

CSE276C - Mathematics for Robotics

Henrik I. Christensen



Computer Science and Engineering
University of California, San Diego

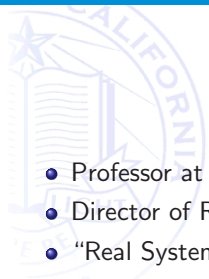
<http://cri.ucsd.edu>

October 2020

Introduction



- Lecturer
- Structure
- Materials
- Information Sources
- Transformations



- Professor at UCSD
- Director of Robotics
- “Real Systems for Real Problems”
- Multi-Robot coordination
- Autonomous Driving Vehicles
- First commercial robot vacuum cleaner
- Working with Boeing, GM, Qualcomm, Robust.AI, ...

Structure of course




- Lectures
- Homework
- Discussions

Structure of course

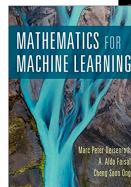
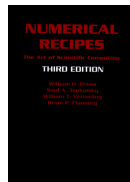


- Lectures
- Homework
- Discussions
- Linear Systems
- Subspace Methods
- Optimization
- Root Finding
- Integration
- Differential Geometry
- Space & Search

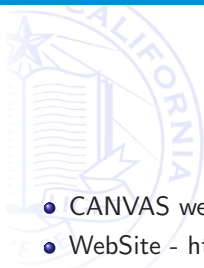
Objectives

- 
- Basic tools for study of robotics
 - Core mathematical concepts
 - A few example applications
 - What are key tools for perception, planning and basic control

- W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. "Numerical Recipes". Cambridge University Press. (Any edition.)
- T. Bewley, Numerical Renaissance: simulation, optimization, & control
- M. Deisenroth, A. Aldo Faisal, and C. Soon Ong, "Mathematics for Machine Learning", Cambridge University Press, 2019

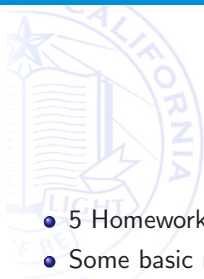


Information Sources



- CANVAS website
- WebSite - <http://www.hichristensen.com/CSE276C-20>
- Piazza - Did you all get an invite?
- Office Hours - Henrik & TA
- Video Lectures available from CANVAS site

Homework

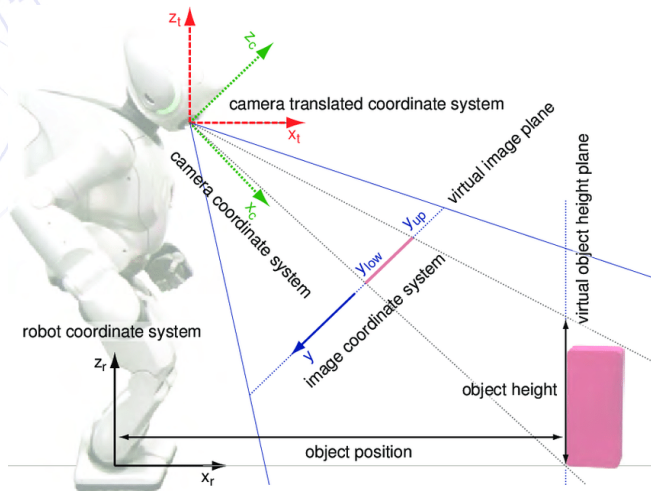


- 5 Homework assignments \approx two weeks
- Some basic math - analysis by manual or automated
- Simple math problems in robotics (could be Python/Numpy or MatLab)
- Analysis of sample robotics data

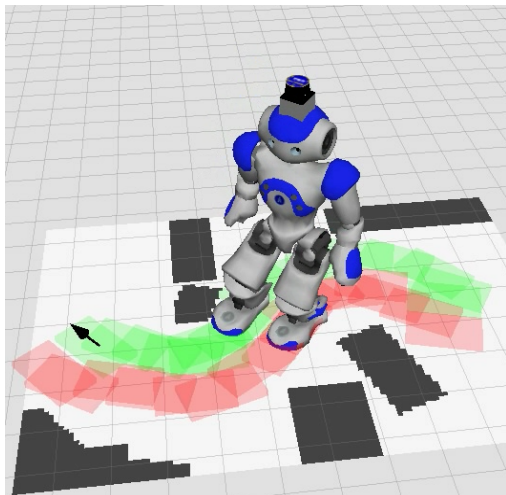


QUESTIONS?

Use of Math in Robotics



Use of Math in Robotics?



Space and Rotations

- How do you represent the position of a robot in space?

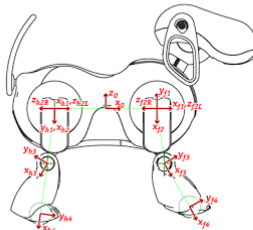
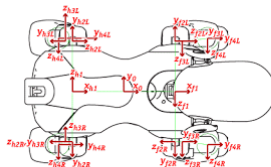
Space and Rotations

- How do you represent the position of a robot in space?

$${}^j p_i = \begin{pmatrix} {}^j p_{x_i} \\ {}^j p_{y_i} \\ {}^j p_{z_i} \end{pmatrix}$$

- The position of i with respect to j
- examples
 - World reference frame
 - Position of the robot
 - Sensor position or a sensor point

Example Reference Frames

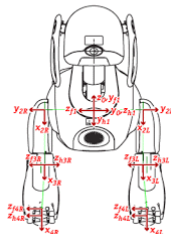


ERS-7 Legs

	Δx	Δy	Δz
1. - shoulder	65	0	0
2. - elevator	0	0	62.5
3. - knee	69.5	0	9
h4. - ball	69.987	-4.993	4.7
h4. - ball	67.681	-18.503	4.7

Diameter of ball of foot is 23.433mm
Each link offset is relative to previous link

The shins shown in this diagram appear to be slightly distorted compared to a real robot. Corresponding measurements have been taken from actual models.



Rotation between two reference frames - i, j



$${}^jR_i = \begin{pmatrix} \vec{x}_i \vec{x}_j & \vec{y}_i \vec{x}_j & \vec{z}_i \vec{x}_j \\ \vec{x}_i \vec{y}_j & \vec{y}_i \vec{y}_j & \vec{z}_i \vec{y}_j \\ \vec{x}_i \vec{z}_j & \vec{y}_i \vec{z}_j & \vec{z}_i \vec{z}_j \end{pmatrix}$$

where $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ and $(\vec{x}_j, \vec{y}_j, \vec{z}_j)$ are
basis vectors for the two coordinate frames

Elementary Rotations

- Rotation around Z-axis

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Elementary Rotations

- Rotation around Z-axis

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- the same for Y and X

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

Considerations for rotations

- We can do combinations

$${}^kR_i = {}^kR_j {}^jR_i$$

Considerations for rotations

- We can do combinations

$${}^kR_i = {}^kR_j {}^jR_i$$

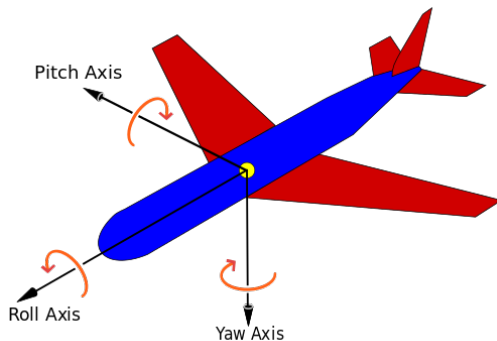
- Note the order is important

$${}^kR_j {}^jR_i \neq {}^jR_i {}^kR_j$$

- The order and reference frames are very important

Euler Angles

- We frequently use Euler angles in robotics



Euler Angles

- The convention used is $R_z R_y R_x$ with respect to $(\alpha, \beta, \gamma)^T$

$${}^jR_i = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$$

Derivation of Euler angles

- If we have the rotation matrix

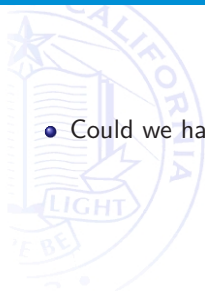
$${}^jR_i = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{pmatrix}$$

- derivation of the Euler Angles

$$\begin{aligned}\beta &= \operatorname{atan2} \frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}} \\ \alpha &= \operatorname{atan2} \frac{r_{21} / \cos \beta}{r_{11} / \cos \beta} \\ \gamma &= \operatorname{atan2} \frac{r_{32} / \cos \beta}{r_{33} / \cos \beta}\end{aligned}$$

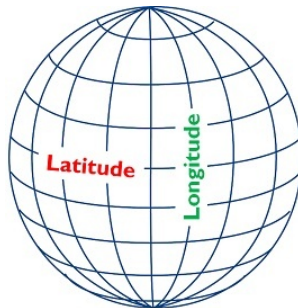
When may we have problems?

- Could we have problems? / When?



When may we have problems?

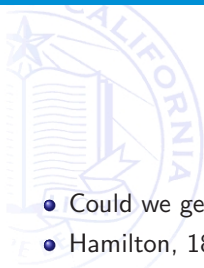
- Could we have problems? / When?
- What about singularities?





QUESTIONS?

Quaternions



- Could we generate a representation that has no singularities?
- Hamilton, 1843.
- A 3-parameter family is not adequate (proved by now)
- A 4-parameter model is a possibility
- Quaternions is a possible representation (not the most intuitive)

Quaternions

- Imagine 3-D imaginary numbers - three basis vectors - $\vec{i}, \vec{j}, \vec{k}$
- We can represent a quaternion as

$$\vec{\epsilon} = \epsilon_0 + \epsilon_1 \vec{i} + \epsilon_2 \vec{j} + \epsilon_3 \vec{k}$$

or $(\epsilon_0, \vec{\epsilon})$

Quaternions

- Imagine 3-D imaginary numbers - three basis vectors - $\vec{i}, \vec{j}, \vec{k}$
- We can represent a quaternion as

$$\vec{\epsilon} = \epsilon_0 + \epsilon_1 \vec{i} + \epsilon_2 \vec{j} + \epsilon_3 \vec{k}$$

- or $(\epsilon_0, \vec{\epsilon})$
- we have three basis vectors

$$\vec{i}\vec{i} = \vec{j}\vec{j} = \vec{k}\vec{k} = -1$$

- mixed products

$$\begin{aligned}\vec{i}\vec{j} &= \vec{k}, \quad \vec{j}\vec{k} = \vec{i}, \quad \vec{k}\vec{i} = \vec{j} \\ \vec{j}\vec{i} &= -\vec{k}, \quad \vec{k}\vec{j} = -\vec{i}, \quad \vec{i}\vec{k} = -\vec{j}\end{aligned}$$

- Null quaternion

$$\vec{0} = 0 + 0\vec{i} + 0\vec{j} + 0\vec{k}$$

- Unit quaternion

$$\vec{1} = 1 + 0\vec{i} + 0\vec{j} + 0\vec{k}$$

Quaternion operations

- Product of two quaternions

$$\begin{aligned}\vec{a}\vec{b} &= a_0b_0 - a_1b_1 - a_2b_2 - a_3b_3 \\ &+ (a_0b_1 + a_1b_0 + a_2b_3 + a_3b_2)\vec{i} \\ &+ (a_0b_2 + a_2b_0 + a_3b_1 - a_1b_3)\vec{j} \\ &+ (a_0b_3 + a_3b_0 + a_1b_2 - a_2b_1)\vec{k}\end{aligned}$$

- The good news there are standard libraries

Rotations w. Quaternions

- Rotating at an angle θ around the vector \vec{a} expressed as a quaternion:

$$\vec{\epsilon} = \cos \frac{\theta}{2} + a_x \sin \frac{\theta}{2} \vec{i} + a_y \sin \frac{\theta}{2} \vec{j} + a_z \sin \frac{\theta}{2} \vec{k}$$

- or

$$\vec{\epsilon} = \left(\cos \frac{\theta}{2}, \vec{a} \sin \frac{\theta}{2} \right)$$

Mapping quaternions to rotation matrices

- The rotation of a quaternion $\vec{\epsilon}$ can be written as

$${}^jR_i = \begin{pmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1\epsilon_2 - \epsilon_0\epsilon_3) & 2(\epsilon_1\epsilon_3 + \epsilon_0\epsilon_2) \\ 2(\epsilon_1\epsilon_2 + \epsilon_0\epsilon_3) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2\epsilon_3 - \epsilon_0\epsilon_1) \\ 2(\epsilon_1\epsilon_3 - \epsilon_0\epsilon_2) & 2(\epsilon_2\epsilon_3 + \epsilon_0\epsilon_1) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{pmatrix}$$

- As I said the good news there are standard libraries

From a rotation matrix to quaternions

- Direct computing the quaternions from a rotation matrix (R)

$$\begin{aligned}\epsilon_0 &= \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \\ \epsilon_1 &= \frac{r_{32} - r_{33}}{4\epsilon_0} \\ \epsilon_2 &= \frac{r_{13} - r_{31}}{4\epsilon_0} \\ \epsilon_3 &= \frac{r_{21} - r_{12}}{4\epsilon_0}\end{aligned}$$

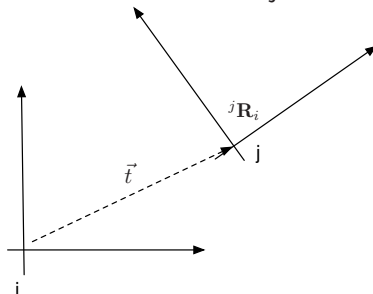
- Quaternions frequently used in graphics, computer vision and robotics



QUESTIONS?

Coordinate transformations

- To move between transformation or move an object we frequently encounter



- We can write this as ${}^j\vec{p} = {}^jR_i {}^i\vec{p} + \vec{t}$

Homogeneous Transformations

- We can do this more easily with homogeneous coordinates

$$\vec{P} = \begin{pmatrix} \vec{p} \\ 1 \end{pmatrix}$$

- given this our transformation can now be written as

$$\begin{pmatrix} {}^j p \\ 1 \end{pmatrix} = \begin{pmatrix} {}^j R_i & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^i p \\ 1 \end{pmatrix}$$

- or ${}^j P = {}^j T_i {}^i P$

Standard Joints

- Revolute joint

$${}^jR_i = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Prismatic joint

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & 0 \\ I & 0 \\ & d \\ 0 & 1 \end{pmatrix}$$

Standard Joints (cont.)

- Cylindrical

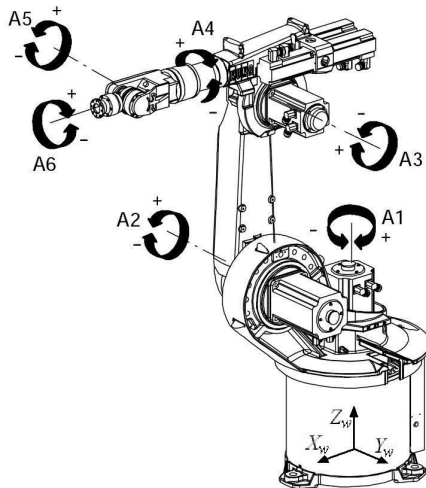
$$T = \begin{pmatrix} & 0 \\ R_\theta & 0 \\ & d \\ 0 & 1 \end{pmatrix}$$

- Spherical

$$T = \begin{pmatrix} R_{\alpha,\beta,\gamma} & 0 \\ 0 & 1 \end{pmatrix}$$

- Most other joints can be constructed from these basic models

Example Robot KUKA KR15



- This is an entire field of its own.
- How can we computer the velocity of a robot?
- If we know a desired velocity, how fast should we turn the wheels?
- In general we will refer to the robot actuators as q_i
- Forward kinematics

$$v = \Phi \dot{q}$$

- Inverse kinematics

$$\dot{q} = \Phi^{-1} v$$

- Not get into the much of the details until we talk about differential geometry (end of course)

Wrap-up



- Basic course information
 - Books, topics, websites, ...
- Introduction to positions, rotations, transformations, ...
- Next time we will talk about linear systems of equations



Questions