Homework 8

This homework is a bit longer, so be sure to start on it early.

1. **Portfolio optimization.** In this problem you will optimize a set of holdings to minimize risk for various average returns. The file `portfolio_optimization.jl` contains `train_returns` and `test_returns` which are return matrices for 20 assets over 2000 days and 500 days, respectively. Using `train_returns` find asset allocation weights for 4 portfolios that minimize risk for annualized returns of 5%, 10%, 20%, and 40%. Report the annualized return and risk on the training and test sets for all 4 portfolios, and comment briefly. Plot the cumulative value for each portfolio over time, starting from the conventional initial investment of $10000, for both the train and test sets of returns. You can use the `cumprod` function to compute the product $V_1(1 + r_1^Tw)(1 + r_2^Tw)\cdots(1 + r_T^Tw)$. For each of the 4 portfolios, report the leverage, defined as $|w_1| + \cdots + |w_n|$. (Several other definitions of leverage are used.) This number is always at least one, and it is exactly one only if the portfolio has no short positions.

2. **Equalizer design from training message.** We consider a communication system, with message to be sent given by an $N$-vector $s$, whose entries are 0 or 1, and received signal $y$, where $y = c \ast s$, where $c$ is an $n$-vector, the channel impulse response. The receiver applies equalization to the received signal, which means that it computes $\tilde{y} = h \ast y = h \ast c \ast s$, where $h$ is an $n$-vector, the equalizer impulse response. The receiver then estimates the original message using $\hat{s} = \text{round}(\tilde{y}_{1:N})$. This works well if $h \ast c \approx e_1$.

In some cases, the channel impulse response $c$ can be measured. Once we know $c$, we can design or choose $h$, for example, by least squares. In many other cases (especially when the channel is wireless), it’s not practical to measure $c$, and in addition, $c$ changes over time. In those cases a different method for choosing $h$ is used, which we explore in this problem.

The method directly designs $h$ without estimating or measuring $c$. The sender first sends a message that is known to the receiver, called the **training message**, $s^{tr}$. (From the point of view of communications, this is wasted transmission, and is called overhead.) The receiver receives the signal $y^{tr} = c \ast s^{tr}$ from the training message, and then chooses $h$ to minimize $\|(h \ast y^{tr})_{1:N} - s^{tr}\|^2$. In practice, this equalizer is used until the bit error rate increases (which means the channel has changed), at which point another training message is sent.

Finally we get to the problem. The file `channel_equalizer.jl` contains the training message $s^{tr}$, the value of $n$, and the signal received from the training message, $y^{tr}$. Your first task is to design an equalizer $h$ using this data, and plot it.
The file also includes the received signal $y$ from a message $s$ that is sent. First, round $y$ to get an estimate of the message, and print it as a sting. Then estimate the message $s$ using your equalizer, and print it as text. You'll know when your equalizer is working.

Hints.

- In Julia you can round to 0 or 1 using
  \[
s\text{\_round} = \text{int}(y \geq 0.5);
\]
- You can turn a Boolean vector, a vector with entries only 0 or 1, into a string using the function `binary2string`. (While not needed for this problem, the function `string2binary` converts a string to a Boolean vector.)

3. Closest solution to a given point. Suppose the wide matrix $A$ has independent rows. Find an expression for the point $x$ that is closest to a given vector $\tilde{x}$ (i.e., minimizes $\|x - \tilde{x}\|^2$) among all vectors that satisfy $Ax = b$.

Remark. This problem comes up when $x$ is some set of inputs to be found, $Ax = b$ represents some set of requirements, and $\tilde{x}$ is some nominal value of the inputs. For example, when the inputs represent actions that are re-calculated each day (say, because $b$ changes every day), $\tilde{x}$ might be yesterday’s action, and the today’s action $x$ found as above gives the least change from yesterday’s action, subject to meeting today’s requirements.

4. Rendezvous. The dynamics of two vehicles, at sampling times $t = 1, 2, \ldots$, are given by
  \[
x_{t+1} = Ax_t + Bu_t, \quad z_{t+1} = Az_t + Bv_t
  \]
where the $n$-vectors $x_t$ and $z_t$ are the states of vehicles 1 and 2, and the $m$-vectors $u_t$ and $v_t$ are the inputs of vehicles 1 and 2. The $n \times n$ matrix $A$ and the $n \times m$ matrix $B$ are known.

The position of vehicle 1 at time $t$ is given by $Cx_t$, where $C$ is a known $2 \times n$ matrix. Similarly, the position of vehicle 2 at time $t$ is given by $Cz_t$.

The initial states of the two vehicles are fixed and given:
  \[
x_1 = x_{\text{start}}, \quad z_1 = z_{\text{start}}.
  \]

We are interested in finding a sequence of inputs for the two vehicles over the time interval $t = 1, \ldots, T-1$ so that they rendezvous at time $t = T$, i.e., $x_T = z_T$. You can select the inputs to the two vehicles,
  \[
u_1, u_2, \ldots, u_{T-1}, \quad v_1, v_2, \ldots, v_{T-1}.
  \]
Among choices of the sequences $u_1, \ldots, u_{T-1}$ and $v_1, \ldots, v_{T-1}$ that satisfy the rendezvous condition, we want the one that minimizes the weighted sum of squares of the
two vehicle inputs,

\[ J = \sum_{t=1}^{T-1} \|u_t\|^2 + \lambda \sum_{t=1}^{T-1} \|v_t\|^2, \]

where \( \lambda > 0 \) is a parameter that trades off the two objectives.

(a) Explain how to find the sequences \( u_1, \ldots, u_{T-1} \) and \( v_1, \ldots, v_{T-1} \) that minimize \( J \) while satisfying the rendezvous condition by solving a constrained least-squares problem.

(b) The problem data \( A, B, C, x_{start}, \) and \( z_{start} \) are defined in \texttt{rendezvous.jl}. Use \texttt{LinearLeastSquares} to find \( u_1, \ldots, u_{T-1} \) and \( v_1, \ldots, v_{T-1} \) for \( \lambda = 0.1, \lambda = 1, \) and \( \lambda = 10. \) Plot the vehicle trajectories (i.e., their positions) for each \( \lambda \) using the plotting code in \texttt{rendezvous.jl}.

(c) Give a simple expression for \( x_T \) in the limit where \( \lambda \to \infty \) and for \( z_T \) in the limit where \( \lambda \to 0. \) Assume that for any \( w \in \mathbb{R}^n \) there exist sequences \( u_1, \ldots, u_{T-1} \) and \( v_1, \ldots, v_{T-1} \) such that the rendezvous constraints are satisfied with \( w = z_T = x_T. \)

5. A linear regulator for a linear dynamical system. We consider a linear dynamical system with dynamics \( x_{t+1} = Ax_t + Bu_t, \) where the \( n \)-vector \( x_t \) is the state at time \( t \) and the \( m \)-vector \( u_t \) is the input at time \( t. \) We assume that \( x = 0 \) represents the desired operating point; the goal is to find an input sequence \( u_1, \ldots, u_{T-1} \) and \( v_1, \ldots, v_{T-1} \) that results in \( x_T = 0, \) given the initial state \( x_1. \) Choosing an input sequence that takes the state to the desired operating point at time \( T \) is called regulation.

(a) Find an explicit formula for the sequence of inputs that yields regulation, and minimizes \( \|u_1\|^2 + \cdots + \|u_{T-1}\|^2, \) in terms of \( A, B, T, \) and \( x_1. \) This control is called the minimum energy regulator.

\[ C = \begin{bmatrix} B & AB & \cdots & A^{T-2}B \end{bmatrix}, \]

and the vector \( u = (u_{T-1}, u_{T-2}, \ldots, u_1) \) (which is the input sequence in reverse order). You do not need to expand expressions involving \( C, \) such as \( CC^T \) or \( C^t, \) in terms of \( A \) and \( B; \) you are welcome to simply give your answers using \( C. \) You may assume that \( C \) is wide and has independent rows.

(b) Show that \( u_t \) (the \( t \)th input in the sequence found in part (a)) can be expressed as \( u_t = K_t x_1, \) where \( K_t \) is an \( m \times n \) matrix. Show how to find \( K_t \) from \( A \) and \( B. \)

(But feel free to use the matrix \( C \) in your answer.)

\[ \text{Hint. Your expression for } K_t \text{ can include submatrices of } C \text{ or } C^t. \]

(c) A constant linear regulator. A very common regulator strategy is to simply use \( u_t = K_1 x_t \) for all \( t, t = 1, 2, 3, \ldots. \) This is called a (constant) linear regulator, and \( K_1 \) is called the state feedback gain (since it maps the current state into the
control input). Using this control strategy can be interpreted as always carrying out the first step of minimum energy control, as if we were going to steer the state to zero $T$ steps in the future. This choice of input does not yield regulation in $T$ steps, but it typically achieves asymptotic regulation, which means that $x_t \to 0$ as $t \to \infty$.

Find the state feedback gain $K_1$ for the specific system with

$$A = \begin{bmatrix} 1.003 & 0 & -0.008 \\ 0.005 & .997 & 0 \\ 0 & 0.005 & 1.002 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 4 & 5 \\ 6 & 2 \end{bmatrix},$$

using $T = 10$. You may find the code in regulation.jl useful.

(d) Simulate the system given in part (c) from several choices of initial state $x_1$, under two conditions: open-loop, which means $u_t = 0$, and closed-loop, which means $u_t = K_1 x_t$, where $K_1$ is the state feedback gain found in part (c). Use regulation.jl to plot $x$.

6. **Price optimization.** We have $n$ different products, with (positive) prices given by the $n$-vector $p$. The prices, which we can control, are held constant over some period, say, a day. The (positive) manufacturing costs of the products are given by the $n$-vector $c$, which is known and constant. The (positive) demands for the products over the day is given by the $n$-vector $d$. The profit made on product $i$ over the day is $d_i(p_i - c_i)$. The total profit for that day is $R = d^T(p - c)$. The demand in any particular day varies, but it is thought to be (approximately) a function of the prices.

The nominal prices are given by the $n$-vector $p^{\text{nom}}$. You can think of these as the prices that have been charged in the past for the products. The nominal demand is the $n$-vector $d^{\text{nom}}$. This is the average value of the demand, when the prices are set to $p^{\text{nom}}$. (The actual daily demand fluctuates around the value $d^{\text{nom}}$.) You know both $p^{\text{nom}}$ and $d^{\text{nom}}$. The nominal profit is $R^{\text{nom}} = (d^{\text{nom}})^T(p^{\text{nom}} - c)$. This is the average value of the daily profit with the nominal prices; the actual daily profit varies around this value.

The standard model that connects prices and demands uses the (fractional) deviations from the nominal values. We define $\delta^p$ and $\delta^d$ as the (vectors of) relative price change and demand change:

$$\delta^p_i = \frac{p_i - p_i^{\text{nom}}}{p_i^{\text{nom}}}, \quad \delta^d_i = \frac{d_i - d_i^{\text{nom}}}{d_i^{\text{nom}}}, \quad i = 1, \ldots, n.$$  

So $\delta^p_3 = +0.05$ means that the price for product 3 has been increased by 5% over its nominal value, and $\delta^d_5 = -0.04$ means that the demand for product 5 in some day is 4% below its nominal value. The standard demand model is $\delta^d = E \delta^p$, where $E$ is the $n \times n$ elasticity matrix. We will assume that this matrix is known, or has been estimated from prior data.
Your job is to choose new product prices, each no more than 5\% from the nominal values, so as to maximize your profit. This means you can choose $\delta p$ as any vector that satisfies $|\delta p_i| \leq 0.05$ for $i = 1, \ldots, n$.

(a) Derive an expression for the profit $R$ in terms of the price changes $\delta p$, assuming the demand model $\delta d = E \delta p$ holds exactly. You might want to use the notation $P = \text{diag}(p^{\text{nom}})$, $D = \text{diag}(d^{\text{nom}})$.

(b) Simplify the expression found in part (a) by dropping any term that involves the product of two or more price changes. (This is justified since these will be small.) Your simplified expression for the profit should be an affine function of $\delta p$, i.e., $R = a^T \delta p + b$.

(c) Now explain how to find the choice of $\delta p_i$ that maximizes the profit expression from part (b), subject to each of these satisfying $|\delta p_i| \leq 0.05$. In other words, how should you change each price, by up to 5\%, so as to maximize profit, assuming the demand model is correct.

(d) Find the specific $\delta p$ for the particular problem with data given in price_optimization.jl, and give the new profit you predict with the changed price. Simulate the effect of your proposed price changes to verify that the price changes lead to an increase in (average) profit. To do this you will use the function $\text{demand}(p)$, also given in the file. This function simulates the demands of the products, as a function of the prices. The function includes natural variation of the demand, so you will not get the same demand if you call the function twice with the same prices. Use this function to simulate 100 days of demand at the nominal price, and also 100 days with the new prices. Plot the profits for these 200 days. Report the average profit over the first 100 days (with nominal prices) and the second 100 days (with your new prices). Did your new prices lead to an increase in profits? Was the change close to your prediction?

Remark. The function $\text{demand}(p)$ is just a simulation, but it uses a realistic model of demand (including fluctuation of demand). You’re welcome to look inside the function, but you cannot use anything you learn by looking inside it in any of the problems, or your code. (We will be checking for this.)