Homework 1 Solution

EE263 Stanford University, Fall 2017

Due: Wednesday 10/4/17 11:59pm

1. **A simple power control algorithm for a wireless network.** First some background. We consider a network of n transmitter/receiver pairs. Transmitter i transmits at power level p_i (which is positive). The path gain from transmitter j to receiver i is G_{ij} (which are all nonnegative, and G_{ii} are positive). The signal power at receiver i is given by $s_i = G_{ii}p_i$. The noise plus interference power at receiver i is given by

$$q_i = \sigma^2 + \sum_{j \neq i} G_{ij} p_j$$

where $\sigma^2 > 0$ is the self-noise power of the receivers (assumed to be the same for all receivers). The *signal to interference plus noise ratio* (SINR) at receiver i is defined as $S_i = s_i/q_i$. For signal reception to occur, the SINR must exceed some threshold value γ (which is often in the range $3-10$). Various *power control algorithms* are used to adjust the powers p_i to ensure that $S_i \geq \gamma$ (so that each receiver can receive the signal transmitted by its associated transmitter). In this problem, we consider a simple power control update algorithm. The powers are all updated synchronously at a fixed time interval, denoted by $t = 0, 1, 2, \ldots$. Thus the quantities p, q, and S are discrete-time signals, so for example $p_3(5)$ denotes the transmit power of transmitter 3 at time epoch $t = 5$. What we'd like is

$$S_i(t) = s_i(t)/q_i(t) = \alpha\gamma,$$

where $\alpha > 1$ is an SINR safety margin (of, for example, one or two dB). Note that increasing $p_i(t)$ (power of the ith transmitter) increases S_i but decreases all other S_j. A very simple power update algorithm is given by

$$p_i(t+1) = p_i(t)(\alpha\gamma/S_i(t)). \tag{1}$$

This scales the power at the next time step to be the power that would achieve $S_i = \alpha\gamma$, if the interference plus noise term were to stay the same. But unfortunately, changing the transmit powers also changes the interference powers, so it's not that simple! Finally, we get to the problem.

a) Show that the power control algorithm (1) can be expressed as a linear dynamical system with constant input, *i.e.*, in the form

$$p(t+1) = Ap(t) + b,$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are constant. Describe A and b explicitly in terms of σ, γ, α and the components of G.

b) *matlab simulation.* Use matlab to simulate the power control algorithm (1), starting from various initial (positive) power levels. Use the problem data

$$G = \begin{bmatrix} 1 & .2 & .1 \\ .1 & 2 & .1 \\ .3 & .1 & 3 \end{bmatrix}, \qquad \gamma = 3, \qquad \alpha = 1.2, \qquad \sigma = 0.1.$$

Plot S_i and p as a function of t, and compare it to the target value $\alpha\gamma$. Repeat for $\gamma = 5$. Comment briefly on what you observe. *Comment:* You'll understand what you see later in the course.

Solution.

a) The power update rule for a single transmitter can be found by manipulating the definitions given in the problem.

$$\begin{aligned} p_i(t+1) &= \frac{\alpha\gamma p_i(t)}{S_i(t)} = \frac{\alpha\gamma p_i(t) q_i(t)}{s_i(t)} = \frac{\alpha\gamma p_i(t)\left[\sigma^2 + \sum_{j\neq i} G_{ij} p_j(t)\right]}{G_{ii} p_i(t)} \\ &= \frac{\alpha\gamma\left[\sigma^2 + \sum_{j\neq i} G_{ij} p_j(t)\right]}{G_{ii}} \end{aligned}$$

In matrix form the equations look like this:

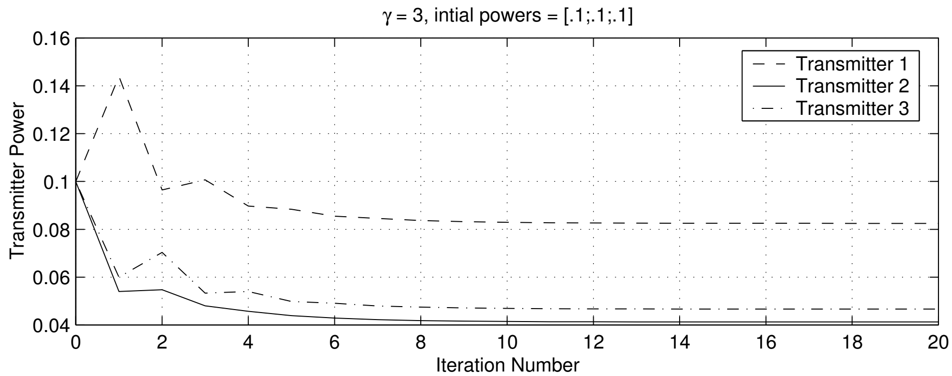
$$\underbrace{\begin{bmatrix} p_1(t+1) \\ p_2(t+1) \\ p_3(t+1) \\ \vdots \\ p_n(t+1) \end{bmatrix}}_{p(t+1)} = \underbrace{\begin{bmatrix} 0 & \frac{\alpha\gamma G_{12}}{G_{11}} & \frac{\alpha\gamma G_{13}}{G_{11}} & \cdots & \frac{\alpha\gamma G_{1n}}{G_{11}} \\ \frac{\alpha\gamma G_{21}}{G_{22}} & 0 & \frac{\alpha\gamma G_{23}}{G_{22}} & \cdots & \frac{\alpha\gamma G_{2n}}{G_{22}} \\ \frac{\alpha\gamma G_{31}}{G_{33}} & \frac{\alpha\gamma G_{32}}{G_{33}} & 0 & \cdots & \frac{\alpha\gamma G_{3n}}{G_{33}} \\ \vdots & & & \ddots & \\ \frac{\alpha\gamma G_{n1}}{G_{nn}} & \frac{\alpha\gamma G_{n2}}{G_{nn}} & \frac{\alpha\gamma G_{n3}}{G_{nn}} & \cdots & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ \vdots \\ p_n(t) \end{bmatrix}}_{p(t)} + \underbrace{\begin{bmatrix} \frac{\alpha\gamma\sigma^2}{G_{11}} \\ \frac{\alpha\gamma\sigma^2}{G_{22}} \\ \frac{\alpha\gamma\sigma^2}{G_{33}} \\ \vdots \\ \frac{\alpha\gamma\sigma^2}{G_{nn}} \end{bmatrix}}_{b}.$$

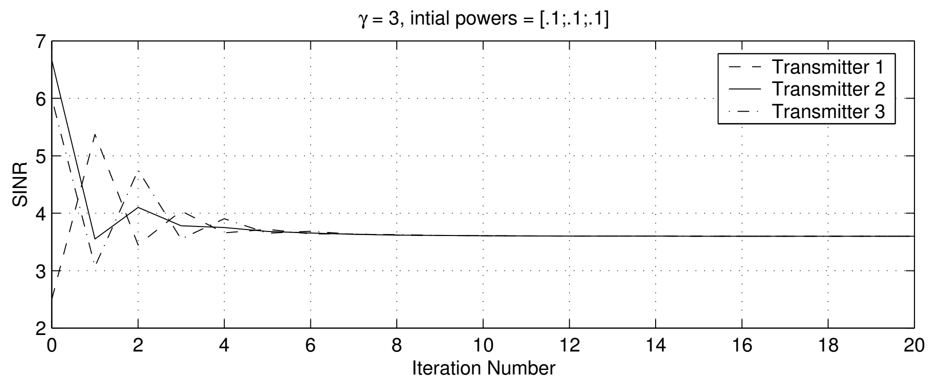
b) The following matlab code simulates the system for $\gamma = 3$ and an initial power of 0.1 for each transmitter.

```
clear all; close all;
G = [1 .2 .1; .1 2 .1; .3 .1 3];% Gain matrix
gamma = 3;                      % minimum SINR
alpha = 1.2;                    % safety margin
sigma = 0.01;                   % Noise power (same for all receivers)
A = zeros(3,3); for i = 1:3
for j = 1:3
if (i~=j)
A(i,j) = alpha*gamma*G(i,j)/G(i,i);
end
end
end
```

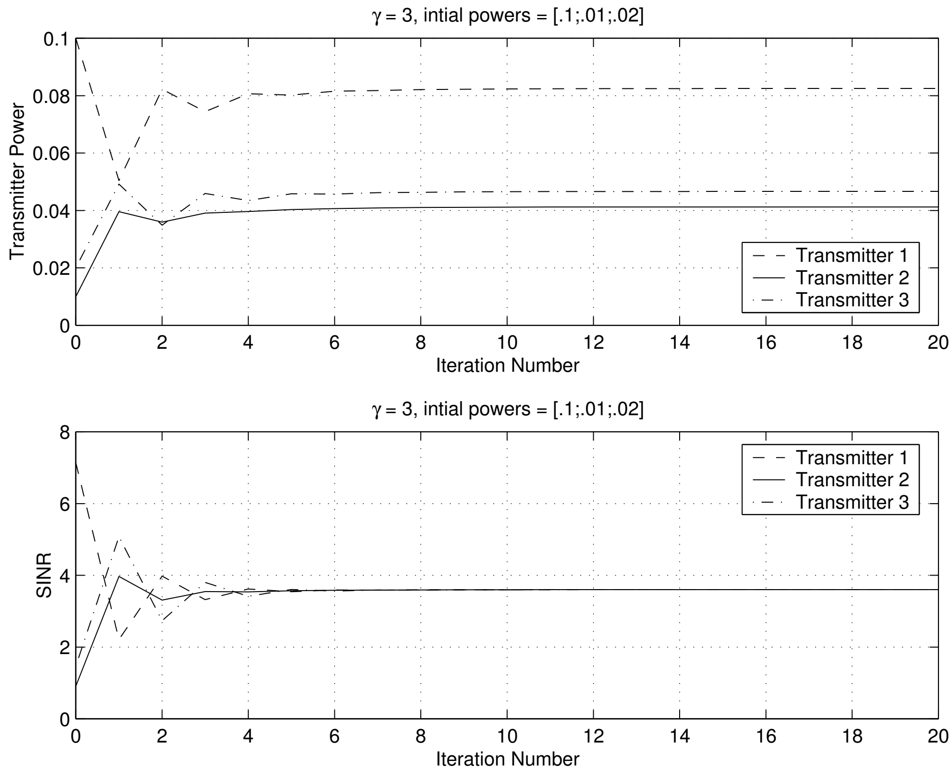
```
b = zeros(3,1); for i = 1:3
b(i) = alpha*gamma*sigma/G(i,i);
end
num_iterations = 20;
p_i = [.1;.1;.1]; % Initialized to p(0)
S = [G(1,1)*p_i(1)/(sigma+G(1,2)*p_i(2)+G(1,3)*p_i(3)); \
G(2,2)*p_i(2)/(sigma+G(2,1)*p_i(1)+G(2,3)*p_i(3)); \
G(3,3)*p_i(3)/(sigma+G(3,1)*p_i(1)+G(3,2)*p_i(2))];
p = p_i;      % matrix to store the powers versus time
for i = 1:num_iterations
p_i = A*p_i+b;
p = [p p_i]; % Find the new powers and save
SINR_current = [G(1,1)*p_i(1)/(sigma+G(1,2)*p_i(2)+G(1,3)*p_i(3)); \
G(2,2)*p_i(2)/(sigma+G(2,1)*p_i(1)+G(2,3)*p_i(3)); \
G(3,3)*p_i(3)/(sigma+G(3,1)*p_i(1)+G(3,2)*p_i(2))];
S = [S SINR_current];
end
figure(1); temp = 0:num_iterations; subplot(2,1,1);
plot(temp,p(1,:),'--', temp,p(2,:),'-',temp,p(3,:),'-.');
xlabel('Iteration Number'); ylabel('Transmitter Power');
title('\gamma = 3, intial powers = [.1;.1;.1]');
legend('Transmitter 1', 'Transmitter 2', 'Transmitter 3',0); grid;
subplot(2,1,2); plot(temp,S(1,:),'--',
temp,S(2,:),'-',temp,S(3,:),'-.'); xlabel('Iteration Number');
ylabel('SINR'); title('\gamma = 3, intial powers = [.1;.1;.1]');
legend('Transmitter 1', 'Transmitter 2', 'Transmitter 3',0); grid;
```

The figure below shows the SINR and transmitter power as a function of iteration number.

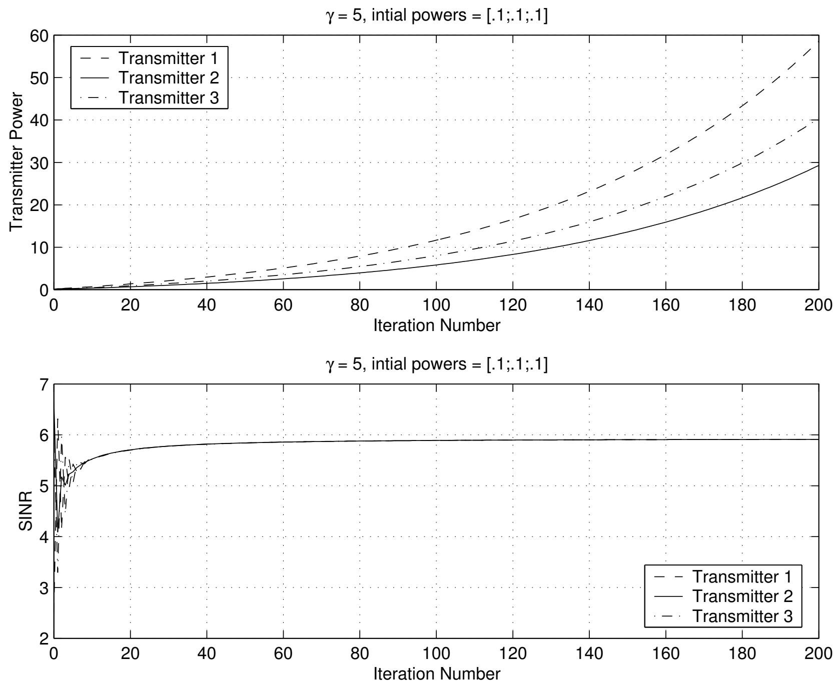


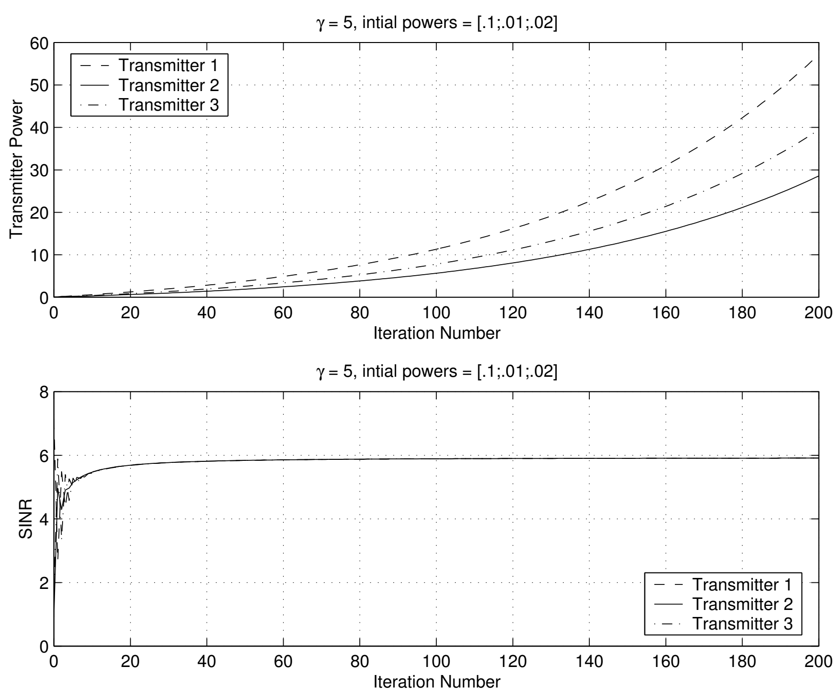


Similar matlab code can be used to try other initial transmitter powers. For example, the simulation shown below used initial transmitter powers of .1, .01, and .02 for the first, second, and third transmitter respectively. In both cases, the final transmitter powers approach .083, .041, and .047. The SINR appoaches $3.6 = \alpha\gamma$. The algorithm appears to work.



Testing the system for $\gamma = 5$ and the same initial conditions (see graphs below) shows that the algorithm does not always succeed. For both initial conditions tried, the transmitter powers grow exponentially. Also, the SINR approaches $5.92 < \alpha\gamma = 6$.





2. Some standard time-series models. A time series is just a discrete-time signal, *i.e.*, a function from $\mathbf{Z}_+$ into $\mathbb{R}$. We think of $u(k)$ as the value of the signal or quantity u at time (or *epoch*) k. The study of time series predates the extensive study of state-space linear systems, and is used in many fields (*e.g.*, econometrics). Let u and y be two time series (input and output, respectively). The relation (or *time series model*)

$$y(k) = a_0 u(k) + a_1 u(k-1) + \cdots + a_r u(k-r)$$

is called a *moving average (MA) model,* since the output at time k is a weighted average of the previous r inputs, and the set of variables over which we average 'slides along' with time. Another model is given by

$$y(k) = u(k) + b_1 y(k-1) + \cdots + b_p y(k-p).$$

This model is called an *autoregressive (AR) model*, since the current output is a linear combination of (*i.e.*, regression on) the current input and some previous values of the output. Another widely used model is the *autoregressive moving average (ARMA) model*, which combines the MA and AR models:

$$y(k) = b_1 y(k-1) + \cdots + b_p y(k-p) + a_0 u(k) + \cdots + a_r u(k-r).$$

Finally, the problem: Express each of these models as a linear dynamical system with input u and output y. For the MA model, use state

$$x(k) = \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-r) \end{bmatrix},$$

and for the AR model, use state

$$x(k) = \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-p) \end{bmatrix}.$$

You decide on an appropriate state vector for the ARMA model. (There are many possible choices for the state here, even with different dimensions. We recommend you choose a state for the ARMA model that makes it easy for you to derive the state equations.) **Remark:** multi-input, multi-output time-series models (*i.e.*, $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^p$) are readily handled by allowing the coefficients a_i, b_i to be matrices.

Solution. In this problem we should find matrices A, B, C and D such that

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

- *Moving average model.* We need to express $x(k+1)$ linearly in terms of $x(k)$ and $u(k)$. We have

$$x(k) = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-r) \end{bmatrix}$$

and therefore

$$x(k+1) = \begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(k+1-r) \end{bmatrix}.$$

Note that

$$x(k+1) = \begin{bmatrix} 0 \\ u(k-1) \\ \vdots \\ u(k+1-r) \end{bmatrix} + \begin{bmatrix} u(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

but

$$\begin{bmatrix} 0 \\ u(k-1) \\ \vdots \\ u(k+1-r) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-r) \end{bmatrix}}_{x(k)}$$

so

$$x(k+1) = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{A} x(k) + \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B} u(k).$$

$y(k)$ should be expressed in terms of $x(k)$ and $u(k)$. This is easy from the relation $y(k) = a_0 u(k) + a_1 u(k-1) + \cdots + a_r u(k-r)$ and we get

$$y(k) = \underbrace{\begin{bmatrix} a_1 & a_2 & \cdots & a_r \end{bmatrix}}_{C} x(k) + \underbrace{a_0}_{D} u(k).$$

(Note: the matrix A with ones on its subdiagonal is called a *shift matrix* because it shifts down the elements of the input vector.)

- *Autoregressive model.* In this case

$$x(k) = \begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-p) \end{bmatrix}$$

so

$$x(k+1) = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k+1-p) \end{bmatrix} = \begin{bmatrix} 0 \\ y(k-1) \\ \vdots \\ y(k+1-p) \end{bmatrix} + \begin{bmatrix} y(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Now

$$\begin{bmatrix} 0 \\ y(k-1) \\ \vdots \\ y(k+1-p) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-p) \end{bmatrix}}_{x(k)}$$

and

$$\begin{bmatrix} y(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \cdots & b_p \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 0 \end{bmatrix} \underbrace{\begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-p) \end{bmatrix}}_{x(k)} + \begin{bmatrix} u(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Thus

$$x(k+1) = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} b_1 & b_2 & \cdots & b_p \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$

or

$$x(k+1) = \underbrace{\begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_p \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{A} x(k) + \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B} u(k)$$

and

$$y(k) = \underbrace{\begin{bmatrix} b_1 & b_2 & \cdots & b_p \end{bmatrix}}_{C} x(k) + \underbrace{1}_{D} u(k).$$

- *Autoregressive moving average model.* One simple choice for $x(k)$ is

$$x(k) = \left[\begin{array}{c} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-r) \\ \hline y(k-1) \\ y(k-2) \\ \vdots \\ y(k-p) \end{array}\right]$$

and therefore

$$x(k+1) = \left[\begin{array}{c} 0 \\ u(k-1) \\ \vdots \\ u(k+1-r) \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] + \left[\begin{array}{c} u(k) \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] + \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline y(k) \\ y(k-1) \\ \vdots \\ y(k+1-p) \end{array}\right].$$

For similar reasons to the previous parts

$$\left[\begin{array}{c} 0 \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k+1-r) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 & \vdots & & & & \vdots \\ 0 & 1 & 0 & \cdots & 0 & \vdots & & & & \vdots \\ \vdots & & \ddots & \ddots & \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \end{array}\right] x(k)$$

and

$$\left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline y(k) \\ y(k-1) \\ y(k-2) \\ \vdots \\ y(k+1-p) \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 0 & 0 & & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & & & \cdots & 0 & \vdots & & & & \vdots \\ 0 & & & \cdots & 0 & 0 & 0 & & \cdots & 0 \\ \hline a_1 & a_2 & a_3 & \cdots & a_r & b_1 & b_2 & b_3 & \cdots & b_p \\ 0 & & & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & & & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots & & \ddots & \ddots & \vdots & \\ 0 & & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{array}\right] x(k) + \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline a_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] u(k).$$

Thus

$$x(k+1) = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 & \vdots & & & & \vdots \\ 0 & 1 & 0 & \cdots & 0 & \vdots & & & & \vdots \\ \vdots & & \ddots & \ddots & \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & & \cdots & 0 \\ \hline a_1 & a_2 & a_3 & \cdots & a_r & b_1 & b_2 & b_3 & \cdots & b_p \\ 0 & & & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & & & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots & & \ddots & \ddots & \vdots & \\ 0 & & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{array}\right] x(k) + \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \hline a_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] u(k)$$

and

$$y(k) = \underbrace{\left[\begin{array}{cccc|cccc} a_1 & a_2 & \cdots & a_r & b_1 & b_2 & \cdots & b_p \end{array}\right]}_{C} x(k) + \underbrace{a_0}_{D} u(k).$$

(Note: it is possible to give state-space models with the state dimension smaller than the ones given here. But our selection of state here makes writing the equations easier.)

3. **Representing linear functions as matrix multiplication.** Suppose that $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is linear. Show that there is a matrix $A \in \mathbb{R}^{m\times n}$ such that for all $x \in \mathbb{R}^n$, $f(x) = Ax$. (Explicitly describe how you get the coefficients A_{ij} from f, and then verify that $f(x) = Ax$ for any $x \in \mathbb{R}^n$.) Is the matrix A that represents f unique? In other words, if $\tilde{A} \in \mathbb{R}^{m\times n}$ is another matrix such that $f(x) = \tilde{A}x$ for all $x \in \mathbb{R}^n$, then do we have $\tilde{A} = A$? Either show that this is so, or give an explicit counterexample.

Solution. Any $x \in \mathbb{R}^n$ can be written as

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 e_1 + x_2 e_2 + \cdots + x_n e_n,$$

where e_i is the ith standard unit vector in $\mathbb{R}^n$. From linearity of f we have

$$f(x) = x_1 f(e_1) + x_2 f(e_2) + \cdots + x_n f(e_n),$$

or in (block) matrix form

$$f(x) = \begin{bmatrix} f(e_1) & f(e_2) & \cdots & f(e_n) \end{bmatrix} x.$$

Therefore, we simply take

$$A := \begin{bmatrix} f(e_1) & f(e_2) & \cdots & f(e_n) \end{bmatrix}.$$

So to determine A we only need to find $f(e_i)$, for $i = 1, \ldots, n$.

Suppose the matrix A is not unique, which means there is another $\tilde{A} \in \mathbb{R}^{m\times n}$ such that $f(x) = \tilde{A}x$. Then $Ax = \tilde{A}x$ or $(A - \tilde{A})x = 0$ for all $x \in \mathbb{R}^n$. When $x = e_i$, $(A - \tilde{A})e_i = 0$ implies that the ith column of $(A - \tilde{A})$ is zero. Repeating this argument for $i = 1, 2, \ldots, n$ proves that *all* columns of $(A - \tilde{A})$ are zero and hence $A = \tilde{A}$. Therefore the choice of A is *unique*.

4. **A mass subject to applied forces.** Consider a unit mass subject to a time-varying force $f(t)$ for $0 \le t \le n$. Let the initial position and velocity of the mass both be zero. Suppose that the force has the form $f(t) = x_j$ for $j - 1 \le t < j$ and $j = 1, \ldots, n$. Let y_1 and y_2 denote, respectively, the position and velocity of the mass at time $t = n$.

   a) Find the matrix $A \in \mathbb{R}^{2\times n}$ such that $y = Ax$.

   b) For $n = 4$, find a sequence of input forces $x_1, \ldots, x_n$ that moves the mass to position 1 with velocity 0 at time n.

Solution. Let $p(t)$ and $v(t)$ denote, respectively, the position and velocity of the mass at time t.

a) The velocity is the integral of the applied force:

$$\begin{aligned}
v(t) &= v(0) + \int_0^t f(\tau)\,d\tau \\
&= v(0) + \sum_{j=1}^{\lfloor t\rfloor} \int_{j-1}^{j} f(\tau)\,d\tau + \int_{\lfloor t\rfloor}^{t} f(\tau)\,d\tau \\
&= v(0) + \sum_{j=1}^{\lfloor t\rfloor} \int_{j-1}^{j} x_j\,d\tau + \int_{\lfloor t\rfloor}^{t} x_{\lfloor t\rfloor+1}\,d\tau \\
&= v(0) + \sum_{j=1}^{\lfloor t\rfloor} (\tau x_j)\big|_{\tau=j-1}^{\tau=j} + (\tau x_{\lfloor t\rfloor+1})\big|_{\tau=\lfloor t\rfloor}^{\tau=t} \\
&= v(0) + \sum_{j=1}^{\lfloor t\rfloor} x_j + (t-\lfloor t\rfloor)x_{\lfloor t\rfloor+1}.
\end{aligned}$$

In particular, because the mass is initially at rest (that is, $v(0)=0$), the final velocity is

$$y_2 = v(n) = \sum_{j=1}^{n} x_j.$$

Similarly, the position is the integral of the velocity:

$$\begin{aligned}
p(t) &= p(0) + \int_0^t v(\tau)\,d\tau \\
&= p(0) + \int_0^t (v(0) + (v(\tau)-v(0)))\,d\tau \\
&= p(0) + v(0)t + \int_0^t (v(\tau)-v(0))\,d\tau \\
&= p(0) + v(0)t + \sum_{j=1}^{\lfloor \tau\rfloor} \int_{j-1}^{j} (v(\tau)-v(0))\,d\tau + \int_{\lfloor t\rfloor}^{t} (v(\tau)-v(0))\,d\tau \\
&= p(0) + v(0)t + \sum_{j=1}^{\lfloor t\rfloor} \int_{j-1}^{j} \left(\sum_{k=1}^{\lfloor \tau\rfloor} x_k + (\tau-\lfloor \tau\rfloor)x_{\lfloor \tau\rfloor+1}\right) d\tau \\
&\qquad + \int_{\lfloor t\rfloor}^{t} \left(\sum_{k=1}^{\lfloor \tau\rfloor} x_k + (\tau-\lfloor \tau\rfloor)x_{\lfloor \tau\rfloor}+1\right) d\tau \\
&= p(0) + v(0)t + \sum_{j=1}^{\lfloor t\rfloor} \int_{j-1}^{j} \left(\sum_{k=1}^{j-1} x_k + (\tau-(j-1))x_j\right) d\tau
\end{aligned}$$

$$
\begin{aligned}
&\quad + \int_{\lfloor t\rfloor}^{t} \left(\sum_{k=1}^{\lfloor t\rfloor} x_k + (\tau - \lfloor t\rfloor) x_{\lfloor t\rfloor+1} \right) d\tau \\
&= p(0) + v(0)t + \sum_{k=1}^{\lfloor t\rfloor} \left(\sum_{k=1}^{j-1} \tau x_k + \frac{1}{2}(\tau - (j-1))^2 x_j \right) \Bigg|_{\tau=j-1}^{\tau=j} \\
&\quad + \left(\sum_{k=1}^{\lfloor t\rfloor} \tau x_k + \frac{1}{2}(\tau - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \right) \Bigg|_{\tau=\lfloor t\rfloor}^{\tau=t} \\
&= p(0) + v(0)t + \sum_{j=1}^{\lfloor t\rfloor} \left(\sum_{k=1}^{j-1} x_k + \frac{1}{2} x_j \right) + \left(\sum_{k=1}^{\lfloor t\rfloor} x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \right) \\
&= p(0) + v(0)t + \sum_{j=1}^{\lfloor t\rfloor} \sum_{k=1}^{j-1} x_k + \sum_{j=1}^{\lfloor t\rfloor} \frac{1}{2} x_j + \sum_{k=1}^{\lfloor t\rfloor} x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \\
&= p(0) + v(0)t + \sum_{k=1}^{\lfloor t\rfloor} \sum_{j=k+1}^{\lfloor t\rfloor} x_k + \sum_{k=1}^{\lfloor t\rfloor} \frac{1}{2} x_k + \sum_{k=1}^{\lfloor t\rfloor} x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \\
&= p(0) + v(0)t + \sum_{k=1}^{\lfloor t\rfloor} (\lfloor t\rfloor - k) x_k + \sum_{k=1}^{\lfloor t\rfloor} \frac{1}{2} x_k + \sum_{k=1}^{\lfloor t\rfloor} x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \\
&= p(0) + v(0)t + \sum_{k=1}^{\lfloor t\rfloor} \left((\lfloor t\rfloor - k) + \frac{1}{2} + (t - \lfloor t\rfloor) \right) x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1} \\
&= p(0) + v(0)t + \sum_{k=1}^{\lfloor t\rfloor} \left(t - k + \frac{1}{2} \right) x_k + \frac{1}{2}(t - \lfloor t\rfloor)^2 x_{\lfloor t\rfloor+1}.
\end{aligned}
$$

In particular, because the mass is initially at rest at the origin (that is, $p(0) = 0$ and $v(0) = 0$), the final position is

$$y_1 = p(n) = \sum_{j=1}^{n} \left(n - k + \tfrac{1}{2}\right) x_j.$$

Thus, we obtain the following system of linear equations:

$$
\begin{aligned}
y_1 &= \sum_{j=1}^{n} (n - j + \tfrac{1}{2}) x_j, \\
y_2 &= \sum_{j=1}^{n} x_j.
\end{aligned}
$$

Since A_{ij} gives the coefficient of x_j in our expression for y_i, we have that

$$A_{1j} = n - j + \tfrac{1}{2} \qquad \text{and} \qquad A_{2j} = 1, \qquad j = 1, \ldots, n.$$

More concretely, we have that

$$A = \begin{bmatrix} n-\frac{1}{2} & n-\frac{3}{2} & \cdots & \frac{3}{2} & \frac{1}{2} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}.$$

b) We want to solve the following system of linear equations:

$$\begin{bmatrix} \frac{7}{2} & \frac{5}{2} & \frac{3}{2} & \frac{1}{2} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This system is underdetermined, and has infinitely many solutions. Suppose we choose $x_2 = x_3 = 0$. Then, we are left with the system

$$\begin{bmatrix} \frac{7}{2} & \frac{1}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The second equation implies that $x_4 = -x_1$. Then, the first equation becomes

$$\tfrac{7}{2}x_1 + \tfrac{1}{2}x_4 = \tfrac{7}{2}x_1 - \tfrac{1}{2}x_1 = 3x_1 = 1.$$

Solving this equation, we find that $x_1 = \frac{1}{3}$. Substituting this value into our expression for x_4 gives $x_4 = -x_1 = -\frac{1}{3}$. Thus, one sequence of input forces that moves the mass to position 1 with velocity 0 at time n is

$$x = \begin{bmatrix} \frac{1}{3} \\ 0 \\ 0 \\ -\frac{1}{3} \end{bmatrix}.$$

5. Express the following statements in matrix language. You can assume that all matrices mentioned have appropriate dimensions. Here is an example: "Every column of C is a linear combination of the columns of B" can be expressed as "$C = BF$ for some matrix F".

There can be several answers; one is good enough for us.

a) Suppose Z has n columns. For each i, row i of Z is a linear combination of rows $i, \ldots, n$ of Y.

b) W is obtained from V by permuting adjacent odd and even columns (*i.e.*, 1 and 2, 3 and 4, ...).

c) Each column of P makes an acute angle with each column of Q.

d) Each column of P makes an acute angle with the corresponding column of Q.

e) The first k columns of A are orthogonal to the remaining columns of A.

Solution.

a) $Z = UY$, where U is upper triangular, *i.e.*, $U_{ij} = 0$ for $i > j$.

b) $W = VS$, where S is the odd-even switch matrix, defined as

$$S = \begin{bmatrix} e_2 & e_1 & e_4 & e_3 & \cdots & e_m & e_{m-1} \end{bmatrix}.$$

c) All entries of the matrix $P^\mathsf{T} Q$ are positive.

d) The diagonal entries of the matrix $P^\mathsf{T} Q$ are positive.

e) $A^\mathsf{T} A$ has the block diagonal form

$$A^\mathsf{T} A = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix},$$

where $B_{11} \in \mathbb{R}^{k \times k}$.

6. Interest and amortization. The scalar first order difference equation

$$y(k+1) = ay(k) + b, \qquad y \in \mathbb{R} \tag{2}$$

arises in many important applications, and its analysis motivates much of the general theory of difference equations. The equation is linear, has a constant coefficient a, and a constant forcing term b. The general solution to this equation is easily deduced. The most straightforward solution procedure is to determine successive values recursively.

a) Assuming $y(0) = C$, use a recursive method to find the general solution to eqn. (2). Separately consider the cases of $a = 1$ and $a \neq 1$ to simplify your solution.

b) When one borrows money at an interest rate i, the total debt increases just as would the balance in a bank account paying the same interest rate. Amortization is a method for repaying an initial debt, including interest and original principal, by a series of payments (usually at equal intervals and of equal magnitude). Assuming an initial debt of $d(0) = D$ with annual interest rate i, and a payment of B made at the end of each year, write a first order difference equation describing the dynamics of $d(k)$.

c) If it is desired to amortize the debt so that it is paid off at the end of n years, it is necessary to select B such that $d(n) = 0$. Use the general solution you found in part (a) to find the amortization formula for B.

Solution.

a) Assuming $y(0) = C$ we immediately get the following successive values for $y(k)$:

$$\begin{aligned} y(0) &= C \\ y(1) &= ay(0) + b = aC + b \\ y(2) &= ay(1) + b = a^2C + ab + b \\ y(3) &= a^3C + a^2b + ab + b \end{aligned}$$

The general term is:

$$y(k) = a^k C + (a^{k-1} + a^{k-2} + \cdots + a + 1)b$$

For $a = 1$, the expression reduces simply to:

$$y(k) = C + kb$$

For $a \neq 1$, the expression can be simplified by collapsing the geometric series, using:

$$1 + a + a^2 + \cdots + a^{k-1} = \frac{1 - a^k}{1 - a}$$

Therefore, the desired solution in closed-form would be:

$$y(k) = a^k C + \frac{1 - a^k}{1 - a} b$$

When $a \neq 1$, another way of displaying the general solution above is sometimes more convenient:

$$y(k) = Da^k + \frac{b}{1 - a}$$

where D is related to the earlier constant C by $D = C - [b/(1 - a)]$. In this form, it is apparent that the solution function is the sum of two elementary functions: the constant function $b/(1 - a)$ and the geometric sequence Da^k.

b) If a payment B is made at the end of each year, the total debt will satisfy the following first order difference equation:

$$d(k + 1) = (1 + i)d(k) - B$$

c) The general solution developed in part (a) implies:

$$d(n) = D(1 + i)^n - \frac{1 - (1 + i)^n}{1 - (1 + i)} B$$

Setting $d(n) = 0$ yields:

$$B \frac{1 - (1 + i)^n}{-i} = D(1 + i)^n$$

which simplifies to the standard amortization formula:

$$B = \frac{iD}{1 - (1 + i)^{-n}}$$

7. Proof of Cauchy-Schwarz inequality. You will prove the Cauchy-Schwarz inequality.

a) Suppose $a \geq 0$, $c \geq 0$, and for all $\lambda \in \mathbb{R}$, $a + 2b\lambda + c\lambda^2 \geq 0$. Show that $|b| \leq \sqrt{ac}$.

b) Given v, $w \in \mathbb{R}^n$ explain why $(v + \lambda w)^\mathsf{T}(v + \lambda w) \geq 0$ for all $\lambda \in \mathbb{R}$.

c) Apply (a) to the quadratic resulting when the expression in (b) is expanded, to get the Cauchy-Schwarz inequality:

$$|v^\mathsf{T} w| \leq \sqrt{v^\mathsf{T} v}\sqrt{w^\mathsf{T} w}.$$

d) When does equality hold?

Solution.

a) If the equation $a + 2b\lambda + c\lambda^2 = 0$ has no real roots (with odd degree) for λ then it never changes sign for $\lambda \in \mathbb{R}$. Since a and c are positive, the value of $a + 2b\lambda + c\lambda^2$ is non-negative at zero and infinity respectively, so the necessary and sufficient condition for $a + 2b\lambda + c\lambda^2$ to be non-negative is the condition for which $a + 2b\lambda + c\lambda^2 = 0$ has no (simple) real roots for λ. Therefore we should have

$$4b^2 - 4ac \leq 0$$

and since $a, c \geq 0$ this gives $|b| \leq \sqrt{ac}$.

b) Clearly $(v + \lambda w)^\mathsf{T}(v + \lambda w) = \|v + \lambda w\|^2$, and the norm of any vector (here $v + \lambda w$) is non-negative. Therefore $(v + \lambda w)^\mathsf{T}(v + \lambda w) \geq 0$ and equality holds when $v + \lambda w = 0$ or $v = -\lambda w$ (*i.e.*, v is a scalar multiple of w.)

c) From the previous part we know that $(v + \lambda w)^\mathsf{T}(v + \lambda w) \geq 0$ and since

$$(v + \lambda w)^\mathsf{T}(v + \lambda w) = v^\mathsf{T} v + 2(v^\mathsf{T} w)\lambda + (w^\mathsf{T} w)\lambda^2,$$

applying the result of problem (a) with $a = v^\mathsf{T} v \geq 0$, $b = v^\mathsf{T} w$ and $c = w^\mathsf{T} w \geq 0$ gives

$$|v^\mathsf{T} w| \leq \sqrt{v^\mathsf{T} v}\sqrt{w^\mathsf{T} w}.$$

d) According to (b), equality holds if and only if v is a scalar multiple of w. If v is a positive scalar multiple of w, then $v^\mathsf{T} w > 0$ so $|v^\mathsf{T} w| = v^\mathsf{T} w$ and we have $v^\mathsf{T} w = \sqrt{v^\mathsf{T} v}\sqrt{w^\mathsf{T} w}$. If v is a negative scalar multiple of w, then $v^\mathsf{T} w < 0$ and $|v^\mathsf{T} w| = -v^\mathsf{T} w$, so $v^\mathsf{T} w = -\sqrt{v^\mathsf{T} v}\sqrt{w^\mathsf{T} w}$.

8. Right inverses. This problem concerns the specific matrix

$$A = \begin{bmatrix} -1 & 0 & 0 & -1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

This matrix is full rank (*i.e.*, its rank is 3), so there exists at least one right inverse. In fact, there are many right inverses of A, which opens the possibility that we can seek right inverses that in addition have other properties. For each of the cases below, either find a specific matrix $B \in \mathbb{R}^{5\times 3}$ that satisfies $AB = I$ and the given property, or explain why there is no such B. In cases where there is a right inverse B with the required property, you must briefly explain how you found your B. You must also attach a printout of some Julia scripts that show the verification that $AB = I$. (We'll be very angry if we have to type in your 5×3 matrix into matlab to check it.) When there is no right inverse with the given property, briefly explain why there is no such B.

a) The second row of B is zero.

b) The nullspace of B has dimension one.

c) The third column of B is zero.

d) The second and third rows of B are the same.

e) B is upper triangular, *i.e.*, $B_{ij} = 0$ for $i > j$.

f) B is lower triangular, *i.e.*, $B_{ij} = 0$ for $i < j$.

Solution.

a) The second row of B is zero. This means that the second column of A isn't used in forming AB. Let $\tilde{A}$ be the matrix A with its second column removed, and let $\tilde{B}$ denote the matrix B with its second row (which is supposed to be zero) removed. We have $\tilde{A}\tilde{B} = AB = I$, so $\tilde{B}$ is a right inverse of $\tilde{A}$. There is such a matrix if and only if $\tilde{A}$ is full rank, which it is. We can take $\tilde{B} = \tilde{A}^\mathsf{T}(\tilde{A}\tilde{A}^\mathsf{T})^{-1}$. Finally to construct B we simply insert a zero second row, moving rows 2, 3, 4 down by one. This gives the matrix

$$B = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 1 \end{bmatrix}.$$

There are other possible choices as well.

b) The nullspace of B has dimension one. This means that B has rank 2, so the rank of AB is at most 2, which rules out the possibility that $AB = I$. So this is impossible.

c) The third column of B is zero. This implies B has a nullspace with dimension at least one, so by part (b) above, this is impossible too.

d) The second and third rows of B are the same. Let $\tilde{B}$ denote B with one of the (identical) rows 2 and 3 deleted. Then we have $AB = \tilde{A}\tilde{B}$, where $\tilde{A}$ is obtained from the matrix A by replacing its second column with the sum of its second and third columns, and deleting its third column. Thus, we need to find a right inverse for $\tilde{A}$, provided it is full rank. It is, so we can take $\tilde{B} = \tilde{A}^\mathsf{T}(\tilde{A}\tilde{A}^\mathsf{T})^{-1}$. Finally to construct B we simply insert a second copy of the second row of $\tilde{B}$ as a new third row. This gives

$$B = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 1 \end{bmatrix}.$$

This matrix also happens to be the pseudo-inverse of A, $B = A^\mathsf{T}(AA^\mathsf{T})^{-1}$, and some of you noticed this immediately and used the pseudo-inverse to answer this question. That's a fine answer; it was our mistake to choose A so that the pseudo-inverse satisfied this condition. In general, of course, it would not.

e) B is upper triangular, *i.e.*, $B_{ij} = 0$ for $i > j$. If B is upper triangular, then it has the form

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix},$$

where $\tilde{B}$ is square and upper triangular. If $AB = I$, then $\tilde{A}\tilde{B} = I$, where $\tilde{A}$ is the matrix formed from the first 3 columns of A. Thus we have $\tilde{A} = \tilde{B}^{-1}$. But the inverse of an upper triangular matrix is also upper triangular, so unless $\tilde{A}$ is upper triangular (and it isn't, in this case), we can't possibly have $\tilde{A}\tilde{B} = I$. So there is no such B in this case.

f) B is lower triangular, *i.e.*, $B_{ij} = 0$ for $i < j$. Let's label the columns of B as

$$b_1, \qquad b_2 = \begin{bmatrix} 0 \\ \tilde{b}_2 \end{bmatrix}, \qquad b_3 = \begin{bmatrix} 0 \\ 0 \\ \tilde{b}_3 \end{bmatrix},$$

where $\tilde{b}_2 \in \mathbb{R}^4$ and $\tilde{b}_3 \in \mathbb{R}^3$. To say that $AB = I$ is the same as saying that $Ab_1 = e_1$, $Ab_2 = e_2$, and $Ab_3 = e_3$, where e_1, e_2, e_3 are the unit vectors. We can solve these equations separately. The first equation is easy; the second we reduce to $\tilde{A}\tilde{b}_2 = e_2$, where here $\tilde{A}$ is A with its first column removed. The third is handled similarly. These equations do have a solution; we get

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Another way: we set it up as a set of 9 linear equations (one for each entry of $AB = I$) in $5+4+3 = 12$ variables. The variables are the first column of B (with 5 entries), the nonzero part of the second column of B (with 4 entries), and the nonzero part of the third second column of B (with 3 entries). We then attempt to solve these 9 equations in 12 variables. Some equations immediately give us the B matrix coefficients, while the others can be solved by inspection to obtain a rather simple matrix

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$