Lecture 7.
Representing Rotation

Matthew T. Mason

Mechanics of Manipulation
Fall 2013

Today's outline

Kinematic representation: goals, overview

Planar displacements

Spatial rotations

- Preview
- Axis-angle
- Rodrigues's formula
- Rotation matrices
- Euler angles

Readings, etc.

- We are starting chapter 3 of the text
- Lots of stuff online on representing rotations
- Murray, Li, and Sastry for matrix exponential
- Roth, Crenshaw, Ohwovoriole, Salamin, all cited in text

Analytic geometry

So far, Euclidean geometry. Why?

- Insight
- Visualization
- Economy of expression

Now Cartesian, analytic geometry. Why?

- Beyond 2D, beyond 3D. We need to work with high dimensional configuration spaces!
- For implementation
- For additional insight

The best of all possible worlds: use both. Understand geometrical or physical meaning for all terms.

Agenda for kinematic representation

Following the kinematic agenda:

- Planar displacements
- Spherical displacements
- Spatial displacements
- Constraints

Representing planar displacements

Obvious idea 1

- Displacement is rotation or translation
- Choose a coordinate frame $(O, \hat{x}, \hat{y})$
- For rotation, $(\text{center}, \text{angle})$, i.e. $((x, y), \theta)$
- For translation, (Δ_x, Δ_y)
- Ugly

Obvious idea 2

- Given O, displacement is rotation about O; translation
- Choose a coordinate frame $(O, \hat{x}, \hat{y})$
- $(\Delta_x, \Delta_y, \theta)$

What are representations for?

- Obvious ideas didn't yield homogeneous coordinate transform matrices?
- We didn't consider all the uses of representations!

Uses of representations

- Communicate with humans and computers
- Operate on points, lines and stuff
- Compose
- Sample, interpolate, average, smooth
- Differentiate, integrate

Operating on stuff

- $(\Delta_x, \Delta_y, \theta)$ is good for communication. How would you operate on points? Composition? Averaging? Sampling?
- To operate on points:
  - Represent points by Cartesian coordinates: (x, y)
  - Rotate using rotation matrix
  - Translate using component-wise addition
  - Tidy it up using homogeneous coordinates
  - We will revisit later

Why representing rotations is hard.

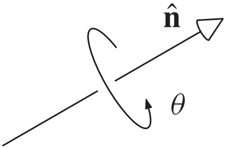
- Rotations do not commute. Vectors are out.
- For computation we like to represent things with real numbers, so our representations all live in $\mathbb{R}^n$.
- Even though $SO(3)$ is a three-dimensional space, it has the topology of projective three space $\mathbb{P}^3$, which cannot be smoothly mapped to $\mathbb{R}^3$.
- *And,* we have lots of different applications, with different requirements: communication, operating on things, composition, interpolation, etc.

Choices

- Axis-angle
  - Good for communication, geometrical insight
- Rotation matrices
  - Good for operating on stuff, composition, analytical insight
- Unit quaternions (aka Euler parameters)
  - Good for composition, analytical insight, sampling
- Euler angles
  - Good for communication, geometrical insight

Axis-angle

Euler's theorem: every spatial rotation has a rotation axis.



- Let O, $\hat{\mathbf{n}}$, θ, be ...
- Let $\text{rot}(\hat{\mathbf{n}}, \theta)$ be the corresponding rotation.
  - Many to one:
    - $\text{rot}(-\hat{\mathbf{n}}, -\theta) = \text{rot}(\hat{\mathbf{n}}, \theta)$
    - $\text{rot}(\hat{\mathbf{n}}, \theta + 2k\pi) = \text{rot}(\hat{\mathbf{n}}, \theta)$, for any integer k.
    - So, restrict θ to $[0, \pi]$. But not smooth at the edges.
    - When $\theta = 0$, the rotation axis is indeterminate, giving an infinity-to-one mapping.
    - Again you can fix by adopting a convention for $\hat{\mathbf{n}}$, but result is not smooth.
    - (Or, what about using the product, $\theta\hat{\mathbf{n}}$? Later.)

What do we want from axis-angle?

- Operate on points
  - Rodrigues's formula
- Compose rotations, average, interpolate, sampling, . . . ?
  - Not using axis-angle
- Convert to other representations? There aren't any yet. But, later we will use axis-angle *big time*. It's very close to *quaternions*.

Rodrigues's formula

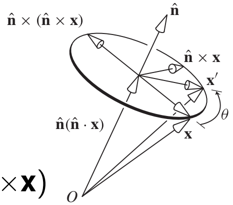
Others derive Rodrigues's formula using rotation matrices: ugly and messy. The geometrical approach is clean and insightful.

- Given point $\mathbf{x}$, decompose into components parallel and perpendicular to the rotation axis

$$\mathbf{x} = \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \mathbf{x}) - \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{x})$$

- Only $\mathbf{x}_\perp$ is affected by the rotation, yielding *Rodrigues's formula*:

$$\mathbf{x}' = \hat{\mathbf{n}}(\hat{\mathbf{n}}\cdot\mathbf{x}) + \sin\theta\ (\hat{\mathbf{n}}\times\mathbf{x}) - \cos\theta\ \hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x})$$



- A common variation:

$$\mathbf{x}' = \mathbf{x} + (\sin\theta)\ \hat{\mathbf{n}}\times\mathbf{x} + (1-\cos\theta)\ \hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x})$$

Rotation matrices

- Choose O on rotation axis. Choose frame $(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)$.
- Let $(\hat{\mathbf{u}}'_1, \hat{\mathbf{u}}'_2, \hat{\mathbf{u}}'_3)$ be the image of that frame.
- Write the $\hat{\mathbf{u}}'_i$ vectors in $\hat{\mathbf{u}}_i$ coordinates, and collect them in a matrix:

$$\hat{\mathbf{u}}'_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}'_1 \\ \hat{\mathbf{u}}_2 \cdot \hat{\mathbf{u}}'_1 \\ \hat{\mathbf{u}}_3 \cdot \hat{\mathbf{u}}'_1 \end{pmatrix}$$

$$\hat{\mathbf{u}}'_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}'_2 \\ \hat{\mathbf{u}}_2 \cdot \hat{\mathbf{u}}'_2 \\ \hat{\mathbf{u}}_3 \cdot \hat{\mathbf{u}}'_2 \end{pmatrix}$$

$$\hat{\mathbf{u}}'_3 = \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}'_3 \\ \hat{\mathbf{u}}_2 \cdot \hat{\mathbf{u}}'_3 \\ \hat{\mathbf{u}}_3 \cdot \hat{\mathbf{u}}'_3 \end{pmatrix}$$

$$A = (a_{ij}) = (\hat{\mathbf{u}}'_1 | \hat{\mathbf{u}}'_2 | \hat{\mathbf{u}}'_3)$$

So many numbers!

- A rotation matrix has nine numbers,
- but spatial rotations have only three degrees of freedom,
- leaving six excess numbers ...
- There are six constraints that hold among the nine numbers.

$$|\hat{\mathbf{u}}'_1| = |\hat{\mathbf{u}}'_2| = |\hat{\mathbf{u}}'_3| = 1$$
$$\hat{\mathbf{u}}'_3 = \hat{\mathbf{u}}'_1 \times \hat{\mathbf{u}}'_2$$

- *i.e.* the $\hat{\mathbf{u}}'_i$ are unit vectors forming a right-handed coordinate system.
- Such matrices are called *orthonormal* or *rotation* matrices.

Rotating a point

- Let (x_1, x_2, x_3) be coordinates of $\mathbf{x}$ in frame $(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)$.
- Then $\mathbf{x}'$ is given by the same coordinates taken in the $(\hat{\mathbf{u}}'_1, \hat{\mathbf{u}}'_2, \hat{\mathbf{u}}'_3)$ frame:

$$
\begin{aligned}
\mathbf{x}' &= x_1\hat{\mathbf{u}}'_1 + x_2\hat{\mathbf{u}}'_2 + x_3\hat{\mathbf{u}}'_3 \\
&= x_1 A\hat{\mathbf{u}}_1 + x_2 A\hat{\mathbf{u}}_2 + x_3 A\hat{\mathbf{u}}_3 \\
&= A(x_1\hat{\mathbf{u}}_1 + x_2\hat{\mathbf{u}}_2 + x_3\hat{\mathbf{u}}_3) \\
&= A\mathbf{x}
\end{aligned}
$$

- So rotating a point is implemented by ordinary matrix multiplication.

Sub- and superscript notation for rotating a point

- Let A and B be coordinate frames.
- Let ${}^{A}\mathbf{x}$ be coordinates in frame A.
- Let ${}^{B}_{A}R$ be the rotation matrix that rotates frame B to frame A.
- Then (see previous slide) ${}^{B}_{A}R$ represents the rotation of the point x:

$$
{}^{B}\mathbf{x}' = {}^{B}_{A}R \; {}^{B}\mathbf{x}
$$

- Note presuperscripts all match. Both points, and xform, must be written in same coordinate frame.

Coordinate transform

There is another use for ${}^B_A R$:

- ${}^A\mathbf{x}$ and ${}^B\mathbf{x}$ represent the same point, in frames A and B resp.
- To transform from A to B:

$$ {}^B\mathbf{x} = {}^B_A R \; {}^A\mathbf{x} $$

- For coord xform, matrix subscript and vector superscript "cancel".

Rotation from B to A is the same as coordinate transform from A to B.

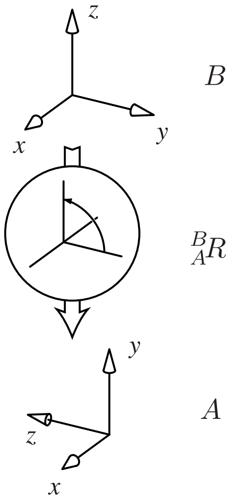
Example rotation matrix

$$ {}^B_AR = \left(\; {}^B\mathbf{x}_A \;\middle|\; {}^B\mathbf{y}_A \;\middle|\; {}^B\mathbf{z}_A \; \right) $$

$$ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} $$

How to remember what ${}^B_AR$ does? Pick a coordinate axis and see. The x axis isn't very interesting, so try y:

$$ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} $$



Nice things about rotation matrices

- Composition of rotations: $\{R_1; R_2\} = R_2 R_1$. ($\{x; y\}$ means do x then do y.)
- Inverse of rotation matrix is its transpose ${}^B_A R^{-1} = {}^A_B R = {}^B_A R^T$.
- Coordinate xform of a rotation matrix:

$$ {}^B R = {}^B_A R \, {}^A R \, {}^A_B R $$

Converting rot($\hat{\mathbf{n}}, \theta$) to R

- Ugly way: define frame with $\hat{\mathbf{z}}$ aligned with $\hat{\mathbf{n}}$, use coordinate xform of previous slide.
- Keen way: Rodrigues's formula!

$$\mathbf{x}' = \mathbf{x} + (\sin\theta)\ \hat{\mathbf{n}} \times \mathbf{x} + (1 - \cos\theta)\ \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{x})$$

- Define "cross product matrix" N:

$$N = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}$$

so that

$$N\mathbf{x} = \hat{\mathbf{n}} \times \mathbf{x}$$

. . . using Rodrigues's formula . . .

- Substituting the cross product matrix N into Rodrigues's formula:

$$\mathbf{x}' = \mathbf{x} + (\sin\theta)N\mathbf{x} + (1 - \cos\theta)N^2\mathbf{x}$$

- Factoring out $\mathbf{x}$

$$R = I + (\sin\theta)N + (1 - \cos\theta)N^2$$

- That's it! Rodrigues's formula in matrix form. If you want to you could expand it:

$$\begin{pmatrix} n_1^2 + (1 - n_1^2)c\theta & n_1 n_2(1 - c\theta) - n_3 s\theta & n_1 n_3(1 - c\theta) + n_2 s\theta \\ n_1 n_2(1 - c\theta) + n_3 s\theta & n_2^2 + (1 - n_2^2)c\theta & n_2 n_3(1 - c\theta) - n_1 s\theta \\ n_1 n_3(1 - c\theta) - n_2 s\theta & n_2 n_3(1 - c\theta) + n_1 s\theta & n_3^2 + (1 - n_3^2)c\theta \end{pmatrix}$$

where $c\theta = \cos\theta$ and $s\theta = \sin\theta$. Ugly.

Rodrigues's formula for differential rotations

Consider Rodrigues's formula for a differential rotation $\text{rot}(\hat{\mathbf{n}}, d\theta)$.

$$\begin{aligned}\mathbf{x}' &= (I + \sin d\theta N + (1 - \cos d\theta) N^2)\mathbf{x} \\ &= (I + d\theta N)\mathbf{x}\end{aligned}$$

so

$$\begin{aligned} d\mathbf{x} &= N\mathbf{x}\, d\theta \\ &= \hat{\mathbf{n}} \times \mathbf{x}\, d\theta\end{aligned}$$

It follows easily that differential rotations are vectors: you can scale them and add them up. We adopt the convention of representing angular velocity by the unit vector $\hat{\mathbf{n}}$ times the angular velocity.

Converting from R to rot$(\hat{\mathbf{n}}, \theta) \ldots$

- Problem: $\hat{\mathbf{n}}$ isn't defined for $\theta = 0$.
- We will do it indirectly. Convert R to a unit quaternion (next lecture), then to axis-angle.

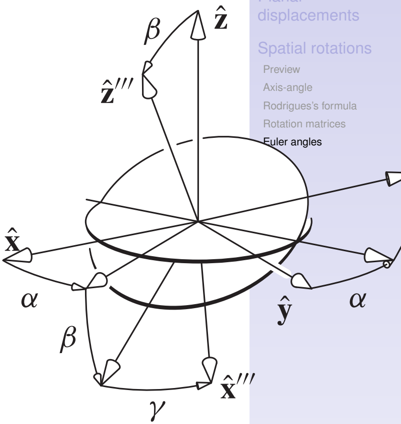
Euler angles

- Three numbers to describe spatial rotations. *ZYZ* convention:

$$(\alpha, \beta, \gamma) \mapsto \text{rot}(\gamma, \hat{\mathbf{z}}'')\,\text{rot}(\beta, \hat{\mathbf{y}}')\,\text{rot}(\alpha, \hat{\mathbf{z}})$$

- Can we represent an arbitrary rotation?

  Rotate α *about* $\hat{\mathbf{z}}$ *until* $\hat{\mathbf{y}}' \perp \hat{\mathbf{z}}'''$;
  Rotate β *about* $\hat{\mathbf{y}}'$ *until* $\hat{\mathbf{z}}'' \parallel \hat{\mathbf{z}}'''$;
  Rotate γ *about* $\hat{\mathbf{z}}''$ *until* $\hat{\mathbf{y}}'' = \hat{\mathbf{y}}'''$.



- Note two choices for $\hat{\mathbf{y}}'$ ...
- ...except sometimes infinite choices.

Converting from Euler angles to rotation matrices

notation

- Define frames $\{0\},\{1\},\{2\},\{3\}$ so that
- rot$(\alpha, \hat{\mathbf{z}})$ maps $\{0\}$ to $\{1\}$, etc.,
- As before ${}^{i}_{j}R$ is the rotation matrix rotating frame $\{i\}$ to frame $\{j\}$, written in frame $\{i\}$ coordinates.
- Let ${}^{k}({}^{i}_{j}R)$ be the same matrix, written in frame $\{k\}$ coordinates.
- Then the correct sequence, written in a common coordinate frame, would be

$$ {}^{0}_{3}R = {}^{0}({}^{2}_{3}R)\,{}^{0}({}^{1}_{2}R)\,{}^{0}({}^{0}_{1}R) $$

Moving frame versus fixed frame

Switch to moving frame

- Use the coordinate transform of a matrix formula:

$$
\begin{aligned}
{}^0_3R &= {}^0({}^2_3R)\,{}^0({}^1_2R)\,{}^0_1R \\
&= ({}^0_2R\,{}^2_3R\,{}^2_0R)\,({}^0_1R\,{}^1_2R\,{}^1_0R)\,{}^0_1R \\
&= {}^0_1R\,{}^1_2R\,{}^2_3R
\end{aligned}
$$

- *Wow!* You can switch between moving frame and fixed frame, if you also switch the order!
- You could also have derived the above, just by interpreting ${}^0_3R$ as a coordinate transform.

From (α, β, γ) to R

$$
\begin{aligned}
{}^0_3R &= {}^0_1R\,{}^1_2R\,{}^2_3R \\
&= \begin{pmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{pmatrix} \begin{pmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} c\alpha\, c\beta\, c\gamma - s\alpha\, s\gamma & -c\alpha\, c\beta\, s\gamma - s\alpha\, c\gamma & c\alpha\, s\beta \\ s\alpha\, c\beta\, c\gamma + c\alpha\, s\gamma & -s\alpha\, c\beta\, s\gamma + c\alpha\, c\gamma & s\alpha\, s\beta \\ -s\beta\, c\gamma & s\beta\, s\gamma & c\beta \end{pmatrix}
\end{aligned}
$$

From R to (α, β, γ) the ugly way

- Case 1: $r_{33} = 1$, $\beta = 0$. $\alpha - \gamma$ is indeterminate.

$$R = \begin{pmatrix} \cos(\alpha+\gamma) & -\sin(\alpha+\gamma) & 0 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Case 2: $r_{33} = -1$, $\beta = \pi or - \pi$. $\alpha + \gamma$ is indeterminate.

$$R = \begin{pmatrix} -\cos(\alpha-\gamma) & -\sin(\alpha-\gamma) & 0 \\ -\sin(\alpha-\gamma) & \cos(\alpha-\gamma) & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

- For generic case: solve 3rd column for β. (Sign is free choice.) Solve third column for α and third row for γ.
- . . . but there are numerical issues . . .

From R to (α, β, γ) the clean way

- Let

$$\sigma = \alpha + \gamma$$
$$\delta = \alpha - \gamma$$

- Then

$$r_{22} + r_{11} = \cos\sigma(1 + \cos\beta)$$
$$r_{22} - r_{11} = \cos\delta(1 - \cos\beta)$$
$$r_{21} + r_{12} = \sin\delta(1 - \cos\beta)$$
$$r_{21} - r_{12} = \sin\sigma(1 + \cos\beta)$$

(No special cases for $\cos\beta = \pm 1$?)

- Solve for σ and δ, then for α and γ, then finally

$$\beta = \tan^{-1}(r_{13}\cos\alpha + r_{23}\sin\alpha, r_{33})$$

But what about $\sin\beta = 0$?

- How can this method work without explicitly addressing the singularity?
- When $\beta = 0$, σ is determined and δ is not. When $\beta = \pi$, δ is determined and σ is not.
- If your *tan*$^{-1}$ handles $(0, 0)$, you can just let it go!