Linear Algebra Summary

Outline

- Vectors and vector spaces
- Matrices
- Matrix operations
- Matrix inversion

Linear Algebra: Why do we need this?

- Fundamental representation for data in robotics applications is a matrix
  - e.g. transform matrix

- Robotics algorithms often deal with multi-dimensional data

- Many algorithms try to fit a model to data, e.g. using least-squares fitting based on linear algebra

- Not knowing basic linear algebra is like being *illiterate* in robotics (and much of Engineering)
  - You will have no idea what is going on in this course or more advanced robotics courses

Vectors

- Scalar: a single number

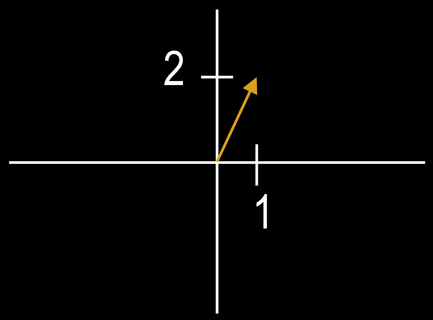
e.g. $a = 5.297$

- Vector: an ordered list of *n* scalars (*n* is the dimensionality)

e.g. $\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \end{bmatrix}$

- Can be interpreted as arrows (or coordinates) in an n-dimensional vector space

e.g. $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \in \boldsymbol{R}^2$



What can I do with vectors and scalars?

- For vectors with the <u>same dimensionality</u>
  - Can add them

$$\boldsymbol{v} + \boldsymbol{w} = \begin{bmatrix} v_1 + w_1 \\ v_2 + w_2 \\ \vdots \end{bmatrix}$$

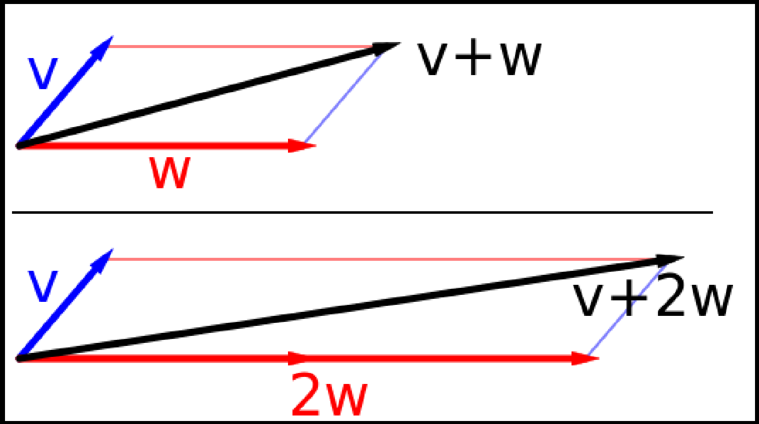
  - Can subtract them

$$\boldsymbol{v} - \boldsymbol{w} = \begin{bmatrix} v_1 - w_1 \\ v_2 - w_2 \\ \vdots \end{bmatrix}$$

  - Can multiply vectors <u>by scalars</u>

$$a\boldsymbol{v} = \begin{bmatrix} av_1 \\ av_2 \\ \vdots \end{bmatrix}$$

- NO multiplication
- NO division



Vector addition with arrows

Vector norm

- Intuitively, a norm measures the "length" of a vector
- Formally, a *p-norm* is a function of the form

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}$$

  - L1-norm:

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i|$$

  - L2-norm (aka the Euclidean Norm):

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \ \dots}$$

  - L∞-norm:

$$\|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

This is the most common norm, often people write $\|\boldsymbol{v}\|$ and mean $\|\boldsymbol{v}\|_2$

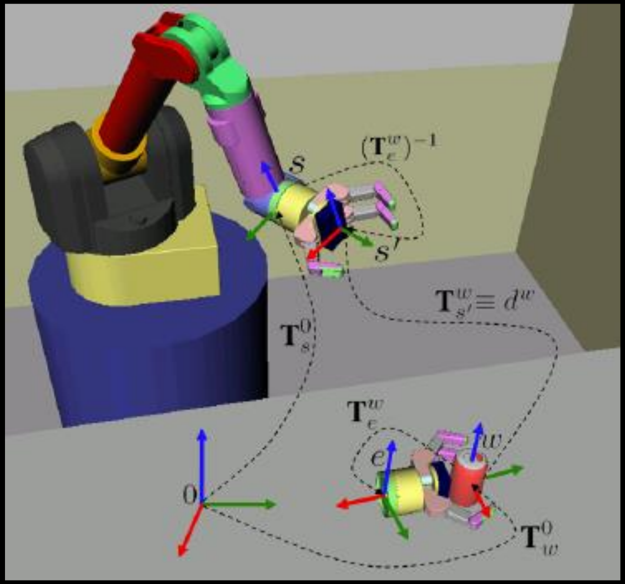
Unit Vectors

- A unit vector is a vector with Euclidean norm 1

$$||\boldsymbol{v}|| = \mathbf{1}$$

- We will use unit vectors to describe directions when we do coordinate frames and transforms

$$x = \begin{bmatrix}1\\0\\0\end{bmatrix} \quad y = \begin{bmatrix}0\\1\\0\end{bmatrix} \quad z = \begin{bmatrix}0\\0\\1\end{bmatrix}$$



Basis Vectors

- Suppose we have an *n*-dimensional vector space $\boldsymbol{R}^n$
- A set of vectors $\mathrm{B} = \{\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots\}$ is a set of basis vectors if
  1. It spans the whole space:

     B spans the space if for any $\boldsymbol{v} \in \boldsymbol{R}^n$

     $$\boldsymbol{v} = a_1\boldsymbol{b}_1 + a_2\boldsymbol{b_2} + \cdots$$

     This is called a linear combination

     where a_i are scalars

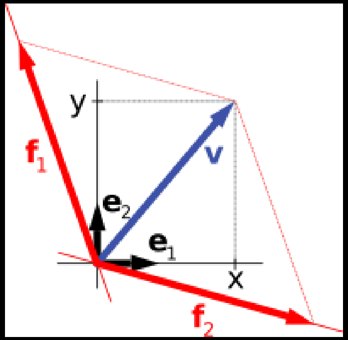
  2. All the vectors in B are linearly independent
     - I.e. no $\boldsymbol{b}_i$ is a linear combinations of the other vectors in B
- So we can represent any $\boldsymbol{v} \in \boldsymbol{R}^n$ as a set of scalars and basis vectors

Basis Vectors

- Example: $\left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right\}$ is a basis for $\boldsymbol{R}^2$

$$\begin{bmatrix}1.3\\-2.4\end{bmatrix} = 1.3\begin{bmatrix}1\\0\end{bmatrix} - 2.4\begin{bmatrix}0\\1\end{bmatrix}$$

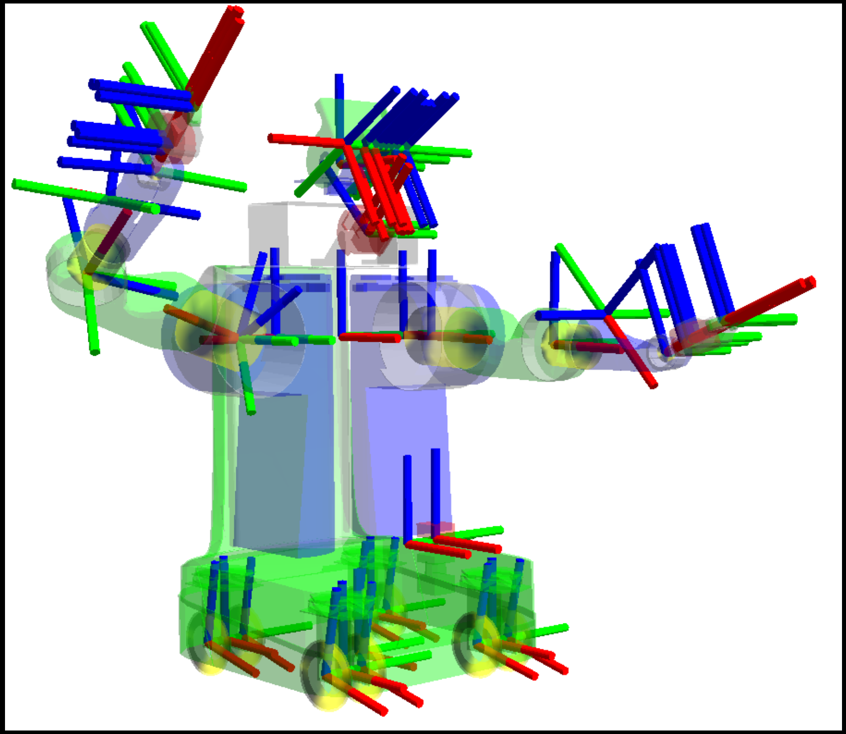
- Basis vectors are not unique!



Both $\{\mathbf{e}_1, \mathbf{e}_2\}$ and $\{\boldsymbol{f}_1, \boldsymbol{f}_2\}$ are bases for $\boldsymbol{R}^2$

- We will use this fact extensively for transformation matrices

Example: PR2 Robot

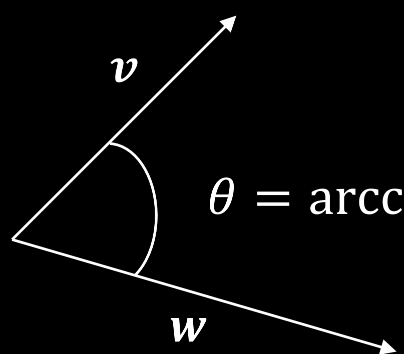


Vector Dot Product

- Dot product of vectors (sometimes called inner product)

$$\boldsymbol{v} \cdot \boldsymbol{w} = \langle \boldsymbol{v}, \boldsymbol{w} \rangle = \sum_{i=1}^{n} v_i w_i$$

- Angle between two vectors:



$$\theta = \arccos\left(\frac{\boldsymbol{v} \cdot \boldsymbol{w}}{\|\boldsymbol{v}\|\,\|\boldsymbol{w}\|}\right)$$

- $\boldsymbol{v}$ and $\boldsymbol{w}$ are called orthogonal if $\boldsymbol{v} \cdot \boldsymbol{w} = \boldsymbol{0}$
  - What is the angle between two orthogonal vectors?

Vector Dot Product

- Dot products and scalars:

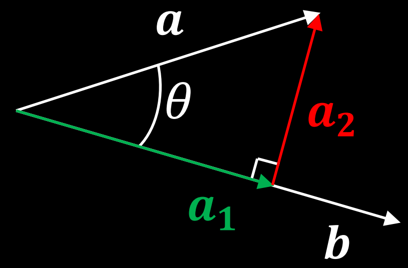
$$a(\boldsymbol{v} \cdot \boldsymbol{w}) = (a\boldsymbol{v}) \cdot \boldsymbol{w} = \boldsymbol{v} \cdot (a\boldsymbol{w})$$

- Can distribute dot product:

$$\boldsymbol{v} \cdot (\boldsymbol{w} + \boldsymbol{p}) = \boldsymbol{v} \cdot \boldsymbol{w} + \boldsymbol{v} \cdot \boldsymbol{p}$$

Vector Projection and Rejection

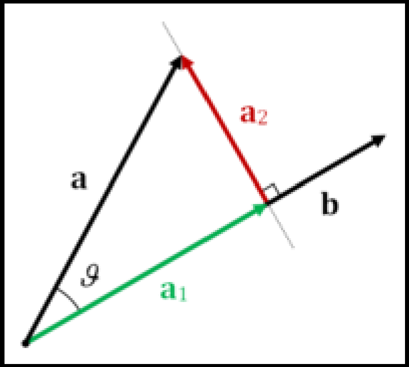
- We sometimes need to divide a vector into components that are parallel and orthogonal to another vector (example later)

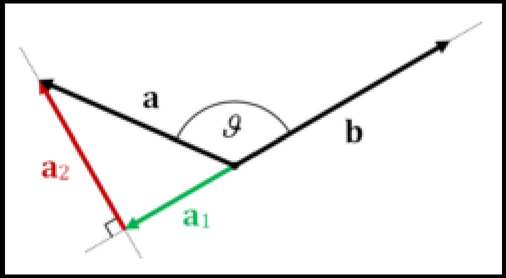


- Scalar projection (from trigonometry):

$$||\boldsymbol{a_1}|| = ||\boldsymbol{a}||cos(\theta)$$

- Angle between $\boldsymbol{a_1}$ and $\boldsymbol{b}$ may be 0 or π!



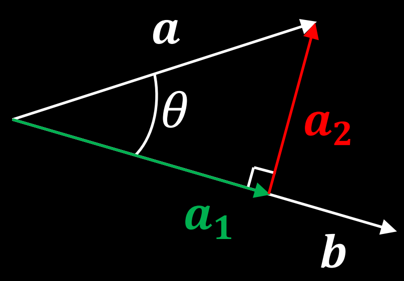


Vector Projection and Rejection

- Vector Projection

$$\boldsymbol{a_1} = \|\boldsymbol{a}\| cos(\theta) \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\boldsymbol{b} \cdot \boldsymbol{b}} \boldsymbol{b}$$

How did we get this?



- Vector Rejection

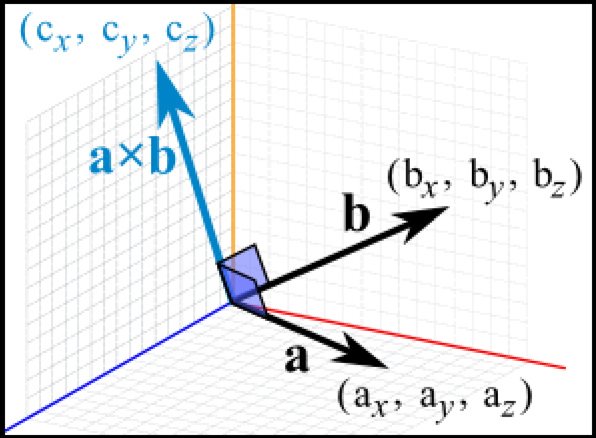
$$\boldsymbol{a_2} = \boldsymbol{a} - \boldsymbol{a_1}$$

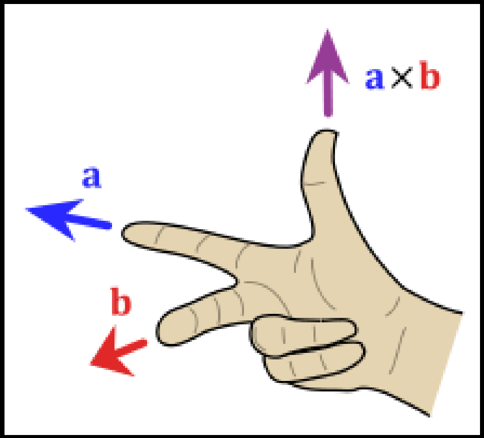
Vector Cross Product

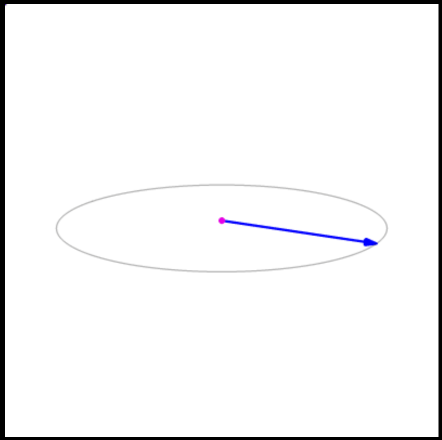
- Given two <u>3D vectors</u> $\boldsymbol{a}$ and $\boldsymbol{b}$, find a vector $\boldsymbol{c}$ orthogonal to both of them:

$$c = a \times b$$

$$\begin{aligned} c_x &= a_y b_z - a_z b_y \\ c_y &= a_z b_x - a_x b_z \\ c_z &= a_x b_y - a_y b_x \end{aligned}$$







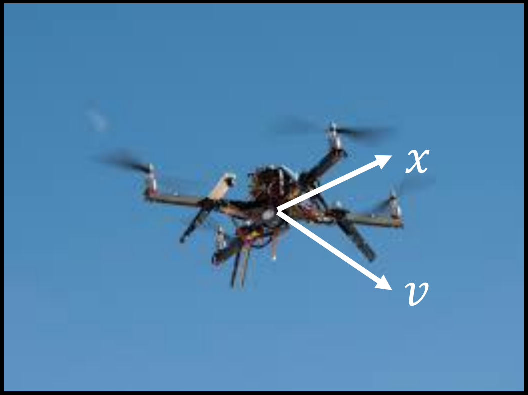
$$\|\boldsymbol{c}\| = \|\boldsymbol{a}\|\|\boldsymbol{b}\| sin(\theta)$$

angle between $\boldsymbol{a}$ and $\boldsymbol{b}$

- Only defined for 3D!

Example: Quadrotor

- We'd like to get a quadrotor to point its camera at a target. We calculate that, at the current position of the quadrotor, it should point the camera along a unit vector $\boldsymbol{v}$.



- However, the camera is rigidly attached to the quadrotor, so we need to command a rotation for the quadrotor. The camera points along x in the quadrotor frame.
  - Assume the camera frame and the quadrotor frame are the same
- To form a rotation matrix, we need three orthogonal unit basis vectors for $\boldsymbol{R}^3$:

$$\mathrm{B} = \{\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{p}\}$$

- How do we find $\boldsymbol{w}$ and $\boldsymbol{p}$?

Example: Quadrotor

$$\mathrm{B} = \{\boldsymbol{v}, \overset{?}{\boldsymbol{w}}, \overset{?}{\boldsymbol{p}}\}$$

- Let's do this one vector at a time
- Notice that $\boldsymbol{w}$ can be any unit vector that is orthogonal to $\boldsymbol{v}$
- So, start with a random 3 x 1 vector $\widetilde{\boldsymbol{w}}$
  - How do we make $\widetilde{\boldsymbol{w}}$ into a vector $\boldsymbol{w}$ that is orthogonal to $\boldsymbol{v}$?

$$\boldsymbol{w} = \widetilde{\boldsymbol{w}} - \frac{\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v}}{\boldsymbol{v} \cdot \boldsymbol{v}} \boldsymbol{v}$$

  - Let's check if we're right. We should have:

$$\boldsymbol{w} \cdot \boldsymbol{v} = \mathbf{0}$$

$$\left(\widetilde{\boldsymbol{w}} - \frac{\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v}}{\boldsymbol{v} \cdot \boldsymbol{v}} \boldsymbol{v}\right) \cdot \boldsymbol{v} = \mathbf{0}$$

$$\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v} - \frac{\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v}}{\cancel{\boldsymbol{v} \cdot \boldsymbol{v}}} \cancel{\boldsymbol{v} \cdot \boldsymbol{v}} = \mathbf{0}$$

$$\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v} - \widetilde{\boldsymbol{w}} \cdot \boldsymbol{v} = \mathbf{0}$$

  - Finally, normalize $\boldsymbol{w}$ to make it a unit vector: $\boldsymbol{w} = \boldsymbol{w} / ||\boldsymbol{w}||$

Example: Quadrotor

$$B = \{\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{p}\}$$

- Now we need a $\boldsymbol{p}$ that is orthogonal to both $\boldsymbol{v}$ and $\boldsymbol{w}$
- Can do this with vector projections, but in 3D it's easier to use the cross product:

$$\boldsymbol{p} = \boldsymbol{v} \times \boldsymbol{w}$$

- Do we need to normalize $\boldsymbol{p}$?

$$||\boldsymbol{p}|| = ||\boldsymbol{v}||\,||\boldsymbol{w}||sin(\theta)$$

$$||\boldsymbol{p}|| = sin(\theta)$$

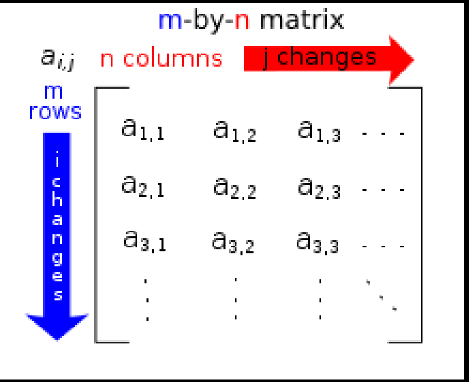
$$||\boldsymbol{p}|| = sin(\pi/2)$$

$$||\boldsymbol{p}|| = 1$$

BREAK

Matrices and Vectors

- Matrix:



A square matrix has $m = n$

- Vector: an m-by-1 matrix

$$\begin{array}{cc} & \text{1 column} \\ \begin{array}{r}\text{m} \\ \text{rows}\end{array} & \begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix} \end{array}$$

Matrix operations

- For matrices with the <u>same dimensions</u>
  - Can add them elementwise, e.g.:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \end{bmatrix}$$

- Can scale them, e.g.:

$$2.4 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} 2.4a_{11} & 2.4a_{12} & 2.4a_{13} \\ 2.4a_{21} & 2.4a_{22} & 2.4a_{23} \end{bmatrix}$$

Matrix Multiplication

- For matrices **A** and **B**, their product is written as **AB**
- Each element in **AB** is *the dot product* of <u>a row</u> of **A** with <u>a column</u> of **B**

Number of columns of **A**

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{n_a} a_{ik} b_{kj}$$

- Example:

A	**B**	**C**
<u>2 x 3</u>	<u>3 x 3</u>	<u>2 x 3</u>

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} \cdot & b_{12} & \cdot \\ \cdot & b_{22} & \cdot \\ \cdot & b_{32} & \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & c_{22} & \cdot \end{bmatrix}$$

$$c_{22} = a_{2:} \cdot b_{:2} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32}$$

Matrix Multiplication

- Matrix multiplication is not commutative!

$$\mathbf{AB} \neq \mathbf{BA}$$

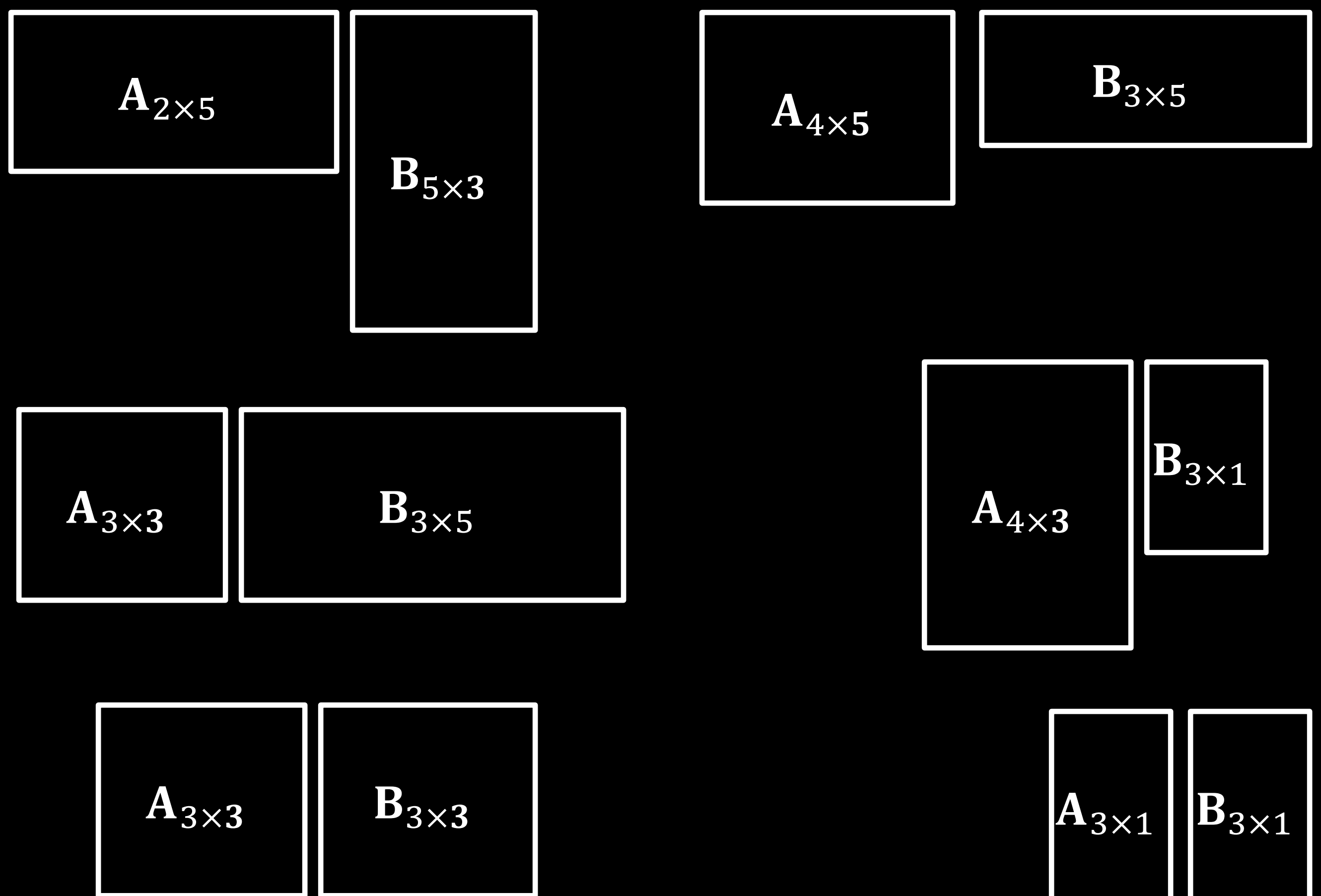
- You can only multiply matrices if they have <u>compatible dimensions</u>

$$\mathbf{A}_{m_a \times n_a} \mathbf{B}_{m_b \times n_b} = \mathbf{C}_{m_a \times n_b}$$

$n_a = m_b$ <u>must be true</u>

Matrix Multiplication

- Which multiplications are valid? What is the dimension of **AB**?



Matrix Transpose

- The transpose of an m-by-n matrix **A** is denoted $\mathbf{A}^T$
- Transpose is done by flipping the matrix about the diagonal; i.e. swap rows and columns:

$$[\mathbf{A}^T]_{ij} = \mathbf{A}_{ji}$$

- Example:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}^T}_{2 \times 3} = \underbrace{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}}_{3 \times 2}$$

- Distributing transpose:

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

- $(\mathbf{A}^T)^T = \mathbf{A}$

Identity Matrix

- The identity matrix $\mathbf{I}_n$ is an *n* x *n* matrix that does no change when multiplied
  - Diagonal elements ($i = j$) are 1
  - Off-diagonal elements ($i \neq j$) are 0

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- For an *m* x *n* matrix **A**

$$\mathbf{A}\mathbf{I}_n = \mathbf{A}$$

$$\mathbf{I}_m\mathbf{A} = \mathbf{A}$$

Matrix Inversion

- Matrix inversion is a common way to solve problems in linear algebra
  - Used everywhere in robotics, from vision to kinematics
- $\mathbf{A}^{-1}$ is the inverse of $\mathbf{A}$ if

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

  - $\mathbf{A}$ must be square (n x n)
  - $\mathbf{A}$ must be *invertible…*

Matrix Invertability

- A square matrix is called singular if it is *not* invertible
- An n x n matrix $\mathbf{A}$ is invertible if (the statements below are equivalent – only need to check one)
  - $\mathbf{A}$ has rank n
    - Rank is the number of linearly independent columns
  - The determinant of $\mathbf{A}$ is not 0
    - The determinant can be viewed as how much the transformation described by the matrix scales an input
  - Many many other ways to check invertability….
- Rank, determinant, and inversion implementations are easy to find in scipy and Matlab

Using Matrix Inversion

- Probably the most common problem in linear algebra: Given a matrix $\mathbf{A}$ and vector b, and the following equation

$$\mathbf{A}x = b$$

solve for the vector x

- Use this to solve a system of linear equations. For example:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned} \quad \Rightarrow \quad \begin{array}{cccc} \mathbf{A} & x & & b \\ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & = & \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \end{array}$$

Solving $\mathbf{A}x = b$

- If $\mathbf{A}$ is <u>*n* x *n*</u> and b is *n* x 1
  - Check rank or determinant of A to see if it is invertible.
  - If so, use the matrix inverse:

$$\mathbf{A}^{-1}\mathbf{A}x = \mathbf{A}^{-1}b$$
$$\mathbf{I}x = \mathbf{A}^{-1}b$$
$$x = \mathbf{A}^{-1}b$$

  - If not, no solution

- What if $\mathbf{A}$ is not square?

The Pseudo-inverse

- The Moore-Penrose Pseudo-inverse is defined as

$$\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T} \qquad (\textit{left} \text{ pseudo-inverse})$$

  - Has some of the properties of the inverse, most importantly:

$$\mathbf{A}^{+}\mathbf{A} = \mathbf{I}$$

  - Derivation:

$$\mathbf{I} = (\mathbf{A}^{T}\mathbf{A})^{-1}(\mathbf{A}^{T}\mathbf{A})$$
$$\mathbf{I} = [(\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}]\mathbf{A}$$
$$\mathbf{I} = \mathbf{A}^{+}\mathbf{A}$$

- The *right* pseudo-inverse is derived similarly to get $\mathbf{A}\mathbf{A}^{+} = \mathbf{I}$

The Pseudo-inverse

$$\mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-\mathbf{1}}\mathbf{A}^{T}$$

- Works even when $\mathbf{A}$ is not square
- What about $(\mathbf{A}^{T}\mathbf{A})^{-\mathbf{1}}$?
  - $(\mathbf{A}^{T}\mathbf{A})$ is automatically square
  - But we need to check if $(\mathbf{A}^{T}\mathbf{A})$ is invertable

- If $\mathbf{A}$ is square and invertable, then $\mathbf{A}^{+} = \mathbf{A}^{-\mathbf{1}}$
  - We don't loose any generality by always using the pseudo-inverse

The Pseudo-inverse

- Can use the pseudo-inverse like an inverse to solve $\mathbf{A}x = b$ when $\mathbf{A}$ is m x n and b is m x 1 :

$$\mathbf{A}^{+}\mathbf{A}x = \mathbf{A}^{+}b$$
$$\mathbf{I}x = \mathbf{A}^{+}b$$
$$x = \mathbf{A}^{+}b$$

- This is known as the least-squares solution
- Remember that $(\mathbf{A}^{T}\mathbf{A})$ must be invertable

$$x = \mathbf{A}^{+}b$$

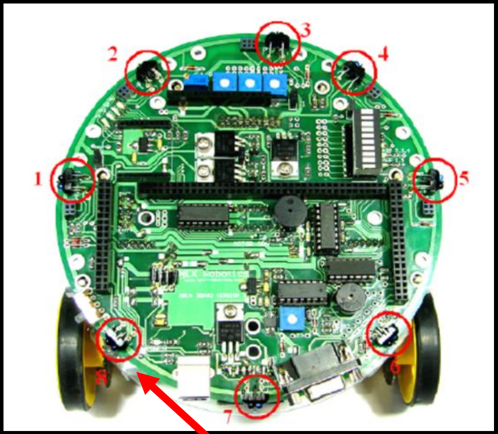
- What does this mean for solving linear systems of equations represented by $\mathbf{A}$ (m x n) and b (m x 1)?
  - m is the number of equations
  - n is the number of unknowns (x is n x 1)

- If m = n
  - $x = \mathbf{A}^{+}b$ is the *exact* solution to the system of equations
- If m < n (*underdetermined*; many solutions are possible)
  - $x = \mathbf{A}^{+}b$ outputs an $\boldsymbol{x}$ that minimizes $||\boldsymbol{x}||_2$
- If m > n (*overdetermined*; no exact solution in general)
  - $x = \mathbf{A}^{+}b$ outputs an $\boldsymbol{x}$ that minimizes the sum of squared errors

Example: Calibration using Least-Squares

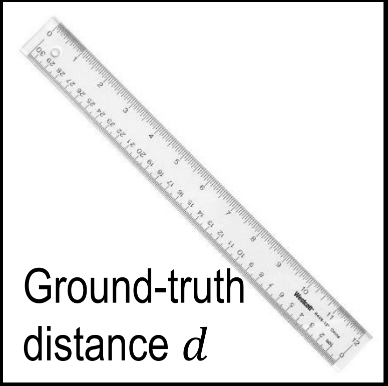
- Suppose you have a IR range sensor that reports a distance, but the sensor is noisy
- You'd like to calibrate this sensor using a ruler as ground-truth
- You measure distance to an object from a series of positions and record the measurements as v
- The corresponding ground-truth distances are d

Fire Bird V Mobile Robot



Measures distance v

- Goal: Find x_1 and x_2 for the function $x_1 v + x_2 = d$ that minimize the sum of squared errors.
  - This is the "Linear Least-Squares" problem



Ground-truth distance d

Example: Calibration using Least-Squares

- How do we turn this into an $\mathbf{A}x = b$ problem?
  - Let's start by thinking about a single datapoint: v_1, d_1
  - We want an equation of the form $x_1 v_1 + x_2 = d_1$
  - We rewrite it as

$$[v_1 \ 1]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = d_1$$

  - Let x be the vector of the unknowns, so $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Then,

$$[v_1 \ 1]x = d_1$$

  - Then define $\mathbf{A} = [v_1 \ 1]$ and $b = d_1$ and we have

$$\mathbf{A}x = b$$

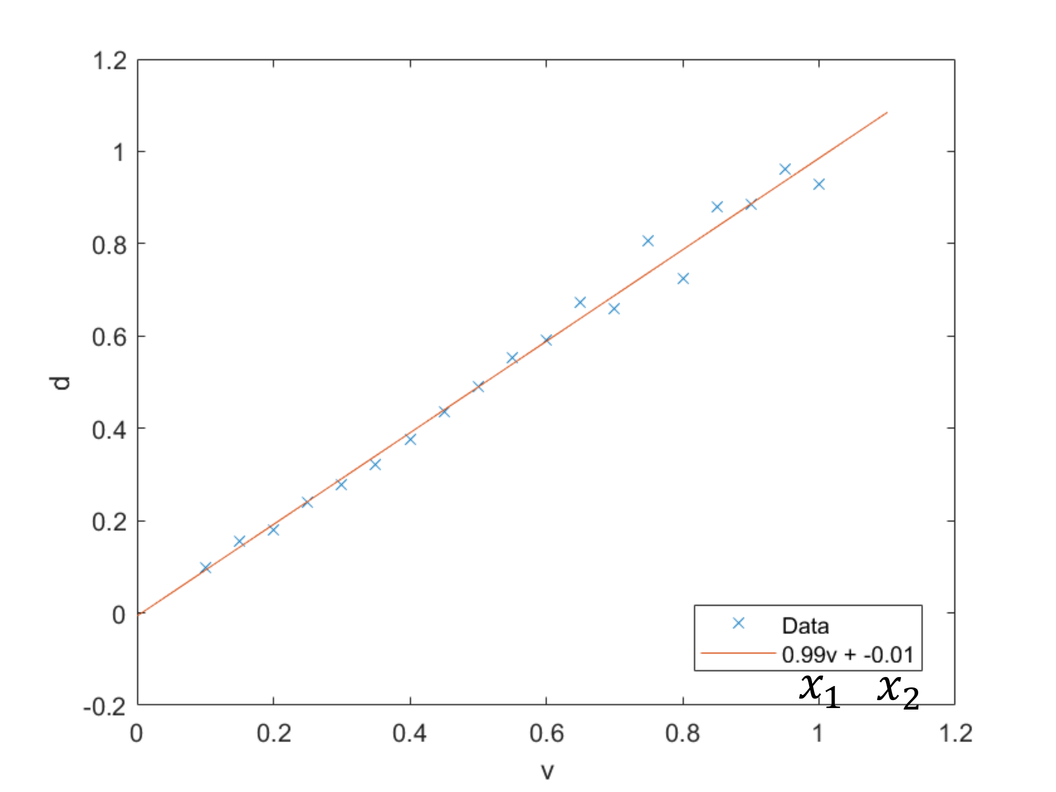
Example: Calibration using Least-Squares

- What about multiple datapoints?

$$\begin{aligned} [v_1\ 1]x &= d_1 \\ [v_2\ 1]x &= d_2 \\ [v_3\ 1]x &= d_3 \\ &\vdots \end{aligned} \Rightarrow \underbrace{\begin{bmatrix} v_1 & 1 \\ v_2 & 1 \\ v_3 & 1 \\ & \vdots \end{bmatrix}}_{\mathbf{A}} x = \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \end{bmatrix}}_{b} \Rightarrow \mathbf{A}x = b$$

Example: Calibration using Least-Squares

Solve $\mathbf{A}x = b$ for x $\Longrightarrow$ $x = \mathbf{A}^{+}b$



- Suppose the IR sensor outputs $v = 0.9\text{m}$
- Then the true distance is estimated as $0.99 * 0.9\text{m} - 0.01 = 0.88\text{m}$

Homework

- Read LaValle Book Chapter 3.2