

ACTIVE DAMPING OF TALL BUILDING VIBRATIONS

Franz Hover, 2 November 2007

Overview

The high-rise building is a modern miracle - miles of steel beams and welds, thousands of fasteners allowing graceful structures of one hundred or more stories in height. Like any high-aspect ratio structure, the skyscraper is flexible. You might not notice this until a strong wind-storm sets up large-scale vibrations in the first bending mode. Then the motions will make you ill, or at a minimum cause fatigue. The motions certainly cause damage to the building, notably in the loss of windows which can crack or fall, and in long-term fatigue life reduction. Among potential remedies for building sway, the most common today is the passive or active mass concept. In fact, our own Hancock Tower in Boston has two 300-ton masses near the top floor, that damp out vibrations caused by wind.

In these final three lab sessions you will study in detail a physical structure with a similar dynamic response, creating a linear model from first principles and using Simulink to characterize it, synthesizing an active control system design based on the model, and testing your controller on the actual device. The teaching staff will act as consultants.

- Lab 7: Create a simplified model of the open-loop system, using the attached notes and data on the physical properties of the plant. You will write this model in state-space form, and use Matlab to (numerically) convert it to a transfer function.
- Lab 8: Employing Matlab, design an active damping system built on the PID controller you have used for the flywheel plant. You will model your system in Simulink, and test out your controller in simulation.
- Lab 9: Test your controller on the real plant; prepare and turn in a report detailing your model, the controller design, and your results. Be sure to document the process and rationale for your controller design!

Supplemental Notes

1. The Step Response vs. the Impulse Response

The prototypical input we have used in this class is the step function, $u(s) = 1/s$ or $u(t) = 1$, for all $t > 0$, and zero otherwise. You should be familiar with the step of multiplying in the LaPlace domain the step function with the transfer function, to obtain the LaPlace transform of the step response:

$$y(s) = G(s)u(s) = \frac{G(s)}{s}$$

where $G(s)$ is the transfer function, and $y(s)$ is the output. The step response is meaningful for many controlled systems, for example corresponding to a step change in torque in the open loop, or a step change in reference voltage in the closed-loop.

In the tower labs, we will excite the physical system with an impulse function. The impulse, or delta function $\delta(t)$ is many things:

- (a) the derivative of the step function,
- (b) an infinitely tall and infinitesimally narrow spike, whose area is exactly one.
- (c) a strong but instantaneous hit to the system, like a hammer or a lightning strike!

Using the differentiation rule in the LaPlace domain, we have

$$L\{\delta(t)\} = 1,$$

leading to the observation that *a transfer function is its own impulse response!*

2. Suggested State Variables in Your Model

Your 2.003 and 2.004 experience tells you that the lab structure is a coupled pair of masses. The mass of the building itself, assumed to be concentrated at the top, is connected to the ground through an effective spring and damper; the "counterweight" is connected to the building mass through a spring and damper. The control is an additional force exerted between the two masses, in parallel with the counterweight spring and damper.

Because the physical system consists of two masses, connected by springs and dampers, it is of fourth-order - the position and velocity of each mass is needed to completely define the state of the system. Using " $F = ma$ ", develop the equations this way:

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

where the derivative of the vector is the same as the vector of derivatives, $a(t)$ is the control action (i.e., force from the voice coil), and $w(t)$ is the external input. It is important to distinguish these two input gain vectors B and G because the two inputs enter the system differently: the actuator force works to separate the two masses, whereas the external input pushes on the building from outside.

3. Obtaining Model Parameters

- (a) The voice coil has similar properties to an electric motor, namely

$$\begin{aligned}\text{force} &= k \times \text{current} \\ \text{back emf} &= k \times \text{velocity}\end{aligned}$$

For each of the voice coils here, $k = 7.1N/A = 7.1V/(m/s)$. Don't forget these numbers (or the gain of the current amplifier) when you implement your design in hardware!

- (b) We weighed the entire upper assembly of the building without the sliding mass (m_1), and then the sliding mass alone (m_2) and found

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

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

- (c) Then we attached the upper assembly without the sliding mass to the aluminum struts - this comprises "the building." An accelerometer was placed to record lateral accelerations, and we observed the vibrations of the structure when released from a non-equilibrium condition. These voltages with the time vector are recorded in the file

`tower_base_1.m`

which you can run within Matlab in the usual way. Your task is to estimate the natural frequency and the damping ratio of this structure, and thereby compute k_1 and b_1 . For your reference, the accelerometer has the approximate calibration $0.0453V/(m/s^2)$.

- (d) Finally, we fixed the building and studied the behavior of the sliding mass, when released from a non-equilibrium initial position. We used the voice coil as a velocity sensor in this case. The data is found in the file

`actuator_A_0.m`

This particular file has the spring clamp at its furthest extension, and no additional masses affixed to the shaft. As above, your task is to estimate the natural frequency and damping ratio of the sliding mass, with the rest of the structure fixed. This will give you k_2 and b_2 .

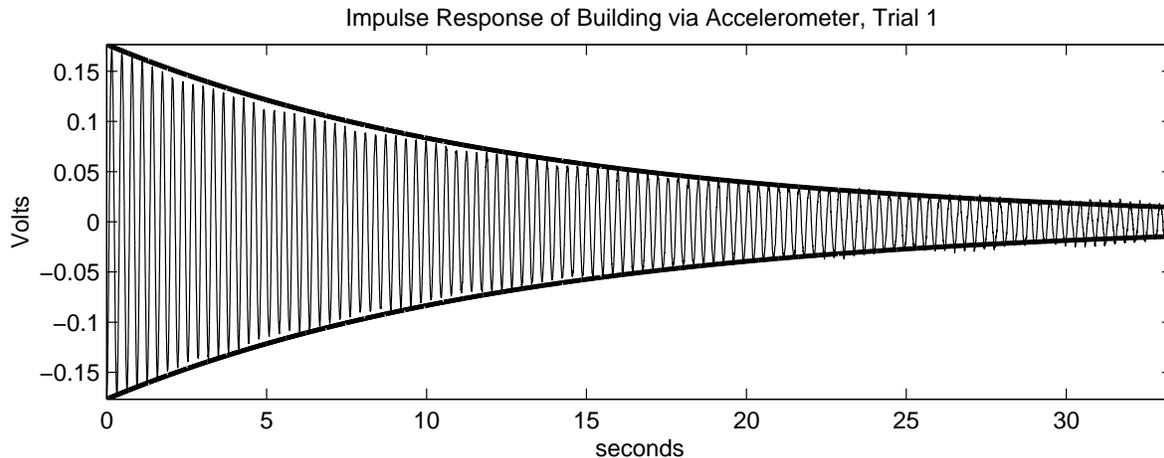


Figure 1: Accelerometer measurement of building response to non-zero initial conditions, with an exponential envelope fitted. The sliding mass is not installed for this test, file tower_base_1.m.

- (e) For your reference, we have also included 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.

4. 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
```

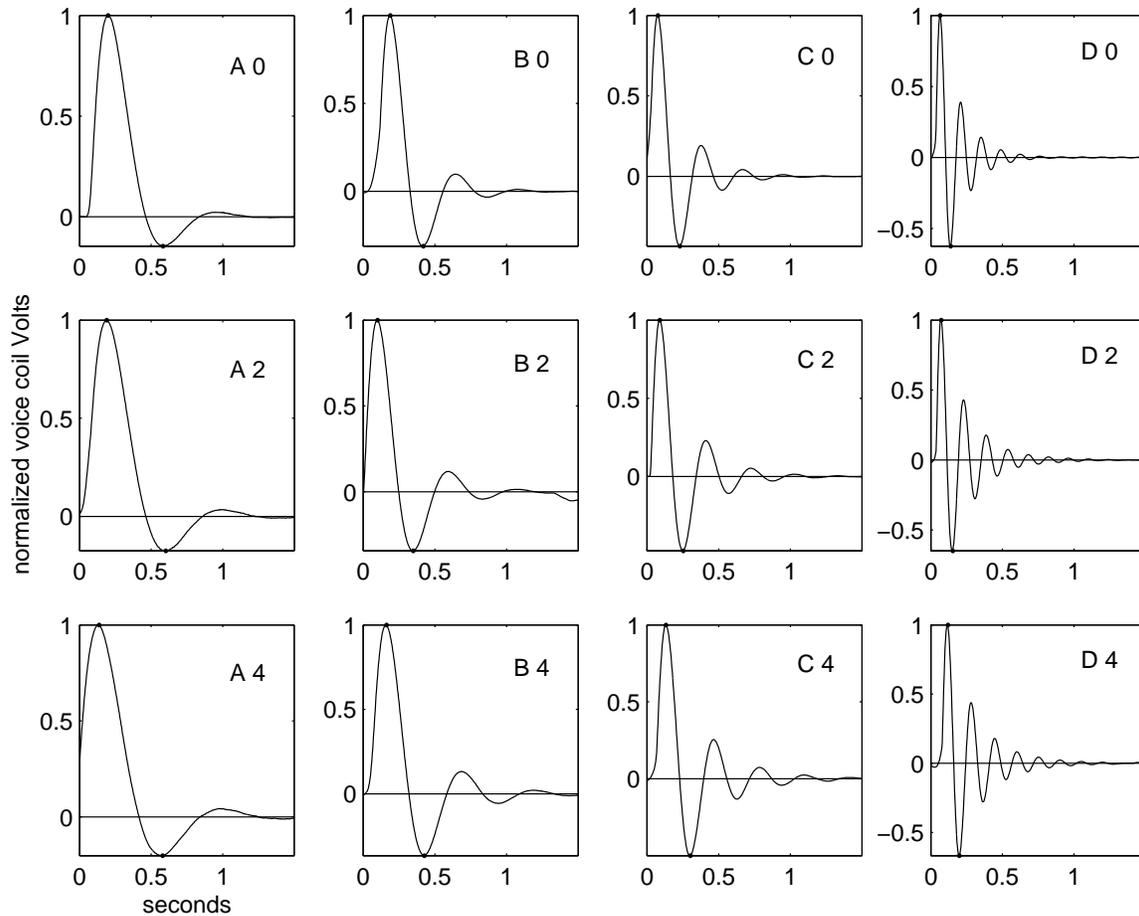


Figure 2: Typical responses of the sliding mass when the building is fixed. The spring configuration and number of additional collets is indicated.

```
[z,p,k] = zpndata(sys) ; % reports zeros, poles, and gain of the system
sysCP = sys1 * sys2 ; % carries out frequency-domain multiplication of systems:
sys1 operating on the output of sys2
sysCL = feedback(sys1*sys2,1) ; % creates the closed-loop system with 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
```

5. A Final Word About Your Report

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!