

MAE 143B - Homework 1

1.1

a)

This block-diagram shows an *open-loop* system with voltage *input* and speed (velocity) *output*. There are no *disturbances*. The remainder of the question depends on what the intermediate signal between the two blocks is. If we choose it to be torque, the sensor block should be dynamic, whereas if we choose it to be velocity, the sensor block should be static. In either case, most motor models relating input voltage to output torque or velocity are dynamic. If we choose the intermediate signal to be velocity, an example transfer function for the motor block could be

$$M(s) = \frac{K_m}{(Ls + R)(Js + B) + K_m K_b},$$

where K_m is the motor-torque constant, L the armature inductance, R the armature resistance, J the moment of inertia for the motor, B the viscous-friction coefficient for the motor and K_b the back-EMF constant, respectively. An ideal sensor could be modeled using the transfer function

$$S_1(s) = 1.$$

b)

If we choose our intermediate signal to be velocity, everything is as in the previous part except for the sensor model now having to be dynamic. For example,

$$S_2(s) = s$$

would be a model for an ideal sensor generating the acceleration output signal from the intermediate velocity signal.

e)

This block-diagram shows an *open-loop* system with hot, cold and two position *inputs* as well as a water *output*. There are no *disturbances*. The two faucets could be modeled using as static components such as

$$w_{f,1}(t) = \alpha_1(t)w_{in,1}(t), \quad w_{f,2}(t) = \alpha_2(t)w_{in,2}(t),$$

where $\alpha_i \in [0, 1]$ are the position inputs, $w_{in,i}$ the hot and cold inputs and $w_{f,i}$ the intermediate signals after the two faucet blocks. The shower could be modeled by a dynamic component to account for the delay experienced under a shower, such as

$$w_{out}(t) = w_{f,1}(t - \tau) + w_{f,2}(t - \tau)$$

for some $\tau > 0$, where w_{out} is the water output.

f)

This block-diagram shows a *closed-loop* system with reference temperature *input* and water (temperature) *output*. There are no *disturbances*. The thermostat compares the reference temperature with the actual water temperature and controls the heater. A simple model for the thermostat could be

$$e_h(t) = t_r(t) - t_w(t),$$

where t_r is the reference temperature, t_w the measured water temperature and e_h the temperature difference to be bridged by the heater. A simple dynamic model for the heater could then be

$$H(s) = \frac{1}{s + \gamma}$$

for some $\gamma > 0$.

1.4

There are many possible, more-or-less detailed ways of drawing this block-diagram. One simple example is the following diagram.

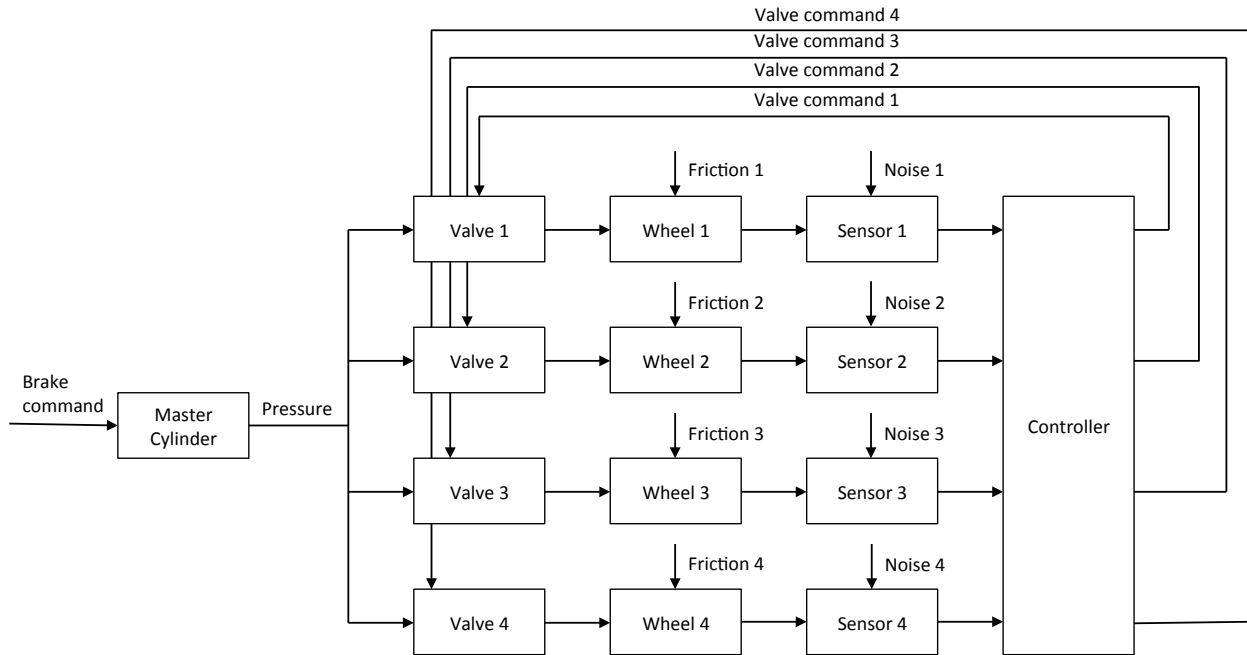


Figure 1: Block-diagram for Problem 1.4

1.7

a)

$$H(s) = G_1(s)G_2(s)$$

b)

$$H(s) = G_1(s) + G_2(s)$$

c)

We have

$$Y(s) = G(s)K(s)(U(s) - F(s)Y(s)),$$

such that

$$H(s) = \frac{Y(s)}{U(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)F(s)}.$$

d)

If we denote the output of $G_1(s)$ by $X_1(s)$, we get

$$X_1(s) = G_1(s)(U(s) - K_1(s)X_1(s) - K_2(s)Y(s)),$$

such that

$$X_1(s) = \frac{G_1(s)}{1 + G_1(s)K_1(s)}(U(s) - K_2Y(s)).$$

By $Y(s) = G_2(s)X_1(s)$, this implies

$$Y(s) = \frac{G_2(s)G_1(s)}{1 + G_1(s)K_1(s)}(U(s) - K_2Y(s))$$

and

$$H(s) = \frac{Y(s)}{U(s)} = \frac{G_2(s)G_1(s)}{1 + G_1(s)K_1(s) + G_2(s)G_1(s)K_2(s)}.$$

e)

We have

$$Y(s) = U(s) + G(s)K(s)(U(s) - Y(s)),$$

such that

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1 + G(s)K(s)}{1 + G(s)K(s)}.$$

1.8

You could use something like the code below. The resulting plots are displayed in Figure 2. The quadratic fit used here approximates the experimental data well within the given range and confirms the behavior one would expect when conducting this experiment. The intuition behind h being a quadratic function of d is reached after analyzing the idealized physics of this experiment. While the projectile accelerates down the ramp, potential energy is converted to kinetic energy as in

$$mgh = \frac{1}{2}mv^2.$$

That is, we would expect the horizontal velocity of the projectile when leaving the ramp to be proportional to the square root of h . Now notice that when approximating air resistance and similar influences, the horizontal velocity of the projectile is constant until it eventually hits the ground, where the time it takes the projectile to hit the ground does not depend on its horizontal velocity and should be approximately the same over all trials. Thus, we expect the distance d to be proportional to the velocity v of the projectile when

leaving the ramp, which by our earlier observation is proportional to the square root of the ramp height h . Inverting this relationship, we expect the ramp height h to be proportional to d^2 , which is confirmed by our data and the quadratic curve fit displayed in Figure 2.

We can approximate the ramp base height h_0 from the data based on our observations above. Thus, knowing it does not allow us to approximate additional information in our idealized model of the experiment. However, we could use the base height to improve our fits, validate the data or approximate air resistance.

Here is the MATLAB code.

```
clear all, close all;

% Read and reorganize data
D_134 = [13+15/16, 19+13/16, 27+11/16, 33+3/8 ;
        13+7/8 , 19+13/16, 27+3/4 , 33+5/16;
        14+1/16 , 19+13/16, 27+3/4 , 33+3/16;
        14 , 19+3/4 , 27+9/16 , 33+7/16;
        13+15/16, 19+3/4 , 27+9/16 , 33+5/8 ];
d_134 = reshape(D_134,size(D_134,1)*size(D_134,2),1);
l_134 = [1*ones(5,1);2*ones(5,1);4*ones(5,1);6*ones(5,1)];

D_67 = [10+11/6 , 14+1/2 , 20+3/4 , 25+7/16, 29+5/8 ;
        10+11/6 , 14+9/16 , 20+3/4 , 25+1/2 , 29+1/2 ;
        10+11/6 , 14+1/2 , 20+3/4 , 25+3/4 , 29+1/2 ;
        10+11/6 , 14+1/2 , 20+3/4 , 25+1/2 , 29+5/16;
        10+11/6 , 14+9/16 , 20+3/16 , 25+5/8 , 29+1/2 ];
d_67 = reshape(D_67,size(D_67,1)*size(D_67,2),1);
l_67 = [1*ones(5,1);2*ones(5,1);4*ones(5,1);6*ones(5,1);8*ones(5,1)];

% Convert to heights in inches
h_134 = 12*l_134*sin(13.4/180*pi);
h_67 = 12*l_67*sin(6.7/180*pi);

% Stack data and add points at (0,0)
d_data = [0*ones(5,1); d_134 ; d_67];
h_data = [0*ones(5,1); h_134 ; h_67];

% Sort data
[d_data,ind] = sort(d_data);
h_data = h_data(ind);

% Fit quadratic curve
dvec = linspace(0,40,1e4+1);

f_fit = fit(d_data,h_data,'poly2');
f_fit = f_fit.p1*dvec.^2 + f_fit.p2*dvec + f_fit.p3;

% Plot data and fit
figure;
hold all;
plot(dvec,f_fit,'LineWidth',2);
plot(d_data,h_data,'kx','LineWidth',3);
ylabel('h in inches');
xlabel('d in inches');
grid on;
ylim([0,20]);
xlim([0,35]);
```

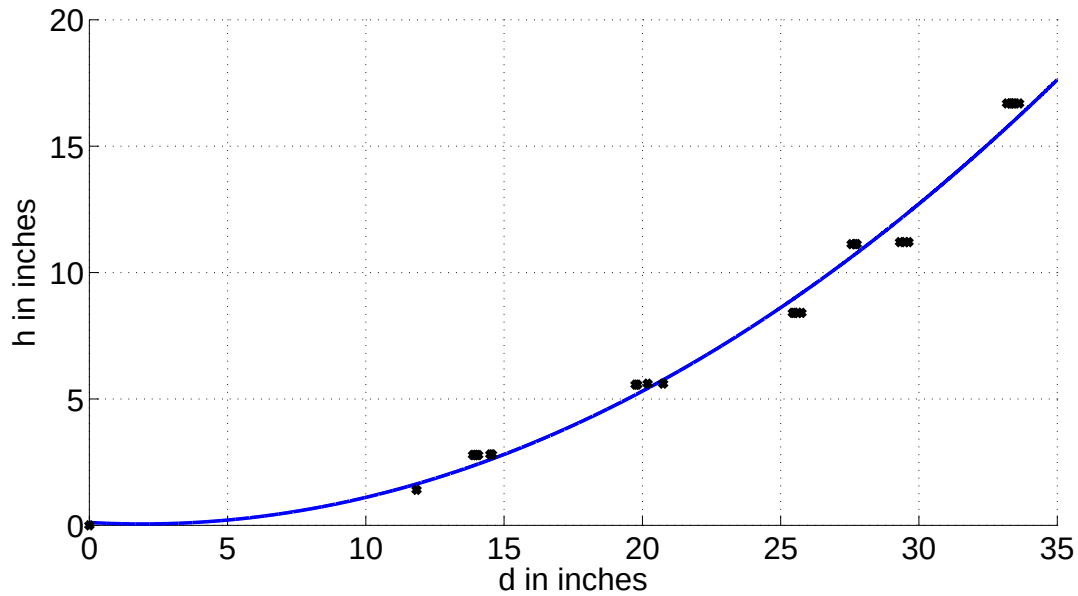


Figure 2: Plots for Problem 1.8

2.1

With Newton's help, we have

$$mg - bv = F = ma = m\dot{v},$$

or $m\dot{v} + bv = mg$.

2.2

Let's assume $g = 10m/s^2$. Following the discussion in Section 2.3 of the reader, we then have

$$\tilde{v} = \frac{gm}{b} = 1m/s, \quad \lambda = -\frac{b}{m} = 10/s, \quad \beta = v_0 - \tilde{v} = v_0 - 1m/s.$$

We can use MATLAB to plot the resulting solution using something like the following code, which results in the plot shown in Figure 3.

```
clear all, close all;

% Parameters
m = 1;           % Mass
b = 10;          % Damping
g = 10;          % Gravitational acceleration

v0 = [0;1;-1];  % Initial conditions

% Compute solution
v_ss = m*g/b;    % Steady state velocity,
                 % derivative = 0
lam = -b/m;      % Constant lambda for homogeneous
                 % solution
beta = v0 - v_ss*ones(size(v0)); % Constants beta to generate
                 % solution
```

```

t = linspace(0,1,1e4 + 1);          % Time vector
for kk = 1:length(v0)                % Get solution
    y(kk,:) = v_ss*ones(size(t)) + beta(kk).*exp(lam*t);
end; clear kk;

% Plot solution
figure;
hold all;
for kk = 1:length(v0)
    lvec{kk} = ['v(0) = ' num2str(v0(kk)) ' m/s'];
    plot(t,y(kk,:), 'LineWidth',2);
end; clear kk;
ylim([-1.5 1.5]);
legend(lvec, 'Location', 'SouthEast');
ylabel('v(t) in m/s')
xlabel('t in s')
grid on;

```

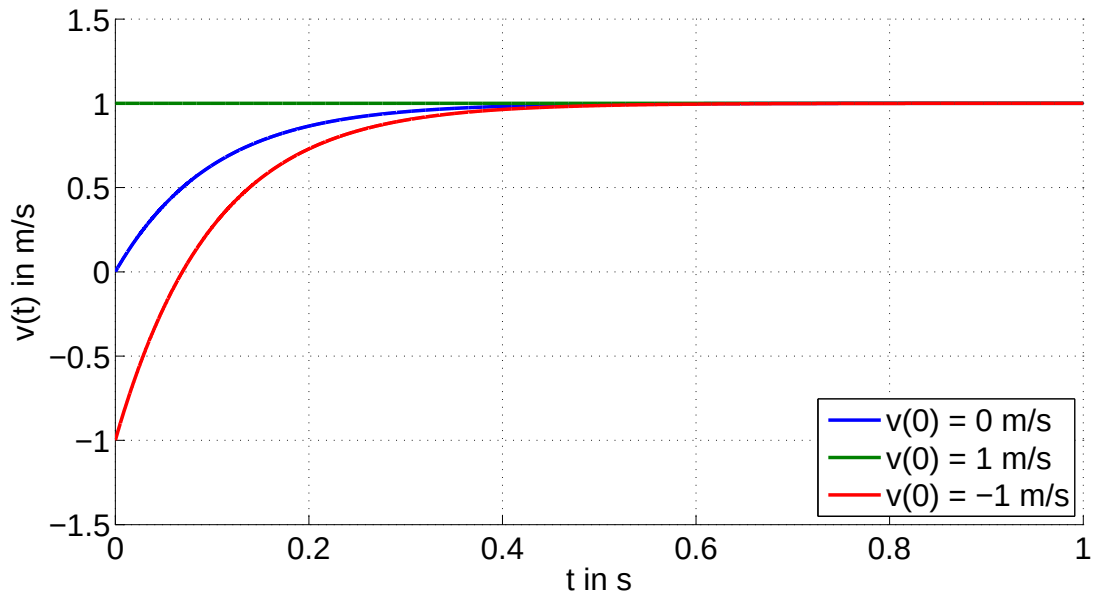


Figure 3: Solutions for Problem 2.2