Homework 1

EE 263 Stanford University Summer 2019

Due: July 03, 2019 11:59 PM

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 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) *simulation* Simulate the power control algorithm, 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.

2. State equations for a linear mechanical system. The equations of motion of a lumped mechanical system undergoing small motions can be expressed as

$$M\ddot{q} + D\dot{q} + Kq = f$$

where $q(t) \in \mathbb{R}^k$ is the vector of deflections, M, D, and K are the *mass*, *damping*, and *stiffness* matrices, respectively, and $f(t) \in \mathbb{R}^k$ is the vector of externally applied forces. Assuming M is invertible, write linear system equations for the mechanical system, with state

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix},$$

input $u = f$, and output $y = q$.

3. 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*)

$$y(k) = a_0u(k) + a_1u(k-1) + \cdots + a_ru(k-r)$$

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

$$y(k) = u(k) + b_1y(k-1) + \cdots + b_py(k-p).$$

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:

$$y(k) = b_1y(k-1) + \cdots + b_py(k-p) + a_0u(k) + \cdots + a_ru(k-r).$$

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.

4. **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.

5. **Counting sequences in a language or code.** We consider a language or code with an alphabet of n symbols $1, 2, \ldots, n$. A sentence is a finite sequence of symbols, $k_1, \ldots, k_m$ where $k_i \in \{1, \ldots, n\}$. A language or code consists of a set of sequences, which we will call the *allowable sequences*. A language is called *Markov* if the allowed sequences can be described by giving the allowable transitions between consecutive symbols. For each symbol we give a set of symbols which are allowed to follow the symbol. As a simple example, consider a Markov language with three symbols $1, 2, 3$. Symbol 1 can be followed by 1 or 3; symbol 2 must be followed by 3; and symbol 3 can be followed by 1 or 2. The sentence 1132313 is allowable (*i.e.*, in the language); the sentence 1132312 is not allowable (*i.e.*, not in the language). To describe the allowed symbol transitions we can define a matrix $A \in \mathbb{R}^{n\times n}$ by

$$A_{ij} = \begin{cases} 1 & \text{if symbol } i \text{ is allowed to follow symbol } j \\ 0 & \text{if symbol } i \text{ is not allowed to follow symbol } j \end{cases}.$$

   a) Let $B = A^r$. Give an interpretation of B_{ij} in terms of the language.

   b) Consider the Markov language with five symbols $1, 2, 3, 4, 5$, and the following transition rules:

      - 1 must be followed by 2 or 3
      - 2 must be followed by 2 or 5
      - 3 must be followed by 1
      - 4 must be followed by 4 or 2 or 5
      - 5 must be followed by 1 or 3

      Find the total number of allowed sentences of length 10. Compare this number to the simple code that consists of all sequences from the alphabet (*i.e.*, all symbol transitions are allowed). In addition to giving the answer, you must explain how you solve the problem. Do not hesitate to use matlab.

c) Consider the Markov language of part (b), among all allowed sequences of length 10, find the most common value for the seventh symbol. In principle you could solve this problem by writing down all allowed sequences of length 10, and counting how many of these have symbol i as the seventh symbol, for $i = 1, \dots 5$. (We're interested in the symbol for which this count is largest.) But we'd like you to use a smarter approach. Explain clearly how you solve the problem, as well as giving the specific answer. Hint: you may find the interpretation of A^k helpful.

6. 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 $i, \dots, n$ 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.

7. Proof of the Cauchy-Schwarz inequality. Let x and y be vectors in $\mathbb{R}^n$. Since the square of a real number is nonnegative, we have that

$$\|(\|y\|x \pm \|x\|y)\|^2 \geq 0.$$

Use this fact to prove the Cauchy-Schwarz inequality:

$$|x^{\mathsf{T}}y| \leq \|x\|\|y\|.$$

When does equality hold in the Cauchy-Scwarz inequality?