EE128 Problem Set 1 Solutions

Justin Hsia
University of California at Berkeley, Fall 2008

Problem 1: Lotka-Volterra Predator-Prey Equations. Count Vito Volterra was an Italian mathematician (1860-1940), who developed a mathematical model to explain the results of a statistical study of fish populations in the Adriatic Sea. In particular, his model explains the increase in predator fish (and corresponding decrease in prey fish) which he observed during the World War I period. Volterra produced a series of models for the interaction of two or more species. Alfred J. Lotka was an American biologist and actuary who independently produced many of the same models.

One of the simplest of their models takes the form:

$$\dot{x}_1 = ax_1 - bx_1x_2 \tag{1}$$
$$\dot{x}_2 = -dx_2 + cx_1x_2 \tag{2}$$

where $x_1 \geq 0$ denotes the sardine (prey) population and $x_2 \geq 0$ denotes the shark (predator) population. a, b, c, and d are all positive constants. Note that the equations model the facts that: sardines multiply faster as they increase in number; the number of sardines decreases as both the sardine and shark population increases; sharks increase in number at a rate proportional to the number of shark-sardine encounters.

(a) Determine all equilibria of this system. **(2 pts)**

Set $\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}$ and we get, $\begin{matrix} ax_1 = bx_1x_2 \\ dx_2 = cx_1x_2 \end{matrix}$, which leads to the following **two** equilibrium points:

$$\boxed{e_1 = \begin{bmatrix}0\\ 0\end{bmatrix}, \quad e_2 = \begin{bmatrix}d/c\\ a/b\end{bmatrix}}$$

(b) Linearize the system about each equilibrium that you found in part (a), and write the results in the form of a first order vector differential equation $\delta\dot{x} = A\delta x$. **(4 pts)**

Let $\begin{matrix} f_1(x_1, x_2) = \dot{x}_1 \\ f_2(x_1, x_2) = \dot{x}_2 \end{matrix}$ and take the partial derivatives: $\begin{matrix} \frac{\delta f_1}{\delta x_1} = a - bx_2 & \frac{\delta f_1}{\delta x_2} = -bx_1 \\ \frac{\delta f_2}{\delta x_1} = cx_2 & \frac{\delta f_2}{\delta x_2} = -d + cx_1 \end{matrix}$

Plugging in the values at e_1, we get: $\boxed{\delta\dot{x} = \begin{bmatrix} a & 0 \\ 0 & -d \end{bmatrix}\delta x}$

Plugging in the values at e_2, we get: $\boxed{\delta\dot{x} = \begin{bmatrix} 0 & -bd/c \\ ac/b & 0 \end{bmatrix}\delta x}$

(c) Program your model into MATLAB, choosing representative values of a, b, c, and d. Show, using simulation, how the shark and sardine populations evolve for the following three initial conditions: **(3 pts)**

- no sardines, a few sharks
- a few sardines, no sharks
- d/c sardines, a/b sharks

The **key to this problem** is rewriting the linearized equations **in terms of $\dot{x}$ and NOT $\delta\dot{x}$** (remember that $\delta x_1 = x_1 - x_{1,e}$ and $\delta x_2 = x_2 - x_{2,e}$):

For e_1, $\delta x_1 = x_1 - 0 = x_1$ and $\delta x_2 = x_2 - 0 = x_2$. So we get $\dot{x} = \begin{bmatrix} a & 0 \\ 0 & -d \end{bmatrix} x$.

For e_2, $\delta x_1 = x_1 - \frac{d}{c}$ and $\delta x_2 = x_2 - \frac{a}{b}$. So we get $\dot{x} = \begin{bmatrix} -\frac{bd}{c}\left(x_2 - \frac{a}{b}\right) \\ \frac{ac}{b}\left(x_1 - \frac{d}{c}\right) \end{bmatrix} = \begin{bmatrix} 0 & -bd/c \\ ac/b & 0 \end{bmatrix} x + \begin{bmatrix} ad/c \\ -ad/b \end{bmatrix}$.

There were 3 different ways to go about this in MATLAB. I used the arbitrary values a = 1.5, b = 0.5, c = 0.4, d = 0.8. Different values of these numbers will simply change the rate of decay/growth of the populations as well as the initial conditions for the 3rd situation. For the first two sets of initial conditions, I used 5 to represent "a few."

1) Simulink

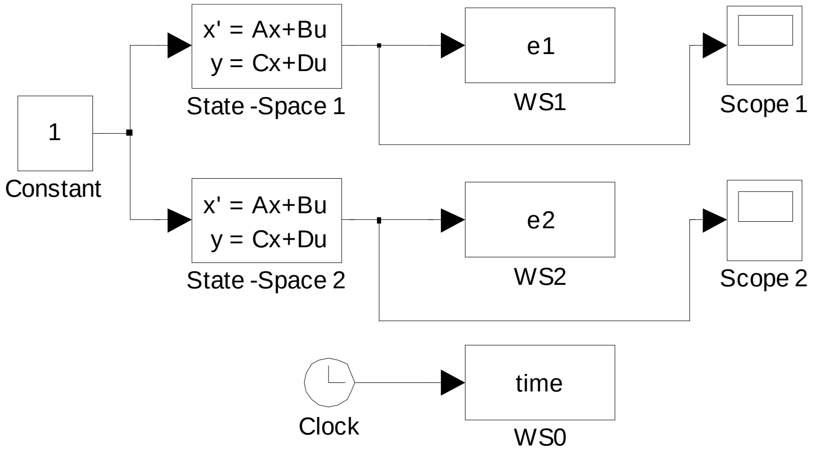

Figure 1: Simulink block diagram of Lotka-Volterra predator-prey equations

The top blocks are for equilibrium 1 (0,0) and the bottom blocks are for equilibrium 2 (d/c, a/b). The state-space A matrices are constructed as shown above. The input to State-Space1 is irrelevant because the B and D matrices are zeroed out, so I hooked it up to the input of State-Space2. The input to State-Space2 DOES matter and is set to 1 so that the B matrix is set to the stand-alone matrix shown above in the equation for $\dot{x}$ for e_2. In both cases, the C matrix was chosen to be $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ so that the TWO output variables are just the state variables. The outputs are then scoped for quick viewing and sent to the workspace for better plotting.

The initial conditions need to be changed in BOTH state-space blocks in-between simulations.

2) lsim

The following code is set up essentially the same as the Simulink diagram described above. Everything is written out, so the matrices for the state-space blocks above will match the matrices that are the inputs to the ss() function calls. This time, however, we need to manually define the time and input vectors. Again, the input to sys2 needs to be 1 for all time, and for convenience, we will feed the same input signal into sys1, even though it is irrelevant in that case.

To save space, I have excluded all code related to labeling the graphs (xlabel, ylabel, title, legend).

```
%% EE128 Problem Set 1, 1c (lsim) – hw1_1c_lsim.m
%% Justin Hsia - Fall 2008
%%  - Creates the linearized state-space models around e1 and e2, then runs
%%  - all 3 sets of ICs on both systems using lsim and plots on figures
a = 1.5; b = 0.5; c = 0.4; d = 0.8;

sys1 = ss([a 0; 0 -d],[0; 0],[1 0; 0 1],[0; 0]);                  %e1
sys2 = ss([0 -b*d/c; a*c/b 0],[a*d/c; -a*d/b],[1 0; 0 1],[0; 0]); %e2

t = 0:0.01:10;              %time steps
u = ones(1,length(t));      %input is 1 over all time

x0(:,1) = [0; 5];           % set the initial conditions
x0(:,2) = [5; 0];
x0(:,3) = [d/c; a/b];

for i=1:3                   % loop across ICs
    y1 = lsim(sys1,u,t,x0(:,i));
    y2 = lsim(sys2,u,t,x0(:,i));

    figure(2*i-1);          % plot on separate figures
    plot(t,y1);
    figure(2*i);
    plot(t,y2);
end
```

3) ode23

To use ode23, you need to define separate functions to represent both systems. As you can see, they are the same linearized equations as seen in the other two methods:

```
%% EE128 Problem Set 1, 1c (ode23) – eq1.m
function [xdot] = eq1(t,x)
    global a b c d
    xdot = [a 0; 0 -d]*x;
end
```

```
%% EE128 Problem Set 1, 1c (ode23) – eq2.m
function [xdot] = eq2(t,x)
    global a b c d
    xdot = [0 -b*d/c; a*c/b 0]*x + [a*d/c; -a*d/b];
end
```

And now we call on these functions using ode23 in a loop. The only major difference is that the initial conditions are now row vectors instead of column vectors.

Again, all labeling code has been excluded.

```
%% EE128 Problem Set 1, 1c (ode23) – hw1_1c_ode23.m
%% Justin Hsia - Fall 2008
%%  - Uses separate function (defined elsewhere) to represent e1 and e2,
%%  - then runs all 3 sets of ICs using ode23 and plots on figures
global a b c d
a = 1.5; b = 0.5; c = 0.4; d = 0.8;

x0 = [0 5; 5 0; d/c a/b];     % set the initial conditions

for i=1:3                     % loop across ICs
    [t1 x1] = ode23('eq1',[0 10],x0(i,:));
    [t2 x2] = ode23('eq2',[0 10],x0(i,:));

    figure(2*i-1)
    plot(t1,x1);
    figure(2*i)
    plot(t2,x2);
end
```

The graphs are on the following page. But what do we expect to see?

- No sardines, a few sharks

Life cannot spring up out of nowhere. The sardine population should remain at zero and the sharks, deprived of their food source, should die out.

- A few sardines, no sharks

The shark population should remain at zero and the sardine population, left unchecked by predators, should grow out of control (exponentially).

- d/c sardines, a/b sharks

This initial condition is defined to match one of the equilibrium points you solved for part a. Since it is an equilibrium point, you expect the populations to remain the same.

So why the difference in simulations?

The important point to take away here is that **linearization only works around the chosen operating (equilibrium) point**. In the first two initial conditions, we expect at least one of the populations to be at zero, which matches up better with the first equilibrium point. Looking at the linearized matrices A_1 and A_2, the sardine and shark populations change based on their OWN populations in A_1, while they change based on the OTHER population in A_2. This is why a zero population stays at zero using equilibrium point 1 and why it doesn't stay at zero using equilibrium point 2. You end up getting negative populations using equilibrium point 2, which doesn't make physical sense.

In the third initial condition, we are at the other equilibrium point, so our second linearization works perfectly. Using the first linearization, the shark population always dies out (-d) and the sardine population, if not at zero, will always grow uncontrollably (a).

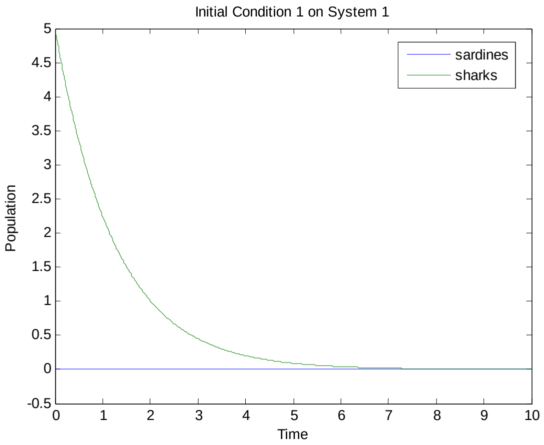

Figure 2: 0 sardines, 5 sharks using equilibrium 1

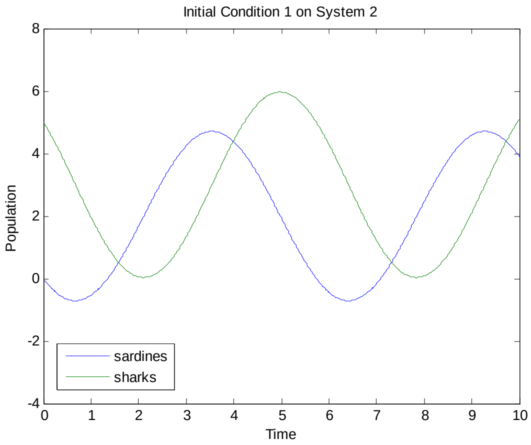

Figure 3: 0 sardines, 5 sharks using equilibrium 2

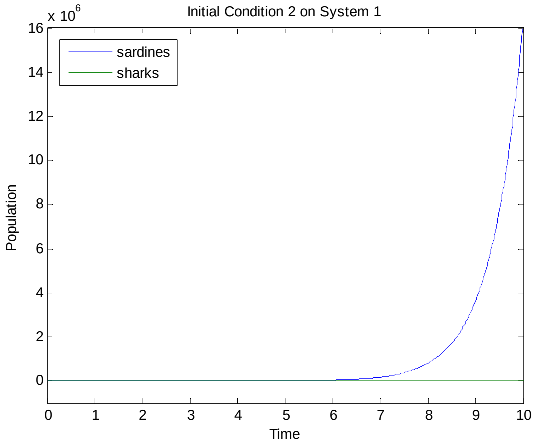

Figure 4: 5 sardines, 0 sharks using equilibrium 1

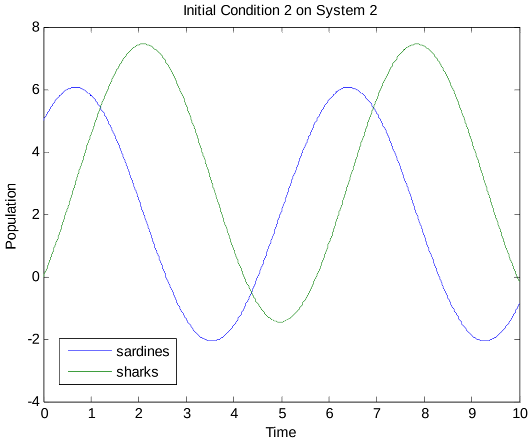

Figure 5: 5 sardines, 0 sharks using equilibrium 2

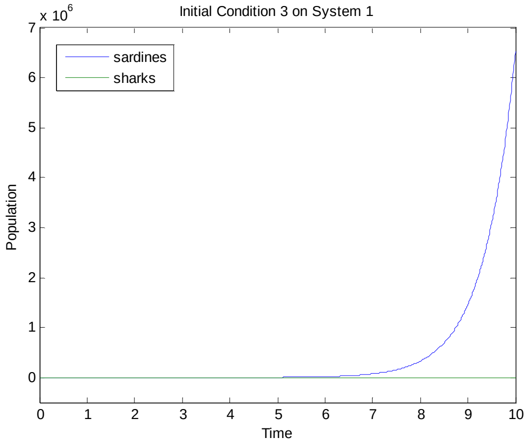

Figure 6: 2 sardines, 3 sharks using equilibrium 1

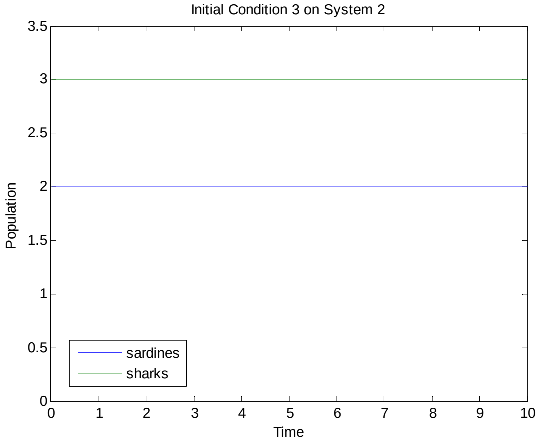

Figure 7: 2 sardines, 3 sharks using equilibrium 2

Problem 2: A magnetically suspended steel ball, linearized.

The simplified dynamics of a magnetically suspended steel ball are given by:

$$m\ddot{y} = mg - c\frac{u^2}{y^2} \tag{3}$$

where the input u represents the current supplied to the electromagnet, y is the vertical position of the ball, which may be measured by a position sensor, g is gravitational acceleration, m is the mass of the ball, and c is a positive constant such that the force on the ball due to the electromagnet is $c\frac{x^2}{y^2}$. Assume a normalization such that $m = g = c = 1$.

(a) Using the states $x_1 = y$ and $x_2 = \dot{y}$ write down a nonlinear state space description of this system. **(2 pts)**

We have $x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$ and $\dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix}$, so $\dot{x}_1 = x_2$ and $\dot{x}_2 = g - \frac{c}{m}\frac{u^2}{{x_1}^2}$:

$$\boxed{\dot{x} = \begin{bmatrix} x_2 \\ g - \dfrac{c}{m}\dfrac{u^2}{{x_1}^2} \end{bmatrix}}$$

(b) What *equilibrium control input* u_e must be applied to suspend the ball at y = 1 m? **(2pts)**

Want the ball at y = 1m, so $x_{1,e}$ = 1. Now solve for $\dot{x}_e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

We get $x_{2,e} = 0$ and $0 = g - \frac{c}{m}\frac{{u_e}^2}{1^2}$, so $\boxed{u_e = \sqrt{\dfrac{mg}{c}} = 1}$.

<u>Side-note</u>: Why do we ignore $u_e = -1$?

Theoretically, $\sqrt{1} = \pm 1$. But think about the physical system. We have a steel ball suspended in the air by a magnetic field. The magnetic field is generated by a current running through an electromagnet. What happens when you run the current in the opposite direction (negative I)? The magnetic field reverses direction as well. So physically, a current of -1 would do the opposite of levitating the ball; it would push it towards the ground.

On further thought, the logic above applies if the ball were a charged particle. It's entirely possible that the ball would self-polarize and be attracted to the electro magnet under either magnetic field direction. In general, we will try to set problems up in a way so that we are interested in positive values. If it needs to go negative, you probably should just redefine the direction.

(c) Write the linearized state space equations for state and input variables representing perturbations away from the equilibrium of part (b). **(4 pts)**

Let $f(x_1, x_2, u) = \dot{x}_2 = g - \frac{c}{m}\frac{u^2}{{x_1}^2}$. Take the partial derivatives:

$$\left.\frac{\delta f}{\delta x_1}\right|_{x_{1,e},x_{2,e},u_e} = 2\frac{c}{m}\frac{u^2}{{x_1}^3}\bigg|_{x_{1,e},x_{2,e},u_e} = \frac{2c}{m}; \qquad \left.\frac{\delta f}{\delta x_2}\right|_{x_{1,e},x_{2,e},u_e} = 0; \qquad \frac{\delta f}{\delta u} = -2\frac{c}{m}\frac{u}{{x_1}^2}\bigg|_{x_{1,e},x_{2,e},u_e} = -\frac{2c}{m}$$

$$\delta\dot{x} = \begin{bmatrix} \delta\dot{x}_1 \\ \delta\dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \delta f/\delta x_1 & \delta f/\delta x_2 \end{bmatrix}\begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta f/\delta u \end{bmatrix}\delta u$$

$$\boxed{\delta\dot{x} = \begin{bmatrix} 0 & 1 \\ \dfrac{2c}{m} & 0 \end{bmatrix}\delta x + \begin{bmatrix} 0 \\ -\dfrac{2c}{m} \end{bmatrix}\delta u = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}\delta x + \begin{bmatrix} 0 \\ -2 \end{bmatrix}\delta u}$$

Problem 3: Linearity.

(a) For each of the following systems H, indicate whether or not the system is (i) Linear, (ii) Time-Invariant. Justify your answers. **(2pts each)**

Linearity: Show that $H(\alpha u_1(t) + \beta u_2(t)) = \alpha H(u_1(t)) + \beta H(u_2(t))$.

Time-invariance: Show that $H(u(t-\tau)) = y(t-\tau)$
(output of delayed input is same as delaying regular output)

i) $H(u(t)) = e^{u(t)}$

$$H(\alpha u_1 + \beta u_2) = e^{\alpha u_1} e^{\beta u_2} \neq \alpha H(u_1) + \beta H(u_2)$$

$$H(u(t-\tau)) = e^{u(t-\tau)} = y(t-\tau)$$

Nonlinear

Time Invar.

ii) $H(u(t)) = au(t) + b$

$$H(\alpha u_1 + \beta u_2) = a\alpha u_1 + a\beta u_2 + b \neq \alpha H(u_1) + \beta H(u_2)$$

$$H(u(t-\tau)) = au(t-\tau) + b = y(t-\tau)$$

Nonlinear

Time Invar.

iii) $H(u(t)) = au_1(t) + bu_2(t), u(t) = [u_1(t) \quad u_2(t)]^T$

$$H(\alpha u_1 + \beta u_2) = a(\alpha u_{11} + \beta u_{21}) + b(\alpha u_{12} + \beta u_{22})$$

$$= \alpha(au_{11} + bu_{12}) + \beta(au_{21} + bu_{22}) = \alpha H(u_1) + \beta H(u_2)$$

$$H(u(t-\tau)) = au_1(t-\tau) + bu_2(t-\tau) = y(t-\tau)$$

Linear

Time Invar.

iv) $H(u(t)) = u(-t)$

$$H(\alpha u_1 + \beta u_2) = \alpha u_1(-t) + \beta u_2(-t) = \alpha H(u_1) + \beta H(u_2)$$

$$H(u(t-\tau)) = u(-t-\tau) \neq y(t-\tau) = u(-(t-\tau))$$

Linear

Time Var.

This one is a little confusing for time invariance. Think about it visually: if you delay a signal by τ and then reverse the signal in time, it's the same as reversing the signal in time and then ADVANCING the signal by τ.

v) $H(u(t)) = \int_0^t e^{-\sigma} u(t-\sigma) d\sigma$

$$H(\alpha u_1 + \beta u_2) = \int_0^t e^{-\sigma}(\alpha u_1(t-\sigma) + \beta u_2(t-\sigma)) d\sigma = \alpha \int_0^t e^{-\sigma} u_1(t-\sigma) d\sigma +$$
$$\beta \int_0^t e^{-\sigma} u_2(t-\sigma) d\sigma = \alpha H(u_1) + \beta H(u_2)$$

Linear

$$H(u(t-\tau)) = \int_0^t e^{-\sigma} u(t-\tau-\sigma) d\sigma \neq y(t-\tau) = \int_0^{t-\tau} e^{-\sigma} u(t-\tau-\sigma) d\sigma$$

Time Var.

(b) Consider an amplifier having input voltage $v_i(t)$ and output voltage $v_o(t)$ related by $v_o(t) = 8\left(v_i(t)\right)^2$. Show that the amplifier is not linear. Derive a linearized model around operating point (v_{iQ}, v_{oQ}). **(3 pts)**

Let $H(v_i(t)) = v_o(t)$, and check linearity:

$$H(\alpha v_{i1} + \beta v_{i2}) = 8(\alpha v_{i1} + \beta v_{i2})^2 \neq \alpha H(v_{i1}) + \beta H(v_{i2}) = \alpha * 8(v_{i1})^2 + \beta * 8(v_{i2})^2$$

Nonlinear

Linearize equation about (v_{iQ}, v_{oQ}):

$$\delta v_o = \left.\frac{\delta H}{\delta v_i}\right|_{v_{iQ}} \delta v_i \, ; \qquad \frac{\delta H}{\delta v_i} = 16 v_i$$

$$\delta v_o = 16 v_{iQ} \delta v_i \qquad \boxed{(v_o(t) - v_{oQ}) = 16 v_{iQ}(v_i(t) - v_{iQ})}$$

Problem 4: Transfer function models. Derive the transfer functions $\frac{V_o(s)}{V_i(s)}$ for each of the RC networks (i) and (ii) below. Assume that $R_1 > 0$, $R_2 > 0$ denote resistor values, and $C > 0$ denotes capacitor values. **(4 pts each)**

These problems can be approached in two ways: 1) using the normal electrical system modeling equations and then the Laplace transform, or 2) taking the Laplace transform first (impedance) and then analyzing the system. I will show both methods here.

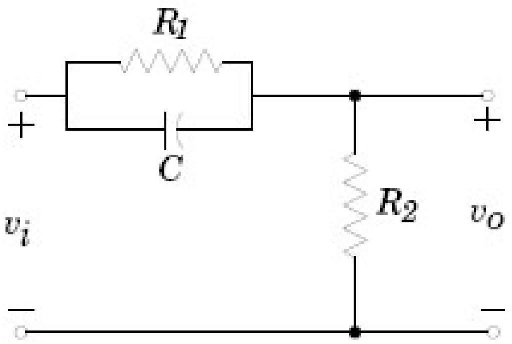

Figure 8: RC network (i)

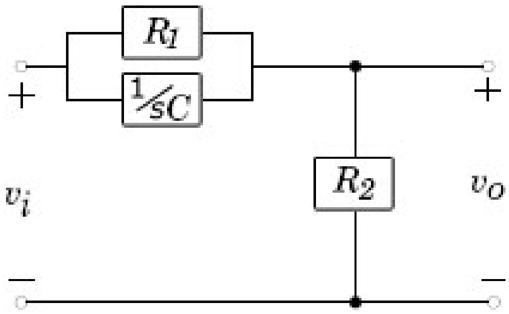

Figure 9: RC network (i) - impedance

Modeling Equations:

Define current through R_1 to be i_1, current through C to be i_2, and current through R_2 to be i_3.

We have: $i_3 = i_1 + i_2, \quad i_3 = \frac{v_o}{R_2}, \quad i_1 = \frac{v_i - v_o}{R_1}, \quad i_2 = C\frac{d}{dt}(v_i - v_o).$

Substituting:

$$\frac{v_o}{R_2} = \frac{v_i - v_o}{R_1} + C\dot{v}_i - C\dot{v}_o$$

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right) v_o + C\dot{v}_o = \frac{1}{R_1} v_i + C\dot{v}_i$$

Laplace:

$$\left(\frac{1}{R_1} + \frac{1}{R_2} + sC\right) V_o(s) = \left(\frac{1}{R_1} + sC\right) V_i(s)$$

$$\frac{V_o(s)}{V_i(s)} = \frac{\left(\frac{1}{R_1} + sC\right)}{\left(\frac{1}{R_1} + \frac{1}{R_2} + sC\right)} * \frac{R_1R_2}{R_1R_2} = \boxed{\frac{V_o(s)}{V_i(s)} = \frac{R_2(1 + sR_1C)}{(R_1 + R_2) + sR_1R_2C}}$$

Impedance:

Define parallel impedance of R_1 and C to be Z.

Solve for Z:

$$\frac{1}{Z} = \frac{1}{R_1} + \frac{1}{\frac{1}{sC}} = \frac{1}{R_1} + sC$$

$$Z = \frac{1}{\frac{1}{R_1} + sC} = \frac{R_1}{1 + sR_1C}$$

Treat as voltage divider:

$$\frac{V_o(s)}{V_i(s)} = \frac{R_2}{R_2 + Z} = \frac{R_2}{R_2 + \frac{R_1}{1 + sR_1C}} * \frac{1 + sR_1C}{1 + sR_1C} = \boxed{\frac{V_o(s)}{V_i(s)} = \frac{R_2(1 + sR_1C)}{(R_1 + R_2) + sR_1R_2C}}$$

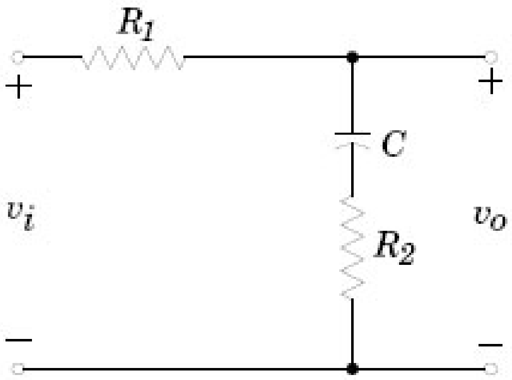

Figure 10: RC network (ii)

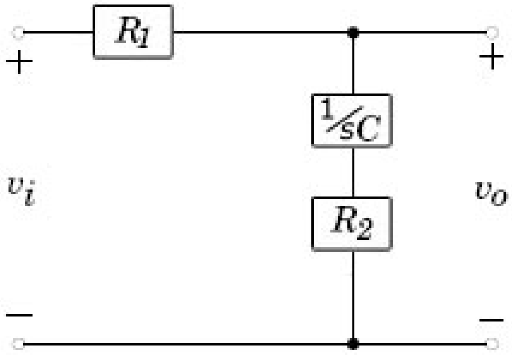

Figure 11: RC network (ii) - impedance

Modeling Equations:

Single current i running through circuit. Define intermediate voltage $\tilde{v}$ to be voltage between C and R_2.

We have: $i = \frac{\tilde{v}-0}{R_2} = \frac{\tilde{v}}{R_2}, \;\; i = \frac{v_i - v_o}{R_1}, \;\; i = C\frac{d}{dt}(v_o - \tilde{v})$

Solve for $\tilde{v}$: $\tilde{v} = iR_2 = \frac{R_2}{R_1}(v_i - v_o)$

Substitute $\tilde{v}$ and i in C equation:

$$\frac{v_i - v_o}{R_1} = C\frac{d}{dt}\left(v_o - \frac{R_2}{R_1}(v_i - v_o)\right) = C\left(\left(1 + \frac{R_2}{R_1}\right)\dot{v}_o - \frac{R_2}{R_1}\dot{v}_i\right)$$

$$\frac{1}{R_1}v_i + C\frac{R_2}{R_1}\dot{v}_i = \frac{1}{R_1}v_o + C\left(1 + \frac{R_2}{R_1}\right)\dot{v}_o$$

Laplace:

$$\left(\frac{1 + sR_2C}{R_1}\right)V_i(s) = \left(\frac{1}{R_1} + sC\left(1 + \frac{R_2}{R_1}\right)\right)V_o(s)$$

$$(1 + sR_2C)V_i(s) = \big(1 + sC(R_1 + R_2)\big)V_o(s)$$

$$\boxed{\frac{V_o(s)}{V_i(s)} = \frac{1 + sR_2C}{1 + sC(R_1 + R_2)}}$$

Impedance:

Define series impedance of C and R_2 to be Z.

Solve for Z: $Z = \frac{1}{sC} + R_2$

Treat as voltage divider:

$$\frac{V_o(s)}{V_i(s)} = \frac{Z}{R_1 + Z} = \frac{\frac{1}{sC} + R_2}{R_1 + \frac{1}{sC} + R_2} * \frac{sC}{sC} = \boxed{\frac{V_o(s)}{V_i(s)} = \frac{1 + sR_2C}{1 + sC(R_1 + R_2)}}$$

Problem 5. Given $Y(s) = \frac{a}{s^2 + a^2}$, what is $\lim_{t\to\infty} y(t)$? **(1 pt)**

You can use the Final Value Theorem (FVT) ($\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s)$) to obtain $\lim_{t\to\infty} y(t) = 0$. The fine print here, however, is that **the FVT applies only if $\lim_{t\to\infty} y(t)$ exists**.

Looking at a table of Laplace Transforms, we can see that $y(t) = \sin(at)$.

$\boxed{\lim_{t\to\infty} y(t) \text{ does not exist}}$