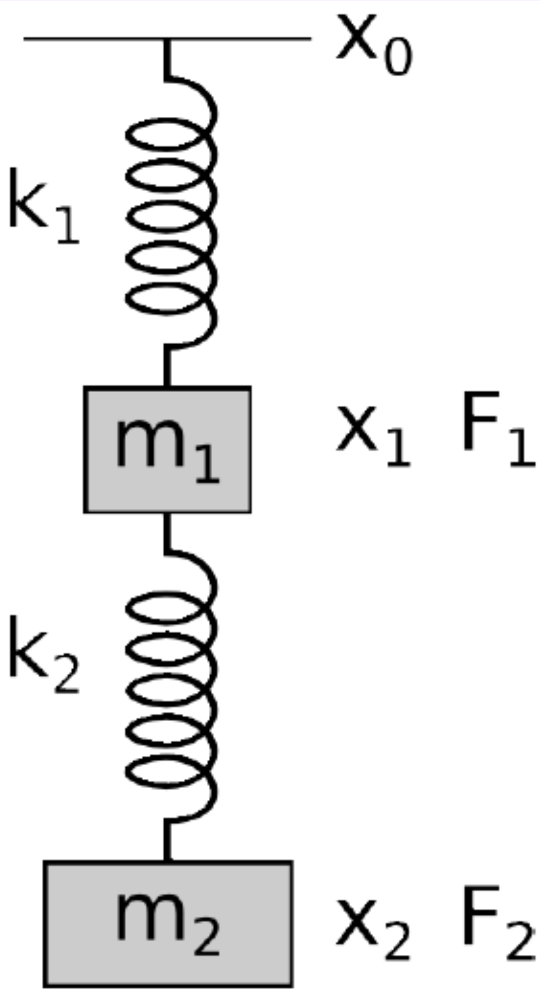
$$\boxed{K=R^{-1}B^TP} \qquad (12)$$

Plugging this back into Eq. (7) (just after we substituted in the system dynamics)

$$\boxed{A^TP+PA-PBR^{-1}B^TP+Q=0} \qquad (13)$$

Gives us the so called 'Algebraic Riccati Equation' (ARE), which we can solve for to get P for the complete solution. The ARE is generally solved for numerically. Only some very simple cases can be solved analytically.

Example 1 – Double Pendulum Test Mass Control



See MATLAB Example 1

Problem:
- design a controller to position the test mass using the two actuators
- the test mass actuator has 1/10 the range of the top mass.

m_1	10 kg
m_2	20 kg
k_1	100 N/m
k_2	500 N/m
g_1	0.4 Ns/m
g_2	0.4 Ns/m

Courtesy Gabriele Vajente

Example 1 – Double Pendulum Test Mass Control

$$\dot{x} = Ax + B_u u + B_d x_0$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -60 & 50 & -0.04 & 0 \\ 25 & -25 & 0 & -0.02 \end{bmatrix} \qquad B_u = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.05 \end{bmatrix} \qquad B_d = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 : \text{ top mass displacement [m]} \\ x_2 : \text{ test mass displacement [m]} \\ x_3 : \text{ top mass velocity [m/s]} \\ x_4 : \text{ test mass velocity [m/s]} \end{bmatrix} \qquad u = \begin{bmatrix} u_1 : \text{ top mass actuator force [N]} \\ u_2 : \text{ test mass actuator force [N]} \end{bmatrix}$$

$$x_0 = \text{suspension point displacement [m]}$$

- We are interested in controlling the test mass position using the two actuators to compensate for suspension point motion.

Example 1 – Double Pendulum Test Mass Control

$$\dot{x} = Ax + B_u u + B_d x_0$$

State space equation

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt$$

Cost function

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

For the states, just weight the test mass displacement

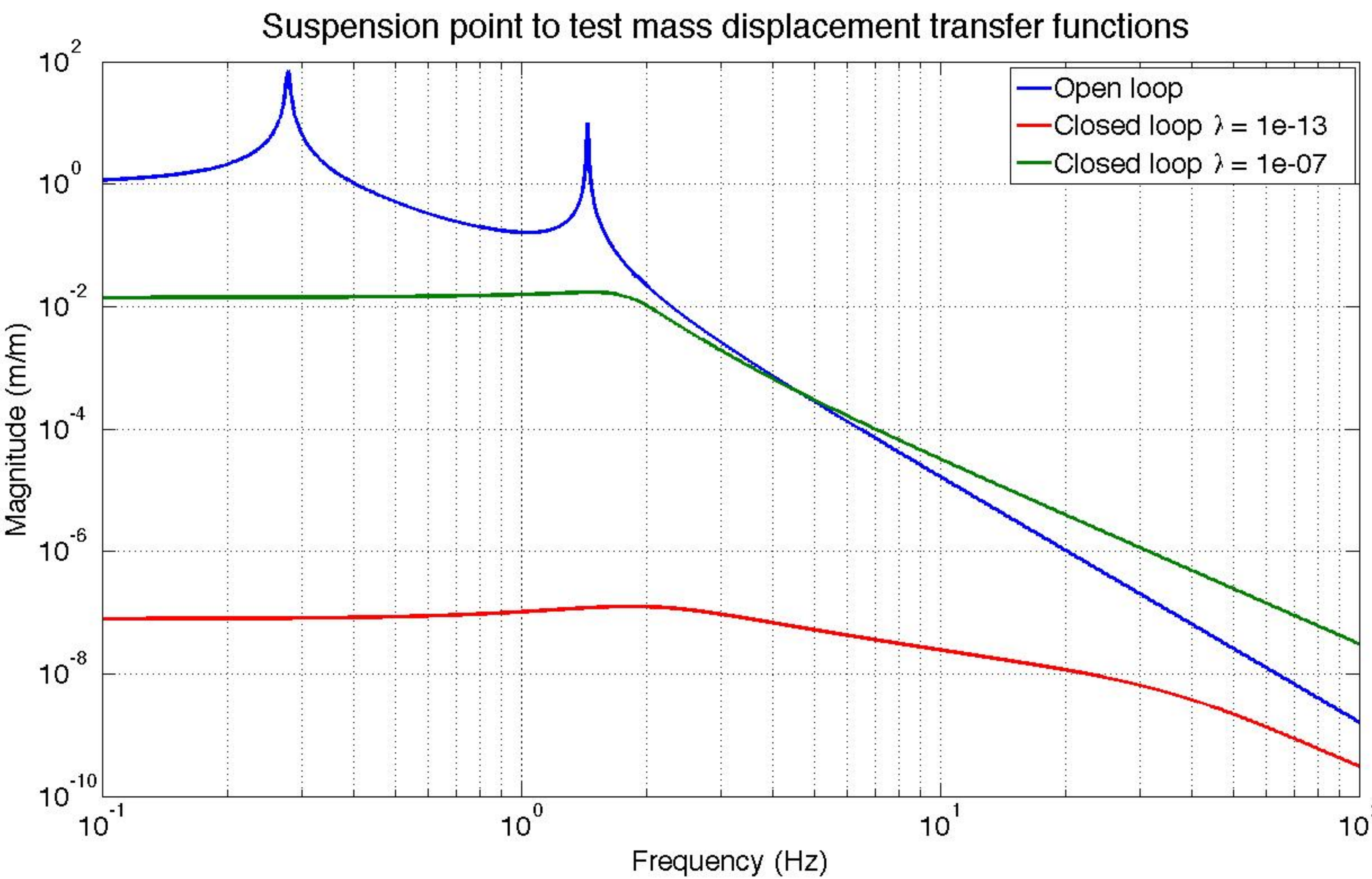
$$R = \lambda R_0 = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$$

- We try putting 10 times the weight on the test mass actuator since it has 10 times less range. Not necessarily an obvious relation to actuator range.
- We will also include in *R* a scalar tuning factor λ.

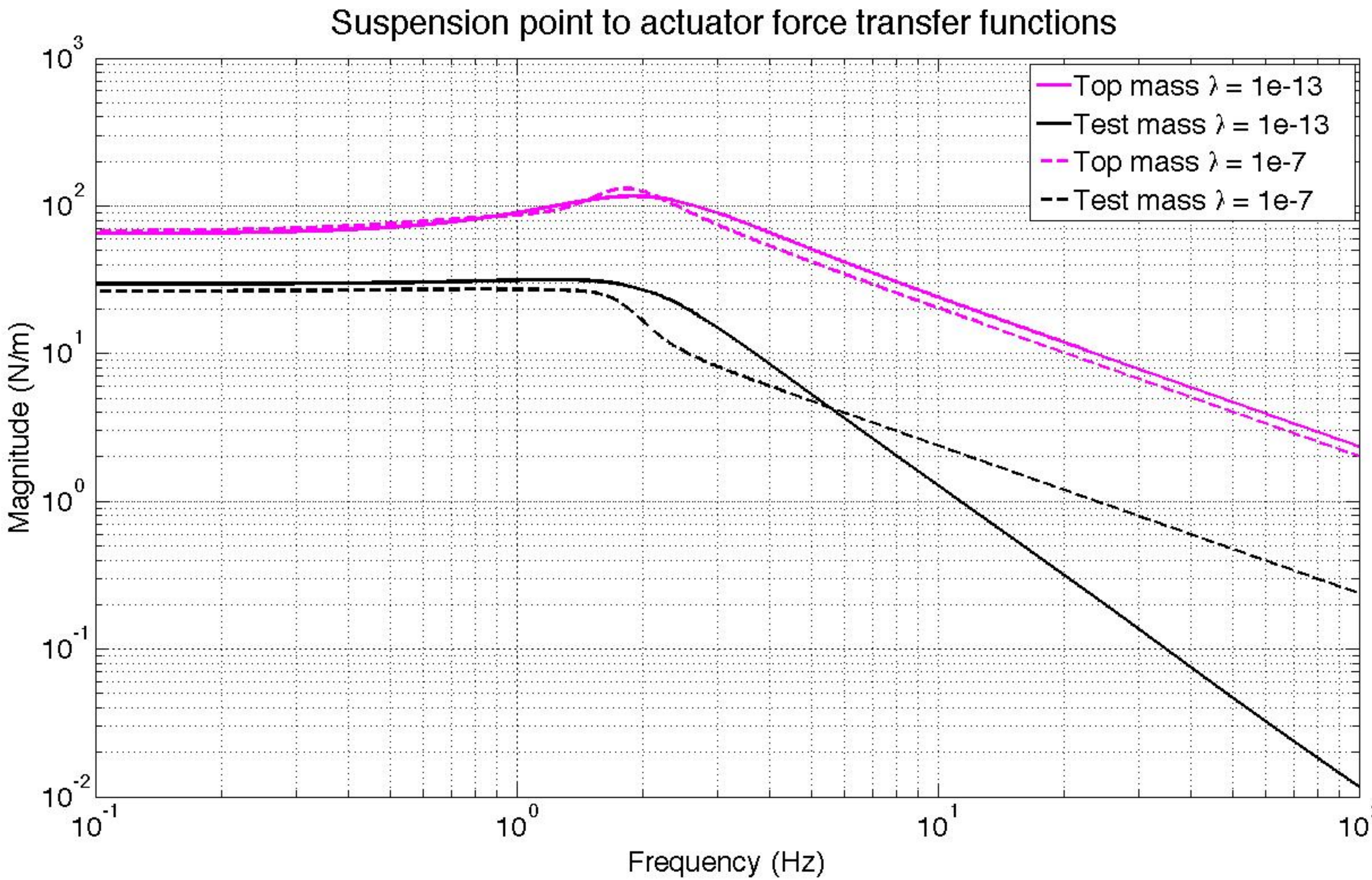
Example 1 – Double Pendulum Test Mass Control

```
% State space matrices
...
system = ss(A,B,C,D);

% LQR weighting matrices
Q = diag([0 1 0 0]);
lambda = 1e-13;
R = lambda*diag([1 10]);

% LQR feedback matrix
K = lqr(A,Bu,Q,R);

% closed loop state space matrices
A_closed_loop = A-B*K;
C_closed_loop = [C;K];
D_closed_loop = zeros(size(A,1),size(C_closed_loop,2))
system_closed_loop =
ss(A_closed_loop,B,C_closed_loop,D_closed_loop)
```

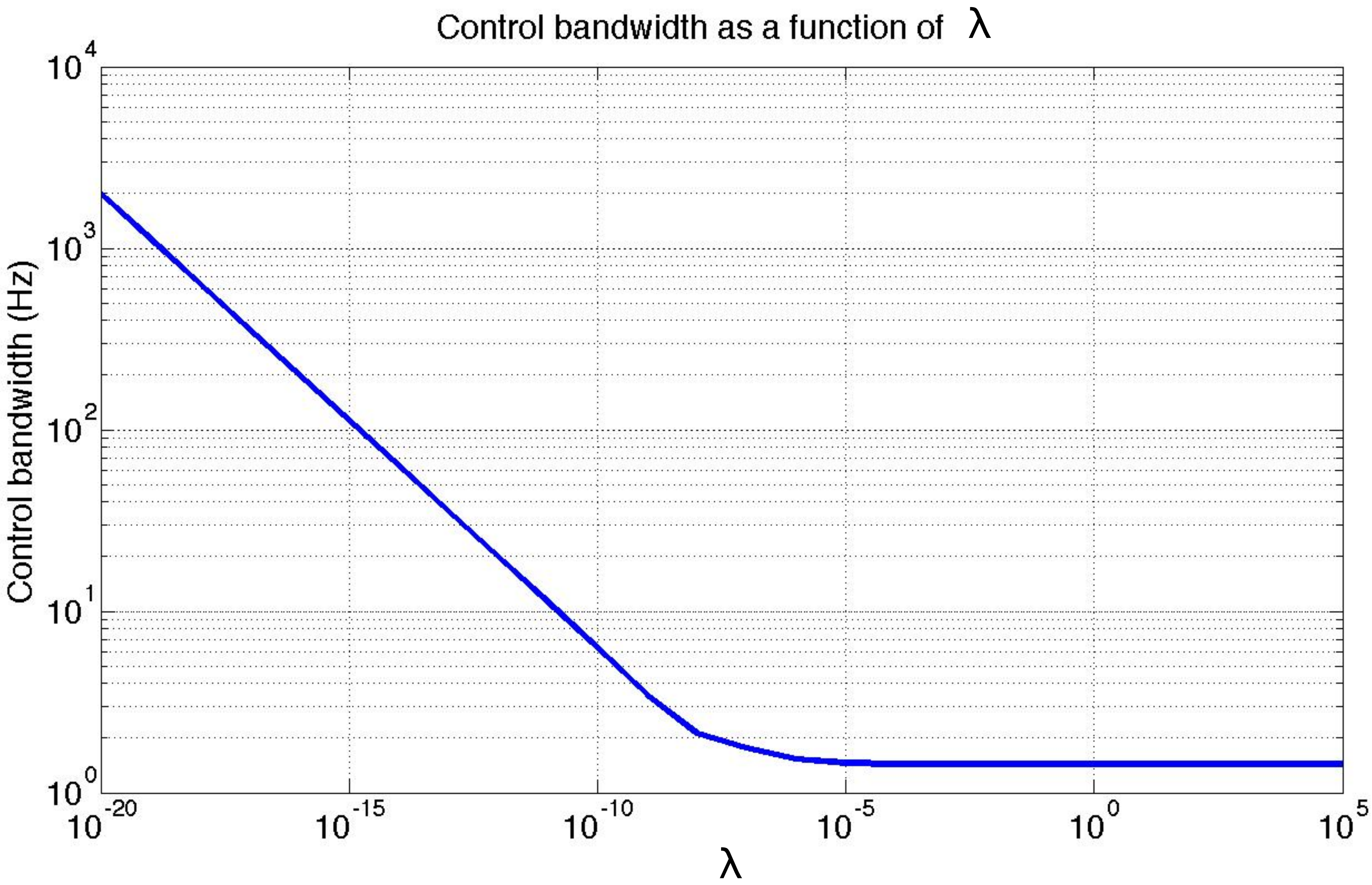
Example 1 – Double Pendulum Test Mass Control



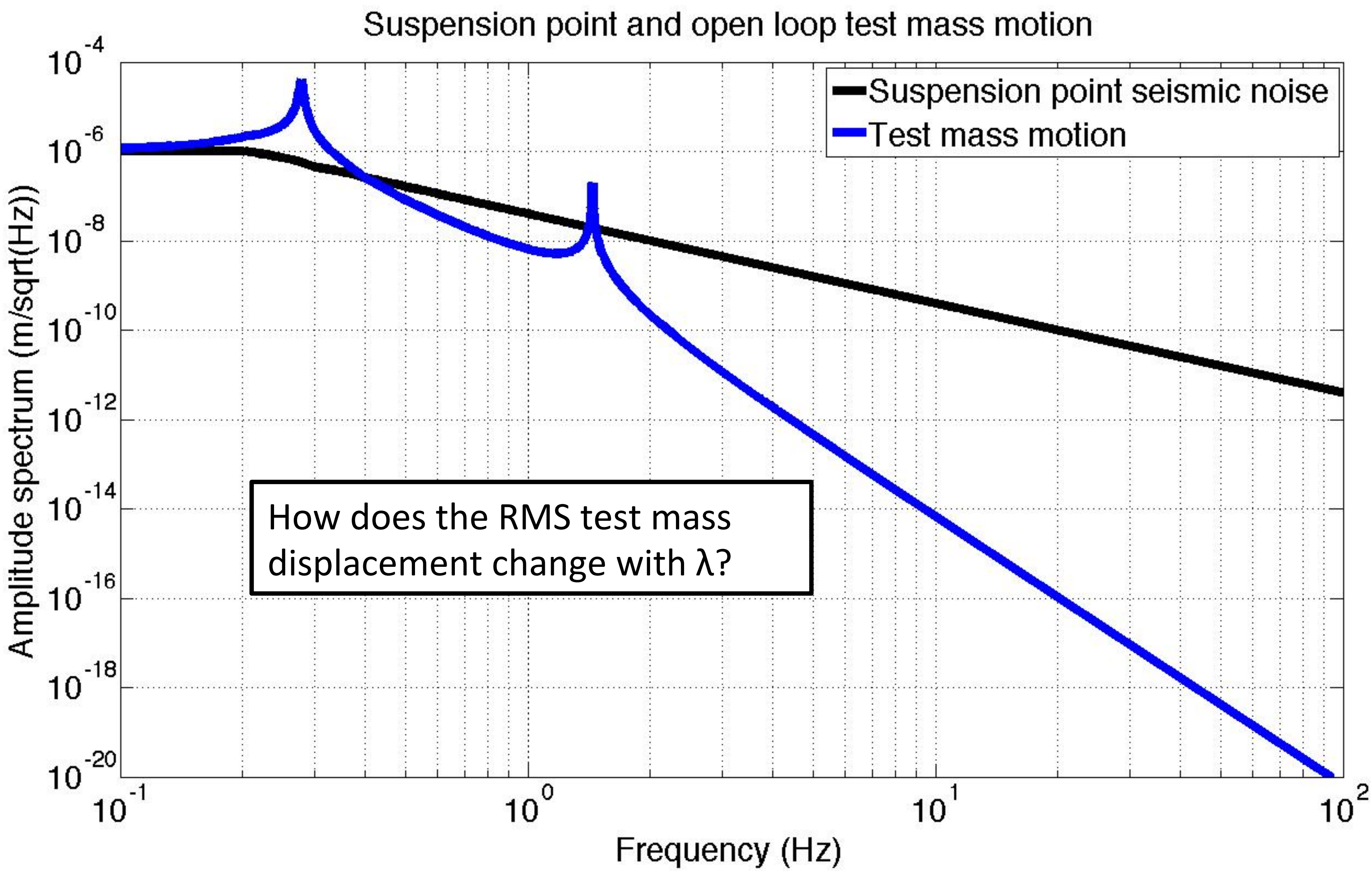
Example 1 – Double Pendulum Test Mass Control



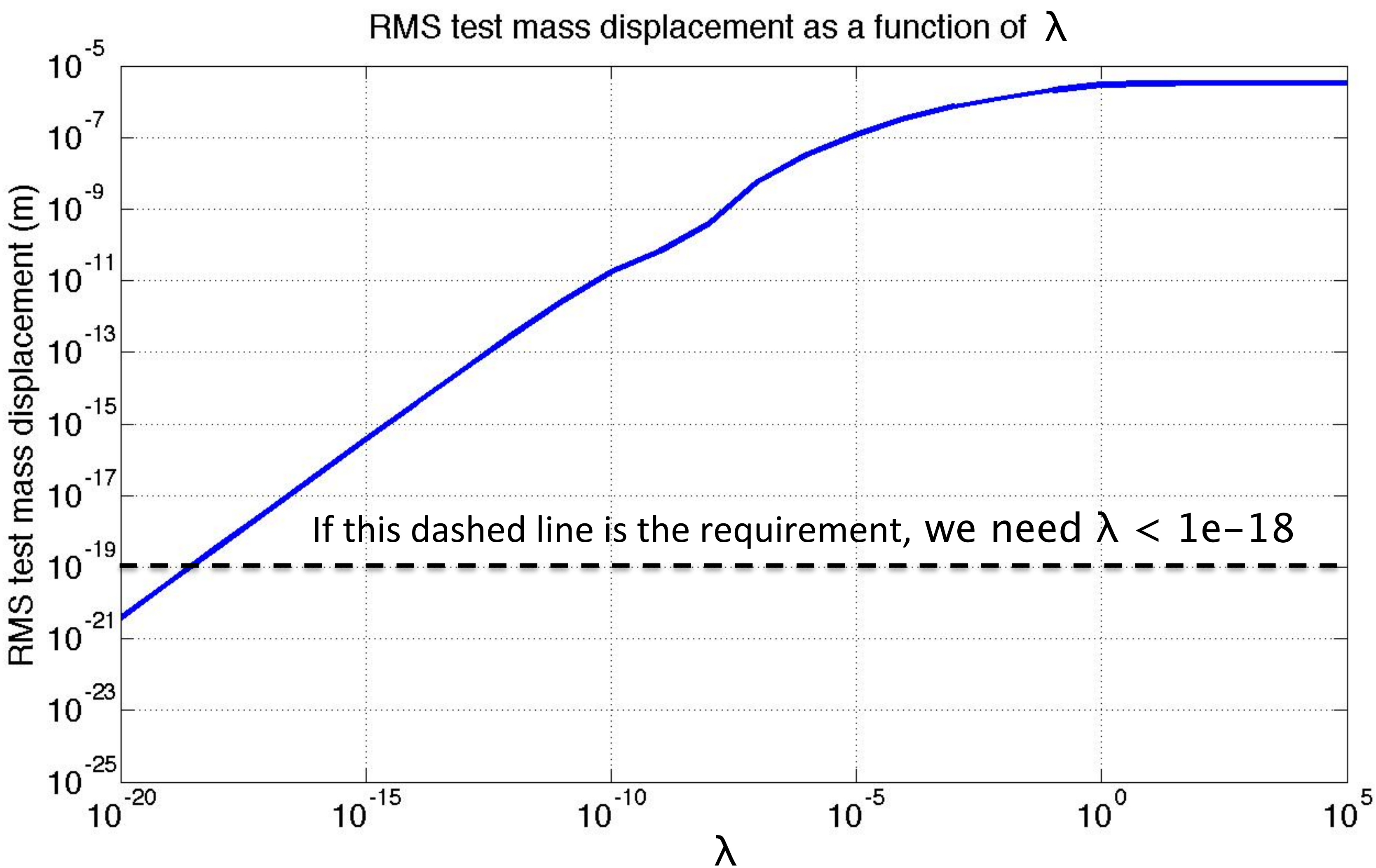
Example 1 – Double Pendulum Test Mass Control



Example 1 – Double Pendulum Test Mass Control



Example 1 – Double Pendulum Test Mass Control



Observer Design with LQR

Recall from yesterday, a system of the form

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

$$y = Cx + Du$$

has an observer of the form

$$\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - C\hat{x}\right)$$

where $\hat{x}$ is the estimated state.

The observer error is then,

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ax - A\hat{x} + Bu - Bu - L\left(y - C\hat{x}\right)$$

$$\dot{e} = (A - LC)e$$

Control/Observer Separation Principal

For control feedback using the estimated state, the system state space is

$$\dot{x} = Ax - BK\hat{x}$$

Augmenting this plant with the state estimation error

$$\dot{e} = \dot{x} - \dot{\hat{x}} = (A - LC)e$$

We get the full closed loop system/compensator state space as

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}$$

Because this matrix is block-triangular, the eigenvalues (poles) simple the combined set of the eigenvalues of *(A-BK)* and *(A-LC).* Thus, the poles of the control and estimator design are **independent**. So, one can design each piece individually, and be confident of where the closed loop poles will be.

Observer – Controller Duality

Observer error

Control error

$$\dot{e} = (A - LC)e$$

$$\dot{x} = (A - BK)x$$

Note that the poles of $(A - LC)$ are the same as its transpose $(A^T - C^T L^T)$

Thus, we can think of the observer feedback design as the ‘control’ design of the fictitious system

$$\dot{\chi} = A^T \chi + C^T \mu$$

$$\mu = -L^T \chi$$

Control-Observer Duals

Control design	Observer design
A	A^T
B	C^T

Observer LQR design

So, we can use LQR to chose how the estimation error evolves

$$J = \int_0^\infty \left(\chi^T Q_e \chi + \mu^T R_e \mu\right) dt$$

- Q_e weights the estimation error
- R_e weights the feedback of the noisy measured output

$$\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - C\hat{x}\right)$$

Recall that

$$\dot{\chi} = A^T \chi + C^T \mu$$

$$\mu = -L^T \chi$$

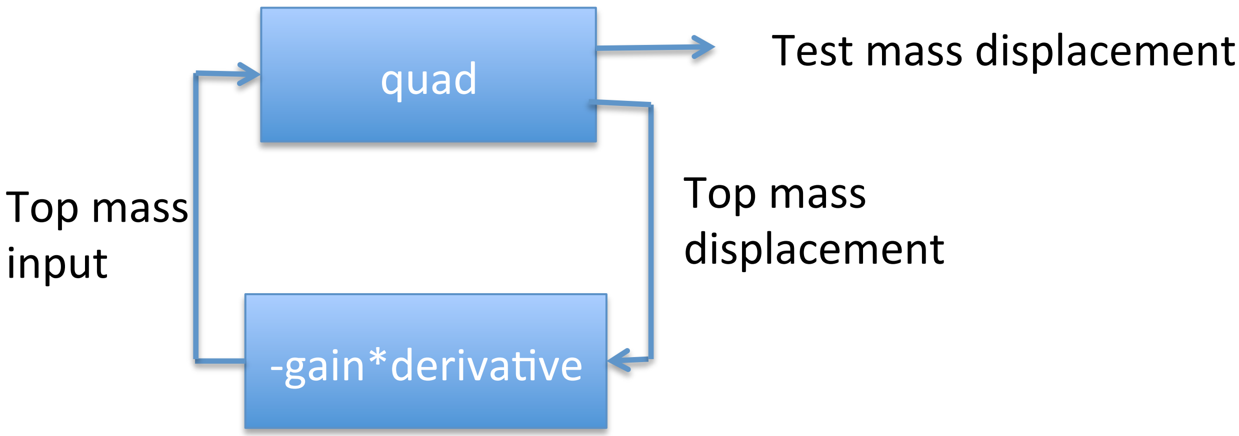
Where the dynamics χ are the same as *e*.

Matlab code

```
% LQR observer feedback matrix
L_transpose = lqr(transpose(A),transpose(C),Qe,Re);
L = transpose(L_transpose);
```

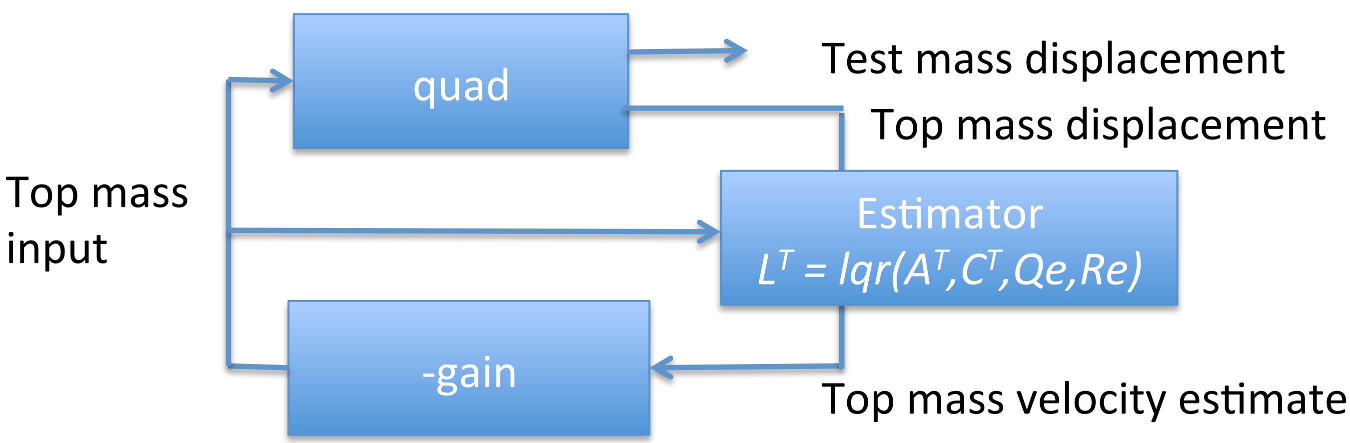
Example 2 – quad damping

- Reference case, pure velocity feedback with no estimation

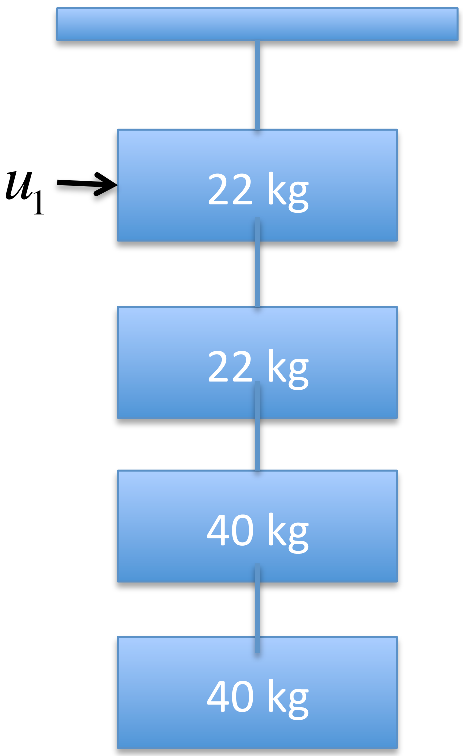


See MATLAB Example 2

- Test case, velocity feedback using state estimation



Example 2 – quad damping



- We only consider the single degree of freedom along the laser beam axis for each stage (longitudinal motion). Thus, we have 8 states with 4 displacements and 4 velocities
- We only care about estimating the top mass velocity. Thus,

$$Q_e = \mathrm{diag}\left(\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}\right)$$

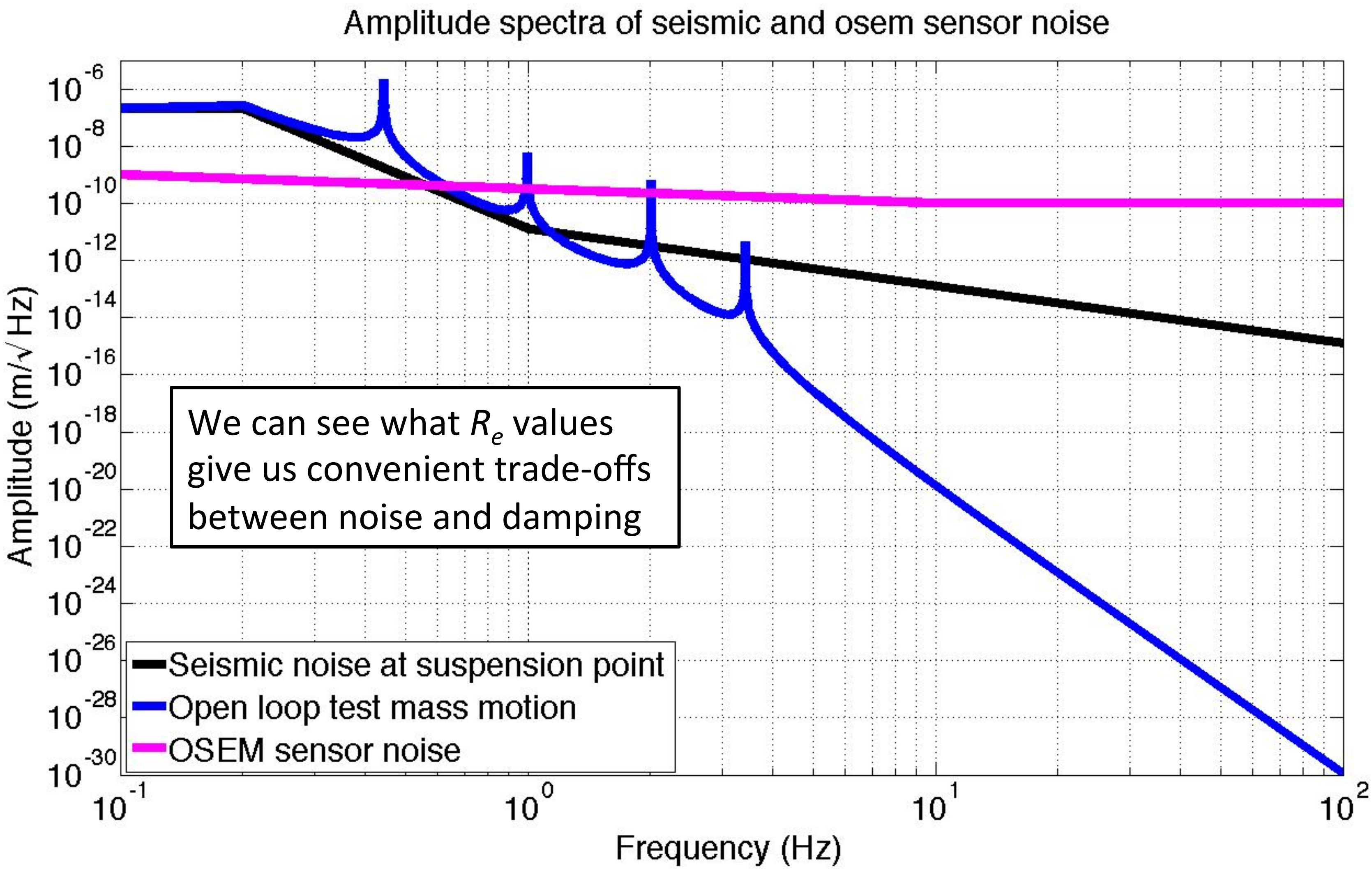
- We then scan for desirable values for the scalar

$$R_e$$

- when we have this pure velocity feedback law

$$u_1 = -200 \frac{d}{dt} x_1 = -200 x_5$$

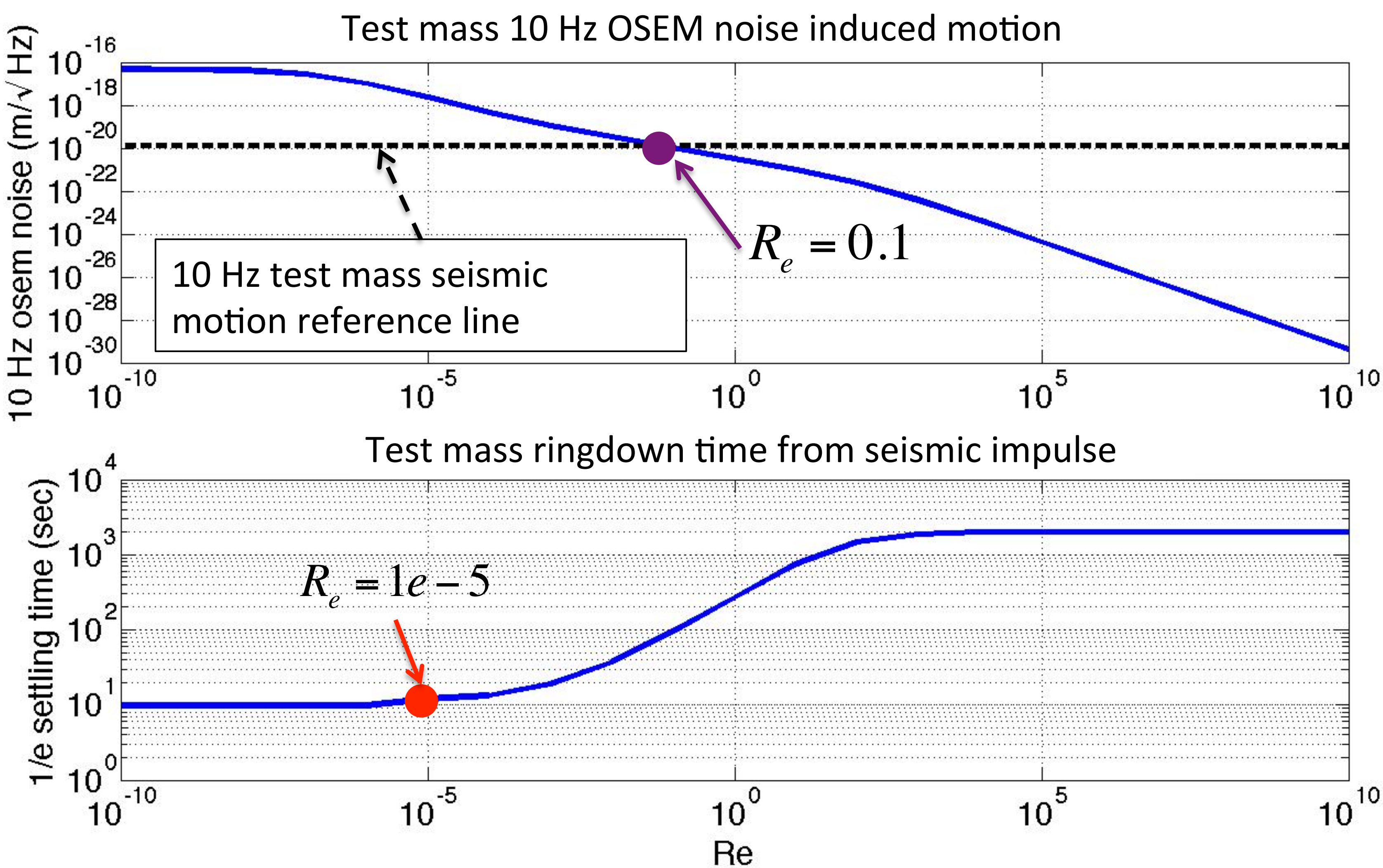
Example 2 – quad damping



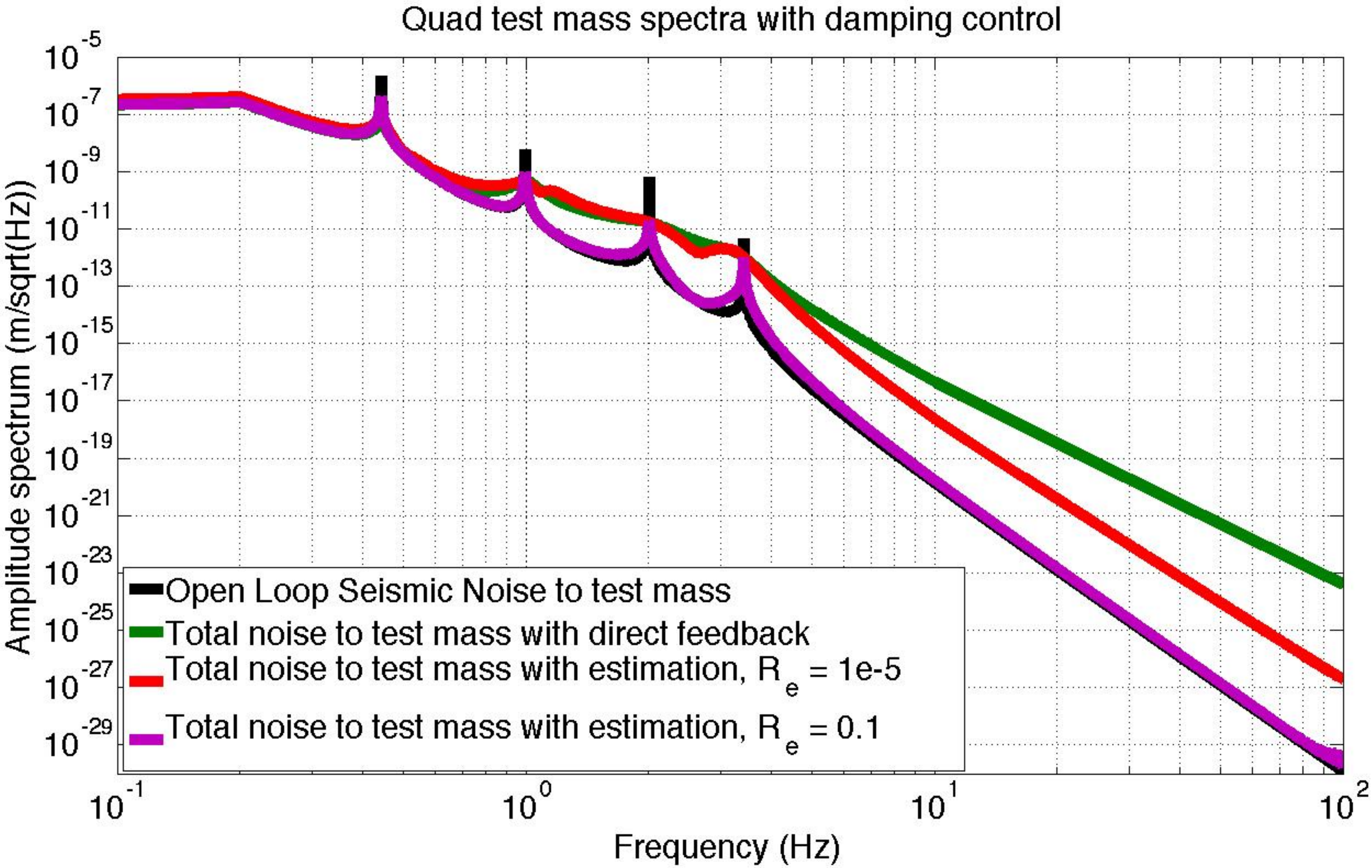
We can see what R_e values give us convenient trade-offs between noise and damping

Example 2 – quad damping

Scanning test mass noise and damping time performance for R_e



Example 2 – quad damping

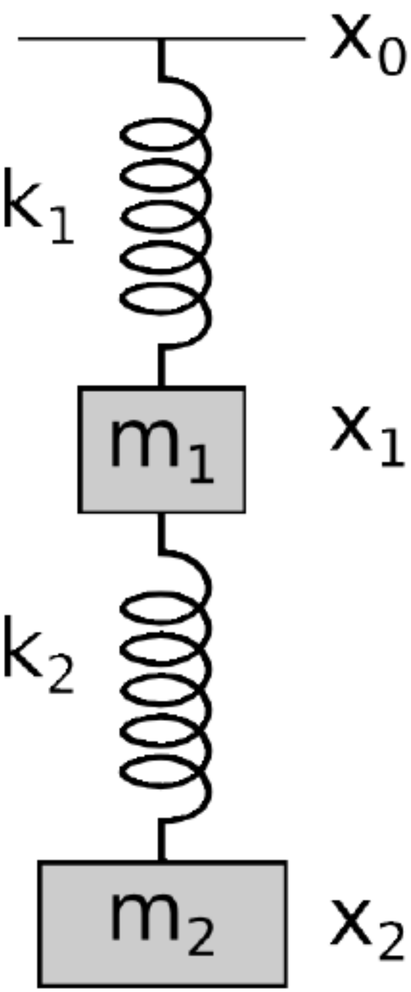


Adding complexity – plant augmentation

- One of LQR's limitations is that the controller has no more complexity than the plant. Consequently, the controller design has finite degrees of freedom.

- If we augment the plant with some fictitious states, we can achieve some more design degrees of freedom.

Example – double pendulum plant augmentation

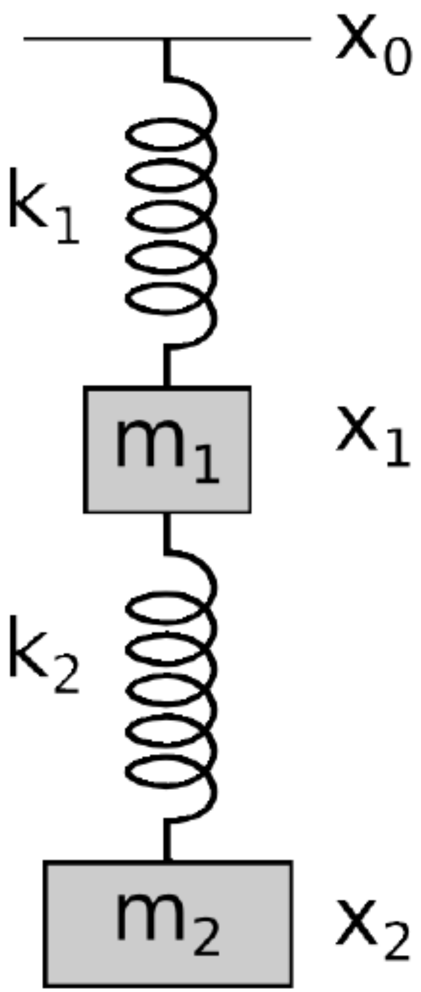


Problem: we want to lock a cavity with this double pendulum

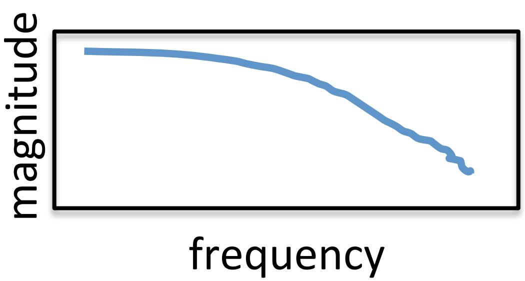
- there is very large seismic noise around the microseism
- The test mass actuator has limited range, but is most effective at high frequencies

See [3]

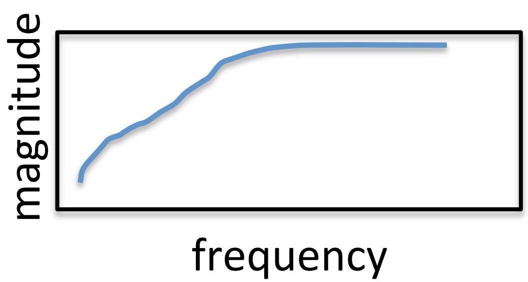
Example – double pendulum plant augmentation

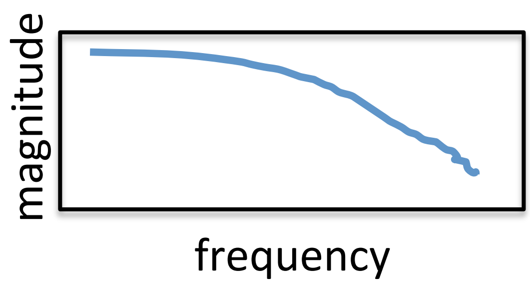


- Choose a low pass weighting function for the test mass displacement



- Choose a high and low pass weighting function for the top and bottom forces respectively





Combine the state spaces of the plant and weighting filters

See the details in T1300301 [3]

For a system state space $\dot{x} = Ax + Bu$

$$\dot{\chi} = F_x \chi + G_x x \qquad \bar{x} = H_x \chi + S_x x$$

Weight function for system state

$$\dot{\mu} = F_u \mu + G_u u \qquad \bar{u} = H_u \mu + S_u u$$

Weight function for control inputs

$$\bar{A} = \begin{bmatrix} A & 0 & 0 \\ G_x & F_x & 0 \\ 0 & 0 & F_u \end{bmatrix}$$

Augmented A matrix

Note: the Matlab functions *zp2ss* or *tf2ss* will take you from the frequency domain to state space matrices. This makes it easy to design weights with poles and zeros while still augmenting the state space.

Combine the state spaces of the plant and weighting filters

Augmented B matrix

$$\bar{B} = \begin{bmatrix} B \\ 0 \\ G_u \end{bmatrix}$$

Augmented state vector

$$z = \begin{bmatrix} x \\ \chi \\ \mu \end{bmatrix}$$

We then solve the LQR problem of the augmented state space

$$\dot{z} = \overline{A}z + \overline{B}u$$

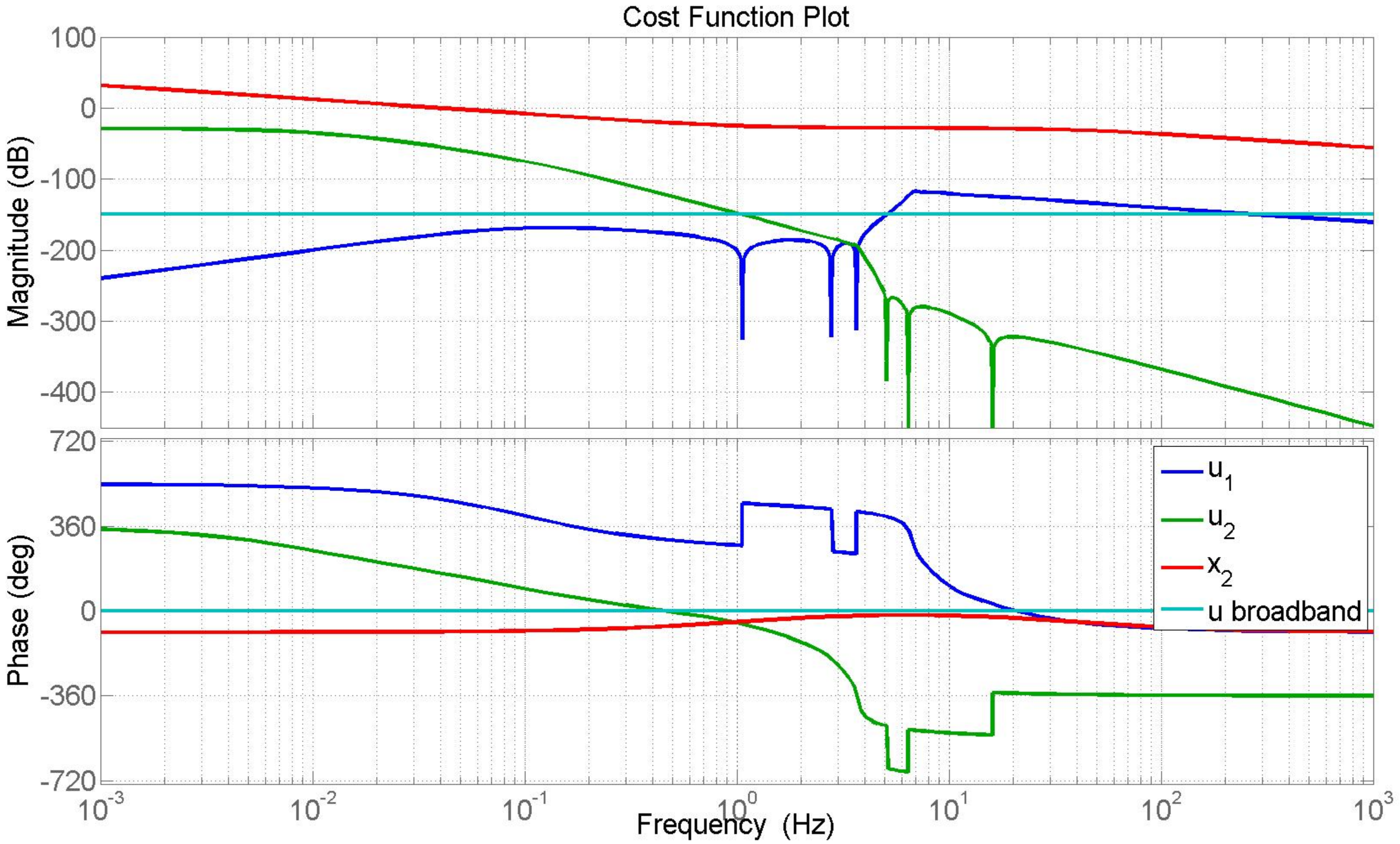
with the following cost function

$$J = \int_0^\infty \left(z^T Q z + u^T R u + 2 z^T N u \right) dt$$

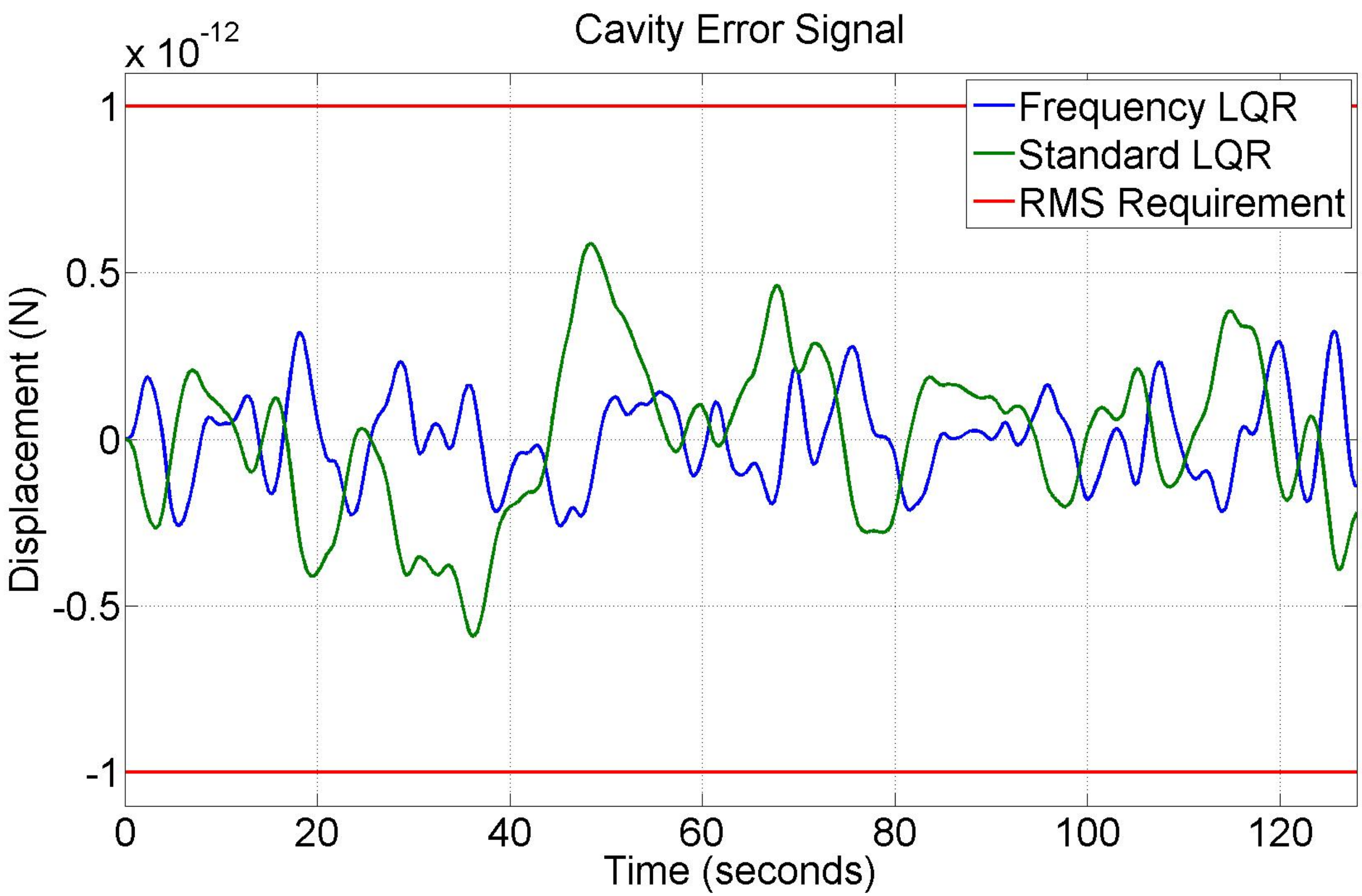
where

$$Q = \begin{bmatrix} S_x^T S_x & S_x^T H_x & 0 \\ H_x^T S_x & H_x^T H_x & 0 \\ 0 & 0 & H_u^T H_u \end{bmatrix} \quad R = S_u^T S_u \quad N = \begin{bmatrix} 0 \\ 0 \\ H_u^T S_u \end{bmatrix}$$

T1300301 Example

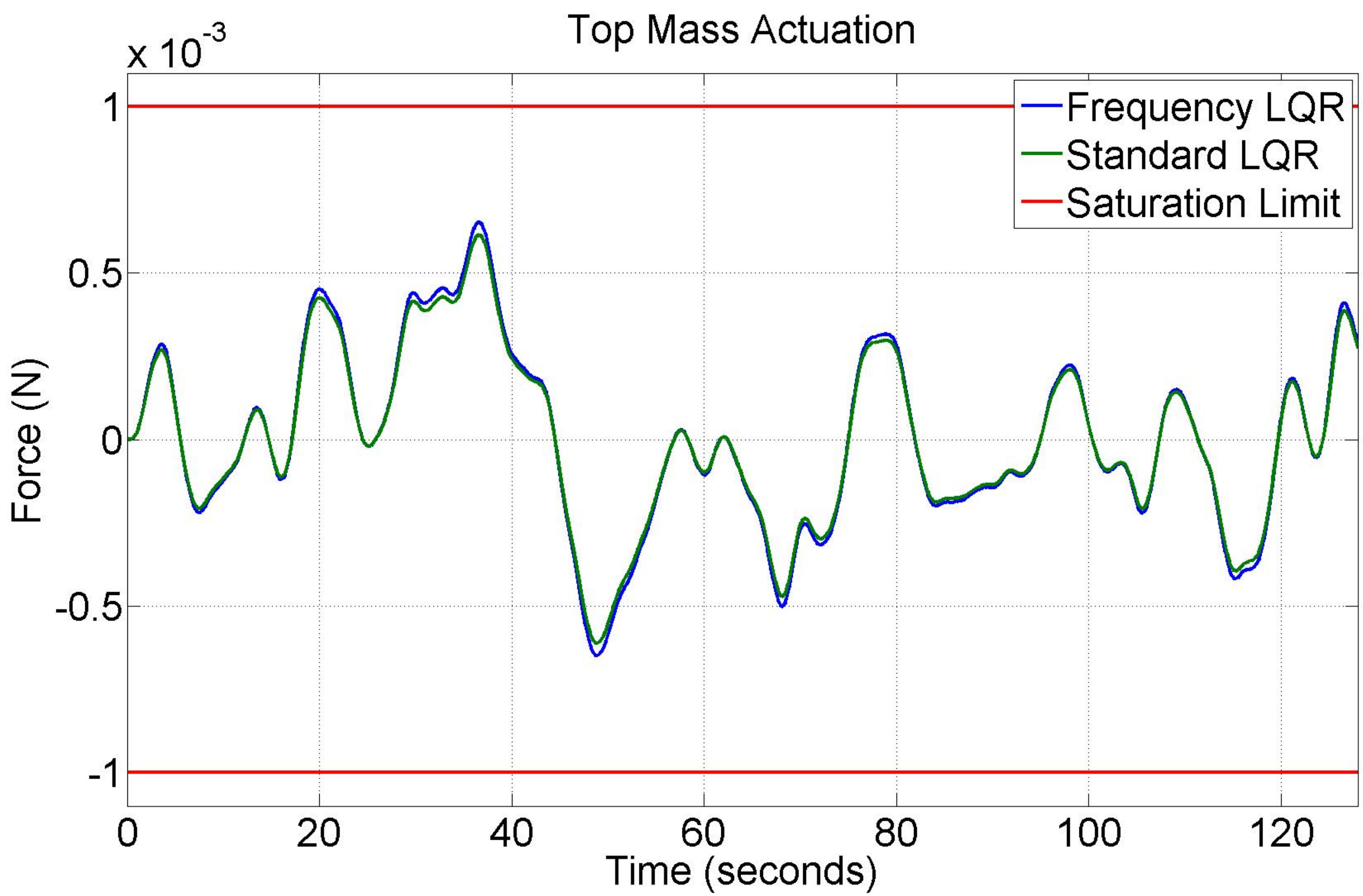


T1300301 Example



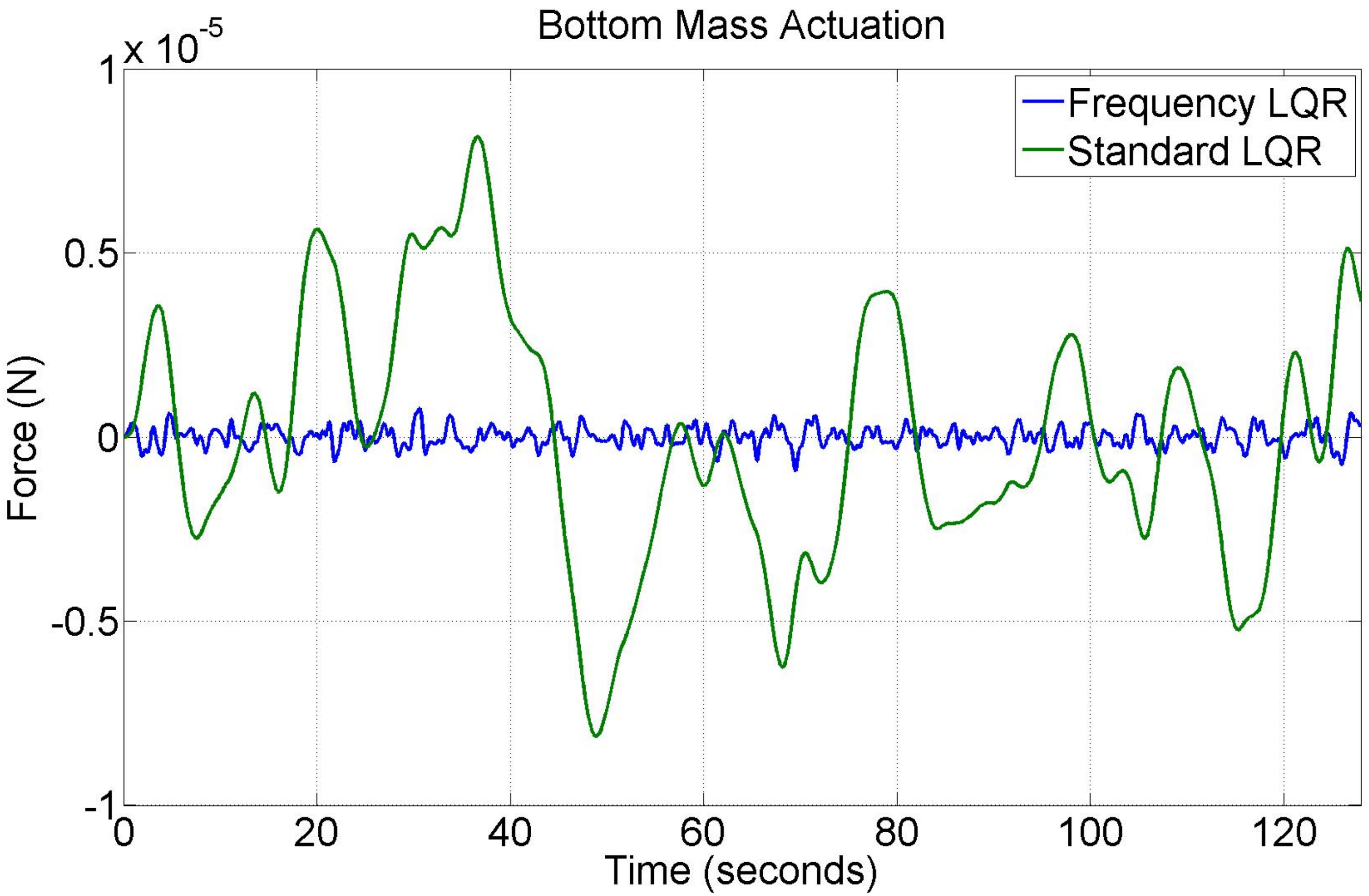
Cavity closed loop error chosen be the same for both standard and frequency weighted LQR

T1300301 Example



The top mass actuation is also approximately the same

T1300301 Example



The frequency weighted approach is successful at reducing the test mass drive without compromising the error signal

Conclusions

- LQR Benefits
  - Optimal solution to control problem
  - Automated (and very fast) method to design
  - Always stable with a good model
  - The controllers are typically robust
- LQR Limitations
  - Only has as much complexity as the plant
    - This can be overcome by augmenting the plant
  - Identifying the best weighting matrices is an art

Backups

Intro to Observers/Estimators

- A state observer estimates the state of a system based on the system measurements.
  - Useful if not all states are known, and/or they are measured with noisy sensors
  - The measurements can involve states and/or inputs
  - The system must be observable
  - Observability (dual of controllable) means that the full state can be reconstructed from the outputs in finite time (recall, controllability is the ability to drive the system to any state in finite time)

Ref 1, Chapter 12-5.