EECS 106B/206B
Robotic Manipulation and Interaction

Instructor: Ruzena Bajcsy and Shankar Sastry
GSIs: Valmik Prabhu, Tiffany Cappellari,
Amay Saxena, Jun Zeng, Ron Thalanki

Goals of the Course

Survey some major research areas in modern robotics, and prepare you to do research in those fields or others.

- Areas Primarily Covered: Nonlinear control, path planning, mobile robotics, grasping, soft robotics
- Areas Somewhat Covered: Legged robotics, active safety, HRI
- Areas Not Covered: Learning, Estimation, SLAM, Vision, Mechatronics

Give you the skills you need to succeed in research.

- Reading papers, presenting your work, writing reports

course LOGISTICS

Prerequisites

- 106A is **strongly recommended**
- EE 128 is nice to have
- Knowledge of **linear algebra**
- Programming in **Python and/or Matlab**
- **Curiosity** about how things work
- Interest in **experimental work**
- **Willingness to explore**

Who to Ask

- Questions regarding **homeworks** should be directed to **Valmik**, **Amay**, and **Jun**
- Questions regarding **course logistics** should be directed to **Tiffany**
- Questions regarding **projects** can be directed to **any TA**
- Please post on **Piazza** for the fastest response
- If sending an email, please prefix it with **[EECS 106B]**
- All staff emails can be found on the course website

Lectures

- **Tuesday/Thursday** from **3:30-5:00 pm**, **3106 Etch**
- Lectures from Ruzena Bajcsy, Shankar Sastry, and Valmik Prabhu, with a multitude of guest lectures (TBA)

Lectures

Review: 2 Lectures

Control: 4 Lectures

Nonholonomic Systems: 2 Lectures

Planning: 3 Lectures

Grasping: 3 Lectures

Soft Robotics: 2 Lectures

Guest Lectures and Special Topics: 12 Lectures

Course Grading

- **15%** for 5 homeworks
- **2%** for scribing
- **3%** for paper presentation
- **50%** for 4 projects
- **30%** for the final project

Homework

Homework will be assigned **biweekly** during January, February, and March. Homeworks are **posted Saturdays** (except the first) and due on Gradescope **two weeks later** on **Friday at 11:59pm**.

Assignment Number	Date Assigned	Date Due
Homework 1	1/23	1/31
Homework 2	2/1	2/14
Homework 3	2/15	2/28
Homework 4	2/29	3/13
Homework 5	3/14	4/3

You have 5 slip days for homework, and may use up to two on any assignment.

Scribing

You will be expected to scribe (write notes for) one lecture this semester. Two people will be assigned to each lecture, and will be collectively responsible for producing the notes.

- Scribed notes are due on Gradescope by the start of the following lecture.
- Scribed notes must be in LaTeX (we will provide a template)
- We want both the .pdf file and the .tex file
- Notes will be given a zero, a one, or a two based on their quality.

Sign up on the google sheet, which has been posted on Piazza.

Paper Presentations

Every student will be responsible for presenting one paper to their lab section.

- The paper presentation has the same grade weight as a homework assignment, and we expect you to spend a similar amount of time (6-8 hrs)
- David McPherson and Tyler Westenbroek will be giving you all some tips on reading papers as well as two example presentations during lab in week 2.
- Talks can be slide-based or board-based, and should take around 20 minutes
- Please practice your talk a couple times beforehand and prepare to answer questions.

Sign up on the google sheet, which will be posted on Piazza

Projects

There will be **four projects**. Each will be assigned over approximately 3 weeks and will require a report (5-10 pages, plus graphics and code snippets).

These projects are **very different from labs in EECS 106A/206A**; they are much more open-ended and are closer to **conducting experiments** than implementing algorithms. They should take approximately 40-60 man-hours.

Each project should be performed by groups of 3 students. It’s recommended that **each group have at least one student who took EECS C106A/206A last fall, and at least one student who has some controls experience**.

The robotics lab will be open 24/7 for students to use. We will set up calendars so that groups do not overlap.

Reserving Robots

Calendars to reserve robots can be found on the course website. Please do not use the hardware until you have completed Lab 0.

Each team may only reserve **3 hours at a time** and can only make a new reservation once one of the team's existing reservation times have ended. If a team is caught abusing this policy and overbooking, a **5% penalty** will be deducted from the group's project grade.

Project Due Dates

Assignment Number	Date Assigned	Date Due
Project 1a	1/23	2/3
Project 1b	2/3	2/18
Project 2	2/19	3/9
Project 3	3/10	3/30
Project 4	3/31	4/13

Project Descriptions

Project 1: Trajectory tracking with control on Baxter/Sawyer. Develop 3 different trajectory tracking controllers and compare them.

Project 2: Path planning with Turtlebots. Comparing optimization based, sampling based, and constructive path planners.

Project 3: Grasp Planning with Baxter/Sawyer. Analyze object meshes and determine grasp locations for successful grasps.

Project 4: Soft Robotics. Modelling and control of a soft robotic finger.

Final Project Guidelines

Students choose their own final projects, but they **must be approved** by course staff. Proposals will be due **3/20** and project presentations will be on the **Friday of RRR week**. You will have an intermediate presentation during lab in week 14.

- These should be **research** projects instead of just implementation projects (unlike in 106a).
- Projects are expected to apply **multiple aspects of course material**.
- Every project should include some ***sensing***, ***planning***, and ***actuation***.
- Projects should **demonstrate good designer/experimentalist rigor**:
  - What did you measure? What are your assumptions? What did your measurements tell you?
  - How did you evaluate your results? How do you account for error?
  - What lessons did you learn?

Expectations from Graduate Students

Project 1: Implement a learning feedback linearization controller

Project 2: Implement closed loop control for trajectory tracking

Project 3: Parameter identification with robust force closure

Project 4: Custom experiment

Final Project: Content must have enough promise and innovation to be published in a robotics conference (given additional development).

Undergrads may complete these additional tasks for extra credit.

Discussions and Office Hours

Discussion Sections:

W 2:00pm-3:00pm 237 Cory, Valmik & Amay

W 4:00pm-5:00pm 293 Cory, Valmik & Amay

Office Hours:

Prof. Bajcsy: Weds 11-12, Thursday 1-2 (SDH 719)

Prof. Sastry: Tuesday 1:15-2:30 (Cory 333B)

Tiffany: Tuesday 11-12 (Cory 111)

Amay: Monday 4-5 (Cory 111)

Jun: Monday 5-6 (Cory 111)

Valmik: Tuesday 12-1, Wednesday 5-6 (Cory 111)

Lab Sections

- Thursday 5-8pm Cory 111, Tiffany
- Friday 11am-2pm Cory 111, Amay
- Friday 2-5pm Cory 111, Jun

While lab sections are three hours, we expect to only use 1.5-2 hrs most days. The rest of the time will be devoted to office hours, where members of that lab section have priority.

The sections currently predicted to take three hours are:

Week 1 (this week), Week 14

Announcements for this Week

Check out the website: http://inst.eecs.berkeley.edu/~ee106b/sp20

Join Piazza and Gradescope

Discussion 1 is tomorrow

Lab 0 in lab section this week

HW 1 will be assigned tomorrow

Project 1a will be released this week and will be discussed during lab section

Expectations

We want to emphasize **collaboration** and **exploration**

Ask **lots of questions**

It’s **ok to be confused**

Expect to **work very hard**

Review of Rigid Body Motion

Content primarily from MLS Ch 2, 3
More advanced material may be found in MLS Appendix A.3

Goals of this lecture

Examine the geometry underlying rotations and rigid body transforms.

- Why do we use the matrix exponential?
- What is a twist?
- This will contribute to a deeper understanding of velocities, Jacobians, and dynamics

Rotation Matrices

Rotation matrices will rotate any vector *q* about some axis *ω* by an angle *θ*

$$\mathbf{A}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$\mathbf{A}_Y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{A}_Z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

R_{AB} transforms q_B to q_A

$$q_A = R_{AB} q_B$$
$$q_B = R_{BA} q_A$$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Rotation Matrices as Linear Transformations

$$R : \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

A rotation matrix is essentially a change of basis matrix

$$q_A = R_{AB}\, q_B$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_A = \left[\begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix}, \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix}, \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix}\right]_{AB} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B$$

Output

Effect on x

Effect on y

Effect on z

Input

Actions with Linear Transformations

$$T_{AA} : A \to A$$

Left Multiplication: Change the output frame

$$R_{BA}T_{AA} : A \to B$$

Right Multiplication: Change the input frame

$$T_{AA}R_{AB} : B \to A$$

Change of Basis: Change both the input and output frames

$$R_{BA}T_{AA}R_{BA}^{-1} : B \to B$$

Rotations as a Group

SO(n), the group of special orthogonal matrices under matrix multiplication can be used to represent any rotation.

Properties of Rotation Matrices:

Orthogonality:

$$RR^T = R^T R = I$$

Positive Determinant:

$$\det(R) = +1$$

Properties of Groups:

Closure:

$$P, Q \in SO(n) \implies PQ \in SO(n)$$

Associativity:

$$\forall A, B, C \in SO(n), (AB)C = A(BC)$$

Identity

$$\exists I \in SO(n), st AI = IA = A, \forall A \in SO(n)$$

Inverse:

$$\forall A \in SO(n), \exists A^{-1} \in SO(n) \, st \, AA^{-1} = I$$

Rotations as a Lie Group

SO(n) is a *continuous, smooth* group (a **Lie Group**), which means that it has a tangent space, or derivative. The tangent space at the identity element is called the **Lie Algebra**, so(n).

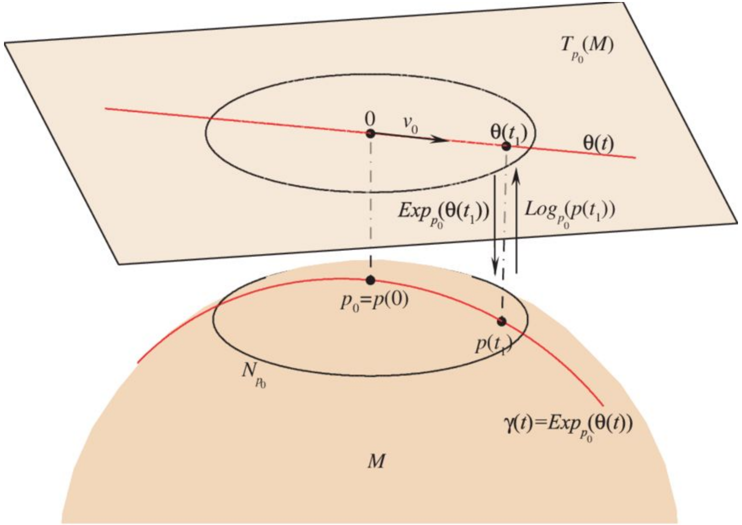


Figure from: Rahman, Inam & Drori, Iddo & Stodden, Victoria & Donoho, David & Schröder, Peter. (2005). Multiscale Representations for Manifold-Valued Data. SIAM Journal on Multiscale Modeling and Simulation. 4. 10.1137/050622729.

The Lie Algebra so(n)

The Lie Algebra, or tangent space, to SO(n) is so(n), the set of all nxn skew-symmetric matrices

$$so(n) = \{S \in \mathbb{R}^{n \times n} : S^T = -S\}$$

Even though skew symmetric matrices have nxn numbers, they only have n(n-1)/2 degrees of freedom. The *hat* operator* is used to go between the parameters and so(n)

$$\hat{}(a) := a \in \mathbb{R}^{\frac{n(n-1)}{2}} \rightarrow \hat{a} \in so(n)$$

For so(2) and so(3), this is

$$so(2) : \omega \in \mathbb{R} \;\; \hat{\omega} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \;\; so(3) : \omega \in \mathbb{R}^3 \;\; \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

* Technically, the hat operator is only defined for so(3), but most people understand it for all so(n) so I'll use it that way here.

The Exponential Map

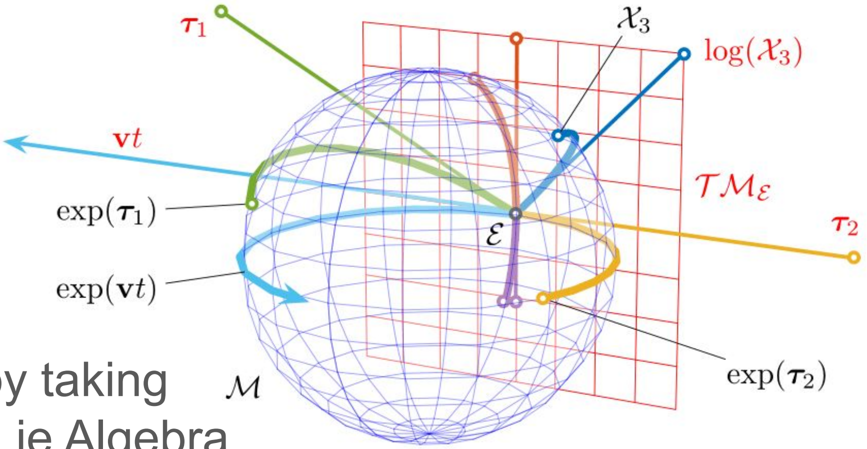
Elements in the Lie Algebra are essentially velocities. You can find any element in the group by starting at the identity and moving at some velocity $\hat{\omega}$ for a time *T*.

The solution to the differential equation

$$\dot{R}(t) = \hat{\omega} R(t)$$

From t = 0 to T is

$$e^{\hat{\omega} T} R(0) = e^{\hat{\omega} T}$$



You can find any element in the Lie Group by taking the matrix exponential of an element in the Lie Algebra.

If $||\hat{\omega}|| = 1$ then $T = \theta$. Thus, $R = e^{\hat{\omega}\theta}, ||\omega|| = 1$

Figure from: Solà, Joan & Deray, Jeremie & Atchuthan, Dinesh. (2018). A micro Lie theory for state estimation in robotics.

Questions:

1. What does the manifold of the Lie Group SO(2) look like? What about so(2)?

2. Let's say you have two rotation matrices R_1 and R_2 with exponential coordinates ω_1 and ω_2. Define R_3 as R_1R_2 with exponential coordinates ω_3. Can you find ω_3 in terms of ω_1 and ω_2?

Answers

1. What does the manifold of Lie Group SO(2) look like? What about so(2)?

The formula for SO(2) is a 2x2 matrix, but we really only need the leftmost vector to describe it. Thus, we can represent it as a circle in two dimensions with unit radius.

Since the leftmost vector has value [1, 0] when the rotation is an identity, so(2) can be represented as the tangent line of the circle at [1, 0]. Note that there is only one degree of freedom on so(2).

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

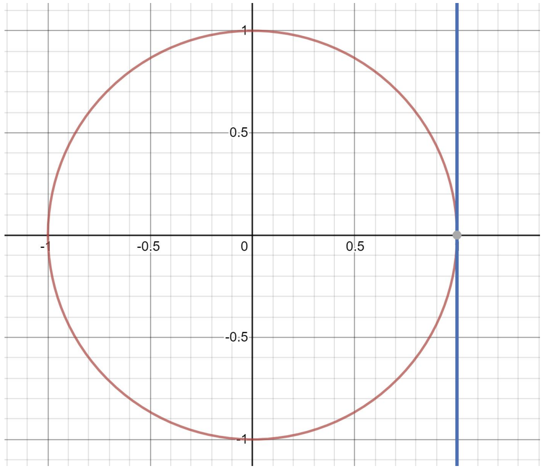


Figure from: desmos.com/calculator

Answers

2. Let's say you have two rotation matrices R_1 and R_2 with exponential coordinates ω_1 and ω_2. Define R_3 as R_1R_2 with exponential coordinates ω_3. Can you find ω_3 in terms of ω_1 and ω_2?

No you cannot. If $\omega_3 = \omega_1 + \omega_2$, then

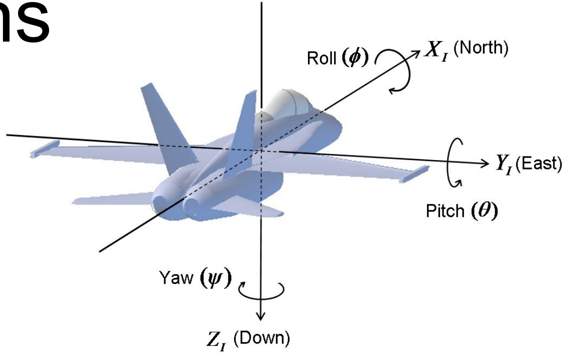
$$e^{\hat{\omega}_1} e^{\hat{\omega}_2} = e^{\hat{\omega}_1 + \hat{\omega}_2}$$

Which is only true if

$$\hat{\omega}_1 \hat{\omega}_2 = \hat{\omega}_2 \hat{\omega}_1$$

Matrix multiplication is not commutative in general so this is usually not true.

Other Parameterizations of Rotations



- Axis Angle:
  - Same as exponential if ||ω|| = 1
- Euler Angles
  - Suffers from the Gimbal Lock singularity
- Unit Quaternions
  - “An unmixed evil” - Lord Kelvin
  - Quaternions are also a Lie Group
  - Every rotation corresponds to two quaternions
  - Quaternions are computationally efficient

$$R_x(\varphi) := e^{\hat{x}\varphi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$

$$R_y(\theta) := e^{\hat{y}\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\psi) := e^{\hat{z}\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \omega_1 \sin(\theta) \\ \omega_2 \sin(\theta) \\ \omega_3 \sin(\theta) \end{bmatrix}, \|\omega\| = 1$$

$$\begin{aligned} R_{ab} &= R_x(\varphi)R_y(\theta)R_z(\psi) \\ &= \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ s_\varphi s_\theta c_\psi + c_\varphi s_\psi & -s_\varphi s_\theta s_\psi + c_\varphi c_\psi & -s_\varphi c_\theta \\ -c_\varphi s_\theta c_\psi + s_\varphi s_\psi & c_\varphi s_\theta s_\psi + s_\varphi c_\psi & c_\varphi c_\theta \end{bmatrix} \end{aligned}$$

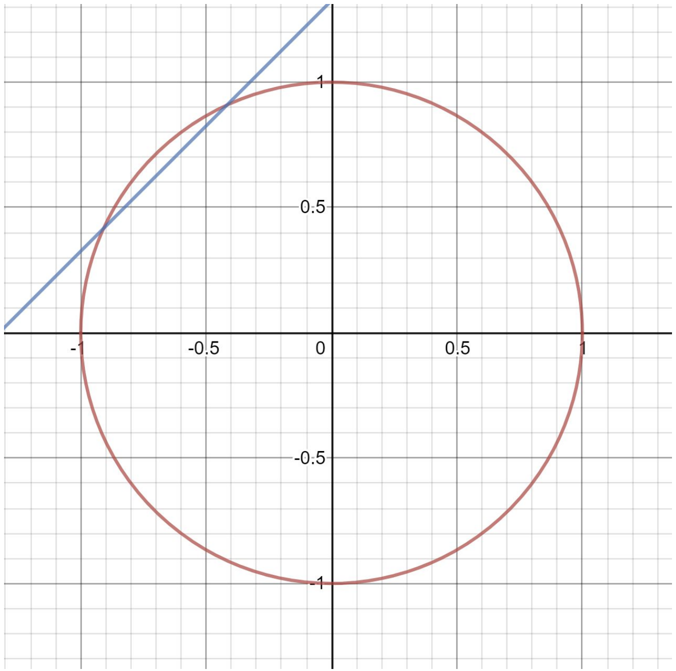
Rotational Velocities

When we take the derivative of a rotation, we get a tangent plane of R at its current value, which is *not* in so(n)

$$q_a = R_{ab} q_b$$

$$\frac{d}{dt} q_a = \dot{R} q_b$$

$$\dot{R} : q_b \to \dot{q}_a$$



Representing Rotational Velocities with so(n)

The tangent plane of $\dot{R}$ looks the same as so(n), but is facing a different direction. However *we can rotate it to be coincident to so(n)*.

Body Velocity:

$$\hat{\omega}^b_{AB} = R^{-1}_{AB}\dot{R}_{AB}$$

Spatial Velocity:

$$\hat{\omega}^s_{AB} = \dot{R}_{AB}R^{-1}_{AB}$$

The spatial velocity is so(n) defined in the A (first) frame. The body velocity is so(n) defined in the B (second) frome.

Rigid Body Transformations

A rigid body transformation is an *affine* transformation consisting of a rotation followed by a translation

$$q_a = R_{ab} q_b + p_{ab}$$

Rigid body transformations have the following properties:

1. Length is preserved: $\|g(p) - g(q)\| = \|p - q\| \quad \forall p, q \in \mathbb{R}^3$
2. The orientation is preserved: $g(v \times w) = g(v) \times g(w) \quad \forall v, w \in \mathbb{R}^3$

Homogeneous Coordinates

Most of the mathematical tricks we have apply only to linear systems. Homogeneous coordinates are a clever bit of notation that make affine rigid body transforms into matrices (linear transforms)

$$q_a = R_{ab} q_b + p_{ab} \longrightarrow \begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} R_a b & p_a b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix}$$

Point: $\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ Vector: $\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$

Point - Point = Vector
Vector ± Vector = Vector
Point ± Vector = Point
Point + Point = n/a

Rigid Body Transforms as a Group

Rigid body transforms are also a Lie Group, the Special Euclidean Group SE(n)

$$SE(n) := \mathbb{R}^n \times SO(n)$$

Closure:

If $g_1, g_2 \in SE(n)$, then $g_1 g_2 \in SE(n)$

Identity:

$I \in SE(n)$

Associativity:

Matrix multiplication is associative

Inverse:

$$g^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \in SE(n)$$

The Lie Algebra se(n)

The Lie Algebra to SE(n) is se(n)

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \qquad \hat{\omega} \in so(n), v \in \mathbb{R}^n$$

While this is an nxn matrix, it can be defined by only n(n-1)/2 + n variables:

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

This representation of se(n) is called a twist. The wedge* operator is used to transform between the two notations.

* While it is technically defined as the wedge operator, everyone calls it a hat

Velocities of a Rigid Body

The velocities for rigid body transforms work exactly the same as the velocities for rotations!

Body Velocity:

$$\hat{\xi}^b_{AB} = g^{-1}_{AB}\dot{g}_{AB}$$

Spatial Velocity:

$$\hat{\xi}^s_{AB} = \dot{g}_{AB}g^{-1}_{AB}$$

Once again, the body velocity is defined as the Lie Algebra in the second frame (B), while the spatial velocity is defined as the Lie Algebra in the first (A).

Computing the Matrix Exponential

The matrix exponential is rather hard to compute for certain matrices, and computational solutions are prone to accumulating error. Thus we use the following formulas to analytically compute the matrix exponential:

SO(3): Rodrigues Formula

$$e^{\hat{\omega}\theta} = I + \hat{\omega}\sin(\theta) + \hat{\omega}^2(1 - \cos(\theta))$$

SE(3): Screw Theory

$$e^{\widehat{\xi}\theta} = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}$$

The Inverse Exponential: The Matrix Logarithm

The inverse of a matrix exponential is the matrix logarithm.

$$\theta = \arccos\left(\frac{tr(R^f (R^i)^{-1}) - 1}{2}\right)$$

$$\omega = \frac{1}{2\sin(T)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$$

$$v = [(I - e^{\hat{\omega}T})\hat{\omega} + \omega\omega^T]^{-1}(p_f - R_f R_i^{-1} p_i)$$

Forward Kinematics

The pose of the end effector is just a position and an orientation. Thus it can be represented as a rigid body transform. To find the forward kinematics of a robot manipulator, simply start with the initial end effector pose $g_{st}(0)$ and left multiply the transform from each joint from the end effector to the base. All twists should be defined in the base frame.

$$g_{st} = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}e^{\hat{\xi}_4\theta_4}g_{st}(0)$$

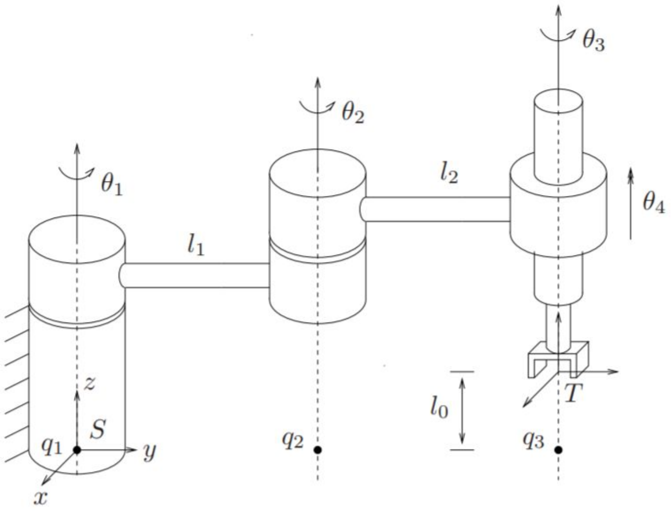


Figure from: MLS Chapter 3.2

Inverse Kinematics

Forward Kinematics is not an injective function; multiple sets of joint angles can result in the same end effector configuration. Thus, the inverse kinematics will have multiple solutions in general.

- An arbitrary 6-DOF manipulator has 16
- An elbow manipulator has 8
- A 7-DOF manipulator (like Baxter) has infinite solutions

There are two main problems with inverse kinematics:

1. How do you find a solution at all?
2. Which solution do you pick?

Sources

MLS Ch 2, Ch 3 Appendix A

Wikipedia:

- https://en.wikipedia.org/wiki/Orthogonal_group
- https://en.wikipedia.org/wiki/Lie_group
- https://en.wikipedia.org/wiki/Lie_algebra

Wolfram Alpha:

- Rowland, Todd and Weisstein, Eric W. "Group." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/Group.html