Principles of Robotic Autonomy
Pre-Knowledge Assessment
Fall 2023

There are 6 questions (5 multiple choice questions, and one programming question). These questions draw upon basic concepts from linear algebra, ODEs, calculus, probability, and very basic programming concepts. If you find these questions difficult, you will likely find this class challenging and so we recommend you to consider taking this class next year when you have a stronger mathematical and programming background. **You are not required to turn this test in; this test will not be considered for your class grade calculation**.

1. (Linear Algebra) What are the eigenvalues of $\begin{bmatrix} 5 & 3 & 1 \\ 0 & -2 & -4 \\ 0 & 0 & 1 \end{bmatrix}$?

   (a) 5, -2, 1, (b) -5, 2, -1 (c) $\frac{3\pm\sqrt{73}}{2}, 1$ (d) 1, -3, -2 (e) Too difficult to compute on paper

2. (Ordinary Differential Equations) Consider the mass-spring-damper system given in Figure 1. The block has mass $m > 0$ and is rigidly attached to a spring and a damper. Assume the x-coordinate is centered at the equilibrium position. The spring has a spring constant $k > 0$ and provides a force negatively proportional to the positional displacement x. The damper has a damping coefficient $b > 0$ and provides a force negatively proportional to the velocity $\dot{x}$. The equation of motion is

$$m\ddot{x} + b\dot{x} + kx = 0. \tag{1}$$

   For appropriate constants $A, B, \omega > 0, \lambda > 0$, what is the general form of the solution to this second-order ordinary differential equation?

   (a) $x(t) = A\sin(\omega t) + B\cos(\omega t)$ (b) $x(t) = Ae^{\lambda t}\sin(\omega t) + Be^{\lambda t}\cos(\omega t)$

   (c) $x(t) = Ae^{-\lambda t} + Be^{\lambda t}\cos(\omega t)$ (d) $x(t) = Ae^{-\lambda t}\sin(\omega t) + Be^{-\lambda t}\cos(\omega t)$

   (e) $x(t) = Ae^{\lambda t}\sin(\omega t) + Be^{-\lambda t}\cos(\omega t)$

3. (Ordinary Differential Equations) Given the ODE in Equation (1), let $m = 1$ and using the change of variables $y_1 = x, y_2 = \dot{x}$, which following linear dynamical system is equivalent?

   (a) $\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -k & -b \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ (b) $\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k & b \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ (c) $\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k & b \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

   (d) $\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -k & -b \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ (e) $\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

4. (Vector Calculus) Let $f(x, y) = 2xy + 6x^2y - 3xy^2$. What is the Hessian of $f(x, y)$?

   (a) $H_f(x,y) = \begin{bmatrix} 12y & 2+12x-6y \\ 2+12x-6y & -6x \end{bmatrix}$ (b) $H_f(x,y) = \begin{bmatrix} 12y & 2+12x-6y \\ -2-12x+6y & -6x \end{bmatrix}$

   (c) $H_f(x,y) = \begin{bmatrix} 2+12x-6y & 12y \\ -6x & 2+12x-6y \end{bmatrix}$ (d) $H_f(x,y) = \begin{bmatrix} -12y & 6x \\ 2+12x-6y & 2+12x-6y \end{bmatrix}$

   (e) $H_f(x,y) = \begin{bmatrix} -12y & 2+12x-6y \\ 2+12x-6y & 6x \end{bmatrix}$

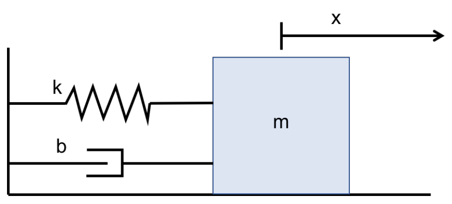

Figure 1: Mass-spring-damper system.

5. (Probability) Bayes' rule says, $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$. You tested positive for a disease, and the test is 80% accurate (i.e., the probability of testing positive given that you have the disease is 0.80, as is the probability of testing negative given that you do not have the disease). This disease strikes one in five people. What is the probability of you having the disease, given that your test was positive?

   (a) $\frac{4}{25}$ (b) $\frac{1}{2}$ (c) $\frac{3}{5}$ (d) 0.2 (e) 0.25

6. (Programming: Recursion[1]) Using your favorite programming language (e.g., Python, Julia, C++), write a recursive function named `even_factorial` that accepts an integer n as a parameter and returns the even factorial of n. An even factorial of an integer is defined as the product of all even integers from 1 through that integer inclusive. For example, the call of `even_factorial(7)` should return $2 \times 4 \times 6$, or 48. The factorial of 0 and 1 are defined to be 1. You may assume that the value passed is non-negative and that its factorial can fit in the range of type int. You may define a helper function if you like.

   Do not use loops or auxiliary data structures to solve the problem recursively. Do not declare any global variables.

[1]Taken from StepByStep https://www.codestepbystep.com/