Kinematics

Stefano Carpin

Department of Computer Science and Engineering
School of Engineering
University of California, Merced
https://sites.ucmerced.edu/scarpin
https//robotics.ucmerced.edu

Motivation and Objectives

Kinematics: the study of motion without considering its causes (position, velocity, acceleration).

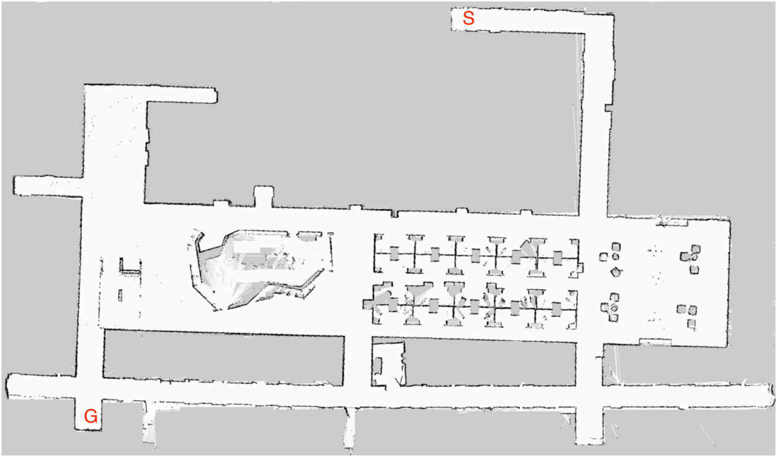
Motivation

- robots operate in the three dimensional world
- algorithms must reason and make decisions about the space in which robots move

Objective

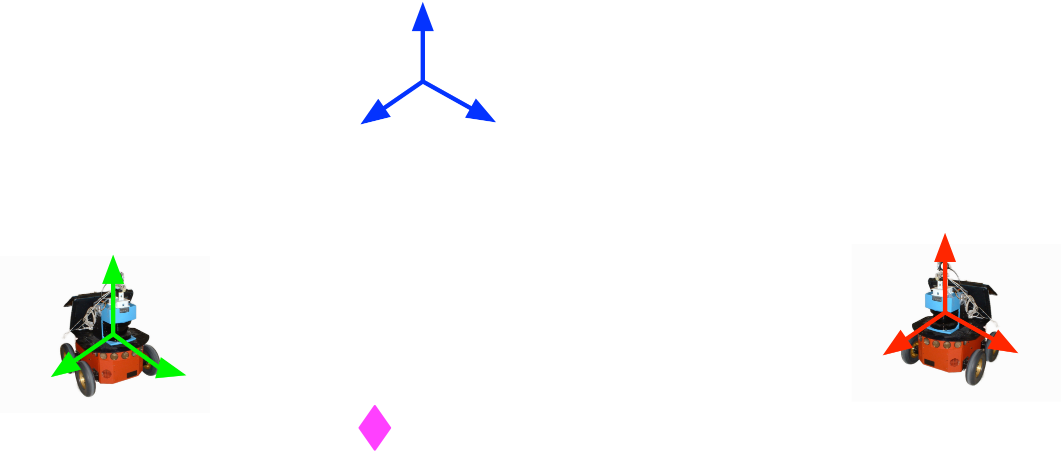
- introduce mathematical models to represent objects in space
- develop efficient algorithms to manipulate these models
- illustrate how these concepts are implemented in ROS

Why Kinematics?

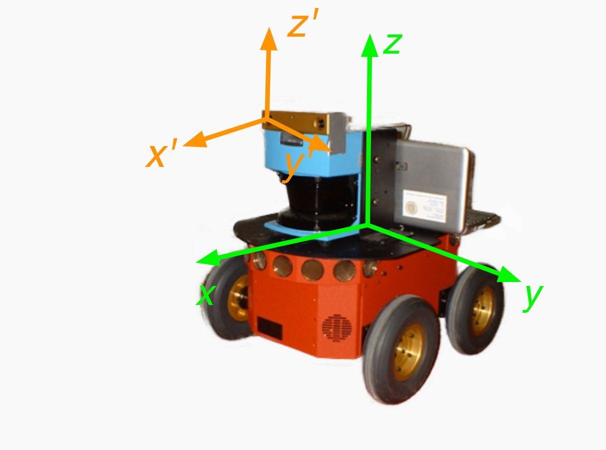


Why Kinematics?

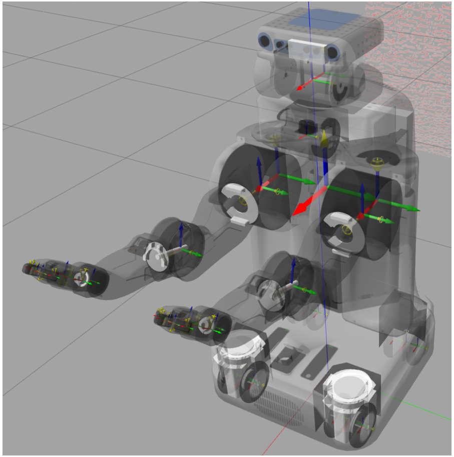
Why Kinematics?



Why Kinematics?



Why Kinematics?



Terminology and Assumptions

- we assume the existence of a so called *world* frame, i.e., a reference frame with respect to which coordinates and angles are expressed
- we abstract the robot as a *rigid object*
- its pose in the world is defined by its *position* and *orientation*
- position and orientation will be collectively called pose
- assumption: a rigid frame is attached to the robot
  - *Rigidly attached* means that whenever the robot moves the frame moves, and vice versa
- we reason about the pose of the frame as a proxy for the pose of the robot

Definition

A body $\mathcal{B}$ is said to be a *rigid body* if the Euclidean distance between any two points in $\mathcal{B}$ is constant.

Definition

A frame of reference is given by an origin point O and three orthonormal vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ called axes satisfying the so-called right hand rule, i.e.,

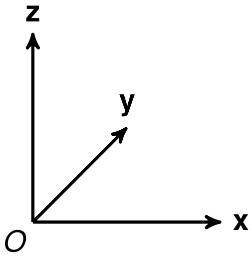
$$\mathbf{x} \times \mathbf{y} = \mathbf{z} \qquad \mathbf{y} \times \mathbf{z} = \mathbf{x} \qquad \mathbf{z} \times \mathbf{x} = \mathbf{y}.$$

Orthonormal definition:

$$\mathbf{x} \cdot \mathbf{x} = 1 \qquad \mathbf{y} \cdot \mathbf{y} = 1 \qquad \mathbf{z} \cdot \mathbf{z} = 1 \qquad \mathbf{x} \cdot \mathbf{y} = 0 \qquad \mathbf{y} \cdot \mathbf{z} = 0 \qquad \mathbf{z} \cdot \mathbf{x} = 0.$$

We indicate a frame as $O - \mathbf{xyz}$ and give symbolic names to frames, like A, B, etc.

Frame Representation

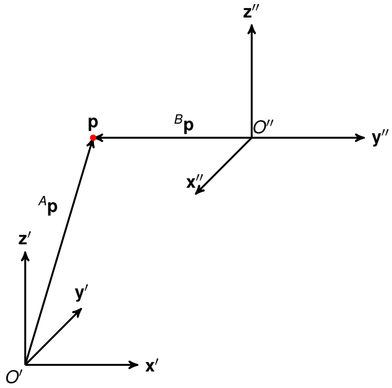


Note the orientation of the axes (right handed)

Coordinates in Multiple Frames

- a point $\mathbf{p}$ in $\mathbb{R}^n$ is represented by n values (coordinates)
- coordinate values are always expressed in a reference frame
- in robotics application there will be many reference systems; so it is important to specify to which reference system the coordinates are referred to
- a leading superscript is used to indicate the reference system
- ${}^{A}\mathbf{p}$ indicates the vector with the coordinates of point $\mathbf{p}$ referred to reference frame A while ${}^{B}\mathbf{p}$ indicates the coordinates of the same point but referred to reference system B

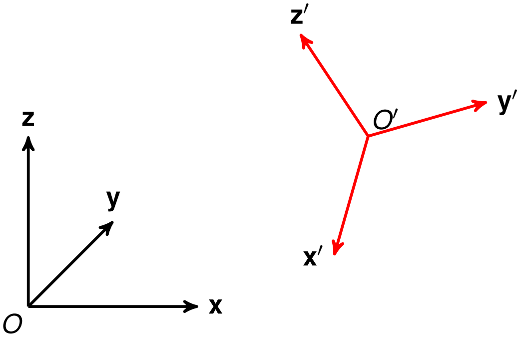
Coordinates in Multiple Frames



In this case $^A\mathbf{p} = [0.6\ 2\ 2]^T$ and $^B\mathbf{p} = [0\ {-1.4}\ 0]^T$

Representing a Frame with Respect to a Different Frame

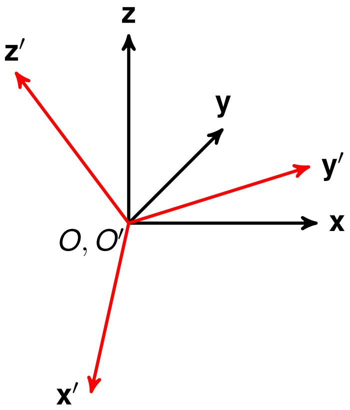
Often we need to represent a frame $B = O' - \mathbf{x}'\mathbf{y}'\mathbf{z}'$ with respect to another frame $A = O - \mathbf{xyz}$.



Representing a Frame with Respect to a Different Frame

- must represent its origin O', i.e., a point. We know how to do that, i.e., ${}^{A}\mathbf{p}$ or ${}^{A}O'$.
- must represent its three orthonormal axes $\mathbf{x}'\mathbf{y}'\mathbf{z}'$
- consider translating frame B so that its origin coincides with frame A. Then take the coordinates of the endpoints of $\mathbf{x}'\mathbf{y}'\mathbf{z}'$

Representing a Frame with Respect to a Different Frame



Representing a Frame with Respect to a Different Frame

Coordinates of the end-point of axis $\mathbf{x}'$

$$^{A}\mathbf{x}' = \begin{bmatrix} \mathbf{x}' \cdot \mathbf{x} \\ \mathbf{x}' \cdot \mathbf{y} \\ \mathbf{x}' \cdot \mathbf{z} \end{bmatrix}$$

Same for $\mathbf{y}'$ and $\mathbf{z}'$

$$^{A}\mathbf{y}' = \begin{bmatrix} \mathbf{y}' \cdot \mathbf{x} \\ \mathbf{y}' \cdot \mathbf{y} \\ \mathbf{y}' \cdot \mathbf{z} \end{bmatrix} \qquad ^{A}\mathbf{z}' = \begin{bmatrix} \mathbf{z}' \cdot \mathbf{x} \\ \mathbf{z}' \cdot \mathbf{y} \\ \mathbf{z}' \cdot \mathbf{z} \end{bmatrix}.$$

Rotation Matrix

These three vectors are compactly represented as the three columns of a 3×3 matrix called the rotation matrix

$$
{}^{A}_{B}\mathbf{R} = \begin{bmatrix} \mathbf{x}' \cdot \mathbf{x} & \mathbf{y}' \cdot \mathbf{x} & \mathbf{z}' \cdot \mathbf{x} \\ \mathbf{x}' \cdot \mathbf{y} & \mathbf{y}' \cdot \mathbf{y} & \mathbf{z}' \cdot \mathbf{y} \\ \mathbf{x}' \cdot \mathbf{z} & \mathbf{y}' \cdot \mathbf{z} & \mathbf{z}' \cdot \mathbf{z} \end{bmatrix}
$$

The subscript B and superscript A indicate that this matrix describes the rotation of frame B with respect to frame A. The order is important: ${}^{A}_{B}\mathbf{R} \neq {}^{B}_{A}\mathbf{R}$.

Representing a Frame with Respect to a Different Frame

In conclusion, a frame can be represented by:

- a vector to describe its origin
- a rotation matrix to describe its axes, i.e., its orientation.

This is the *pose* of the frame (position and orientation), or the pose of the rigid body (robot) attached to the frame.

Inverse Representation

What if we instead want ${}^B_A\mathbf{R}$?
Same reasoning, but we consider the projections of the axis **xyz** along the axis $\mathbf{x}'\mathbf{y}'\mathbf{z}'$

$$
{}^B_A\mathbf{R} = \begin{bmatrix} \mathbf{x}\cdot\mathbf{x}' & \mathbf{y}\cdot\mathbf{x}' & \mathbf{z}\cdot\mathbf{x}' \\ \mathbf{x}\cdot\mathbf{y}' & \mathbf{y}\cdot\mathbf{y}' & \mathbf{z}\cdot\mathbf{y}' \\ \mathbf{x}\cdot\mathbf{z}' & \mathbf{y}\cdot\mathbf{z}' & \mathbf{z}\cdot\mathbf{z}' \end{bmatrix}.
$$

The dot product is commutative, and therefore ${}^B_A\mathbf{R} = {}^A_B\mathbf{R}^T$.

Change of Coordinates

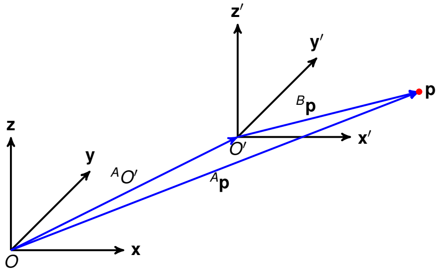
Example: two robots seen in the beginning

Problem statement: Given $A = O - \mathbf{xyz}$ and $B = O' - \mathbf{x'y'z'}$ and ${}^B\mathbf{p} = [p_x \; p_y \; p_z]^T$, determine ${}^A\mathbf{p}$.

We want to *change* the coordinates of $\mathbf{p}$ from frame B to frame A.

Change of Coordinates: case 1

Frames are parallel but do not have the same origin

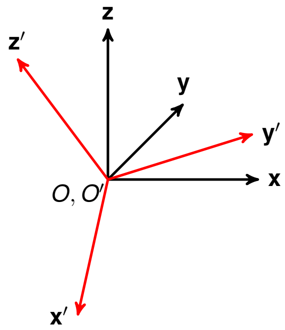


Let $^AO' = [\Delta x \ \Delta y \ \Delta z]^T$. Then coordinates of **p** in frame A are obtained by translation, i.e.,

$$^A\mathbf{p} = {}^B\mathbf{p} + {}^AO' = \begin{bmatrix} p_x + \Delta x \\ p_y + \Delta y \\ p_z + \Delta z \end{bmatrix}$$

Change of Coordinates: case 2

Frames have the same origin but are not parallel



$$ {}^{B}\mathbf{p} = p_x \mathbf{x}' + p_y \mathbf{y}' + p_z \mathbf{z}' $$

Change of Coordinates: case 2

Recall:

$$ {}^{A}_{B}\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

Therefore

$$ \begin{aligned} {}^{A}\mathbf{x}' &= r_{11}\mathbf{x} + r_{21}\mathbf{y} + r_{31}\mathbf{z} \\ {}^{A}\mathbf{y}' &= r_{12}\mathbf{x} + r_{22}\mathbf{y} + r_{32}\mathbf{z} \\ {}^{A}\mathbf{z}' &= r_{13}\mathbf{x} + r_{23}\mathbf{y} + r_{33}\mathbf{z}. \end{aligned} $$

Change of Coordinates: case 2

Substitute into ${}^{B}\mathbf{p}$

$$
\begin{aligned}
{}^{A}\mathbf{p} &= p_x(r_{11}\mathbf{x} + r_{21}\mathbf{y} + r_{31}\mathbf{z}) + p_y(r_{12}\mathbf{x} + r_{22}\mathbf{y} + r_{32}\mathbf{z}) + p_z(r_{13}\mathbf{x} + r_{23}\mathbf{y} + r_{33}\mathbf{z}) \\
&= (p_x r_{11} + p_y r_{12} + p_z r_{13})\mathbf{x} + (p_x r_{21} + p_y r_{22} + p_z r_{23})\mathbf{y} + (p_x r_{31} + p_y r_{32} + p_z r_{33})\mathbf{z}.
\end{aligned}
$$

Recall definition of matrix-vector multiplication: ${}^{A}\mathbf{p} = {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{p}$
To remember: superscripts and subscripts cancel out *diagonally*.

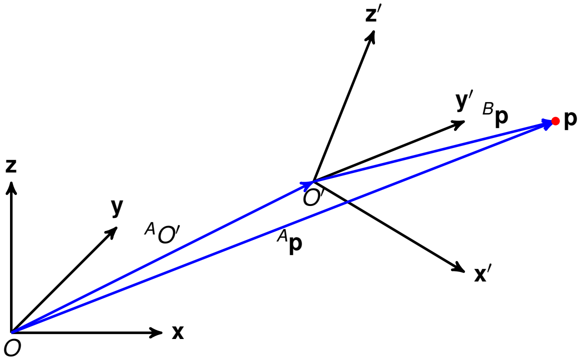
Example

Let ${}^{B}\mathbf{p} = [0\ 2\ 1]^T$ and

$$ {}^{A}_{B}\mathbf{R} = \begin{bmatrix} 0.70710678 & 0 & 0.70710678 \\ 0.61237244 & 0.5 & -0.61237244 \\ -0.35355339 & 0.8660254 & 0.35355339 \end{bmatrix} $$

To get coordinates of **p** with respect to A compute ${}^{A}\mathbf{p} = {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{p} = [0.70710678\ 0.38762756\ 2.0856042]^T$.

Change of Coordinates: General Case



Must first translate (overlap origins) and then rotate (align axes)

$$^A\mathbf{p} = {}^A_B\mathbf{R}\,{}^B\mathbf{p} + {}^AO'$$

Example

Consider the previous figure. Assume ${}^{B}\mathbf{p} = [1\ 4\ 2]^T$. Let us furthermore assume that we have the description of frame B with respect to frame A, i.e., ${}^{A}O' = [7\ \ -2\ 1]^T$ and

$$
{}^{A}_{B}\mathbf{R} = \begin{bmatrix} 0.70710678 & 0 & 0.70710678 \\ 0.61237244 & 0.5 & -0.61237244 \\ -0.35355339 & 0.8660254 & 0.35355339 \end{bmatrix}
$$

What is ${}^{A}\mathbf{p}$? Apply formula ${}^{A}\mathbf{p} = {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{p} + {}^{A}O'$ obtaining ${}^{A}\mathbf{p} = [9.12132034\ \ -0.61237244\ 4.81765501]^T$.

Rotation Matrices

3×3 matrices subject to multiple constraints (*special orthogonal*)

1. each of its columns has length 1;
2. its columns are mutually orthogonal;
3. its determinant is 1.

Despite having 9 elements, they are defined by three parameters (more later)

Inverse of a Rotation Matrix

Theorem

The transpose of a rotation matrix is equal to its inverse, i.e., if $\mathbf{R}$ *is a rotation matrix, then* $\mathbf{R}^T = \mathbf{R}^{-1}$.

Proof: see whiteboard and lecture notes.

Combining theorem and previous result:

$${}^B_A\mathbf{R} = {}^A_B\mathbf{R}^T = [{}^A_B\mathbf{R}]^{-1}$$

Example

Rotation matrix expressing rotation of frame B with respect to A:

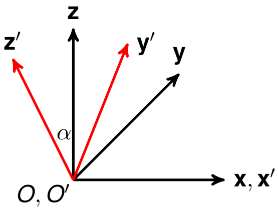
$$
{}^{A}_{B}\mathbf{R} = \begin{bmatrix} 0.9211 & -0.3894 & 0 \\ 0.3894 & 0.9211 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

The matrix expressing the rotation of A with respect to B is its inverse, i.e., ${}^{B}_{A}\mathbf{R} = [{}^{A}_{B}\mathbf{R}]^{-1}$.
According to the theorem: it is ${}^{A}_{B}\mathbf{R}$

$$
{}^{B}_{A}\mathbf{R} = \begin{bmatrix} 0.9211 & 0.3894 & 0 \\ -0.3894 & 0.9211 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

Elementary Rotation Matrices

General rotations can be obtained as composition of *simpler* rotations.
Assume two frames are coincident and then rotate one of the two about one common axis (move red frame $B = O' - \mathbf{x}'\mathbf{y}'\mathbf{z}'$ and keep black frame $A = O - \mathbf{xyz}$ fixed).
Start with rotation of angle α about $\mathbf{x}$



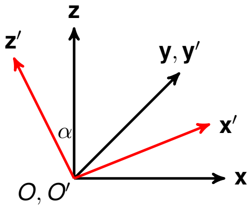
Elementary Rotation Matrices ($\mathbf{x}$ axis)

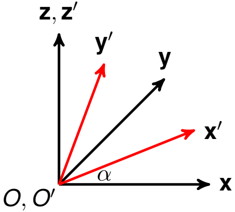
- $\mathbf{x}'$ did not move $\Rightarrow$ coordinates of the end point of axis $\mathbf{x}'$ in frame A are $[1 \quad 0 \quad 0]^T$
- coordinates of $\mathbf{y}'$ in frame A are $[0 \quad \cos\alpha \quad \sin\alpha]^T$.
- coordinates of $\mathbf{z}'$ in frame A are $[0 \quad -\sin\alpha \quad \cos\alpha]^T$.

Putting all together:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Elementary Rotation Matrices (**y** and **z** axes)



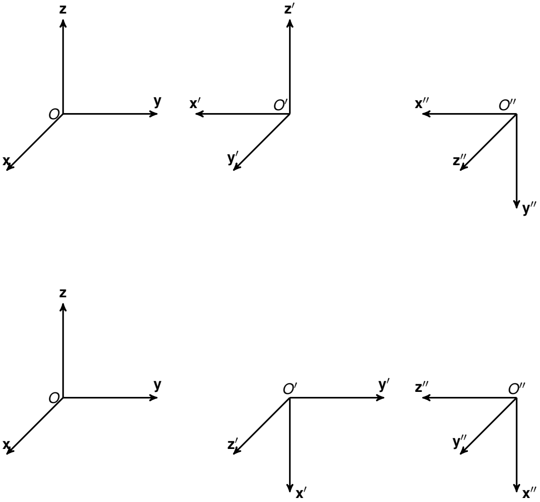


$$\mathbf{R}_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \qquad \mathbf{R}_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

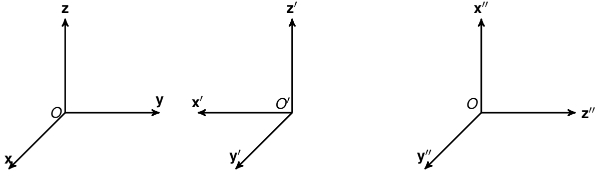
Composite Rotations

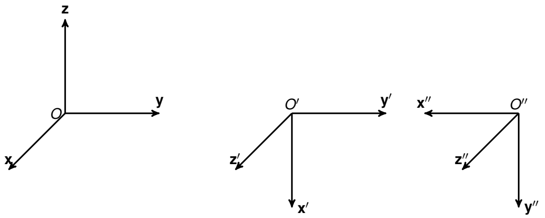
- rigid bodies can rotate about *arbitrary* axes
- generic rotations can be obtained composing elementary rotations
- important fact: rotations do not commute, i.e., the order matters
  - if a frame rotates first about **x** and then about **y**, the result is different from rotating first about **y** and then about **x**
- it is necessary to always specify about which axis rotations are performed, i.e., moving or fixed

Composite Rotations (about fixed axes)



Composite Rotations (about moving axes)





Rotation Matrices for Composite Rotations – Fixed Frame

- two initially aligned frames A and B; A is fixed, B moves
- B rotates of angle α about $\mathbf{x}$ and then of an angle β about $\mathbf{z}$ where $\mathbf{x}$ and $\mathbf{z}$ are the axes of the fixed frame A.
- What is the expression for ${}^A_B\mathbf{R}$?

Solution:

- A and B are aligned; after first rotation we get an intermediate rotation matrix ${}^A_B\mathbf{R}' = \mathbf{R}_x(\alpha)$
- Rotating a matrix corresponds to rotate all its axes $\Rightarrow$ apply second rotation to axes of intermediate matrix

$$ {}^A_B\mathbf{R} = \mathbf{R}_z(\beta)\mathbf{R}_x(\alpha) $$

Rotation Matrices for Composite Rotations – Fixed Frame

General case:

- Let A and B be two initially coincident
- B undergoes a sequence of n rotations $\mathbf{R}_1, \mathbf{R}_2 \ldots \mathbf{R}_n$, all expressed about the fixed frame A
- final expression:

$$^{A}_{B}\mathbf{R} = \mathbf{R}_n \ldots \mathbf{R}_2 \mathbf{R}_1$$

- Important: if rotations are about the fixed axes it is necessary to premultiply

Rotation Matrices for Composite Rotations – Moving Frame

General case:

- Let A and B be two initially coincident
- B undergoes a sequence of n rotations $\mathbf{R}_1, \mathbf{R}_2 \ldots \mathbf{R}_n$, all expressed about the moving frame B
- final expression:

$$^{A}_{B}\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2 \ldots \mathbf{R}_n.$$

- Important: if rotations are about the moving axes it is necessary to postmultiply

Example

Let A and B be two initially coincident frames. Frame B rotates first of α about the fixed axis $\mathbf{x}$, then β about the moving axis $\mathbf{z}$ and finally γ about the fixed axis $\mathbf{y}$.

Determine the final rotation matrix describing B in the frame A.

Answer (details on the whiteboard and on th lecture notes):

$$ {}^{A}_{B}\mathbf{B}''' = \mathbf{R}_y(\gamma)\,{}^{A}_{B}\mathbf{B}'' = \mathbf{R}_y(\gamma)\mathbf{R}_x(\alpha)\mathbf{R}_z(\beta). $$

Composite Rotations

Theorem

Let ${}^A_B\mathbf{R}$ and ${}^B_C\mathbf{R}$ be two rotation matrices describing the orientation of frame B with respect to frame A and the rotation of frame C with respect to frame B, respectively. Then

$$ {}^A_C\mathbf{R} = {}^A_B\mathbf{R}\,{}^B_C\mathbf{R} $$

i.e., their product gives the orientation of frame C with respect to frame A.

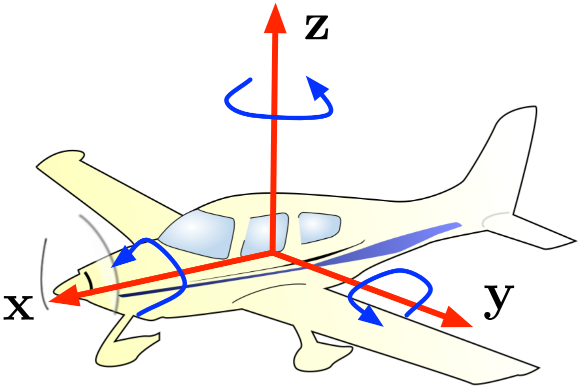
Proof: See lecture notes (repeated change of coordinates on the columns of the second matrix).

Simple rule: cancellations occur diagonally

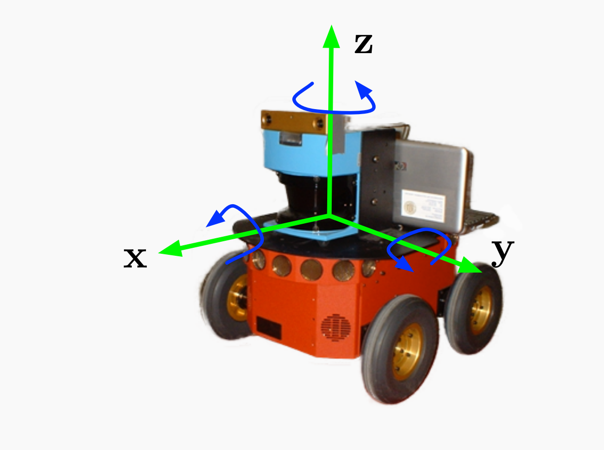
Rotation Parametrizations

- Rotation matrices: 9 elements with many constraints $\Rightarrow$ fully determined by three parameters (i.e., three angles)
- many different choices, but only few used in practice. We will see just two:
  - Euler angles
  - Roll-Pitch-Yaw

Roll-Pitch-Yaw



Roll-Pitch-Yaw



Roll-Pitch-Yaw

- rotation α about the **x** axis (roll)
- followed by a rotation of β about the **y** axis (pitch)
- and then by a rotation of γ about the **z** axis (yaw)

Since all rotations are about the fixed frame,

$$\mathbf{R} = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha).$$

Euler Angles

- three successive rotations about the *moving axes*
- order: ZYZ
- α about Z, then β about Y, and γ about Z
- since all rotations are about the moving axes:

$$\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$$

Inverse problems

Given a desired rotation matrix **R** determine the three angles giving such rotation (either Euler or Roll-Pitch-Yaw)
Input:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Output: α, β, γ.

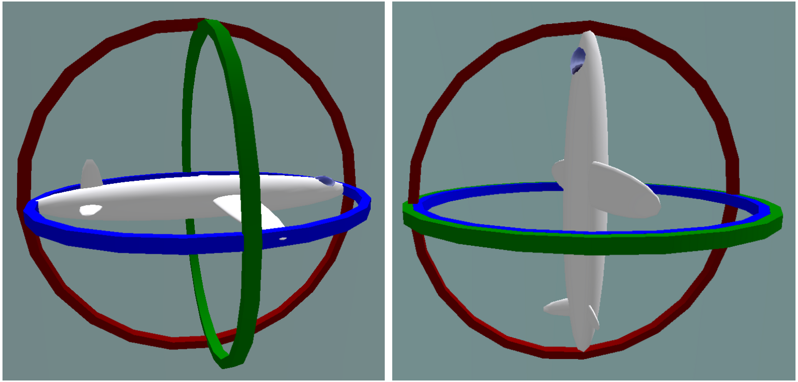
Inverse problems

- Good news: there exist closed form solutions for either parametrization
- Multiple solutions (one to many situation). Why?
- Ex: for Euler one set of solutions is the following (see lecture notes for other set, and roll, pitch, yaw)

$$\alpha = \text{atan2}(r_{23}, r_{13}) \qquad \beta = \text{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \qquad \gamma = \text{atan2}(r_{32}, -r_{31})$$

  - no need to recall them, but must know how to apply them
- in ROS there will be ready to use functions solving the problem for you

Gimbal Lock



Figures from https://en.wikipedia.org/wiki/Gimbal_lock

Gimbal Lock

Why does it happen? Let us explicitly write the rotation matrix for the ZYZ Euler angles

$$\mathbf{R} = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}$$

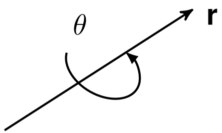
- If $\beta = k\pi$ then $\sin\beta = 0 \Rightarrow$ some terms are 0
- Inverse problem cannot be solved. Why? $\beta = k\pi$ implies first and third rotation axes are aligned, so final result is given by two rotations only.
- problem exists with every parametrization based on three angles
- problem also known as *singularity*

Quaternions

- alternative rotation parametrization that is not affected by singularities
- uses four parameters to represent one rotation
- complex algebraic structure pervasively used in robotics and computer graphics

Theorem (Euler's rotation theorem)

In three dimensions, every rotation is equivalent to a single rotation about an axis through the origin.



Quaternions

- similar to complex numbers, but with three *imaginary* components (i, j, k):
  $\mathbf{q} = a + bi + cj + dk$
- Properties of the imaginary components: $i^2 = j^2 = k^2 = ijk = -1$
- if $b = c = d = 0$ then we have a real number, and if $c = d = 0$ we have a complex number
- norm of a quaternion: $|\mathbf{q}| = \sqrt{a^2 + b^2 + c^2 + d^2}$
- *unit quaternion*: a quaternion of length 1

Why do we care about quaternions? ROS represents rotations using quaternions (not rotation matrices)

From quaternion to matrices

Let $\mathbf{q} = a + bi + cj + dk$ be a unit quaternion. Its associated rotation matrix is (some times written $\mathbf{R}(\mathbf{q})$)

$$\mathbf{R} = \begin{bmatrix} 2(a^2 + b^2) - 1 & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & 2(a^2 + c^2) - 1 & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & 2(a^2 + d^2) - 1 \end{bmatrix}$$

Exercise: verify that it is indeed a rotation matrix.

From quaternions to axis angle

Given a unit quaternion $\mathbf{q} = a + bi + cj + dk$ it is interesting to determine its parameters according to Euler's theorem.

- angle: $\theta = 2 \arccos a$.
- components of the rotation axis $\mathbf{r} = [r_1 \ r_2 \ r_3]^T$

$$r_1 = \frac{b}{\sin\frac{\theta}{2}} \qquad r_2 = \frac{c}{\sin\frac{\theta}{2}} \qquad r_3 = \frac{d}{\sin\frac{\theta}{2}}$$

Not defined when $\sin\frac{\theta}{2} = 0$, i.e., when $\theta = 0$ or $\theta = 2\pi \Rightarrow$
makes sense because for these two values there is no rotation at all.

From axis-angle to quaternions

Given θ and and a unary vector $\mathbf{r} = [r_1 \ r_2 \ r_3]^T$ the associated quaternion is

$$\mathbf{q} = \cos\frac{\theta}{2} + i\sin\frac{\theta}{2}r_1 + j\sin\frac{\theta}{2}r_2 + k\sin\frac{\theta}{2}r_3.$$

Non-uniqueness

- Mapping between quaternions and rotations is not one-to-one
- $\mathbf{q}$ is a unit quaternion $\Rightarrow -\mathbf{q}$ is a unit quaternion
- $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation matrix $\mathbf{R}$ (prove it!)
- as per Euler's theorem, a rotation of θ about $\mathbf{r}$ is equivalent to a rotation of $2\pi - \theta$ about $-\mathbf{r}$

From rotation matrices to quaternions

Given a rotation matrix $\mathbf{R}$, determine the equivalent unit quaternion $\mathbf{q} = a + bi + cj + dk$

$$a = \frac{\sqrt{r_{11} + r_{22} + r_{33} + 1}}{2} \qquad b = \frac{r_{32} - r_{23}}{4a} c = \frac{r_{13} - r_{31}}{4a} \qquad d = \frac{r_{21} - r_{12}}{4a}.$$

Quaternion algebra and rotating points with quaternions

- Given two quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ it is immediate to define the sum and product operation treating them as polynomials
- given $\mathbf{q} = a + bi + cj + dk$, its *conjugate* is $\mathbf{q}^* = a - bi - cj - dk$
- let $\mathbf{p} = [p_x\ p_y\ p_z]^T$ be a point $\Rightarrow \mathbf{p}' = 0 + p_x i + p_y j + p_z k$
- let $\mathbf{q}$ be a unit quaternion
- $\mathbf{p}'' = \mathbf{q}\mathbf{p}'\mathbf{q}^* = p'' + p''_x i + p''_y j + p''_z k$
- the vector $[p''_x\ p''_y\ p''_z]$ is the vector obtained by rotating $\mathbf{p}$ by the rotation associated with $\mathbf{q}$
- important: interpolation between orientations

Homogeneous Coordinates

- points are identified by vectors in $\mathbb{R}^3$
- directions can be also be identified by a (unary) vector in $\mathbb{R}^3$
- homogenous coordinates offer a unified way to represent and distinguish both points and directions
  - will lead to convenient transformation matrices
- idea: represent both points and directions using a *four dimensional* vector

Homogeneous Coordinates

- a point $\mathbf{p} = [x \quad y \quad z]^T$ is represented by the four dimensional vector $\mathbf{p} = [x \quad y \quad z \quad 1]^T$
  - last coordinate is always 1
  - use same symbol for both Cartesian and homogeneous coordinates
- a direction $\mathbf{d} = [x \quad y \quad z]^T$ is represented by the four dimensional vector $\mathbf{d} = [x \quad y \quad z \quad 0]^T$
  - last coordinate is always 0
- so last element allows to distinguish points from directions

Example

Orientation of frame $B - \mathbf{x}'\mathbf{y}'\mathbf{z}'$ with respect to frame $A - \mathbf{xyz}$ is given by the rotation matrix:

$$
{}^{A}_{B}\mathbf{R} = \begin{bmatrix} 0.5721 & 0.0064 & 0.8202 \\ 0.5721 & 0.7135 & -0.4046 \\ -0.5878 & 0.7006 & 0.4045 \end{bmatrix}
$$

- Each column represents the direction of one axis of B in frame A
- direction $\mathbf{x}'$ in homogeneous coordinates is $\mathbf{x}' = [0.5721\ 0.5721\ \ -0.5878\ 0]^T$

Transformation Matrices

Definition

Let $A - \mathbf{xyz}$ and $B - \mathbf{x'y'z'}$ be two frames. Let ${}^{A}\mathbf{p} = [p_x \; p_y \; p_z]^T$ be the coordinates of the origin of $B - \mathbf{x'y'z'}$ expressed in frame A, and let ${}^{A}_{B}\mathbf{R}$ be the rotation matrix expressing the orientation of frame B with respect to A. Then the *transformation matrix* expressing frame $B - \mathbf{x'y'z'}$ with respect to frame $A - \mathbf{xyz}$ is the 4×4 matrix defined as follows:

$$
{}^{A}_{B}\mathbf{T} = \left[\begin{array}{ccc|c} & & & p_x \\ & {}^{A}_{B}\mathbf{R} & & p_y \\ & & & p_z \\ 0 & 0 & 0 & 1 \end{array} \right].
$$

- last row is always [0 0 0 1]

Transformation Matrices

- many interpretations
- first three columns: directions of $\mathbf{x}'\mathbf{y}'\mathbf{z}'$ in homogeneous coordinates
- last column: coordinates of $\mathbf{p}$ in homogeneous coordinates
- Interpretation #1: single object combining all elements necessary to represent a frame (rotation and origin)

Transformation Matrices

- Interpretation #2: operator to transform points and directions expressed in homogeneous coordinates

Let **p** be a point

$$\mathbf{Tp} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11}p_x + r_{12}p_y + r_{13}p_z + t_x \\ r_{21}p_x + r_{22}p_y + r_{23}p_z + t_y \\ r_{31}p_x + r_{32}p_y + r_{33}p_z + t_z \\ 1 \end{bmatrix}$$

- **p** expressed in homogeneous coordinates and result expressed in homogeneous coordinates
- rotate **p** by **R** and then translate by $[t_x \; t_y \; t_z]^T$
- can express rotation, translation, and rototranslation

Transformation Matrices

Let $\mathbf{d} = [d_x \ d_y \ d_z \ 0]^T$ is a direction expressed in homogeneous coordinates.

$$\mathbf{Td} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ 0 \end{bmatrix} = \begin{bmatrix} r_{11} d_x + r_{12} d_y + r_{13} d_z \\ r_{21} d_x + r_{22} d_y + r_{23} d_z \\ r_{31} d_x + r_{32} d_y + r_{33} d_z \\ 0 \end{bmatrix}$$

- result is rotated direction in homogeneous coordinates
- no translation applied (*directions cannot be translated*)

Transformation Matrices

Special transformation matrices (as operators)

$$\mathbf{T}(\Delta_x, \Delta_y, \Delta_z) = \begin{bmatrix} 1 & 0 & 0 & \Delta_x \\ 0 & 1 & 0 & \Delta_y \\ 0 & 0 & 1 & \Delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}(\mathbf{z}, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}(\mathbf{x}, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}(\mathbf{y}, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Interpretation #3: operator to change coordinates
- let ${}^{A}_{B}\mathbf{T}$ is the transformation matrix representing frame B with respect to frame A
- let ${}^{B}\mathbf{p}$ is the vector expressing the coordinates of $\mathbf{p}$ in frame B

$$ {}^{A}\mathbf{p} = {}^{A}_{B}\mathbf{T}\,{}^{B}\mathbf{p}. $$

- result follows from previous two interpretations

Transformation Matrices

- Interpretation #4: operator to transform frames
- consider ${}^A_B\mathbf{T}$ and ${}^B_C\mathbf{T}$
- can multiply them together (dimensions agree)

$$
{}^A_B\mathbf{T}\,{}^B_C\mathbf{T} = \begin{bmatrix} & {}^A_B\mathbf{R} & & \begin{matrix} {}^AO'_x \\ {}^AO'_y \\ {}^AO'_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & {}^B_C\mathbf{R} & & \begin{matrix} {}^BO''_x \\ {}^BO''_y \\ {}^BO''_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} =
$$

$$
= \begin{bmatrix} & {}^A_B\mathbf{R}\,{}^B_C\mathbf{R} & & {}^A_B\mathbf{R}\,{}^BO'' + {}^AO' \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & {}^A_C\mathbf{R} & & {}^AO'' \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^A_C\mathbf{T}
$$

Rotate directions and rototranslates origin

Transformation Matrices: Chaining and Inverse

- Previous relation can be applied multiple times in sequence

$$ {}^{0}_{n}\mathbf{T} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T} \dots {}^{n-2}_{n-1}\mathbf{T}\,{}^{n-1}_{n}\mathbf{T}. $$

- Inverse problem: given ${}^{A}_{B}\mathbf{T}$ we are interested in ${}^{B}_{A}\mathbf{T}$
  - usual requirement: ${}^{A}_{B}\mathbf{T}\,{}^{B}_{A}\mathbf{T} = {}^{B}_{A}\mathbf{T}\,{}^{A}_{B}\mathbf{T} = \mathbf{I}$
- solution via simple linear algebra

$$ {}^{A}_{B}\mathbf{T} = \begin{bmatrix} & {}^{A}_{B}\mathbf{R} & & {}^{A}O' \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow {}^{B}_{A}\mathbf{T} = \begin{bmatrix} & {}^{B}_{A}\mathbf{R} & & -{}^{B}_{A}\mathbf{R}\,{}^{A}O' \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Example: Change of Coordinates

Consider case seen in slide 5 (two robots observing the same object).

- A world frame, B green frame, C red frame
- ${}^{A}_{B}\mathbf{T}$ and ${}^{A}_{C}\mathbf{T}$ are known
- left ${}^{B}\mathbf{p}$ pose of object as observed by left robot. What is the pose of the object with respect to the right robot ${}^{C}\mathbf{p}$?

1. First Solution
   1. determine ${}^{A}\mathbf{p} = {}^{A}_{B}\mathbf{T}\,{}^{B}\mathbf{p}$
   2. then determine ${}^{C}_{A}\mathbf{T}$ inverting ${}^{A}_{C}\mathbf{T}$
   3. overall solution: ${}^{C}\mathbf{p} = {}^{C}_{A}\mathbf{T}\,{}^{A}_{B}\mathbf{T}\,{}^{B}\mathbf{p}$
2. Second Solution
   1. compute ${}^{C}_{B}\mathbf{T}$ (how?)
   2. overall solution: ${}^{C}\mathbf{p} = {}^{C}_{B}\mathbf{T}\,{}^{B}\mathbf{p}$

Transformation Trees

- useful data structure to quickly perform arbitrary changes of coordinates
- supported by ROS via the `tf/tf2` packages

Definition

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ be a set of n frames and let $\mathcal{T} = \{{}^{A_{i(1)}}_{A_{j(1)}}\mathbf{T} \ldots {}^{A_{i(k)}}_{A_{j(k)}}\mathbf{T}\}$ be a set of k transformation matrices between the given frames. The *transformation graph* defined by these frames and transformation matrices is a graph $G = (V, E)$ with n vertices and k edges. Each vertex is labeled with one of the frames, i.e., v_1 is labeled A_1, v_2 is labeled A_2, and so on. Edge $(v_l, v_m) \in E$ if and only if ${}^{A_l}_{A_m}\mathbf{T} \in \mathcal{T}$.

Definition (Transformation tree)

A *transformation tree* is a transformation graph with no cycles.

Transformation Trees

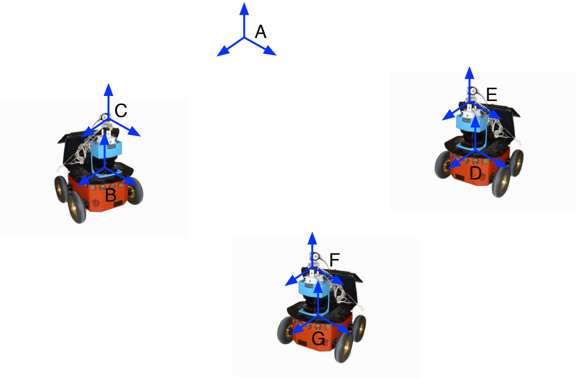
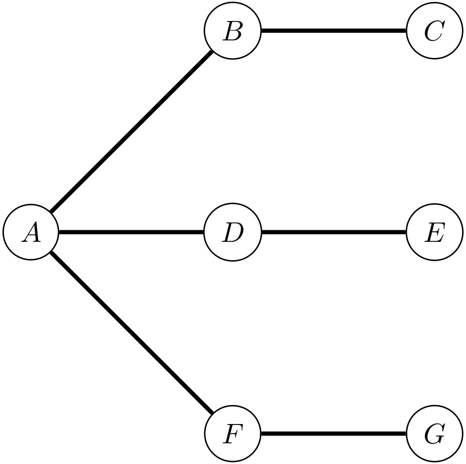


Figure: Three robots with 7 frames.

Transformation Trees



Transformation tree based on $\mathcal{T} = \{{}^{A}_{B}\mathbf{T}, {}^{B}_{C}\mathbf{T}, {}^{A}_{D}\mathbf{T}, {}^{D}_{E}\mathbf{T}, {}^{A}_{F}\mathbf{T}, {}^{F}_{G}\mathbf{T}\}$

Transformation Trees

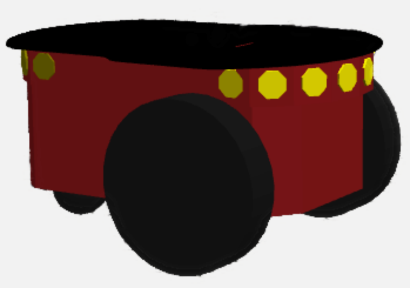
Using transformation trees:

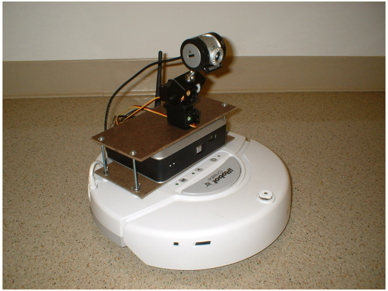
- given ${}^{A}\mathbf{p}$ can I determine ${}^{B}\mathbf{p}$?
- if ${}^{A}_{B}\mathbf{T}$ is in the tree, the answer is obviously *yes*
- if there is a path between A and B in the tree, then ${}^{A}_{B}\mathbf{T}$ can be computed and the answer is *yes*
- if there is no path between A and B in the tree, the answer is *no*

Kinematic Motion Models

- focus on wheeled vehicles
- useful to anticipate the effect of actions (we will need this in planning and navigation)
- related to exported API in many ROS nodes
- Question 1: if I give a certain input, what is the output?
- Question 2: if I desire a certain output, which input should I give?

Differential Drive

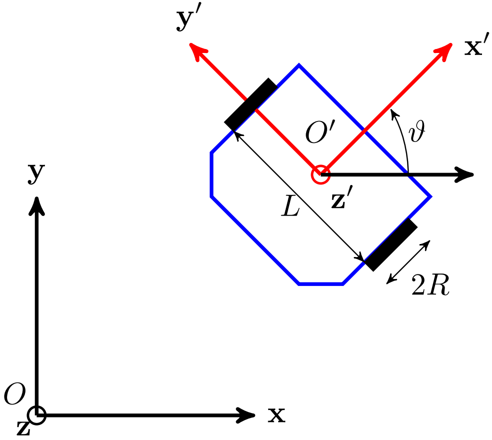




Differential Drive

- two or four actuated wheels
- wheels on the same side are *coupled*, i.e., they receive the same control
- relevant parameters and inputs:
  - R: radius of the wheels
  - L: separation between the wheels
  - ω_R, ω_L: angular velocities for the right and left wheel
- What type of motions can it perform?
  1. forward/backward straight
  2. turn in place
  3. motion along an arc

Differential Drive



Frame $B = O' - \mathbf{x}'\mathbf{y}'\mathbf{z}'$ is attached to the robot. Pose is x, y, ϑ.

Differential Drive Transformation Matrix

$$
{}^{A}_{B}\mathbf{T} = \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 & x \\ \sin\vartheta & \cos\vartheta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Fully specified by the pose x, y, ϑ.

In some instances we write just x, y, ϑ instead of ${}^{A}_{B}\mathbf{T}$ (one-to-one correspondence).

Differential Drive Transition Equations

$$\dot{x} = \frac{R}{2}(\omega_R + \omega_L)\cos\vartheta$$
$$\dot{y} = \frac{R}{2}(\omega_R + \omega_L)\sin\vartheta$$
$$\dot{\vartheta} = \frac{R}{L}(\omega_R - \omega_L)$$

Note: two inputs (ω_R, ω_L), but three state variables (x, y, ϑ)

Often more convenient thinking in terms of translational and rotational velocities

That is what you normally do in ROS

Example

Let $R = 0.05m$ and $L = 0.15m$. Determine ω_L and ω_R so that the translational speed is $0.1m/s$ and the rotational speed is $0.05rad/s$.
From the above relationships the (positive) translational speed can be written as

$$\sqrt{\dot{x}^2 + \dot{y}^2} = \sqrt{\left(\frac{R}{2}(\omega_R + \omega_L)\cos\vartheta\right)^2 + \left(\frac{R}{2}(\omega_R + \omega_L)\sin\vartheta\right)^2}$$
$$= \sqrt{\frac{R^2}{4}(\omega_R + \omega_L)^2} = \frac{R}{2}(\omega_R + \omega_L).$$

We therefore obtain a linear system with two equations in two unknowns:

$$\frac{R}{2}(\omega_R + \omega_L) = 0.1 \qquad \frac{R}{L}(\omega_R - \omega_L) = 0.05$$

The system solves to $\omega_R = 2.075rad/s$ and $\omega_L = 1.925rad/s$.

Approximate Equations – Discrete Time Models

When ω_R, ω_L are given:

$$x(t+\Delta t) \approx x(t) + \dot{x}\Delta t = x(t) + \frac{R}{2}(\omega_R + \omega_L)\cos\vartheta(t)\Delta t$$
$$y(t+\Delta t) \approx y(t) + \dot{y}\Delta t = y(t) + \frac{R}{2}(\omega_R + \omega_L)\sin\vartheta(t)\Delta t$$
$$\vartheta(t+\Delta t) \approx \vartheta(t) + \dot{\vartheta}\Delta t = \vartheta(t) + \frac{R}{L}(\omega_R - \omega_L)\Delta t$$

Approximate Equations – Discrete Time Models

When v_t, $v_r \neq 0$ are given:

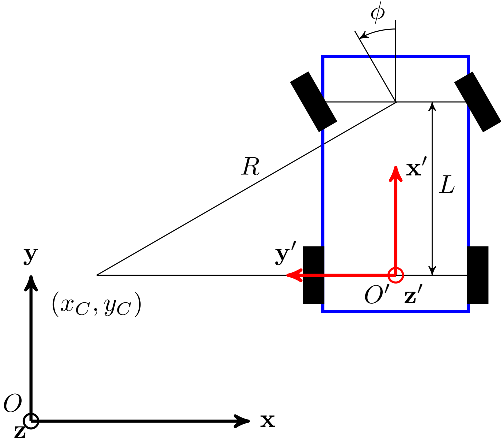
$$
\begin{aligned}
x(t+\Delta t) &\approx x(t) + \left[-\frac{v_t(t)}{v_r(t)} \sin \vartheta(t) + \frac{v_t(t)}{v_r(t)} \sin(\vartheta(t) + v_r(t)\Delta t) \right] \\
y(t+\Delta t) &\approx y(t) + \left[\frac{v_t(t)}{v_r(t)} \cos \vartheta(t) - \frac{v_t(t)}{v_r(t)} \cos(\vartheta(t) + v_r(t)\Delta t) \right] \\
\vartheta(t+\Delta t) &\approx \vartheta(t) + v_r(t)\Delta t
\end{aligned}
$$

If instead $v_r = 0$, the above relationships simplify leading to a straight motion (note that these relationships are not approximated):

$$
\begin{aligned}
x(t+\Delta t) &= x(t) + v_t \sin \vartheta(t) \\
y(t+\Delta t) &= y(t) + v_t \cos \vartheta(t) \\
\theta(t+\Delta t) &= \theta(t).
\end{aligned}
$$

Skid steer drive


Ackerman Steer



Inputs: v_t and ϕ. Radius of the center of rotation: $R = \frac{L}{\tan \phi}$

Ackerman Steer Kinematic Model

Continuous time model:

$$
\begin{aligned}
\dot{x} &= v_t \cos\vartheta \\
\dot{y} &= v_t \sin\vartheta \\
\dot{\vartheta} &= \frac{v_t}{L} \tan\phi
\end{aligned}
$$

Discrete time model

$$
\begin{aligned}
x(t+\Delta t) &\approx x(t) + v_t \Delta t \cos\vartheta \\
y(t+\Delta t) &\approx y(t) + v_t \Delta t \sin\vartheta \\
\vartheta(t+\Delta t) &\approx \vartheta(t) + \frac{v_t \Delta t}{L} \tan\phi
\end{aligned}
$$

Velocity

- so far we have mostly talked about static poses
- for time varying poses, the derivative with respect to time leads to velocities
- position component: simple
- orientation: slightly more complicated – we will consider just special cases (see next)