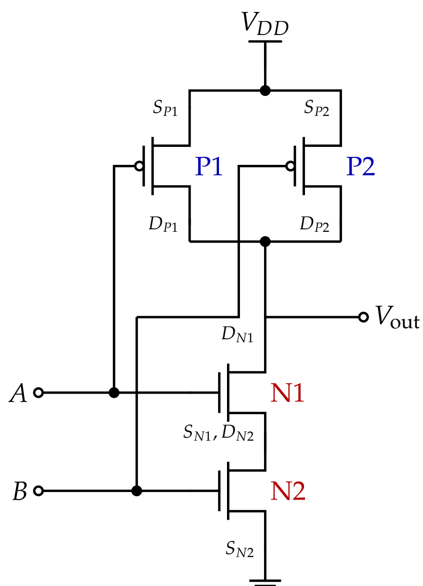
# Homework 8

**This optional homework is "due" on Thursday, August 11th at 5:59 pm PT. There will be no submission for this homework as it will be strictly practice for your final.**

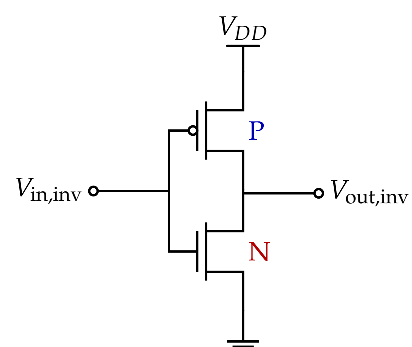
1. **Transistor Switch Model (Spring 2021 Midterm)**

   In this problem, we will analyze the behavior of a NAND gate driving an inverter. Figure 1a shows the transistor model of a NAND gate and Figure 1b shows the transistor model of an inverter.

   In this question assume that $V_{DD}$ is greater than both the NMOS threshold $V_{\text{th,n}}$ and PMOS threshold $|V_{\text{th,p}}|$.




**(a)** NAND schematic with transistors.

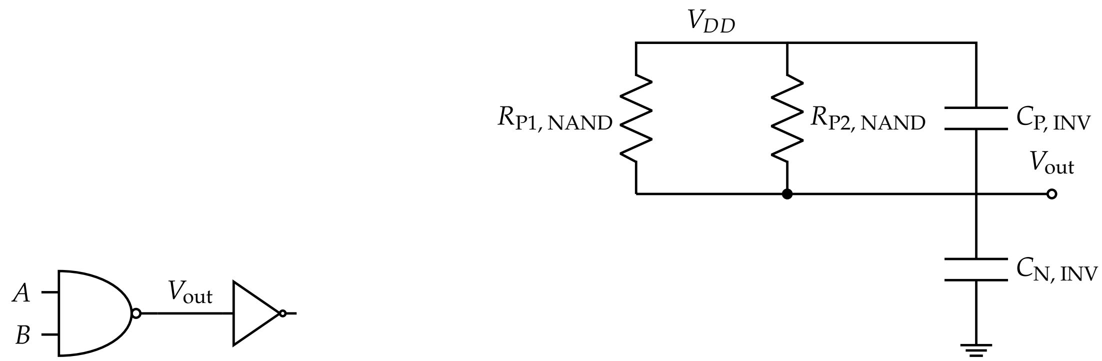



**(b)** Inverter schematic with transistors.

**Figure 1:** Transistor schematics

   (a) A diagram of a NAND gate driving an inverter is shown in Figure 2a. Consider the case where $A = V_{DD}$ and $B = V_{DD}$ for a long time before $t = 0$. Then at $t = 0$, we switch $A$ and $B$ to 0 V. The equivalent simplified circuit after this transition is shown in Figure 2b. **Find $V_{\text{out}}$ at time $t = 0$.**

   **Solution:** $V_{\text{out}}(0) = 0$. Since $A = B = V_{DD}$ for $t < 0$, both NMOS transistors have been switched on due to $V_{GSn} = V_{DD} > V_{\text{th,n}}$ until right before $t = 0$. Similarly, both PMOS transistors are switched off because $|V_{GSp}| = V_{DD} > |V_{\text{th,p}}|$. Therefore, $C_{\text{N, INV}}$ is fully discharged to 0V and $C_{\text{P, INV}}$ is fully charged to $V_{DD}$ right before $t = 0$. When the transition happens the charge on capacitors cannot jump instantaneously. So, at $t = 0$ the voltage at output will remain 0V.




**(a)** NAND driving an Inverter

**(b)** A Simplified Equivalent Circuit for when $A = B = 0\text{V}$

**Figure 2:** Schematic and model of a NAND gate driving an inverter

(b) **Write the differential equation for solving $V_{\text{out}}(t)$ for $t \geq 0$ in the circuit shown in Figure 2b.** Specifically, **find coefficients $\lambda$ and $b$ in the following symbolic differential equation:**

$$\frac{dV_{\text{out}}(t)}{dt} = \lambda V_{\text{out}}(t) + bV_{DD}, \tag{1}$$

as a function of $R_P, C_{\text{P, INV}}, C_{\text{N, INV}}$, and $V_{DD}$. Assume that $R_{\text{P1,NAND}} = R_{\text{P2, NAND}} = R_P$.

**Solution:** The first thing to note is that the two resistors are in parallel, so we can lump them into one resistor with half the value. We can then start by writing a KCL at the output node:

$$C_{\text{N, INV}} \frac{dV_{\text{out}}(t)}{dt} + C_{\text{P, INV}} \frac{d(V_{\text{out}}(t) - V_{DD})}{dt} + \frac{2}{R_P}(V_{\text{out}}(t) - V_{DD}) = 0,$$

$$\frac{dV_{\text{out}}(t)}{dt} = -\frac{2}{R_P(C_{\text{P, INV}} + C_{\text{N, INV}})} V_{\text{out}}(t) + \frac{2}{R_P(C_{\text{P, INV}} + C_{\text{N, INV}})} V_{DD}.$$

Therefore, $\lambda = -\frac{2}{R_P(C_{\text{P, INV}}+C_{\text{N, INV}})}$ and $b = \frac{2}{R_P(C_{\text{P, INV}}+C_{\text{N, INV}})}$.

(c) **Solve $V_{\text{out}}(t)$ in the differential equation eq. (1) and the initial condition $V_{\text{out}}(0)$.** You should leave your answer in terms of $\lambda, b, V_{DD}$, and $V_{\text{out}}(0)$.

**Solution:**

Approach 1: We know from homework that for a differential equation of type $\frac{dv(t)}{dt} = \lambda v(t) + bu(t)$ and initial condition $v(t_0)$, the answer is $v(t) = v(t_0)e^{\lambda t} + b\int_{t_0}^{t} u(\tau)e^{\lambda(t-\tau)}d\tau$. We will apply the same principle here as well:

$$V_{\text{out}}(t) = V_{\text{out}}(0)e^{\lambda t} + b\int_0^t V_{DD}e^{\lambda(t-\tau)}d\tau \tag{2}$$

$$= V_{\text{out}}(0)e^{\lambda t} + bV_{DD}e^{\lambda t}\int_0^t e^{-\lambda\tau}d\tau \tag{3}$$

$$= V_{\text{out}}(0)e^{\lambda t} + bV_{DD}e^{\lambda t} \cdot \frac{e^{-\lambda t} - 1}{-\lambda} \tag{4}$$

$$= V_{\text{out}}(0)e^{\lambda t} + \frac{b}{\lambda}V_{DD}\left(e^{\lambda t} - 1\right). \tag{5}$$

Approach 2: We can use variable substitution to arrive at a homogeneous differential equation and then solve it. Define $\widetilde{V}_{\text{out}}(t) = V_{\text{out}}(t) + \frac{b}{\lambda}V_{DD}$.

$$\frac{d(V_{\text{out}}(t) + \frac{b}{\lambda}V_{DD})}{dt} = \lambda\left(V_{\text{out}}(t) + \frac{b}{\lambda}V_{DD}\right), \tag{6}$$

$$\frac{d\widetilde{V}_{\text{out}}(t)}{dt} = \lambda \widetilde{V}_{\text{out}}(t), \tag{7}$$
$$\widetilde{V}_{\text{out}}(t) = \widetilde{V}_{\text{out}}(0)e^{\lambda t}. \tag{8}$$

We also know that $\widetilde{V}_{\text{out}}(0) = V_{\text{out}}(0) + \frac{b}{\lambda}V_{DD}$. Substituting $V_{\text{out}}$ back will give us the answer.

$$V_{\text{out}}(t) + \frac{b}{\lambda}V_{DD} = \left(V_{\text{out}}(0) + \frac{b}{\lambda}V_{DD}\right)e^{\lambda t} \tag{9}$$
$$= V_{\text{out}}(0)e^{\lambda t} + \frac{b}{\lambda}V_{DD}\left(e^{\lambda t} - 1\right). \tag{10}$$

(d) Now consider the case where $A = 0$V and $B = 0$V for a long time before $t = 0$ in Figure 2a. At $t = 0$ we switch $A$ and $B$ to $V_{DD}$. **Write down the state (ON/OFF) of transistors P1, P2, N1, and N2 in the NAND gate. Draw the equivalent simplified circuit for this transition that will help us with writing the differential equation of $V_{\text{out}}(t)$.**

*Hint: You may find the NAND resistor-switch model in Figure 3 helpful. Don't forget to include the inverter's capacitors, $C_{N,\,INV}$ and $C_{P,\,INV}$,* which are loading the NAND gate.

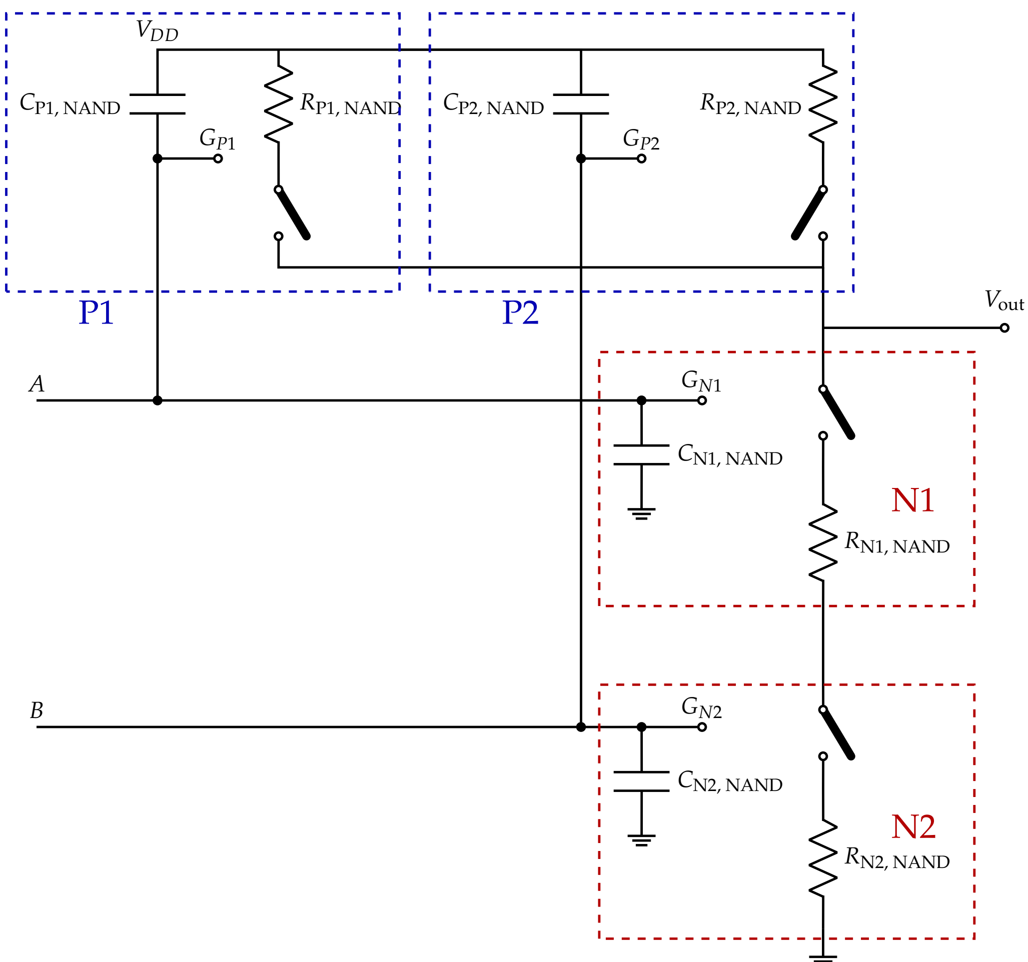



**Figure 3:** NAND Model: Capacitances

**Solution:**

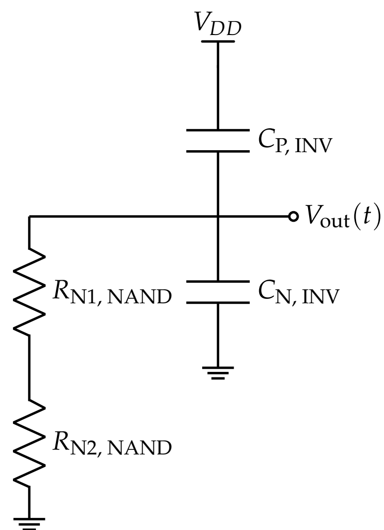
Both of our inputs are high ($V_{\text{DD}}$). Thus $|V_{\text{GSp}}| = 0V \implies$ both PMOS transistors P1 and P2 are OFF. At the same time $V_{\text{GSn}} = V_{\text{DD}} \implies$ both NMOS transistors N1 and N2 are ON.

Since the PMOS transistors are both off, their switches in the equivalent circuit are open. As a result their resistors $R_{\text{P1, NAND}}$ and $R_{\text{P2, NAND}}$ are floating and don't need to be included.

The NMOS transistors, on the other hand, are both on, so in the simplified equivalent circuit, their resistors $R_{\text{N1, NAND}}$ and $R_{\text{N2, NAND}}$ connect the output to ground.

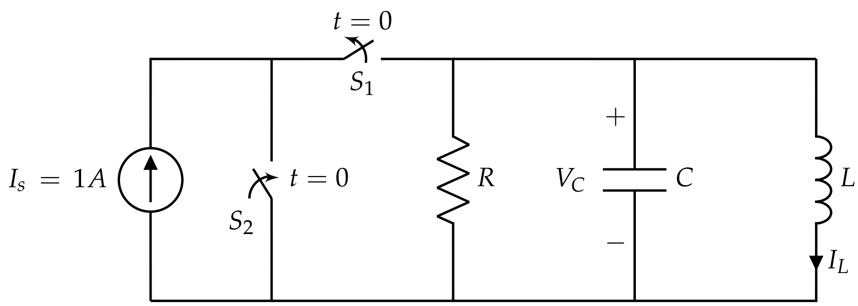
In addition, $V_{\text{out}}$ drives an inverter, which means that $V_{\text{out}}$ is connected to two gate capacitances: $C_{\text{P, INV}}$ to $V_{\text{DD}}$ and $C_{\text{N, INV}}$ to ground.

Thus we end up with the simplified circuit below.



**2. Parallel RLC with Current Source (Fall 2021 Midterm)**

Consider the following circuit:




(a) At $t = 0$, switch $S_1$ became open and switch $S_2$ became closed. We first need to construct our state space system for $t \geq 0$. Our natural state variables are the current through the inductor $x_1(t) = I_L(t)$ and the voltage across the capacitor $x_2(t) = V_C(t)$ since these are the quantities whose derivatives show up in the system of equations governing our circuit.

**Find the system of differential equations in terms of our state variables that describes this circuit for $t \geq 0$. Leave the system symbolic in terms of $I_s, R, L,$ and $C$. Write the system of differential equations in vector/matrix form with the vector state variable:**

$$\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} \tag{11}$$

This should be in the form $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t)$ with a $2 \times 2$ matrix $A$.
*Show your work.*

**Solution:**
Let's first find the relationship between $\frac{dI_L(t)}{dt}$ and $V_C(t)$:

$$V_C(t) = V_L(t) = L\frac{dI_L(t)}{dt} \rightarrow \frac{dI_L(t)}{dt} = \frac{1}{L}V_C(t) \tag{12}$$

Using KCL, we can also find the relationship between $\frac{dV_C(t)}{dt}$ and $I_L(t)$:

$$I_C(t) + I_L(t) + I_R(t) = 0, \quad I_C(t) = C\frac{dV_C(t)}{dt}, \quad I_R = \frac{V_C(t)}{R} \tag{13}$$

$$C\frac{dV_C(t)}{dt} + I_L(t) + \frac{V_C(t)}{R} = 0 \rightarrow \frac{dV_C(t)}{dt} = -\frac{1}{C}I_L(t) - \frac{1}{RC}V_C(t) \tag{14}$$

Note that $I_C(t) + I_L(t) + I_R(t) = 0$ because the current source ($I_s$) is disconnected from the RLC network at $t = 0$.
In summary, we have the following differential equations:

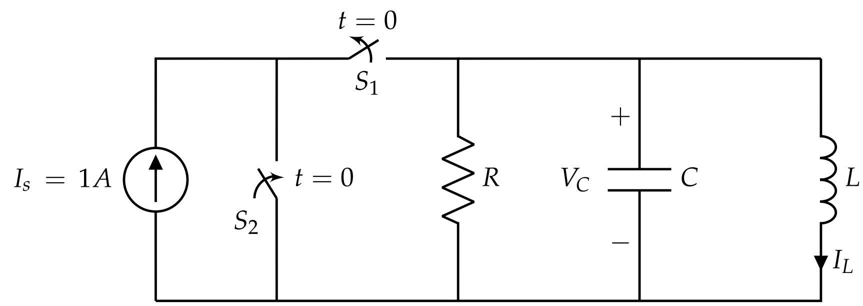
$$\frac{dI_L(t)}{dt} = \frac{1}{L}V_C(t) \tag{15}$$

$$\frac{dV_C(t)}{dt} = -\frac{1}{C}I_L(t) - \frac{1}{RC}V_C(t). \tag{16}$$

Finally, we can represent the above differential equations in the matrix/vector multiplication form as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix}}_{A} \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix}. \tag{17}$$

(b)




For a long time in the past $t < 0$, assume that the switch $S_1$ had remained closed and $S_2$ had remained open. **What is $I_L(0)$ and $V_C(0)$? Give a brief justification as well.**

**Solution:**

If something has been happening for a long time, we expect to see that things are no longer changing. For the current in the inductor to no longer be changing, the voltage across it must be zero. Since the current is connected in parallel with the capacitor, it must be that $V_C(0) = 0V$. With a zero voltage across it, the resistor must have zero current flowing through it. For the voltage across the capacitor to not be changing, there must be zero current flowing through it as well. To satisfy, KCL, this means all the current from the current source must be flowing through the inductor, setting $I_L(0) = 1A$.

The above is a fuller solution than we expected. A shorter one could simply be: if a system has been left as it is for a long time, it should be in steady state. In steady-state, an inductor is a short and a capacitor is an open circuit. This means that all the current goes through the short, making $I_L(0) = 1A$ and since it is a short, the voltage across it and hence the capacitor is $0V$.

Writing the above in math, we'd get: The given RLC network reaches steady state ($\frac{dV_C(t)}{dt} = 0$ and $\frac{dI_L(t)}{dt} = 0$) after we wait for long enough. This can be interpreted as $I_C = 0$ and $V_L = 0$ since $I_C = C\frac{dV_C(t)}{dt}$ and $V_L = L\frac{dI_L(t)}{dt}$. Therefore the capacitor and the inductor behave as an open and a short circuit, respectively. Thus, all of the input current flows through the inductor ($I_L = I_s = 1A$).

A significantly longer answer would be to realize that the nonhomogeneous differential equation that corresponds to the configuration for $t < 0$ is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix}}_{A} \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{C} \end{bmatrix} I_s. \tag{18}$$

This is stable because $A's$ eigenvalues all have real part less than 0 as long as $R > 0$ (as confirmed in part (c)) and hence all transients will have long since decayed away. The only (because $A$ is invertible) steady-state solution (with the derivatives being zero since nothing is changing in steady-state) this has for $I_s = 1A$ is $I_L = 1A$ and $V_C = 0V$.

(c) For the rest of this problem, assume $R = 1\,\text{M}\Omega, L = 25\,\mu\text{H}, C = 10\,\text{nF}$. With these values, we get the following eigenvalues and eigenvectors for $A$ in the differential equation $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t)$.

$$\lambda_1 = -Z_0 + \mathrm{j}\omega_0 \qquad \lambda_2 = -Z_0 - \mathrm{j}\omega_0, \tag{19}$$

$$\text{where } Z_0 = \sqrt{\frac{L}{C}} = 50 \text{ and } \omega_0 = \frac{1}{\sqrt{LC}} = 2\times 10^6$$

$$V = \begin{bmatrix} \vec{v}_{\lambda_1} & \vec{v}_{\lambda_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ a & \overline{a} \end{bmatrix} \tag{20}$$

$$V^{-1} = \begin{bmatrix} b & -\mathrm{j}0.01 \\ \overline{b} & \mathrm{j}0.01 \end{bmatrix} \tag{21}$$

Consider a nice coordinate system for which we can write $\vec{x}(t) = V\vec{\widetilde{x}}(t)$.

**What is the $\widetilde{A}$ so that $\frac{\mathrm{d}}{\mathrm{d}t}\vec{\widetilde{x}}(t) = \widetilde{A}\vec{\widetilde{x}}(t)$? You can leave the answer symbolic in terms of $\lambda_1, \lambda_2$.** Note that you **do not** need to find/evaluate $a, b$. You can still answer all the questions below without knowing these values.

**Solution:**

A calculation is not required as the effect of multiplying $A$ appropriately by $V$ and $V^{-1}$ yields a matrix with the eigenvalues on its diagonal, in the order corresponding to the eigenvectors in the $V$ matrix.

$$\widetilde{A} = V^{-1}AV = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} -Z_0 + \mathrm{j}\omega_0 & 0 \\ 0 & -Z_0 - \mathrm{j}\omega_0 \end{bmatrix} \tag{22}$$

However, a quick way to verify is the following calculation:

$$\widetilde{A} = V^{-1}AV \tag{23}$$

$$= V^{-1}\begin{bmatrix} | & & | \\ A\vec{v}_{\lambda_1} & \cdots & A\vec{v}_{\lambda_n} \\ | & & | \end{bmatrix} \tag{24}$$

$$= V^{-1}\begin{bmatrix} | & & | \\ \lambda_1\vec{v}_{\lambda_1} & \cdots & \lambda_n\vec{v}_{\lambda_n} \\ | & & | \end{bmatrix} \tag{25}$$

$$= \begin{bmatrix} | & & | \\ \lambda_1 V^{-1}\vec{v}_{\lambda_1} & \cdots & \lambda_n V^{-1}\vec{v}_{\lambda_n} \\ | & & | \end{bmatrix} \tag{26}$$

$$= \begin{bmatrix} | & & | \\ \lambda_1\vec{e}_1 & \cdots & \lambda_n\vec{e}_n \\ | & & | \end{bmatrix} \tag{27}$$

$$= \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \tag{28}$$

Note that $\vec{e}_i$ is the $i$-th column of the identity matrix $I$ - this comes from setting the columns of $V^{-1}V = I$ equal to each other after distributing $V^{-1}$ on the columns of $V$.

Followings are for the verification of eigenvalues and eigenvectors:

$$\det(A - \lambda I) = \begin{vmatrix} -\lambda & \frac{1}{L} \\ \frac{-1}{C} & \frac{-1}{RC} - \lambda \end{vmatrix} = \lambda^2 + \frac{1}{RC}\lambda + \frac{1}{LC} = 0 \tag{29}$$

$$\lambda = -\frac{1}{2RC} \pm \frac{1}{2}\sqrt{\left(\frac{1}{RC}\right)^2 - \frac{4}{LC}} = -50 \pm \mathrm{j}\left(2 \times 10^6\right) \tag{30}$$

Assuming the eigenvectors have the following form: $\vec{v} = \begin{bmatrix} 1 \\ y \end{bmatrix}$, then we have:

$$A\vec{v} = \lambda\vec{v} = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} 1 \\ y \end{bmatrix} = \begin{bmatrix} \lambda_i \\ y \times \lambda_i \end{bmatrix} \tag{31}$$

From the above, we can easily find that $y = L\lambda_i$. Based on this, we can calculate the eigenvectors:

$$v_1 = \begin{bmatrix} 1 \\ L\lambda_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.00125 + \mathrm{j}50 \end{bmatrix}, \qquad v_2 = \begin{bmatrix} 1 \\ L\lambda_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.00125 - \mathrm{j}50 \end{bmatrix} \tag{32}$$

1in

(d) Now, suppose that our initial conditions for $I_L(0)$ and $V_C(0)$ have changed to the following:

$$\begin{bmatrix} I_L(0) \\ V_C(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{33}$$

Using the information from (c), **find $I_L(t)$ for $t \geq 0$ and write the answer in a form involving real exponentials and sinusoids. Does $I_L(t)$ converge as $t \to \infty$? If so, what does it converge to?**

**Show your work.**

**Solution:**

$$\vec{\tilde{x}}(0) = V^{-1}\vec{x}(0) = \begin{bmatrix} b & -\mathrm{j}0.01 \\ \overline{b} & \mathrm{j}0.01 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\mathrm{j}0.01 \\ \mathrm{j}0.01 \end{bmatrix} \tag{34}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} \tag{35}$$

$$\tilde{x}_1(t) = K_1 e^{\lambda_1 t} \quad \text{and} \quad \tilde{x}_1(0) = -\mathrm{j}0.01 \tag{36}$$

$$\tilde{x}_2(t) = K_2 e^{\lambda_2 t} \quad \text{and} \quad \tilde{x}_1(0) = \mathrm{j}0.01 \tag{37}$$

Solving above yields follows:

$$\tilde{x}_1(t) = (-\mathrm{j}0.01)\, e^{\lambda_1 t} \tag{38}$$

$$\tilde{x}_2(t) = (\mathrm{j}0.01)\, e^{\lambda_2 t} \tag{39}$$

Finally, we can calculate $x(t)$ by computing $V\vec{\tilde{x}}(t)$:

$$\begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} = \vec{x}(t) = V\vec{\tilde{x}}(t) = \begin{bmatrix} 1 & 1 \\ a & \bar{a} \end{bmatrix} \begin{bmatrix} (-\text{j}0.01)\, e^{\lambda_1 t} \\ (\text{j}0.01)\, e^{\lambda_2 t} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ a & \bar{a} \end{bmatrix} \begin{bmatrix} (-\text{j}0.01)\, e^{(-Z_0+j\omega_0)t} \\ (\text{j}0.01)\, e^{(-Z_0-j\omega_0)t} \end{bmatrix} \tag{40}$$

$$I_L(t) = -\text{j}0.01 e^{-Z_0 t}\left(e^{\text{j}\omega_0 t} - e^{-\text{j}\omega_0 t}\right) = 2\text{j} \times -\text{j}0.01 e^{-Z_0 t}\frac{(e^{\text{j}\omega_0 t} - e^{-\text{j}\omega_0 t})}{2\text{j}} \tag{41}$$

$$\therefore I_L(t) = 0.02 e^{-Z_0 t}\sin(\omega_0 t) \tag{42}$$

Regarding convergence, we have for the current $\lim_{t\to\infty} I_L(t) = 0$ since the $e^{-Z_0 t}$ term goes to 0 as $t \to \infty$.

Verification:

$$\begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix} = V\vec{\tilde{x}}(t) = \begin{bmatrix} 1 & 1 \\ -0.00125 + \text{j}50 & -0.00125 - \text{j}50 \end{bmatrix} \begin{bmatrix} (-\text{j}0.01)\, e^{(-Z_0+j\omega_0)t} \\ (\text{j}0.01)\, e^{(-Z_0-j\omega_0)t} \end{bmatrix} \tag{43}$$

$$I_L(t) = 0.02 e^{-Z_0 t}\sin(\omega_0 t) \tag{44}$$

$$V_C(t) = e^{-Z_0 t}\cos(\omega_0 t) - (25 \times 10^{-6}) e^{-Z_0 t}\sin(\omega_0 t) \tag{45}$$

Let us check if above satisfies $V_C(t) = V_L(t) = L\frac{dI_L(t)}{dt}$.

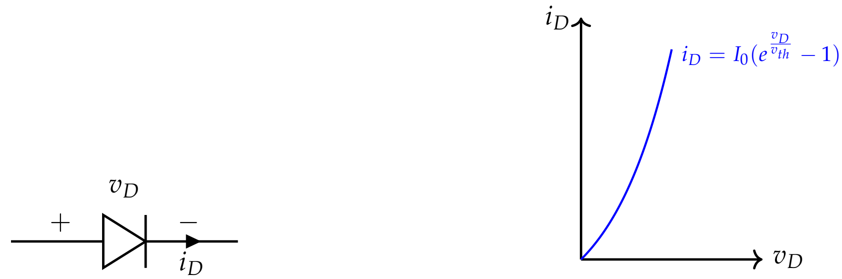
$$L\frac{dI_L(t)}{dt} = (25 \times 10^{-6})(0.02)e^{-Z_0 t}\left(\omega_0\cos(\omega_0 t) - Z_0\sin(\omega_0 t)\right) \tag{46}$$

$$= e^{-Z_0 t}\cos(\omega_0 t) - (25 \times 10^{-6})e^{-Z_0 t}\sin(\omega_0 t) \tag{47}$$

Therefore, the pair ($I_L(t)$ and $V_C(t)$) satisfies the differential equation.

**3. Nonlinear Circuit Analysis and Control (Spring 2021 Final)**

So far, we have mainly focused on analyzing circuits with linear circuit elements, including resistors, capacitors, and inductors. However, we now have the tools to analyze circuits with nonlinear components. One such component is the diode. Diodes show up in many circuit applications, such as a buck-boost converter, which is a DC-to-DC converter commonly used to raise or lower some supply voltage and feed it to some other part of your circuit. We give a circuit diagram of a diode as well as its defining IV relationship below.




**(a)** Diode circuit diagram

**(b)** $i_D = I_0(e^{\frac{v_D}{v_{th}}} - 1)$

**Figure 4:** Diode circuit element description

For simplicity, we will be assuming parameters (perhaps unrealistically) such that the I-V relationship for our diode is:

$$i_D = e^{v_D} - 1. \tag{48}$$

(a)

(b) We want to analyze the circuit below.

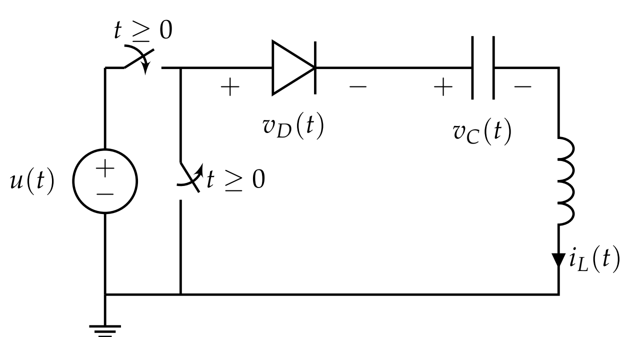



**Figure 5:** Diode LC Circuit Diagram

First, we'll define a model where $\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} v_C(t) \\ i_L(t) \end{bmatrix}$.

**Use KCL, KVL, and the element I-V relationships to get a system of differential equations that describe $\vec{x}(t)$ for $t \geq 0$ as a vector-valued function in terms of $v_C(t), i_L(t), u(t)$:**

$$\frac{d}{dt}\vec{x}(t) = \vec{f}(v_C, i_L, u) = \begin{bmatrix} f_1(v_C, i_L, u) \\ f_2(v_C, i_L, u) \end{bmatrix}.$$

**What are $f_1$ and $f_2$? Note that these may be non-linear functions, but they cannot contain derivatives. Show your work.** **Solution:** The concepts needed to solve this problem were explored in Homework 2 Q4 and EECS 16A.

KCL at the node between the capacitor and inductor gives:

$$C\frac{d}{dt}v_C(t) = i_L(t) \implies \frac{d}{dt}v_C(t) = \frac{1}{C}i_L(t) = f_1$$

KVL gives:

$$\begin{aligned} u(t) &= v_D(t) + v_C(t) + L\frac{d}{dt}i_L(t) \\ \implies u(t) &= \ln(i_L(t)+1) + v_C(t) + L\frac{d}{dt}i_L(t) \\ \implies \frac{d}{dt}i_L(t) &= -\frac{1}{L}v_C(t) - \frac{1}{L}\ln(i_L(t)+1) + \frac{1}{L}u(t) = f_2 \end{aligned}$$

Putting everything together, we get

$$\frac{d}{dt}\vec{x}(t) = \frac{d}{dt}\begin{bmatrix} v_C(t) \\ i_L(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{C}i_L(t) \\ -\frac{1}{L}v_C(t) - \frac{1}{L}\ln(i_L(t)+1) + \frac{1}{L}u(t) \end{bmatrix}$$

(c) Say that one of the equations you got above was in the form:

$$\frac{d}{dt}y(t) = \frac{1}{L}\ln(y(t)+a) + \frac{1}{L}u(t), \tag{49}$$

where $a \in \mathbb{R}$ is a constant and $u(t)$ can be thought of as a control input. (This is not necessarily the correct answer for the earlier part). You choose $y^* = 0$ and $u^* = 1\,\text{V}$ as the operating point. **Linearize the above equation (49) about this operating point.** Recall that $\frac{d}{dz}\ln(z) = \frac{1}{z}$. Show your work.

**Solution:** The concepts needed to solve this problem were explored in Note 15 and Homework 13 Q2.

We have the system

$$\frac{d}{dt}y(t) = f\left(y(t), u(t)\right).$$

The function can be linearized around $y^* = 0$ and $u^* = 1\,\text{V}$ as follows:

$$\begin{aligned} f\left(y(t), u(t)\right) &\approx f(y^*, u^*) + \left(\frac{\partial f}{\partial y}\bigg|_{y(t)=y^*}\right)(y(t) - y^*) + \left(\frac{\partial f}{\partial u}\bigg|_{u(t)=u^*}\right)(u(t) - u^*) && (50) \\ &= \frac{1}{L}\ln(0+a) + \frac{1}{L}\cdot 1 + \frac{1}{L}\left(\frac{1}{y(t)+a}\bigg|_{y(t)=0}\right)(y(t)-0) + \frac{1}{L}(u(t)-1) && (51) \\ &= \frac{1}{aL}y(t) + \frac{1}{L}u(t) + \frac{1}{L}\ln(a) && (52) \\ \implies \frac{d}{dt}y(t) &= \frac{1}{aL}y(t) + \frac{1}{L}u(t) + \frac{1}{L}\ln(a) && (53) \end{aligned}$$

Alternatively, you could have noticed that $f$ is already linear with respect to $u(t)$ and so you only need to linearize the ln term.

(d) Now suppose you chose a capacitance and inductance such that the linearized model for the system in Fig. 5 around a particular equilibrium point looked like:

$$\frac{d}{dt}\vec{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -4 & -4 \end{bmatrix}}_{A} \vec{x}(t) + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u(t) \tag{54}$$

In order to solve this system, you need to convert $A$ into a more convenient form.

**Find an orthonormal matrix $V$ and an upper-triangular matrix $T$ such that $A = VTV^\top$. Show your work.**

Hint: You may use the fact that the eigenvalues of $A$ are $-2, -2$, with eigenspace $\text{span}(\vec{v}_1)$, where $\vec{v}_1 = \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}$.

**Solution:** The concepts needed to solve this problem were explored in Note 11, Homework 10 Q5, and Discussion 10A.

From the algorithm discussed in lecture, we can construct an orthonormal basis recursively, starting with the single eigenvector given to us. We first want to find some vector $\vec{r}_1$ that is orthonormal to $\vec{v}_1$, which can be done either from inspection or Gram-Schmidt.

If you use Gram-Schmidt, then there will be 2 cases as you will orthonormalize $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

**Case 1:**

$$\begin{aligned} \vec{q}_1 &= \vec{e}_1 - \langle \vec{v}_1, \vec{e}_1 \rangle \vec{v}_1 \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{5}} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} \frac{4}{5} \\ \frac{2}{5} \end{bmatrix} \\ \implies \vec{r}_1 &= \frac{\vec{q}_1}{\|\vec{q}_1\|} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} \end{aligned}$$

**Case 2:**

$$\begin{aligned} \vec{q}_1 &= \vec{e}_2 - \langle \vec{v}_1, \vec{e}_2 \rangle \vec{v}_1 \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{2}{\sqrt{5}} \begin{bmatrix} -\frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} \frac{2}{5} \\ \frac{1}{5} \end{bmatrix} \\ \implies \vec{r}_1 &= \frac{\vec{q}_1}{\|\vec{q}_1\|} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} \end{aligned}$$

Then from lecture we know the $V$ basis should upper-triangularize a $2 \times 2$ matrix:

$$V = \begin{bmatrix} \vec{v}_1 & \vec{r}_1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix}$$

$$T = V^\top AV = \begin{bmatrix} \vec{v}_1^\top A\vec{v}_1 & \vec{v}_1^\top A\vec{r}_1 \\ \vec{r}_1^\top A\vec{v}_1 & \vec{r}_1^\top A\vec{r}_1 \end{bmatrix} = \begin{bmatrix} \lambda_1\vec{v}_1^\top\vec{v}_1 & \vec{v}_1^\top A\vec{r}_1 \\ \lambda_1\vec{r}_1^\top\vec{v}_1 & \vec{r}_1^\top A\vec{r}_1 \end{bmatrix} = \begin{bmatrix} -2 & -5 \\ 0 & -2 \end{bmatrix}$$

Alternatively, if we use $\vec{r}_1 = \begin{bmatrix} -\frac{2}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} \end{bmatrix}$

$$V = \begin{bmatrix} \vec{v}_1 & \vec{r}_1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

$$T = V^\top AV = \begin{bmatrix} \vec{v}_1^\top A\vec{v}_1 & \vec{v}_1^\top A\vec{r}_1 \\ \vec{r}_1^\top A\vec{v}_1 & \vec{r}_1^\top A\vec{r}_1 \end{bmatrix} = \begin{bmatrix} \lambda_1\vec{v}_1^\top\vec{v}_1 & \vec{v}_1^\top A\vec{r}_1 \\ \lambda_1\vec{r}_1^\top\vec{v}_1 & \vec{r}_1^\top A\vec{r}_1 \end{bmatrix} = \begin{bmatrix} -2 & 5 \\ 0 & -2 \end{bmatrix}$$

(e) We now want to move the eigenvalues of our linearized system more left in the complex plane to have our state approach the equilibrium point faster. The system is given below again for convenience:

$$\frac{d}{dt}\vec{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -4 & -4 \end{bmatrix}}_{A}\vec{x}(t) + \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{\vec{b}}u(t).$$

**Design a state-feedback controller $u = \vec{k}^\top\vec{x} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}\vec{x}$ to move the eigenvalues of the system to $\lambda = -4, -5$. That is, find $k_1, k_2$ to give the desired eigenvalues.**

**Solution:** The concepts needed to solve this problem were explored in Note 8 and Homework 9 Q2.

If we set $\delta u = \vec{k}^\top\delta\vec{x} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} x$, then the closed loop system becomes

$$\frac{d}{dt}\delta\vec{x} = (A + \vec{b}\vec{k}^\top)\delta\vec{x} = \left(\begin{bmatrix} 0 & 1 \\ -4 & -4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 4k_1 & 4k_2 \end{bmatrix}\right)\delta\vec{x}$$

$$= \begin{bmatrix} 0 & 1 \\ -4+4k_1 & -4+4k_2 \end{bmatrix} \delta\vec{x}$$

We want the eigenvalues to be $-4, -5$ so the desired characteristic polynomial is $(\lambda+4)(\lambda+5) = \lambda^2 + 9\lambda + 20$. The characteristic polynomial of our closed loop matrix is

$$\begin{aligned} -\lambda(-4+4k_2-\lambda) - (-4+4k_1) &= \lambda^2 + (4-4k_2)\lambda + (4-4k_1) \\ &= \lambda^2 + 9\lambda + 20 \end{aligned}$$

Thus, we need $k_1 = -4$, and $k_2 = \frac{-5}{4}$ so $\vec{k}^\top = \begin{bmatrix} -4 & \frac{-5}{4} \end{bmatrix}$.

**4. Minimum Norm Solutions for Circuits involving Resistors (Fall 2021 Final)**

Consider a current $i_s$ flowing into a network of two parallel resistors $R_1$ and $R_2$, as shown in fig. 6 below.

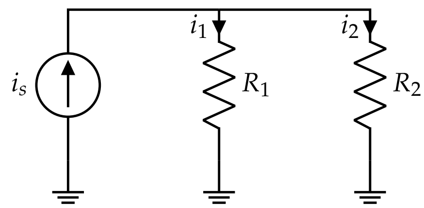



**Figure 6:** Current $i_s$ dividing into $i_1$ and $i_2$.

From EECS 16A, we know that we can equate the voltage drops across the parallel resistors to derive $i_1 = \frac{R_2}{R_1+R_2} i_s$ and $i_2 = \frac{R_1}{R_1+R_2} i_s$. In this problem, we will try to derive the same current division result using the concept of minimum norm instead of voltage analysis.

It turns out that the current $i_s$ will divide into two parts $i_1$ and $i_2$ in such a way that minimizes the total power dissipation $P = i_1^2 R_1 + i_2^2 R_2$ in the resistors.

(a) Argue that the current division result given by $i_1 = \frac{R_2}{R_1+R_2} i_s$ and $i_2 = \frac{R_1}{R_1+R_2} i_s$ minimizes the total power dissipation $P = i_1^2 R_1 + i_2^2 R_2$ using calculus.

Use the fact that KCL gives $i_2 = i_s - i_1$ to express $P$ as a function of $i_1$ only.

*(HINT: Once you solve for the optimal $i_1$, you don't have to do calculus again for $i_2$. Just use KCL.)*

**Solution:** We can minimize $P$ as follows:

$$P = i_1^2 R_1 + i_2^2 R_2 = i_1^2 R_1 + (i_s - i_1)^2 R_2 \tag{55}$$

$$\implies \frac{\mathrm{d}P}{\mathrm{d}i_1} = 2 i_1 R_1 - 2(i_s - i_1) R_2 = 0 \tag{56}$$

$$\implies i_1 = \frac{R_2}{R_1 + R_2} i_s \tag{57}$$

$$\implies i_2 = i_s - i_1 = \frac{R_1}{R_1 + R_2} i_s. \tag{58}$$

Here, we notice that the quadratic has a positive constant multiplying the squared term, and so this must be the unique minimum.

Hence it is proved that $i_1 = \frac{R_2}{R_1+R_2} i_s$ and $i_2 = \frac{R_1}{R_1+R_2} i_s$ minimize $P$.

(b) Instead of using calculus to minimize the total power dissipation $P$, we can represent the current division problem as a minimum norm problem. Consider the vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ where $x_1 = i_1\sqrt{R_1}$ and $x_2 = i_2\sqrt{R_2}$. Notice that $P = i_1^2 R_1 + i_2^2 R_2 = x_1^2 + x_2^2 = \|\vec{x}\|^2$. Find the row vector $A$ so that the KCL constraint $i_1 + i_2 = i_s$ can be written as $A\vec{x} = i_s$.

**Solution:** The total power dissipation given by $P = \|\vec{x}\|^2$ has to be minimized. KCL gives us

$$i_1 + i_2 = i_s \tag{59}$$

$$\implies \frac{x_1}{\sqrt{R_1}} + \frac{x_2}{\sqrt{R_2}} = i_s \tag{60}$$

$$\implies \begin{bmatrix} \frac{1}{\sqrt{R_1}} & \frac{1}{\sqrt{R_2}} \end{bmatrix} \vec{x} = i_s \tag{61}$$

$$\implies A\vec{x} = i_s \tag{62}$$

So $A = \begin{bmatrix} \frac{1}{\sqrt{R_1}} & \frac{1}{\sqrt{R_2}} \end{bmatrix}$.

(c) Using the $A$ matrix you found above, **what is the minimum norm solution to $A\vec{x} = i_s$?** Show your work.

To help you save computation, the compact SVD of a general $1 \times 2$ row vector is given by

$$\begin{bmatrix} a & b \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sqrt{a^2+b^2} \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} & \frac{b}{\sqrt{a^2+b^2}} \end{bmatrix}}_{V^\top} \tag{63}$$

**Solution:**

We know that the minimum norm solution for the system $A\vec{x} = i_s$ is given by $\tilde{\vec{x}} = A^\dagger i_s$, where $A^\dagger$ is the pseudo-inverse of $A$. Using eq. (63), the compact SVD of $A$ is given by

$$A = U\Sigma V^\top \tag{64}$$

$$\begin{bmatrix} \frac{1}{\sqrt{R_1}} & \frac{1}{\sqrt{R_2}} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sqrt{\frac{R_1+R_2}{R_1R_2}} \end{bmatrix}}_{\Sigma} \underbrace{\frac{1}{\sqrt{R_1+R_2}} \begin{bmatrix} \sqrt{R_2} & \sqrt{R_1} \end{bmatrix}}_{V^\top} \tag{65}$$

The pseudo-inverse is given by

$$A^\dagger = V\Sigma^{-1}U^\top \tag{66}$$

$$= \underbrace{\frac{1}{\sqrt{R_1+R_2}} \begin{bmatrix} \sqrt{R_2} \\ \sqrt{R_1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} \sqrt{\frac{R_1R_2}{R_1+R_2}} \end{bmatrix}}_{\Sigma^{-1}} \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{U^\top} \tag{67}$$

$$= \begin{bmatrix} \frac{R_2\sqrt{R_1}}{R_1+R_2} \\ \frac{R_1\sqrt{R_2}}{R_1+R_2} \end{bmatrix} \tag{68}$$

Hence the minimum norm solution is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{R_2\sqrt{R_1}}{R_1+R_2} \\ \frac{R_1\sqrt{R_2}}{R_1+R_2} \end{bmatrix} i_s \tag{69}$$

(d) Transform the minimum norm solution of $A\vec{x} = i_s$ to the original variables $i_1$ and $i_2$, and confirm that the result is $i_1 = \frac{R_2}{R_1+R_2} i_s$ and $i_2 = \frac{R_1}{R_1+R_2} i_s$ as the current-divider formula predicts. Show your work.

**Solution:** Changing variables back to the current division problem, we have

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} \frac{x_1}{\sqrt{R_1}} \\ \frac{x_2}{\sqrt{R_2}} \end{bmatrix} = \begin{bmatrix} \frac{R_2 i_s}{R_1+R_2} \\ \frac{R_1 i_s}{R_1+R_2} \end{bmatrix} \tag{70}$$

This matches the current division ratio from regular voltage analysis.

(e) We can solve any arbitrarily complicated circuit network using KCL and norm minimization, following the same technique that we used for the simple network in fig. 6. Consider a resistor network which has $n$ resistor branches, with currents $i_1, i_2, \ldots, i_n$ across the branch resistances $R_1, R_2, \ldots, R_n$ respectively, and $m$ total nodes each with current sources $i_{s_1}, i_{s_2}, \ldots, i_{s_m}$, which may be positive, negative or zero, as shown in fig. 7. Then the $m$ KCL equations at the $m$ nodes can be written as $K\vec{i} = \vec{i}_s$, where $K \in \mathbb{R}^{m\times n}$, $\vec{i} = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} \in \mathbb{R}^n$, and $\vec{i}_s = \begin{bmatrix} i_{s_1} \\ i_{s_2} \\ \vdots \\ i_{s_m} \end{bmatrix} \in \mathbb{R}^m$. This KCL constraint $K\vec{i} = \vec{i}_s$ completely captures what is visualized in fig. 7, so you don't have to write any additional KCL. Note that fig. 6 is a simple example of fig. 7 with $n = 2$ and $m = 1$.

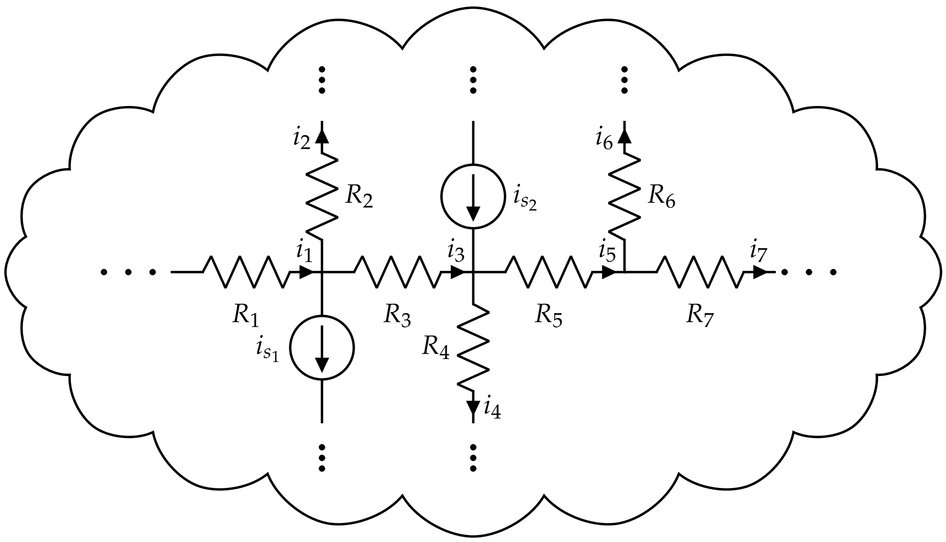



**Figure 7:** A section of an arbitrarily complicated network with $n$ branches and $m$ nodes.

We can change variables to $\vec{x} = D\vec{i}$ to represent the KCL constraint $K\vec{i} = \vec{i}_s$ as $A\vec{x} = \vec{i}_s$, and so the minimization of dissipated power $P = \sum_{j=1}^{n} i_j^2 R_j$ is just the minimization of $\sum_{j=1}^{n} x_j^2 = \|\vec{x}\|^2$. Find the diagonal matrix $D$, and then find the matrix $A$ in terms of $D$ and $K$.

*(HINT: Look at how $\vec{x}$ was defined in the previous part.)* **Solution:** Define a vector $\vec{x} \in \mathbb{R}^n$ where $x_j = i_j\sqrt{R_j}\ \forall j \in [1, n]$. Then we can change variables from $\vec{i}$ to $\vec{x}$ using the relation

$$\vec{x} = \underbrace{\begin{bmatrix} \sqrt{R_1} & 0 & \ldots & 0 \\ 0 & \sqrt{R_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sqrt{R_n} \end{bmatrix}}_{D} \vec{i} \tag{71}$$

$$\implies \vec{i} = D^{-1}\vec{x} \tag{72}$$

The diagonal matrix $D \in \mathbb{R}^{n\times n}$ has non-zero diagonal elements, hence it is invertible. Hence the KCL can be represented as

$$K\vec{i} = \vec{i}_s \tag{73}$$

$$\implies \underbrace{KD^{-1}}_{A}\vec{x} = \vec{i}_s \tag{74}$$

(f) Assume the compact SVD of $A$ is given by $A = U\Sigma V^\top$.

Use the minimum norm solution to $A\vec{x} = \vec{i}_s$ to solve for $\vec{i}$. Recall from the previous part that $\vec{x} = D\vec{i}$. Your final answer for $\vec{i}$ can only use $U$, $\Sigma$, $V$, $D$, $\vec{i}_s$ as well as standard matrix operations like inverses, etc.

**Solution:**

We can find $\vec{i}$ by using the min norm solution for eq. (74) as follows:

$$\vec{x} = A^\dagger \vec{i}_s = V\Sigma^{-1}U^\top \vec{i}_s \tag{75}$$

$$\implies \vec{i} = D^{-1}V\Sigma^{-1}U^\top \vec{i}_s. \tag{76}$$

This connection between power-dissipation minimization and electrical circuits goes even deeper. With the tools that you learn in Math 53 and EECS 127, you can see that the very concept of voltage itself can be understood via Lagrange multipliers (dual variables) associated with the KCL constraints that are active at each node. There is a further sense in which the reason that "state" is associated with inductors and capacitors is precisely because there is energy stored in these circuit elements. This is intimately related to the concept of the Lagrangian formulation of mechanics that you might encounter in Physics and Mechanical Engineering. Different formulations of "minimalist principles" are ubiquitous in science and engineering.

**5. Affine Control (Spring 2022 Midterm)**

In this problem, we will analyze a *affine* model of the form

$$x[i+1] = \alpha x[i] + \beta u[i] + \gamma \tag{77}$$

where $\alpha, \beta, \gamma \in \mathbb{R}$, $x\colon \mathbb{N} \to \mathbb{R}$ is the state, and $u\colon \mathbb{N} \to \mathbb{R}$ is the input. Affine models are ubiquitous in control theory – in fact, our robot car from lab obeys a two-state-variable affine model.

(a) Suppose (for this part only) that:

- $\alpha = 1$,
- $\beta = 0$,
- $\gamma \neq 0$,
- $x[0]$ is anything.

so the model is of the form

$$x[i+1] = x[i] + \gamma. \tag{78}$$

**Is the state $x$ bounded?** *Justify your answer.*

**Solution:** No. In particular we have

$$x[i] = x[0] + \gamma \cdot i. \tag{79}$$

Since $\gamma \neq 0$, we have that

$$\lim_{i\to\infty} |x[i]| = \lim_{i\to\infty} |x[0] + \gamma \cdot i| = \infty. \tag{80}$$

(b) Suppose (for this part only) that the state evolves according to eq. (77), i.e.,

$$x[i+1] = \alpha x[i] + \beta u[i] + \gamma \tag{81}$$

and

- $\alpha \neq 0$,
- $\beta > 0$,
- $\gamma \neq 0$,
- $x[0] = 0$.

Suppose that we supply feedback control of the form

$$u[i] = f \cdot x[i] \tag{82}$$

for $f \in \mathbb{R}$.

i. **For the specific case of $f = \frac{-1-\alpha}{\beta}$, show that the state $x$ is bounded.**

**Solution:** After plugging in feedback control, our model becomes

$$x[i+1] = (\alpha + \beta f)x[i] + \gamma. \tag{83}$$

If $f = \frac{-1-\alpha}{\beta}$, then $\alpha + \beta f = -1$, so we have

$$x[i+1] = -x[i] + \gamma \tag{84}$$

which has state trajectory

$$x[i] = \begin{cases} \gamma & i \text{ is odd} \\ 0 & i \text{ is even} \end{cases} \tag{85}$$

which is bounded by $|\gamma|$. Thus if $f = \frac{-1-\alpha}{\beta}$ then $x$ is bounded.

ii. **In terms of $\alpha$ and $\beta$, give a range of $f$ that keeps the state $x$ bounded.**

**Solution:** The answer is

$$f \in \left[\frac{-1-\alpha}{\beta}, \frac{1-\alpha}{\beta}\right). \tag{86}$$

*The endpoint (non)inclusions are important.*

Quick and dirty reasoning: With feedback control, our model becomes

$$x[i+1] = (\alpha + \beta f)x[i] + \gamma \tag{87}$$

We know that for a regular linear system of the form

$$x[i+1] = \kappa x[i] + u[i] \tag{88}$$

if the input $u$ is bounded and $|\kappa| < 1$, then the state $x$ is bounded; this corresponds to $\alpha + \beta f \in (-1, 1)$, which is if and only if $f \in \left(\frac{-1-\alpha}{\beta}, \frac{1-\alpha}{\beta}\right)$. It turns out that, if we restrict ourselves to *constant* inputs $\gamma = u[i]$ (which is a sub-class of bounded inputs), then $\kappa = -1$ (corresponding to $f = \frac{-1-\alpha}{\beta}$) also keeps the state bounded – and we can see this by part i.

(Not Required) Formal proof

We use the same notation, letting $\kappa := \alpha + \beta f$, so that Equation (83) becomes

$$x[i+1] = \kappa x[i] + \gamma. \tag{89}$$

Indeed, the general trajectory of Equation (89) becomes

$$x[i] = \begin{cases} \gamma \cdot i & \kappa = 1 \\ \gamma \cdot \frac{\kappa^i - 1}{\kappa - 1} & \kappa \neq 1 \end{cases}. \tag{90}$$

Thus

$$|x[i]| = \begin{cases} |\gamma| i & \kappa = 1 \\ |\gamma| \frac{|\kappa^i - 1|}{|\kappa - 1|} & \kappa \neq 1 \end{cases}. \tag{91}$$

From here there are some cases:

- $|\kappa| = 1$. Here there are two more cases:
  - If $\kappa = 1$ then
  $$|x[i]| = |\gamma| i \to \infty \tag{92}$$
  and $x$ is unbounded.
  - If $\kappa = -1$ then
  $$|x[i]| = |\gamma| \frac{|(-1)^i - 1|}{2} = \begin{cases} |\gamma| & i \text{ is odd} \\ 0 & i \text{ is even} \end{cases} \tag{93}$$
  which is bounded by $|\gamma|$.
- $|\kappa| > 1$. Then
  $$|x[i]| = |\gamma| \frac{|\kappa^i - 1|}{|\kappa - 1|} \to \infty \tag{94}$$
  and $x$ is unbounded.

- $|\kappa| < 1$. Then

$$|x[i]| = |\gamma| \frac{|\kappa^i - 1|}{|\kappa - 1|} \to \frac{|\gamma|}{|\kappa - 1|} \tag{95}$$

and $x$ is bounded.

Thus $x$ is bounded if and only if $\kappa \in [-1, 1)$. Since $\kappa = \alpha + \beta f$, this is if and only if

$$f \in \left[\frac{-1-\alpha}{\beta}, \frac{1-\alpha}{\beta}\right). \tag{96}$$

(c) Suppose (for this part only) that the state evolves according to eq. (77), i.e.,

$$x[i+1] = \alpha x[i] + \beta u[i] + \gamma \tag{97}$$

and

- $\alpha$ is anything,
- $\beta$ is anything,
- $\gamma$ is anything,
- $x[0]$ is anything.

Suppose that we are setting up a least-squares system identification procedure to learn $\alpha$, $\beta$, and $\gamma$, and that we have data of the form $(x[i], u[i], x[i+1])$, for $i \in \{0, 1, \ldots, \ell - 1\}$. **Set up a least-squares problem $D\vec{p} \approx \vec{s}$ to learn estimates for $\alpha, \beta, \gamma$. What are $D$, $\vec{p}$, and $\vec{s}$?**

*NOTE*: Your answer for $D$ should be as compact as possible.

*NOTE*: You do not need to solve the least squares problem; just set it up.

**Solution:** We follow the procedure on the hint and the lab. Namely, we have

$$D := \begin{bmatrix} x[0] & u[0] & 1 \\ \vdots & \vdots & \vdots \\ x[\ell-1] & u[\ell-1] & 1 \end{bmatrix} \qquad \vec{p} := \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \qquad \vec{s} := \begin{bmatrix} x[1] \\ \vdots \\ x[\ell] \end{bmatrix}. \tag{98}$$

(d) Suppose (for this part only) that the state evolves according to Equation (77), i.e.,

$$x[i+1] = \alpha x[i] + \beta u[i] + \gamma \tag{99}$$

and

- $\alpha > 1$,
- $\beta > 0$,
- $\gamma > 0$,
- $x[0]$ is anything.

Suppose that we actually got our discrete-time model

$$x[i+1] = \alpha x[i] + \beta u[i] + \gamma \tag{100}$$

by discretizing a continuous-time model

$$\frac{\mathrm{d}}{\mathrm{d}t} x(t) = a x(t) + b u(t) + c \tag{101}$$

where the sampling interval length is $\Delta = 1$, i.e., $x[i] = x(i\Delta)$, and $u(t)$ is piecewise constant over intervals of length $\Delta$, i.e., $u(t) = u(i\Delta) = u[i]$ for $t \in [i\Delta, (i+1)\Delta)$. **In terms of $\alpha, \beta, \gamma$, what are $a, b$, and $c$?**

*(HINT: You can use any discretization formulas we derived in class, as long as they apply. Alternatively, you may use the following formula in your derivation.*

*For a constant input $v$, and a time $t_0$ for which $x(t_0)$ is known, the solution to the differential equation*

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t) + v \qquad t \geq t_0 \tag{102}$$

*is given by*

$$x(t) = \mathrm{e}^{a(t-t_0)}x(t_0) + \frac{\mathrm{e}^{a(t-t_0)} - 1}{a} \cdot v, \qquad t \geq t_0. \tag{103}$$

*when $a \neq 0$. Also, recall from the problem statement above that the sampling interval length $\Delta = 1$.)*

**Solution:** First Solution Using Integral Formula:

Starting from time $t_0 = i\Delta$ and the value $x(t_0) = x[i]$, we use the piecewise constant assumption to simplify the continuous model.

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t) + bu[i] + c, \qquad t \in [i\Delta, (i+1)\Delta). \tag{104}$$

We use Equation (103) to find $x(t) = x((i+1)\Delta) = x[i+1]$. Using the substitution $v = bu[i] + c$, we have

$$x(t) = \mathrm{e}^{a(t-i\Delta)}x[i] + \frac{\mathrm{e}^{a(t-i\Delta)} - 1}{a}(bu[i] + c). \tag{105}$$

Plugging in $t = i\Delta$ gets

$$\begin{aligned}
x[i+1] &= x((i+1)\Delta) && (106)\\
&= \mathrm{e}^{a((i+1)\Delta - i\Delta)}x[i] + \frac{\mathrm{e}^{a((i+1)\Delta - i\Delta)} - 1}{a}(bu[i] + c) && (107)\\
&= \mathrm{e}^{a\Delta}x[i] + \frac{\mathrm{e}^{a\Delta} - 1}{a}(bu[i] + c) && (108)\\
&= \mathrm{e}^{a\Delta}x[i] + \frac{\mathrm{e}^{a\Delta} - 1}{a} \cdot bu[i] + \frac{\mathrm{e}^{a\Delta} - 1}{a} \cdot c && (109)\\
&= \mathrm{e}^{a}x[i] + \frac{\mathrm{e}^{a} - 1}{a} \cdot bu[i] + \frac{\mathrm{e}^{a} - 1}{a} \cdot c && (110)
\end{aligned}$$

Thus we get

$$\begin{aligned}
\alpha &= \mathrm{e}^{a} && (111)\\
\beta &= \frac{\mathrm{e}^{a} - 1}{a} \cdot b && (112)\\
\gamma &= \frac{\mathrm{e}^{a} - 1}{a} \cdot c. && (113)
\end{aligned}$$

Thus

$$\begin{aligned}
a &= \log(\alpha) && (114)\\
b &= \frac{a}{\mathrm{e}^{a} - 1}\beta = \frac{\log(\alpha)}{\alpha - 1}\beta && (115)\\
c &= \frac{a}{\mathrm{e}^{a} - 1}\gamma = \frac{\log(\alpha)}{\alpha - 1}\gamma. && (116)
\end{aligned}$$

Alternative Solution Using Known Discretization Coefficients:

Letting $v(t) = bu(t) + c$, we have that $v(t)$ is piecewise constant over intervals of length $\Delta$. We have the continuous-time model

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t) + v(t) \tag{117}$$

from which we can read off the discretization

$$x[i+1] = \mathrm{e}^{a}x[i] + \frac{\mathrm{e}^{a} - 1}{a}v[i] \tag{118}$$

$$= \mathrm{e}^{a}x[i] + \frac{\mathrm{e}^{a} - 1}{a}(bu[i] + c) \tag{119}$$

$$= \mathrm{e}^{a}x[i] + \frac{\mathrm{e}^{a} - 1}{a}bu[i] + \frac{\mathrm{e}^{a} - 1}{a}c \tag{120}$$

from which the solution proceeds the same as before.

**6. SVD Calculation (Fall 2021 Final)**

(a) Let $A = \begin{bmatrix} 0 & 2 \\ \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}$. The eigenvalues of $AA^\top = \begin{bmatrix} 4 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$ are $5, 2, 0$ with corresponding unnormalized eigenvectors $\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}$.

In addition, we know that:

$$A^\top \begin{bmatrix} 2 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{2} & 0 \\ 5 & 0 & 0 \end{bmatrix} \tag{121}$$

**Write out the singular value decomposition (SVD) of A in any form you choose (outer product form, compact, or full).** (No need to show work.)

**Solution:** The normalized eigenvectors of $AA^\top$ are the columns of $U$:

$$U = \begin{bmatrix} \frac{2\sqrt{5}}{5} & 0 & \frac{-\sqrt{5}}{5} \\ 0 & 1 & 0 \\ \frac{\sqrt{5}}{5} & 0 & \frac{2\sqrt{5}}{5} \end{bmatrix} \tag{122}$$

Also, the singular values of $A$ are the square roots of the eigenvalues of $AA^\top$. Since $\Sigma$ is the same shape as $A$ and contains the singular values along its diagonal (with zeros elsewhere):

$$\Sigma = \begin{bmatrix} \sqrt{5} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix} \tag{123}$$

Finally, we can find the columns of $V$ from the columns of $U$ and the nonzero singular values:

$$\vec{v}_1 = \frac{1}{\sigma_1} A^\top \vec{u}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{124}$$

$$\vec{v}_2 = \frac{1}{\sigma_2} A^\top \vec{u}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{125}$$

$$V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{126}$$

Writing the SVD in outer product form yields:

$$A = \sum_{i=1}^{2} \sigma_i \vec{u}_i \vec{v}_i^\top = \sqrt{5} \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ 0 \\ \frac{\sqrt{5}}{5} \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} + \sqrt{2} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{127}$$

where $\vec{u}_i$ and $\vec{v}_i$ are the columns of $U$ and $V$, respectively. We can also write the full SVD:

$$A = U\Sigma V^\top = \begin{bmatrix} \frac{2\sqrt{5}}{5} & 0 & \frac{-\sqrt{5}}{5} \\ 0 & 1 & 0 \\ \frac{\sqrt{5}}{5} & 0 & \frac{2\sqrt{5}}{5} \end{bmatrix} \begin{bmatrix} \sqrt{5} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{128}$$

or the compact SVD:

$$A = U\Sigma V^\top = \begin{bmatrix} \frac{2\sqrt{5}}{5} & 0 \\ 0 & 1 \\ \frac{\sqrt{5}}{5} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{5} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{129}$$

For these specific numbers, once you had the first two columns of $U$ and know the $\Sigma$, you could also just read off what the $\vec{v}_i$ have to be since you knew that the first column of $A$ is a scaled multiple of $\vec{u}_2$ and the second column of $A$ is a scaled multiple of $\vec{u}_1$. The numbers were made simple enough to let you do this as a valid way of solving the problem.

(b) **What is the best rank 1 approximation of $A$ (i.e., what is the rank 1 matrix $B$ that minimizes $\|A - B\|_F$)? Write your answer as a $3 \times 2$ dimensional matrix.** (No need to show work)

**Solution:** The best rank 1 approximation is $B = \sigma_1 \vec{u}_1 \vec{v}_1^\top$, or the outer product of the vectors $\vec{u}_i$ and $\vec{v}_i$ that correspond to the largest singular value $\sigma_1$.

In this case:

$$B = \sqrt{5} \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ 0 \\ \frac{\sqrt{5}}{5} \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{130}$$

For this specific problem, you could also observe that the two columns of $A$ are clearly orthogonal to each other. Consequently, the best rank 1 approximation can only capture one of them, which means that you just want the heavier one which is the second one. From this, you could have immediately seen that $B$ must have this form. This is also a valid alternative way to solve this problem.

**7. Dynamical system approach to solving Ridge Regression (Spring 2022 Final)**

In this problem, we will derive a dynamical system based approach to solving a modified version of the least-squares problem, commonly known as "ridge regression". This problem attempts to find the $\vec{x}$ that minimizes $\|A\vec{x} - \vec{y}\|^2 + \lambda \|\vec{x}\|^2$. Here we assume $A \in \mathbb{R}^{m \times n}$ is full column rank and scalar $\lambda \geq 0$.

The solution to the ridge regression problem is

$$\vec{\hat{x}} = (A^\top A + \lambda I)^{-1} A^\top \vec{y}. \tag{131}$$

Note that this solution is quite similar to the solution of least-squares. In many cases, direct computation of the solution to ridge regression is too slow, because it requires computing the matrix inverse $(A^\top A + \lambda I)^{-1}$, which is generally very costly for $A$ with very large dimensions. We will instead solve the problem iteratively by using an update rule which turns this particular problem into an analysis of a particular discrete-time state-space dynamical system.

(a) We first connect the ridge regression problem to the familiar ordinary least-squares problem. **State the condition on $\lambda$ in (131) needed to recover the least squares solution.**

**Solution:** In ridge regression the solution is $\vec{\hat{x}} = (A^\top A + \lambda I)^{-1} A^\top \vec{y}$. By setting $\lambda = 0$ we recover the formula to compute the solution for least-squares.

(b) Using (131), **show that $(A^\top A + \lambda I)\vec{\hat{x}} - A^\top \vec{y} = 0$.**

**Solution:** We are given that for ridge regression, the solution is of the form $\vec{\hat{x}} = (A^\top A + \lambda I)^{-1} A^\top \vec{y}$. Plugging this into the left-hand side of the equation that we want to prove, we get

$$(A^\top A + \lambda I)\vec{\hat{x}} - A^\top \vec{y} = (A^\top A + \lambda I)(A^\top A + \lambda I)^{-1} A^\top \vec{y} - A^\top \vec{y} = A^\top \vec{y} - A^\top \vec{y} = 0.$$

(c) In iterative optimization schemes, we will get a sequence of estimates for $\vec{\hat{x}}$ at each timestep. Let $\vec{x}[i]$ denote our estimate for $\vec{\hat{x}}$ at timestep $i$.

In this problem we will consider the following update rule for solving the ridge regression problem:

$$\vec{x}[i+1] = \vec{x}[i] - \alpha\left((A^\top A + \lambda I)\vec{x}[i] - A^\top \vec{y}\right) \tag{132}$$

that gives us an updated estimate $\vec{x}[i+1]$ using the previous one $\vec{x}[i]$. Here $\alpha$ is the "step size" in our update rule which controls how much we update our solution estimate at each time step. For the purposes of this problem, it doesn't matter where we got the update rule, but the important thing to note is that if $\vec{x}[i] = \vec{\hat{x}}$, then by the previous part, $\vec{x}[i+1] = \vec{\hat{x}}$ and the system remains in equilibrium at $\vec{\hat{x}}$ for all time.

To show that $\vec{x}[i] \to \vec{\hat{x}}$, we define a new state variable $\Delta\vec{x}[i] = \vec{x}[i] - \vec{\hat{x}}$. It represents the deviation from where we want to be.

**Derive the discrete-time state evolution equation for $\Delta\vec{x}[i]$, and show that it takes the form:**

$$\Delta\vec{x}[i+1] = (I - \alpha G)\Delta\vec{x}[i]. \tag{133}$$

**What is $G$?**

**Solution:**

$$\begin{aligned}
\Delta\vec{x}[i+1] &= \vec{x}[i+1] - \vec{\hat{x}} && (134)\\
&= \vec{x}[i] - \alpha\left((A^\top A + \lambda I)\vec{x}[i] - A^\top \vec{y}\right) - \vec{\hat{x}} && (135)\\
&= (\vec{x}[i] - \vec{\hat{x}}) - \alpha\left((A^\top A + \lambda I)\vec{x}[i] - A^\top \vec{y}\right) && (136)\\
&= \Delta\vec{x}[i] - \alpha\left((A^\top A + \lambda I)\vec{x}[i] - A^\top \vec{y}\right) && (137)\\
&= \Delta\vec{x}[i] - \alpha\left((A^\top A + \lambda I)\vec{x}[i] - (A^\top A + \lambda I)\vec{\hat{x}}\right) && (138)\\
&= \Delta\vec{x}[i] - \alpha(A^\top A + \lambda I)\Delta\vec{x}[i] && (139)\\
&= \left(I - \alpha(A^\top A + \lambda I)\right)\Delta\vec{x}[i] && (140)
\end{aligned}$$

So $G = A^\top A + \lambda I$.

(d) We would like to select $\alpha$ such that $\Delta\vec{x}[i]$ converges to 0. In particular, we want to make sure that we have a stable system. To do this, we need to understand the eigenvalues of $I - \alpha G$. **Given that $\lambda_k\{G\}$ are the eigenvalues of $G$, for $k \in \{1, 2, \ldots, n\}$ what are the eigenvalues of the matrix $I - \alpha G$?** *(Please fill in one of the circles for the options below. You will only be graded on your final answer.)*

i. $1 - \alpha\lambda_k\{G\}$ for $k \in \{1, 2, \ldots, n\}$

ii. $\alpha\lambda_k\{G\}$ for $k \in \{1, 2, \ldots, n\}$

iii. $1 - \lambda_k\{G\}$ for $k \in \{1, 2, \ldots, n\}$

iv. $1 + \alpha\lambda_k\{G\}$ for $k \in \{1, 2, \ldots, n\}$

| Option | i | ii | iii | iv |
|---|---|---|---|---|
| Answer | ◯ | ◯ | ◯ | ◯ |

**Solution:** Suppose that $(\lambda_k\{G\}, \vec{v}_k\{G\})$ is an eigenvalue-eigenvector pair for $G$. Then

$$(I - \alpha G)\vec{v}_k\{G\} = \vec{v}_k\{G\} - \alpha G\vec{v}_k\{G\} = \vec{v}_k\{G\} - \alpha\lambda_k\{G\}\vec{v}_k\{G\} = (1 - \alpha\lambda_k\{G\})\vec{v}_k\{G\}. \quad (141)$$

Hence, the eigenvalues of $I - \alpha G$ are $1 - \alpha\lambda_k\{G\}$.

(e) For system (133) to be stable, we need all the eigenvalues of $I - \alpha G$ to have magnitudes that are smaller than 1 (since this is a discrete-time system). **State the condition on $\alpha$ that would ensure that system (133) is stable.** You may assume that $\lambda_k\{G\}$ are real and $\lambda_k\{G\} > 0$ for $k \in \{1, 2, \ldots, n\}$.

**Solution:** For stability, we require that $|1 - \alpha\lambda_k\{G\}| < 1$ for $k \in \{1, 2, \ldots, n\}$. This is equivalent to the condition $-1 < 1 - \alpha\lambda_k\{G\} < 1$ for $k \in \{1, 2, \ldots, n\}$. Isolating the $\alpha$ term we obtain the condition $0 < \alpha < \frac{2}{\lambda_k\{G\}}$ for $k \in \{1, 2, \ldots, n\}$.

**8. Brain-Machine Interface and Neuron Sorting (Homework Practice)**

The iPython notebook `pca_BMI.ipynb` will guide you through the process of analyzing brain machine interface data leveraging the SVD. This will help you to prepare for the project.

For SIXT33N (your robot car), you used the SVD and PCA to classify your sound inputs so that your car does what you tell it to do.

Brain-Machine interfaces (BMIs) are a way for a brain to directly communicate to an external device. They're often used for research, mapping, assisting or repairing human cognitive or sensory-motor function.

Data was collected that shows waveform traces of (simulated) brain waves of a subject whose arm is pointing in certain directions over time. These waveform traces are gathered from electrodes inserted into the brain, but the electrodes are generally recording more than 1 neuron at the same time — the brain is crowded after all. To make predictions based on neuron firing rates, we need to be able to distinguish waveforms that come from different neurons.

Each neuron has its own waveform "signature" shape unique to that neuron. This is due to the physical characteristics of neurons, such as their physical shape and structure. It is impossible to know beforehand how many neurons an electrode measures or what each neuron's waveform looks like, so we must first separate the waveforms from the different neurons near the electrode.

The goal of this problem is to see how we can use the SVD (in its PCA manifestation) and clustering to decide which neuron fired. We will first look at a case where there are only two neurons, then a case where there are three neurons. The neurons have also been presorted using a professional software package, so we can check our model against presorted data.

Please complete the notebook by following the instructions given.

**Task 1: Two Neuron Waveform Sorting**

(a) The data you have are traces of length 32 timesteps. The data is arranged along the rows of the matrix, each row is one trace. Import the data sets and see the average waveform for each presorted neuron by running the corresponding cells in the .ipynb. **What is the shape of the training data matrix** `three_neurons_training`?

**Solution:** Part a) of the notebook `pca_brain_machine_interface_sol.ipynb` contains solutions to this exercise. `three_neurons_training.shape = (4398, 32)`

(b) **What do the columns and rows of the training data matrix:** `three_neurons_training` **represent?**

**Solution:**

Each row of the training data matrix corresponds to a captured waveform of a (possibly different) neuron. The columns of the training data set correspond to the times at which the neuron's waveform was sampled. Here, the times are all relative to the "start" of each waveform.

In the real world, this data would have come from a segmentation (i.e., cutting into pieces) of a long recording of a trace from an electrode embedded in the brain. We have not taught you how to do such segmentation in this course. However, even from 16A, you might have a rough idea of how to do it. You could simply look for time-periods in which there is some activity and then cut those out. In the real-world, the problem of segmentation is done iteratively with understanding what the signals of interest look like. This is because we would ideally like to align all the waveforms to start at the right starting point. (This should remind you a bit of the positioning lab that you did in 16A.) For this knowing the waveforms is useful. These kinds of "chicken-egg" problems are solved iteratively using the same philosophy that is done in k-means clustering (which some of you might have seen in 61A's yelp-like project, although it hasn't run recently.).

We can approximate each waveform in a lower dimensional space using PCA. In your implementation of PCA you will decide how to choose these components.

(c) You will be using the SVD in your PCA function. **What matrices does the SVD return? What are the dimensions of these matrices for the SVD of `three_neurons_training`?**

**Solution:** Our training data `three_neurons_training` was of shape (4398, 32). When we take the SVD our three matrices $U$, $\Sigma$ and $V$ will be of shapes (4398 x 4398), (4398 x 32), and (32 x 32), respectively.

(d) To represent each waveform in a lower dimensional space we want a basis for the time signals of the waveforms of the neurons. To construct this basis we start by taking the SVD (of `three_neurons_training`). **Which of the $U$ and $V$ matrices can we use to construct our desired basis?**

**Solution:** Each of our waveforms is a time series of length 32 and thus can be represented as a 32 x 1 vector. Since our data is along the rows of our data matrix, vectors in matrix $V$ corresponds to our basis.

(e) **Read through `PCA_train` and implement `PCA_project`.** We will use these functions throughout the rest of the notebook, so make sure you understand them.

**Solution:** In .ipynb

(f) **Call `PCA_train` on `two_neurons_training` and plot the 2 principal components.** Note that since the dataset is randomized, you might get different plots every time you run the second code cell of this notebook (the `_make_training_set` function). Note that these components have no real "physical" relationship to the actual shape of the neuron plots. They are a part of a basis, not the representation of exemplar traces.

**Solution:** In .ipynb

(g) Before we project onto our principal components, let us project our data onto 2 random 32 length vectors. **Run the corresponding part in the .ipynb file and comment on the separation of the 2 neuron's distinctive trace shapes in this projected basis.** Do you think that it would be easy to tell them apart just by looking at their projection into these two random dimensions?

**Solution:** We can see that since there is a larger overlap between the presorted points this random basis does not create a good separation.

(h) **Project `two_neurons_test` onto the two principal components found using the SVD and produce a 2D scatter plot in the new basis.** We will also try projecting the presorted data containing 2 neurons so we can see how the model behaves on the 2 neurons. **Comment on the difference between the projections here and part (g) where you projected onto random directions.**

**Solution:** In .ipynb. The projections created using PCA have more separation.

(i) Now we will repeat this process for three principal components. **Do you see a significant improvement with 3 principal components?**

**Solution:** In .ipynb. It is not necessary to use 3 principal components since the 3-D plots show us that there are similar clusters in comparison to 2-D.

**Task 2: Three Neuron Sorting**

(j) In the previous part we sorted 2 neurons using two or three principal components. In this part we are now dealing with 3 neurons. **Replicate what we did in the previous parts by finding and projecting our data on the first two principal components of this three neuron dataset in the .ipynb.**

**Solution:** In .ipynb

(k) Now **call** `PCA_train` **on** `three_neurons_training` **and plot the 3 principal components. Do the same classification process as the 2 neuron data, but now with the 3 neuron data. Compare your model's behavior with that of the presorted data.** The `plot_3D` function will be useful to view the results.

**Solution:** In .ipynb

(l) **How many principal components do you actually need to cluster the 3 neurons?** *(HINT: Think about whether there was an increase in separability between the last two parts.)*

**Solution:** 2 - we can see that the clusters are relatively similar at each Z location, so we only need to worry about their location in the x-y plane.

(m) Now that we know how to project our data onto a basis where it is clearly separable we can classify our points and determine which neuron fired. (This is what we need to know if we want to use the firings of different neurons to build a BMI.)

To classify our points we use the following algorithm:

i. Find centroids: points that represent the average waveform of a particular neuron firing, in the basis of principal components.

ii. Classify each incoming point by assigning it the value of the centroid closest to it (i.e., declare that the neuron waveform that a point represents is the same as the neuron waveform of the centroid the point is closest to).

Since we already have some presorted data, we can use this information to aid in classification. We can calculate our centroids by taking the empirical mean of the presorted data corresponding to a particular neuron firing. **Use this method to find centroids in the corresponding section of the .ipynb file. Then use** `which_neuron` **on the** `two_classified` **data set and count the number of times each neuron fired.**

**Solution:** In the .ipynb file

**Contributors:**

- Anirudh Rengarajan.
- Kunmo Kim.
- Ayan Biswas.
- Vladimir Stojanovic.
- Anant Sahai.
- Michael Danielczuk.
- Nikhil Shinde.
- Ashwin Vangipuram.
- Daniel Abraham.