EE263 homework 2

1. *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 $q(t) \in \mathbf{R}^k$ is the vector of deflections, M, D, and K are the *mass*, *damping*, and *stiffness* matrices, respectively, and $f(t) \in \mathbf{R}^k$ 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 $y = q$.

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

3. *Matrix representation of polynomial differentiation.* We can represent a polynomial of degree $< n$,

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0,$$

as the vector $[a_0\ a_1\ \cdots\ a_{n-1}]^T \in \mathbf{R}^n$. Consider the linear transformation $\mathcal{D}$ that differentiates polynomials, *i.e.*, $\mathcal{D}p = dp/dx$. Find the matrix D that represents $\mathcal{D}$ (*i.e.*, if the coefficients of p are given by a, then the coefficients of dp/dx are given by Da).

4. Consider the (discrete-time) linear dynamical system

$$x(t+1) = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t) + D(t)u(t).$$

Find a matrix G such that

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N) \end{bmatrix} = G \begin{bmatrix} x(0) \\ u(0) \\ \vdots \\ u(N) \end{bmatrix}.$$

The matrix G shows how the output at $t = 0, \ldots, N$ depends on the initial state $x(0)$ and the sequence of inputs $u(0), \ldots, u(N)$.

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) 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.

6. *Gradient of some common functions.* Recall that the gradient of a differentiable function $f : \mathbf{R}^n \to \mathbf{R}$, at a point $x \in \mathbf{R}^n$, is defined as the vector

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix},$$

where the partial derivatives are evaluated at the point x. The first order Taylor approximation of f, near x, is given by

$$\hat{f}_{\text{tay}}(z) = f(x) + \nabla f(x)^T (z - x).$$

This function is affine, *i.e.*, a linear function plus a constant. For z near x, the Taylor approximation $\hat{f}_{\text{tay}}$ is very near f. Find the gradient of the following functions. Express the gradients using matrix notation.

(a) $f(x) = a^T x + b$, where $a \in \mathbf{R}^n$, $b \in \mathbf{R}$.

(b) $f(x) = x^T A x$, for $A \in \mathbf{R}^{n \times n}$.

(c) $f(x) = x^T A x$, where $A = A^T \in \mathbf{R}^{n \times n}$. (Yes, this is a special case of the previous one.)

7. *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

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

where $\sigma > 0$ 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 $A \in \mathbf{R}^{n \times n}$ and $b \in \mathbf{R}^n$ 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

$$G = \begin{bmatrix} 1 & .2 & .1 \\ .1 & 2 & .1 \\ .3 & .1 & 3 \end{bmatrix}, \qquad \gamma = 3, \qquad \alpha = 1.2, \qquad \sigma = 0.01.$$

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 soon understand what you see.