

ACTIVE DAMPING OF TALL BUILDING VIBRATIONS: CONTINUED

Franz Hover, 15 November 2007

Review of Objectives

- Lab 7: Create a simplified model of the open-loop system, using the given notes and physical data on the plant. Write this model in state-space form, and use Matlab to create an impulse response plot from the controlled input to the measurement.
- Lab 8: Employing Matlab, design an active damping system built on the PID controller. We recommend to use the nominal plant parameters that are given in this document. Model the system in Simulink, and test out the controller in simulation.

There is nothing to turn in for Lab 8.

- Lab 9: Test the controller on the real plant; prepare and turn in a report detailing the model, the controller design, and results. Does your design meet the objective of reducing tower motion due to wind disturbances?

This report will be due in the week following Lab 9.

Notes

1. State Equations

Let $x_1(t)$ be the absolute lateral position of the tower mass, and $x_2(t)$ be the absolute position of the sliding mass; $\dot{x}_1(t) = v_1(t)$ and $\dot{x}_2(t) = v_2(t)$ are the absolute velocities. Then we can define the state vector as:

$$\vec{q}(t) = \begin{Bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{Bmatrix} = \begin{Bmatrix} x_1(t) \\ v_1(t) \\ x_2(t) \\ v_2(t) \end{Bmatrix}, \text{ and}$$
$$\frac{d\vec{q}(t)}{dt} = A\vec{q}(t) + Ba(t) + Gw(t)$$

where $a(t)$ is the control action (i.e., force from the voice coil), and $w(t)$ is the disturbance input. A , B , and C are constant matrices. It is important to distinguish between

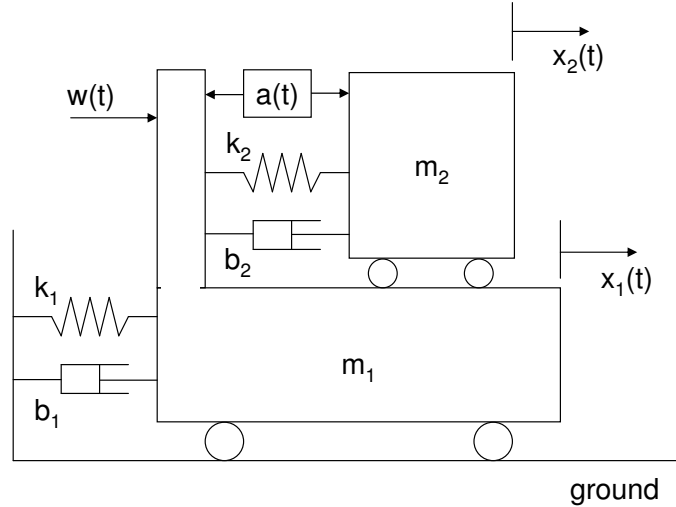


Figure 1: Schematic of the tower physical model. The input a is a force that acts to separate the masses when positive; the disturbance w acts to move x_1 positive. The measurement is relative velocity $y = \dot{x}_2 - \dot{x}_1$.

the two input gain matrices B and G because the two inputs $a(t)$ and $w(t)$ enter the system differently: the actuator force works to separate the two masses, whereas the external input pushes on the building (m_1) from outside.

With masses, springs, and dampers labeled as in the figure, we get for the state equations

$$\frac{d\vec{q}(t)}{dt} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1+k_2}{m_1} & -\frac{b_1+b_2}{m_1} & \frac{k_2}{m_1} & \frac{b_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{b_2}{m_2} & -\frac{k_2}{m_2} & -\frac{b_2}{m_2} \end{bmatrix} \vec{q}(t) + \begin{bmatrix} 0 \\ -\frac{1}{m_1} \\ 0 \\ \frac{1}{m_2} \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix} w(t)$$

The observation with the second voice coil is the relative velocity of the two masses:

$$y(t) = [0 \quad -1 \quad 0 \quad 1] \vec{q}(t)$$

Note that the sign of the entries here is chosen so that positive control action (separating the masses) will lead to a positive measurement (m_2 moving in the direction it is pushed). This is a very good practice to use in designing feedback systems.

2. Model Parameters

- (a) Each of the two voice coils has the constant $k = 7.1 \text{ N/A} = 7.1 \text{ V/(m/s)}$. Don't forget these numbers (or the gain of the current amplifier) when you implement your design in hardware!

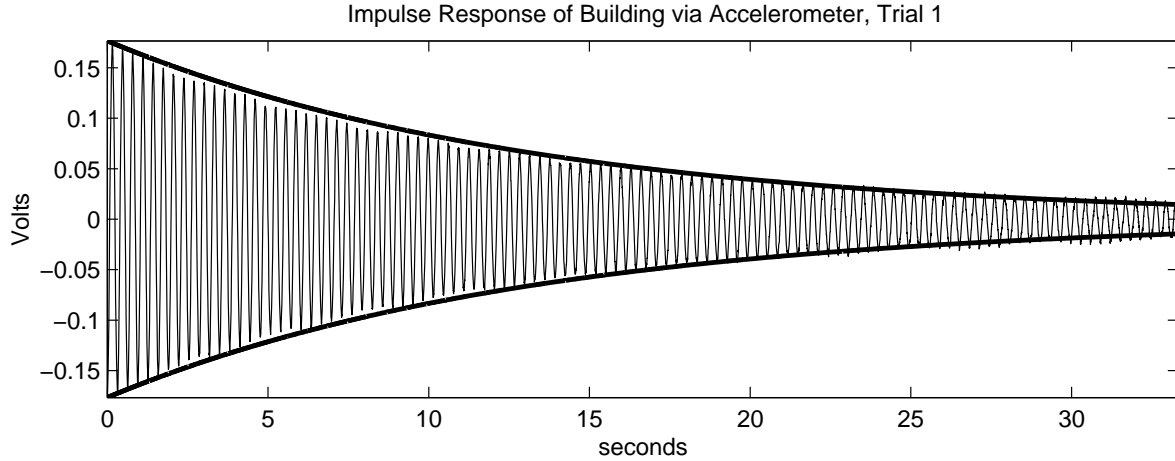


Figure 2: Accelerometer measurement of building response to non-zero initial conditions, with an exponential envelope fitted. The sliding mass is not installed for this test, file tower_base_1.m.

(b) The masses are

$$\begin{aligned} m_1 &= 5.11 \text{ kg} \\ m_2 &= 0.87 + 0.075n \text{ kg}, \end{aligned}$$

where n is the number of additional steel collets placed on the shaft.

(c) Our analysis of the file tower_base_1.m gave the following values for the single mass m_1 :

$$\begin{aligned} \tau &= 13.3 \text{ seconds} \\ \omega_d &= 19.9 \text{ rad/sec} \\ \zeta &= 0.0038 \\ \omega_n &= 19.9 \text{ rad/s} \\ k_1 &= 2020 \text{ N/m} \\ b_1 &= 0.77 \text{ N/(m/s)} \end{aligned}$$

The first two values come directly from inspecting the data. Then:

- i. The time constant of the decay envelope tells us the real part of the poles:
 $\sigma = 1/\tau$.
- ii. The undamped natural frequency is $\omega_n = \sqrt{\omega_d^2 + \sigma^2}$.
- iii. The damping ratio is $zeta = \sigma/\omega_n$.
- iv. $k_1 = m_1\omega_n^2$.
- v. $b_1 = 2\zeta\sqrt{k_1m_1}$.

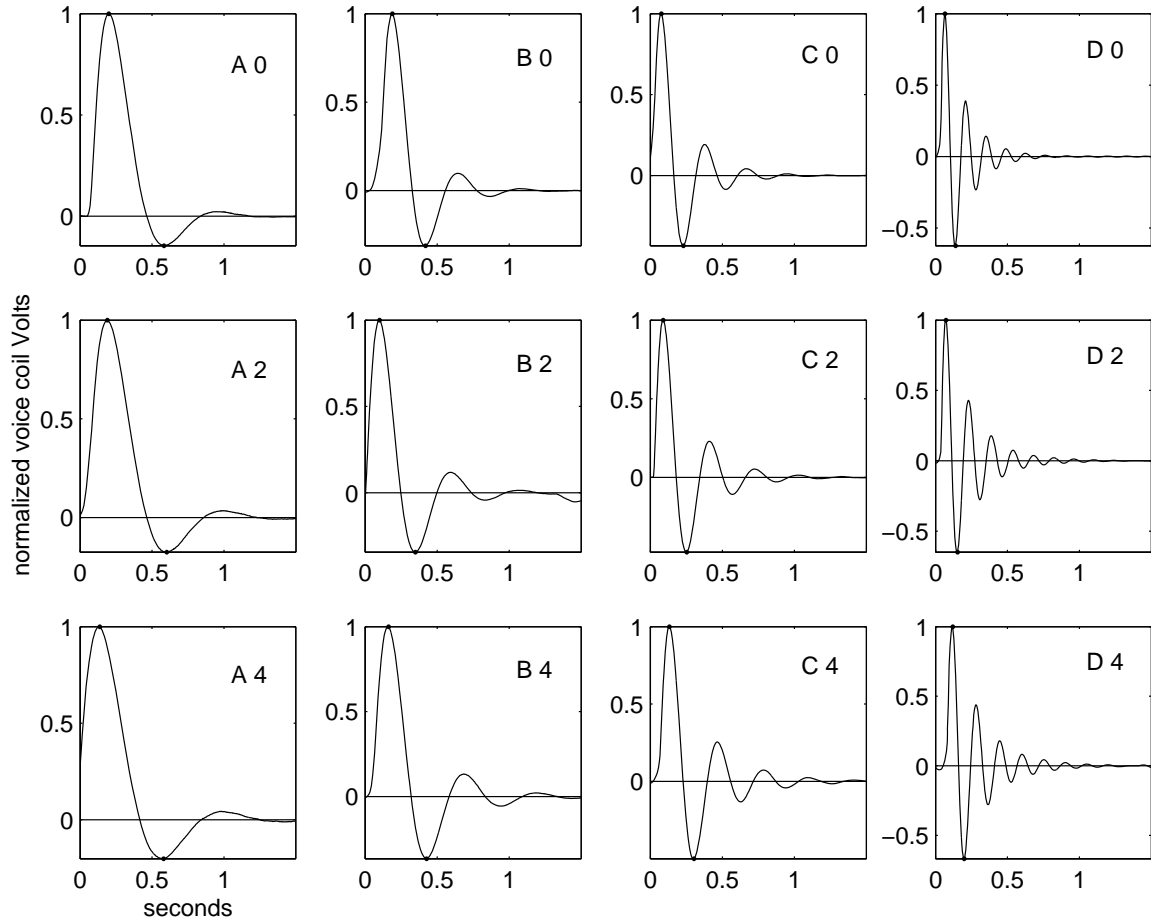


Figure 3: Typical responses of the sliding mass when the building is fixed. The spring configuration and number of additional collets is indicated. Amplitude shown is normalized and therefore dimensionless.

- (d) Our analysis of the file `actuator_A_0.m` gave, for the mass m_2 moving while m_1 is fixed:

$$\begin{aligned}
 \text{overshoot } os &= 0.142 \\
 \omega_d &= 8.2 \text{ rad/sec} \\
 \zeta &= 0.52 \\
 \omega_n &= 9.6 \text{ rad/s} \\
 k_2 &= 81 \text{ N/m} \\
 b_2 &= 8.8 \text{ N/(m/s)}
 \end{aligned}$$

These values are for the case with no additional collets, i.e., $n = 0$ in the above mass formula, and with the spring in configuration A.

The overshoot and damped natural frequency are measured directly from the

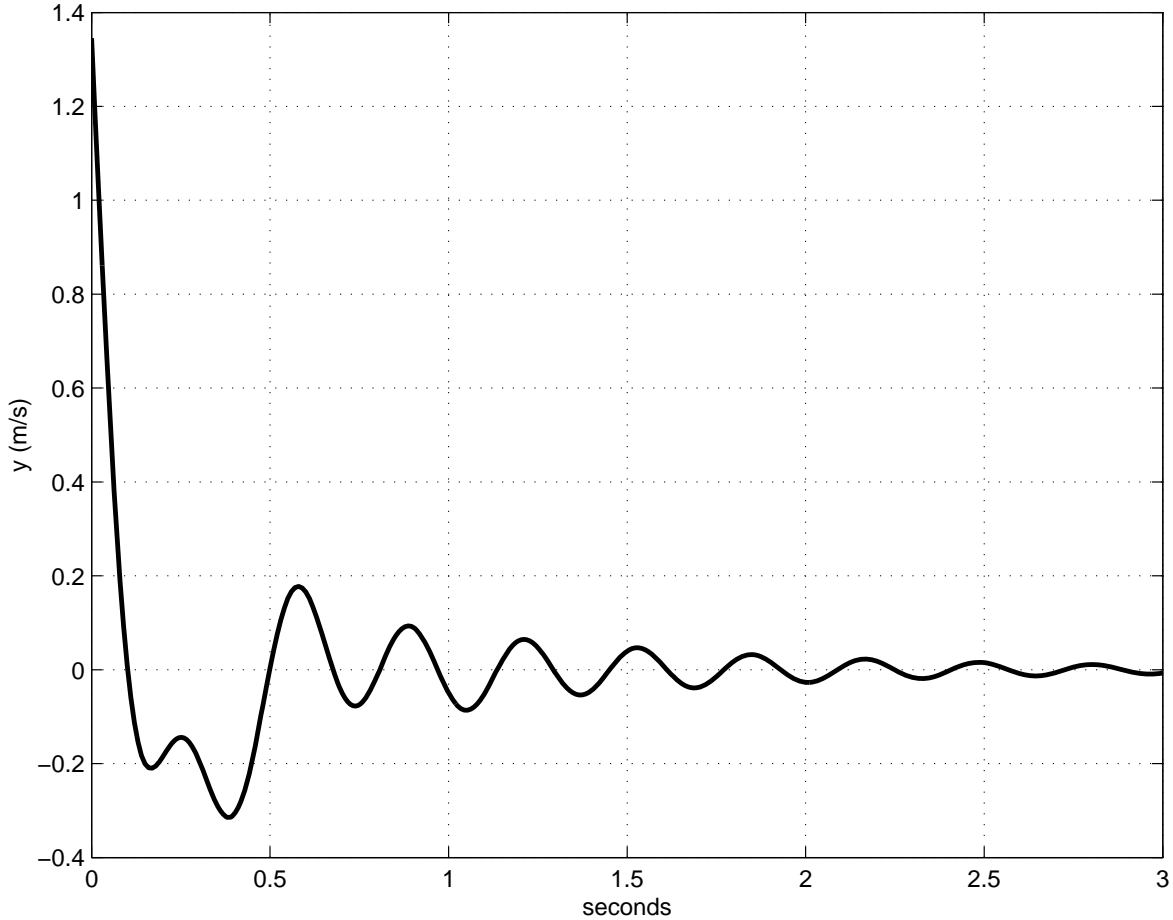


Figure 4: Open-loop impulse response with the nominal plant parameters: measurement $y(t)$ is driven by an impulse in the actuator $a(t)$. Note these are taken in units of m/s and N , respectively.

observed data. Then:

- i. We found iteratively a damping ratio to satisfy the equation $os = \exp(-\zeta\pi/\sqrt{1-\zeta^2})$.
 - ii. The undamped natural frequency is $\omega_n = \omega_d/\sqrt{1-\zeta^2}$.
 - iii. $k_2 = m_2\omega_n^2$.
 - iv. $b_2 = 2\zeta\sqrt{k_2m_2}$.
- (e) For your reference, we have also provided the sliding mass data for a number of other conditions; the letter in the file name indicates the spring length (A,B,C, or D - from longest to shortest), and the number indicates how many steel collets were attached to the shaft (0,2, or 4). The A_0 condition will give you good results with your active controller. FOR EXTRA CREDIT, explore the possibilities of different spring and mass combinations, and discuss how they affect the ability of

the sliding mass to act as a *passive* damper.

Note that a complete analysis of these data shows that for spring positions C and D, there is a marked change in b_2 , suggesting that a nonlinear effect is coming into play, perhaps due to side loading of the air bearings because the spring is so short.

3. A Limitation on Matlab Systems

Matlab does not allow the explicit definition of systems having more zeros than poles. A PID controller has two zeros, and one pole (at the origin), so you need to take an extra step to build and manipulate a distinct PID object. Augment the regular PID with a fast pole as follows:

```
sysPID = tf([kd kp ki], [tr 1 0]) ;
```

where `tr` is set to be a small time constant compared to the other time scales in the system, e.g., 0.01 or less in our case. In the root locus plot, this will add a fast pole which only affects the other loci when the gain is very high.

4. The Effect of Disturbance on the Building Motion

In Lab 7, you focused on the system that maps control input (separation force generated by one of the voice coils) to plant output (relative velocity reported by the second voice coil). In practical terms, however, it is the effect of the disturbance $w(t)$ on the building motion, say $z(t) = x_1(t)$, that is most important in your design. Luckily, putting the closed-loop poles in the right place makes the system behave well from both points of view. For the tower labs, you can explore this property directly using Simulink.

The following discussion shows in more detail, and without resorting to Simulink, how to assess the closed-loop response for input $w(t)$ and output $z(t)$, given a controller design that is based only on the transfer function from $a(t)$ to $y(t)$. *You do not have perform these analyses to receive full credit on the project - but you do have to demonstrate a reduction in building motion caused by a wind disturbance, when your active controller is running.*

First, to calculate how the open-loop system behaves with a disturbance input $w(t)$ and output $z(t)$, use the state-space description

$$\begin{aligned} \frac{d\vec{q}(t)}{dt} &= A\vec{q}(t) + Gw(t) \\ z(t) &= C_2\vec{q}(t) \end{aligned}$$

where $C_2 = [1 \ 0 \ 0 \ 0]$, for example, to pull out $x_1(t)$, the position of mass m_1 . If you make a new system in Matlab using

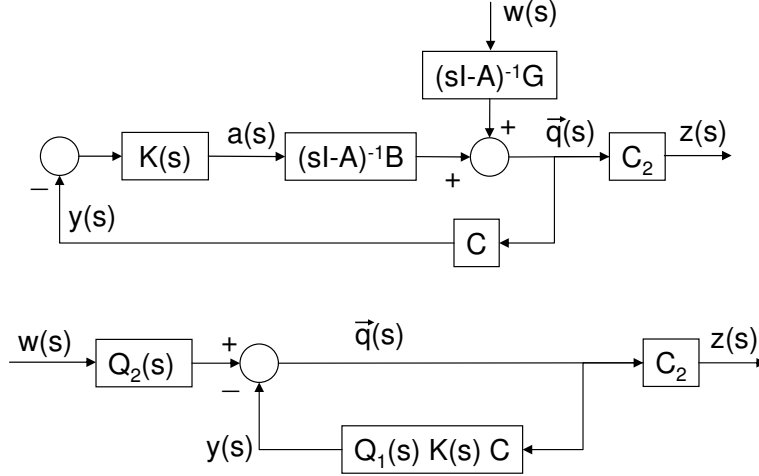


Figure 5: Equivalent block diagrams for assessing the closed-loop transfer function from disturbance $w(s)$ to auxiliary output $z(s)$. The compensator is $K(s)$; in the lower version, $Q_1(s) = (sI - A)^{-1}B$ and $Q_2(s) = (sI - A)^{-1}G$.

```
sys0Lbuilding = ss(A,G,C2,0);
```

then its input is taken as $w(t)$, and the output will be $z(t)$.

To calculate how the closed-loop system behaves with an impulse disturbance, we have to develop a new block diagram, as shown in the figure. We take advantage of the system linearity to write (in the Laplace domain)

$$\begin{aligned}
 \vec{q}(s) &= (sI - A)^{-1}Ba(s) + (sI - A)^{-1}Gw(s) \\
 &= -(sI - A)^{-1}BK(s)y(s) + (sI - A)^{-1}Gw(s) \\
 &= -(sI - A)^{-1}BK(s)C\vec{q}(s) + (sI - A)^{-1}Gw(s) \\
 &= \left(I + (sI - A)^{-1}BK(s)C \right)^{-1} (sI - A)^{-1}Gw(s) \\
 &\quad \text{(use the fact that for square matrices } (XY)^{-1} = Y^{-1}X^{-1} \text{)} \\
 &= \left((sI - A)(I + (sI - A)^{-1}BK(s)C) \right)^{-1} Gw(s) \\
 &= (sI - A + BK(s)C)^{-1} Gw(s) \quad \text{so that} \\
 \frac{z(s)}{w(s)} &= C_2(sI - A + BK(s)C)^{-1}G.
 \end{aligned}$$

This has the familiar structure of an output matrix multiplied by an inverse matrix involving s , multiplied by an input gain matrix. How do you get Matlab to do it? Recognizing that $(sI - A)^{-1}G$ simply premultiplies $w(s)$ coming into the feedback loop, and that C_2 post-multiplies to give the output, you can use the regular feedback construction, which has unity (or the identity matrix) in the forward path and $H(s) = (sI - A)^{-1}BK(s)C$ in the backward path. In Matlab, this looks like

```

sysK = tf([kd kp ki], [tr 1 0]) ; % the controller (a PID as written)
sysQ1 = ss(A, B, eye(4,4), zeros(4,1)) ; % makes  $Q1 = (sI - A)^{-1}B$ 
sysQ2 = ss(A, G, eye(4,4), zeros(4,1)) ; % makes  $Q2 = (sI - A)^{-1}G$ 
sysCLWToZ = C2 * feedback(eye(4,4),sysQ1*sysK*C) * sysQ2 ; % the closed-
loop system taking input w to output z
impulse(sysCLWToZ) ; % plots the impulse response from input w to output z

```

5. Some Matlab Hints

```

% comment the rest of the line
help ss ; % show help on the function called ss
Q = [ 1 2 3 ; 4 5 6 ; 7 8 9] ; % makes a 3x3 matrix Q
sysSS = ss(A,B,C,D) ; % makes the state-space system defined by the four matrices
sysTF = tf([0 1],[1 2]) ; % makes the system  $1/(s+2)$ 
impulse(sys) ; % plots the impulse response of the system (both SS and TF forms
will work)
[A,B,C,D] = ssdata(sysTF) ; % gives one state-space realization of the transfer
function representation
[num,den] = tfdata(sysSS); % gives the (unique) numerator and denominator of
the state-space system
[z,p,k] = zpkdata(sys) ; % reports zeros, poles, and gain of the system
sys12 = sys1 * sys2 ; % carries out frequency-domain multiplication of systems:
sys1 operating on the output of sys2 (not the other way around!)
sysCL = feedback(sys1*sys2,1) ; % creates the closed-loop system with sys1 and
sys2 in the forward path and one in the feedback path
rltool(sys) ; % starts up the root locus design tool
rlocus(sys,kpvec) ; % plots the root locus for the proportional gains in the vector
kpvec

```

6. A Final Word About Your Report

Your task is to document the process and rationale for your controller design.

Your report should be concise and professional. We expect to see good grammar and spelling, organization into sections, and clear, annotated figures that support the text. We can't give you the good grade you deserve if we can't figure out what you did!