ME 7247: Advanced Control Systems **Fall 2022–23**

Lecture 1: Overview and linear algebra review

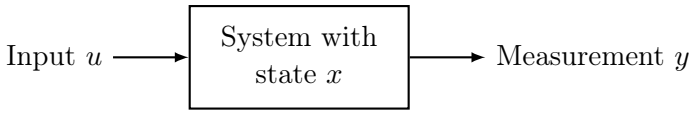
Friday September 9, 2022

Lecturer: Laurent Lessard *Scribe: Laurent Lessard*

First lecture of the semester! Overview of the main topics we will cover in the class, review of some linear algebra material, overview of estimation and control problems.

1 Overview of the class

The main theme of the class is how uncertainty can be modeled and treated in the context of control systems. Here, a *system* is a dynamical system, which has the abstract representation:



Example: A DC motor connected to a battery.

- The input $u(t)$ is the voltage we supply to the motor, which is a function of time.
- The measurement $y(t)$ is the angular speed of the motor shaft, which we measure using a tachometer (and may be inexact or noisy).
- The system is the motor itself. The state $x(t)$ is a vector containing all the quantities that determine the motor's behavior. This may include the current through the windings, the angular position and speed of the shaft, the temperature, etc.

In introductory controls classes, we typically assume a model of the system is known, i.e. we have (linear) differential equations that describe how $y(t)$ is related to $u(t)$. We typically ask: how can we choose $u(t)$ as a function of $y(t)$ so that the closed-loop system has desirable performance?

In this class, we will expand the scope in several directions of practical interest:

- Systems that may have multiple inputs and/or multiple outputs (MIMO systems)
- Cases where the measurements are noisy or otherwise inexact.
- Cases with *hard* constraints, such as a temperature limit that must never be exceeded.
- Cases where the system itself is uncertain, either because the system was approximated (e.g. it isn't actually linear), or the model was obtained experimentally and contains errors.

We will also frame these problems using an *optimization* viewpoint, which is more modern and more flexible than the classical tools you might have seen in controls classes. So we will not use transfer functions, root locus, Bode plots, etc. We will emphasize both mathematical rigor as well as practical implementations using Matlab.

2 Linear algebra review

Key definitions: inner product, outer product, vector norm, angle between vectors, orthogonality, subspace, span, dimension, range, nullspace, basis, orthonormal basis, semi-orthogonal matrix, orthogonal matrix, orthogonal complement (perp space), rank, invertible matrix.

2.1 Matrix multiplication

Vectors are treated as columns by default, and we reference them with lower-case letters. Matrices are referenced with upper-case letters. For example, if $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$ this means:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},$$

where each x_j and each a_{ij} is a real number. Because vectors are columns by default, writing $x \in \mathbb{R}^n$ is the same as writing $x \in \mathbb{R}^{n \times 1}$. Two matrices can be multiplied together whenever the first matrix has as many columns as the second matrix has rows. If $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$, then we can write $C = AB$, where

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}.$$

Matrix multiplication satisfies the following properties:

(i) Not commutative in general: $AB \neq BA$. In fact, AB and BA might not both be defined, and may be different sizes. An exception is when one of the matrices is a scalar, i.e. 1×1.

(ii) Distributive: $A(B + C) = AB + AC$ whenever the dimensions make sense.

(iii) Associative: $(AB)C = A(BC)$. This is useful because some groupings may require fewer arithmetic operations than others (more efficient to compute).

Block partitioning. Consider this example of a 3×3 matrix multiplying a 3×2 matrix:

$$\begin{bmatrix} 1 & 2 & 4 \\ -3 & 1 & 0 \\ 4 & 4 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} 17 & 4 \\ -3 & -7 \\ 0 & 3 \end{bmatrix}.$$

The $(1, 1)$ entry is calculated using the first row of the first matrix and the first column of the second matrix: $17 = 1 \cdot 1 + 2 \cdot 0 + 4 \cdot 4$. If the matrix is partitioned into blocks, we can apply the same multiplication formula, but in a block-wise fashion. Using the same example:

$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \\ -3 & 1 \\ 4 & 4 \end{bmatrix} & \begin{bmatrix} 4 \\ 0 \\ -1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 4 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 17 & 4 \\ -3 & -7 \\ 0 & 3 \end{bmatrix} \end{bmatrix}.$$

For example, the $(1, 1)$ block is calculated using the first block-row of the first matrix and the first block-column of the second matrix:

$$\begin{bmatrix} 17 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 4 \end{bmatrix} \begin{bmatrix} 4 & 1 \end{bmatrix}.$$

Vector operations. Consider two vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. If $m = n$ (both vectors are the same length), we can form their *inner product* by summing the products of the corresponding entries. We can write this in several different ways:

$$\langle x, y \rangle := x^\mathsf{T} y = \sum_{i=1}^{n} x_i y_i$$

The inner product of two real vectors is a real scalar (a number). It is also commutative, so $\langle x, y \rangle = \langle y, x \rangle$. Writing the inner product as $x^\mathsf{T} y$ views it as a matrix-matrix multiplication, where x^T, the transpose of x, is in $\mathbb{R}^{1 \times n}$, while $y \in \mathbb{R}^{n \times 1}$.

We can also multiply a column by a row, which produces a matrix. If $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, then the outer product of x and y is

$$xy^\mathsf{T} = \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & \ddots & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{bmatrix}$$

We will also make frequent use of *vector norms*. The norm of a vector is a scalar that can be interpreted as the "size" of the vector. For all $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, a vector norm satisfies

(i) Positivity: $\|x\| \geq 0$ for all x.

(ii) Definiteness: If $\|x\| = 0$ then $x = 0$.

(iii) Absolute homogeneity: $\|\alpha x\| = |\alpha| \|x\|$.

(iv) Triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$.

Positivity is actually a consequence of the triangle inequality and absolute homogeneity, but we include it anyway because it is a useful property. There are many possible norms, but the one we will use the most is the *Euclidean* or *standard* norm, which we just call "norm". It is:

$$\|x\| := \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}.$$

In addition to the four properties above, the standard norm also satisfies the two properties:

(v) Inner product equivalence: $\|x\|^2 = \langle x, x \rangle$.

(vi) Cauchy–Schwarz inequality: $|\langle x, y \rangle| \leq \|x\| \|y\|$.

We can also define the *angle* between two vectors $x, y \in \mathbb{R}^n$ as the angle $\theta \in [0, \pi]$ such that

$$\cos \theta = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

When $\langle x, y \rangle = \|x\| \|y\|$, the angle between them is 0 and the vectors are aligned. When $\langle x, y \rangle = 0$, the vectors are *orthogonal* and the angle between them is $\frac{\pi}{2}$. When $\langle x, y \rangle = -\|x\| \|y\|$, the angle between them is π and the vectors point in opposite directions.

Matrix product interpretations. If $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$, we can decompose the product in several useful ways. First, we can distribute A across the columns of B:

$$C = AB = A \begin{bmatrix} | & & | \\ b_1 & \cdots & b_n \\ | & & | \end{bmatrix} = \begin{bmatrix} Ab_1 & \cdots & Ab_n \end{bmatrix}.$$

So the j^{th} column of C is Ab_j, where b_j is the j^{th} column of B. Likewise, we can distribute B across the rows of A:

$$C = AB = \begin{bmatrix} — & \tilde{a}_1^\mathsf{T} & — \\ & \vdots & \\ — & \tilde{a}_m^\mathsf{T} & — \end{bmatrix} B = \begin{bmatrix} \tilde{a}_1^\mathsf{T} B \\ \vdots \\ \tilde{a}_m^\mathsf{T} B \end{bmatrix}.$$

So the i^{th} row of C is $\tilde{a}_i^\mathsf{T} B$, where $\tilde{a}_i^\mathsf{T}$ is the i^{th} row of A. We can also simultaneously decompose A by columns and B by rows, leading to a matrix of inner products:

$$C = AB = \begin{bmatrix} — & \tilde{a}_1^\mathsf{T} & — \\ & \vdots & \\ — & \tilde{a}_m^\mathsf{T} & — \end{bmatrix} \begin{bmatrix} | & & | \\ b_1 & \cdots & b_n \\ | & & | \end{bmatrix} = \begin{bmatrix} \tilde{a}_1^\mathsf{T} b_1 & \cdots & \tilde{a}_1^\mathsf{T} b_n \\ \vdots & \ddots & \vdots \\ \tilde{a}_m^\mathsf{T} b_1 & \cdots & \tilde{a}_m^\mathsf{T} b_n \end{bmatrix} = \begin{bmatrix} \langle \tilde{a}_1, b_1 \rangle & \cdots & \langle \tilde{a}_1, b_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{a}_m, b_1 \rangle & \cdots & \langle \tilde{a}_m, b_n \rangle \end{bmatrix}.$$

In this case, $c_{ij} = \tilde{a}_i^\mathsf{T} b_j = \langle \tilde{a}_i, b_j \rangle$. Conversely, we can also decompose A by columns and B by rows, leading to a sum of outer products:

$$C = AB = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_p \\ | & & | \end{bmatrix} \begin{bmatrix} — & \tilde{b}_1^\mathsf{T} & — \\ & \vdots & \\ — & \tilde{b}_p^\mathsf{T} & — \end{bmatrix} = a_1 \tilde{b}_1^\mathsf{T} + \cdots + a_p \tilde{b}_p^\mathsf{T}.$$

Each $a_i \tilde{b}_j^\mathsf{T}$ is a $\mathbb{R}^{m \times n}$ matrix, the same size as C.

2.2 Subspaces

A subspace $S \subseteq \mathbb{R}^n$ is a subset of $\mathbb{R}^n$ with the properties that:

(i) It always contains zero: $0 \in S$.

(ii) Is closed under addition: If $x, y \in S$, then $x + y \in S$.

(iii) Is closed under scalar multiplication: If $x \in S$ and $\alpha \in \mathbb{R}$, then $\alpha x \in S$.

A set of vectors $\{v_1, \ldots, v_k\}$ *spans* S if every vector in S can be written as a linear combination of the v_i's. We write: $S = \text{span}(v_1, \ldots, v_k)$. If a spanning set is *linearly independent*, it is called a *basis*. Every basis of S contains the same number of vectors. This number is called the *dimension* of S, written $\dim(S)$. Given a basis for S, every vector in S can be expressed as a linear combination of the basis vectors in exactly one way.

By convention, if $S = \{0\}$, then $\dim(S) = 0$. A subspace can never be empty; it always contains at least the zero vector. If $\dim(S) = 1$, then S is a line through the origin. If $\dim(S) = 2$, then S is a plane. If $\dim(S) = n$, then $S = \mathbb{R}^n$, the whole space.

Range, nullspace, and rank. Given a matrix $A \in \mathbb{R}^{m \times n}$, two important subspaces are the *range* and *nullspace* of A. The range of A is the set of vectors that can be reached upon multiplication by A. We can define it using set notation as

$$\text{range}(A) := \{Ax \mid x \in \mathbb{R}^n\}.$$

The nullspace of A is the set of vectors that map to zero through A, which we write as

$$\text{null}(A) := \{x \in \mathbb{R}^n \mid Ax = 0\}.$$

Note that $\text{range}(A)$ is a subspace of $\mathbb{R}^m$ while $\text{null}(A)$ is a subspace of $\mathbb{R}^n$.

The *column rank* of a A is the dimension of the span of the columns of A. The *row rank* of A is the dimension of the span of the rows of A. An important fact: the row rank and column rank are equal, so we call them both *rank*. The span of the columns is the range, so $\text{rank}(A) := \dim \text{range}(A)$. Row and column rank being equal means that $\text{rank}(A) = \text{rank}(A^\mathsf{T})$. If $\text{rank}(A) = m$, we say A has *full row rank*. If $\text{rank}(A) = n$, we say A has *full column rank*. In general, $\text{rank}(A) \leq \min(m, n)$. If A has full row rank or full column rank, we simply say that A has *full rank*.

Orthonormal basis. Every subspace S has an *orthonormal basis*. This means that there exists a set $\{u_1, \ldots, u_k\}$ such that each vector has norm 1 and all pairs are orthogonal. In other words,

(i) $\text{span}(u_1, \ldots, u_k) = S$

(ii) $\langle u_i, u_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$.

Given any spanning set $\{v_1, \ldots, v_k\}$ for S, one way to obtain an orthonormal basis for S is via the *Gram–Schmidt process*. An example of an orthonormal basis for $\mathbb{R}^n$ is the *standard basis*:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots \quad e_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

If $\{u_1, \ldots, u_k\}$ is an orthonormal set of vectors, the matrix $U := \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \in \mathbb{R}^{n \times k}$ is a *semi-orthogonal matrix*. It has the property that $U^\mathsf{T} U = I_k$ (the $k \times k$ identity matrix). Multiplication by a semi-orthogonal matrix preserves the inner product and the Euclidean norm. In other words, $\langle Ux, Uy \rangle = \langle x, y \rangle$ and $\|Ux\| = \|x\|$. It's "semi" because $UU^\mathsf{T} \neq I$ in general. In the case where U is square and semi-orthogonal, we have $U^\mathsf{T} U = UU^\mathsf{T} = I_n$, which means $U^{-1} = U^\mathsf{T}$. In this case, U is called *orthogonal*.

Orthogonal complement. The *orthogonal complement* (also called the *perp space*) of $S \subseteq \mathbb{R}^n$ is the set of vectors orthogonal to all vectors in S. We define this as:

$$S^\perp := \{x \in \mathbb{R}^n \mid \langle x, s \rangle = 0 \text{ for all } s \in S\}.$$

$S^\perp$ is a vector space. Every vector $v \in \mathbb{R}^n$ can be decomposed in a unique way as $v = x + y$, where $x \in S$ and $y \in S^\perp$. If $\{u_1, \ldots, u_k\}$ is a basis for S, and $\{u_{k+1}, \ldots, u_n\}$ is a basis for $S^\perp$, then $\{u_1, \ldots, u_n\}$ is a basis for $\mathbb{R}^n$. Finally, we have $\dim(S) + \dim(S^\perp) = n$.

2.3 Linear equations

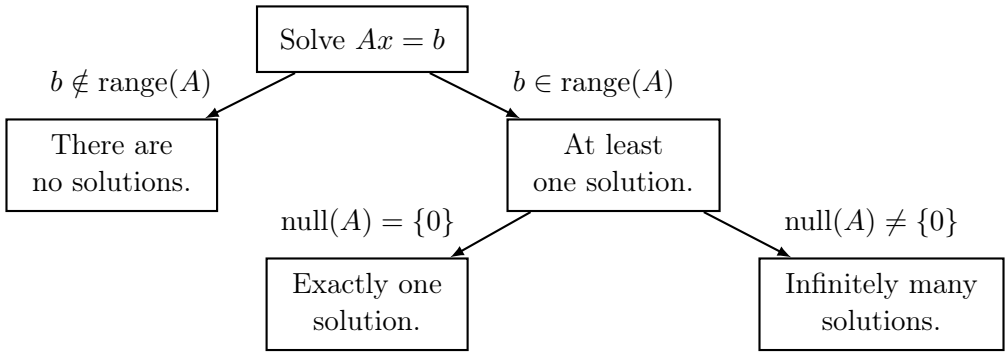
A set of m linear equations in the variables $\{x_1, \ldots, x_n\}$ looks like

$$\left\{\begin{array}{r} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{array}\right\}, \quad \text{or} \quad \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad \text{or} \quad Ax = b.$$

There are precisely three cases that can occur when solving $Ax = b$.

1. There are no solutions. For example, $x_1 + x_2 = 1$ and $x_1 + x_2 = 0$.
2. There is exactly one solution. For example, $x_1 = 1$ and $x_2 = 0$.
3. There are infinitely many solutions. For example, $x_1 + x_2 = 0$.

A solution can exist if and only if $b \in \text{range}(A)$. If a solution $\hat{x}$ does exist and there is a nonzero $v \in \text{null}(A)$, then $\hat{x} + \alpha v$ is also a solution for any $\alpha \in \mathbb{R}$, so there are infinitely many solutions.



The cases where A has full rank are of particular interest, because they imply existence and uniqueness properties for the solutions of $Ax = b$. Here are two results about this.

Theorem 2.1 (existence)**.** *Let $A \in \mathbb{R}^{m \times n}$. The following are equivalent.*

(i) $\text{rank}(A) = m$ *(A has full row rank).*

(ii) $\text{range}(A) = \mathbb{R}^m$.

(iii) For all $b \in \mathbb{R}^m$, there exists a solution to the equation $Ax = b$.

Theorem 2.2 (uniqueness)**.** *Let $A \in \mathbb{R}^{m \times n}$. The following are equivalent.*

(i) $\text{rank}(A) = n$ *(A has full column rank).*

(ii) $\text{null}(A) = \{0\}$.

(iii) If $\hat{x}$ is a solution to $Ax = b$, then it is the only solution.

If both of these conditions hold, i.e. $\text{rank}(A) = m = n$, then A must be square. This corresponds to the case where A is invertible, and the unique solution to $Ax = b$ is $x = A^{-1}b$.

3 Estimation and control problems

Most engineering problems can be boiled down to some version of $Ax = b$, so it's important to know how to solve such equations. We generally encounter two versions, which we will call *estimation problems* and *control problems*. We will look at these in more detail later. Here is an overview.

Estimation problems. In an estimation problem, A is typically a *tall matrix* ($m > n$). Another word for this is *overdetermined* (there are more equations than variables). In such cases, there are typically no solutions to $Ax = b$. Our task is to find an $\hat{x}$ that is a "good fit", so $A\hat{x} \approx b$ in some sense (we will make this more precise later). A typical estimation interpretation is to decompose A into its rows:

$$b = Ax = \begin{bmatrix} — & \tilde{a}_1^\mathsf{T} & — \\ & \vdots & \\ — & \tilde{a}_m^\mathsf{T} & — \end{bmatrix} x = \begin{bmatrix} \tilde{a}_1^\mathsf{T} x \\ \vdots \\ \tilde{a}_m^\mathsf{T} x \end{bmatrix} = \begin{bmatrix} \langle \tilde{a}_1, x \rangle \\ \vdots \\ \langle \tilde{a}_m, x \rangle \end{bmatrix}.$$

Here, each component of b is the inner product of the corresponding row of A with x. In other words, $b_i = \tilde{a}_i^\mathsf{T} x = \langle \tilde{a}_i, x \rangle$. This is a scalar because $\tilde{a}_i^\mathsf{T} \in \mathbb{R}^{1 \times n}$ while $x \in \mathbb{R}^{n \times 1}$. We can interpret $b_1, \ldots, b_m$ as m linear measurements of x, the quantity we're trying to estimate.

For example, suppose we are trying to estimate the epicenter of an earthquake. We want to find its coordinates (latitude, longitude, depth) and we have m seismometers installed at various locations on the Earth. Each seismometer provides a noisy measurement of its distance to the epicenter, which we can relate to the coordinates. If we use many seismometers (large m), we have a hope of overcoming the noise and accurately triangulating the epicenter location.

Control problems. In a control problem, A is typically a *wide matrix* ($m < n$). Another word for this is *underdetermined* (there are more variables than equations). In such cases, there are typically many possible solutions to $Ax = b$. Our task is to *design* x so that b has some desirable properties. A typical control interpretation is to decompose A into its columns:

$$y = Ax = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = a_1 x_1 + \cdots + a_n x_n.$$

Here, each a_i can be interpreted as an *actuator*. Therefore, among all possible combinations of the n actuators that achieves the desired total b, our task is to pick the "best" one. There are many ways to define "best", and we will explore this in greater detail later.

For example, suppose we have a satellite in space and it is outfitted with n small thrusters that point in different directions. Thruster i provides a 3-dimensional force and torque with respect to the center of mass (six degrees of freedom in total), which we write as $a_i \in \mathbb{R}^6$. Let $x_1, x_2, \ldots, x_n$ be the intensities of the n thrusters. If we collect the columns a_i into a matrix $A \in \mathbb{R}^{6 \times n}$, the total force and torque provided by the thrusters is Ax. Among all possible choices that achieve our target $Ax = b$, our goal is to select the one that uses the least fuel. These small thrusters are more efficient at smaller thrust; suppose the fuel used by thruster i is $f_i = x_i^2$. So the total fuel used by all thrusters is $\|x\|^2$. We want the x with smallest possible norm that satisfies $Ax = b$.