Basic Kinematics

• Spatial Descriptions and Transformations
• Introduction to Motion
Objectives of the Lecture

• Learn to represent position and orientation
• Be able to transform between coordinate systems.
• Use **frames** and **homogeneous coordinates**

Reference: Craig, “Introduction to Robotics,” Chapter 2
Almost any introductory book to robotics
Introduction

• Robot manipulation implies movement in space
• Coordinate systems are required for describing position/movement
• Objective: describe rigid body motion
• Starting point: there is a universe/inertial/stationary coordinate system, to which any other coordinate system can be referred
The Descartes Connection

- Descartes – invented the now called Cartesian coordinates

- Descartes was lying on his bed watching a fly. Slowly, it came to him that he could describe the fly's position at any instant by just three numbers. Those three numbers were along the planes of the floor and two adjacent walls, what we now call the x,y,z coordinate system.
Coordinate System in Robotics
Three Problems with CS

- Given 2 CS’s, how do we express one as a function of the other?
- Given a point in one CS, what are the point’s coordinates on a second CS?
- Apply an operation on a vector
a coordinate system

\[ \mathbf{A} \]

- point = position vector

\[ \mathbf{A} \mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \]
Description of an Orientation

Often a point is not enough: need orientation

- In the example, a description of \{B\} with respect to \{A\} suffices to give orientation
- Orientation = System of Coordinates
- Directions of \{B\}: \(X_B\), \(Y_B\) and \(Z_B\)
- In \{A\} coord. system: \(^A\!X_B\), \(^A\!Y_B\) and \(^A\!Z_B\)
From \{A\} to \{B\}

\[
\begin{align*}
\cos(\alpha_X) &= X_A \cdot X_B \\
\cos(\alpha_Y) &= Y_A \cdot X_B \\
\cos(\alpha_Z) &= Z_A \cdot X_B 
\end{align*}
\]

We conclude:

\[
^AX_B = \begin{bmatrix} X_A \cdot X_B \\ Y_A \cdot X_B \\ Z_A \cdot X_B \end{bmatrix}
\]
Rotation Matrix

- Stack three unit vectors to form Rotation Matrix
- $^A_B R$ describes $\{B\}$ with respect to $\{A\}$

$$ ^A_B R = \begin{bmatrix} ^A_X_B & ^A_Y_B & ^A_Z_B \end{bmatrix} $$

- Each vector in $^A_B R$ can be written as dot product of pair of unit vectors: cosine matrix
- Rows of $^A_B R$: unit vectors of $\{A\}$ with respect to $\{B\}$
- What is $^A_B R^{-1}$? What is $det(^A_B R)$?
- Position + orientation = Frame
Description of a Frame

- Frame: set of four vectors giving position + orientation
- Description of a frame: position + rotation matrix
- Ex.:
  \[ \{ B \} = \left\{ {}^A R, {}^A P_{BORG} \right\} \]
  - position: frame with identity as rotation
  - orientation: frame with zero position
Mapping: from frame 2 frame

• If \( \{A\} \) has same orientation as \( \{B\} \), then \( \{B\} \) differs from \( \{A\} \) in a translation: \( ^A\mathbf{p}_{\text{BORG}} \)

\[
^A\mathbf{p} = ^B\mathbf{p} + ^A\mathbf{p}_{\text{BORG}}
\]

• Mapping: change of description from one frame to another. The vector \( ^A\mathbf{p}_{\text{BORG}} \) defines the mapping.
Rotated Frames

\[ \mathbf{B}P = p_x X_B + p_y Y_B + p_z Z_B \]

\[ \mathbf{A}P = p_x^A X_B + p_y^A Y_B + p_z^A Z_B \]

\[ \mathbf{A}P = \begin{bmatrix} X_B & Y_B & Z_B \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \]

\[ \mathbf{A}P = \mathbf{A}R^B \mathbf{P} \]

Description of Rotation = Rotation Matrix
Rotated Frame (cont.)

- The previous expression can be written as

\[ A_P = B^R_P \]

- The rotation mapping changes the description of a point from one coordinate system to another

- The point does not change! only its description
Example (2D rotation)

\[ x_1 = x_0 \cos \theta + y_0 \sin \theta \]
\[ y_1 = -x_0 \sin \theta + y_0 \cos \theta \]
General Frame Mapping

\[ ^A P = _B R^B P + ^A P_{BORG} \]

Replace by the more appealing equation:

\[
\begin{bmatrix}
^A P \\
1
\end{bmatrix} =
\begin{bmatrix}
_A R & _A P_{BORG} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
_B P \\
1
\end{bmatrix}
\]

A “1” added here

A row added here
Homogeneous Coords

• Homogeneous coordinates: embed 3D vectors into 4D by adding a “1”

• More generally, the transformation matrix $T$ has the form:

$$ T = \begin{bmatrix} \text{Rot. Matrix} & \text{Trans. Vector} \\ \text{Perspect. Trans.} & \text{Scaling Factor} \end{bmatrix} $$
Operators: Translation, Rotation and General Transformation

- Translation Operator:

\[ A^P_2 = A^P_1 + A^Q \]

\[ = \text{TRANS} (A^Q,|Q|)(A^P_1) \]
Translation Operator

• Translation Operator:
• Only one coordinate frame, point moves
• Equivalent to mapping point to a 2nd frame
• Point Forward = Frame Backwards

• How does \textit{TRANS} look in homogeneous coordinates?
Operators (cont.)

- Rotational Operator

Rotation around axis:
Rotation Operator

- Rotational Operator

The rotation matrix can be seen as rotational operator
- Takes $^A P_1$ and rotates it to $^A P_2 = R^A P_1$

- $^A P_2 = ROT(K, q)(^A P_2)$

- Write ROT for a rotation around K
Operators (Cont.)

• Transformation Operators
  * A transformation mapping can be viewed as a transformation operator: map a point to any other in the same frame
  * Transform that rotates by $R$ and translates by $Q$ is the same as transforming the frame by $R \& Q$
Compound Transformation

If \( \{C\} \) is known relative to \( \{B\} \), and \( \{B\} \) is known relative to \( \{A\} \). We want to transform \( P \) from \( \{C\} \) to \( \{A\} \):

\[
^{B}P = ^{B}T^{C}P
\]

\[
\Rightarrow ^{A}P = ^{A}T^{C}P
\]

\[
^{A}P = ^{A}T^{B}P
\]

From here define

\[
^{A}T = ^{A}T^{B}T
\]

Write down the compound in homog. coords
More on Rotations

• We saw that a rotation can be represented by a rotation matrix

• Matrix has 9 variables and 6+ constraints (which?)

• Rotations are far from intuitive: they do not commute!

• Rotation matrix can be parameterized in different manners:
  — Roll, pitch and yaw angles
  — Euler Angles
  — Others
Euler’s Theorem

• Euler’s Theorem: Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis.

• Not to be confused with Euler angles, Euler integration, Newton-Euler dynamics, inviscid Euler equations, Euler characteristic…

• Leonard Euler (1707-1783)
XYZ Fixed
Euler Angles

• This means that we can represent an orientation with 3 numbers.
• A sequence of rotations around principle axes is called an Euler Angle Sequence.
• Assuming we limit ourselves to 3 rotations without successive rotations about the same axis, we could use any of the following 12 sequences:

  | XYZ | XZY | XYX | XZX |
  | YXZ | YZX | YXY | YZY |
  | ZXY | ZYX | ZXZ | ZYZ |
Euler Angles

• This gives us 12 redundant ways to store an orientation using Euler angles
• Different industries use different conventions for handling Euler angles (or no conventions)
ZYX Euler
Euler Angles to Matrix Conversion

- To build a matrix from a set of Euler angles, we just multiply a sequence of rotation matrices together:

\[
R_x \cdot R_y \cdot R_z = \begin{bmatrix}
1 & 0 & 0 \\
0 & c_x & s_x \\
0 & -s_x & c_x \\
\end{bmatrix}
\begin{bmatrix}
c_y & 0 & -s_y \\
0 & 1 & 0 \\
s_y & 0 & c_y \\
\end{bmatrix}
\begin{bmatrix}
c_z & s_z & 0 \\
-s_z & c_z & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
c_y c_z & c_y s_z & -s_y \\
S_x S_y c_z - c_x s_z & S_x S_y s_z + c_x c_z & S_x c_y \\
c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y \\
\end{bmatrix}
\]
Euler Angle Order

• As matrix multiplication is not commutative, the order of operations is important
• Rotations are assumed to be relative to fixed world axes, rather than local to the object
• One can think of them as being local to the object if the sequence order is reversed
Using Euler Angles

• To use Euler angles, one must choose which of the 12 representations they want.
• There may be some practical differences between them and the best sequence may depend on what exactly you are trying to accomplish.
Euler Angle, Animated
Orientation Representation

- Euler Angle I

\[
R_{z\phi} = \begin{pmatrix}
\cos \phi & -\sin \phi & 0 \\
\sin \phi & \cos \phi & 0 \\
0 & 0 & 1
\end{pmatrix},
R_{u'\theta} = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \theta & -\sin \theta \\
0 & \sin \theta & \cos \theta
\end{pmatrix},
\]

\[
R_{w''\varphi} = \begin{pmatrix}
\cos \varphi & -\sin \varphi & 0 \\
\sin \varphi & \cos \varphi & 0 \\
0 & 0 & 1
\end{pmatrix}
\]
Euler Angle I

Resultant eulerian rotation matrix:

\[ R = R_{z\phi} R_{u\theta} R_{w\varphi} \]

\[
\begin{pmatrix}
\cos \phi \cos \varphi & -\cos \phi \sin \varphi & \sin \varphi \sin \theta \\
-\sin \phi \sin \varphi \cos \theta & -\sin \phi \cos \varphi \cos \theta & \sin \phi \cos \theta \\
\sin \phi \cos \varphi & -\sin \phi \sin \varphi & -\cos \phi \sin \theta \\
+\cos \phi \sin \varphi \cos \theta & +\cos \phi \cos \varphi \cos \theta & \cos \varphi \sin \theta \\
\sin \varphi \sin \theta & \cos \varphi \sin \theta & \cos \theta
\end{pmatrix}
\]
Note the opposite (clockwise) sense of the third rotation, $\phi$. 
Orientation Representation

- Matrix with Euler Angle II

\[
\begin{pmatrix}
-\sin \phi \sin \varphi & -\sin \phi \cos \varphi & \cos \phi \sin \theta \\
+\cos \phi \cos \varphi \cos \theta & -\sin \phi \cos \varphi \cos \theta & \\
\cos \phi \sin \varphi & \cos \phi \cos \varphi & \sin \varphi \sin \theta \\
+\sin \phi \cos \varphi \cos \theta & -\sin \phi \cos \varphi \cos \theta & \\
-\cos \varphi \sin \theta & \sin \varphi \sin \theta & \cos \theta
\end{pmatrix}
\]

Quiz: How to get this matrix?
Orientation Representation

- Description of Roll Pitch Yaw

Quiz: How to get rotation matrix?
Euler Angles - Summary

• Euler angles are used in a lot of applications, but they tend to require some rather arbitrary decisions
• They also do not interpolate in a consistent way (but this isn’t always bad)
• There is no simple way to concatenate rotations
• Conversion to/from a matrix requires several trigonometry operations
• They are compact (requiring only 3 numbers)
Quaternions
Quaternions

- Quaternions are an interesting mathematical concept with a deep relationship with the foundations of algebra and number theory
- Invented by W.R. Hamilton in 1843
- In practice, they are most useful to us as a means of representing orientations
- A quaternion has 4 components

\[ q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix} \]
Quaternions (Imaginary Space)

• Quaternions are actually an extension to complex numbers
• Of the 4 components, one is a ‘real’ scalar number, and the other 3 form a vector in imaginary ijk space!

\[ q = q_0 + iq_1 + jq_2 + kq_3 \]

\[
\begin{align*}
i^2 &= j^2 = k^2 = ijk = -1 \\
i &= jk = -kj \\
j &= ki = -ik \\
k &= ij = -ji
\end{align*}
\]
Quaternions (Scalar/Vector)

- Sometimes, they are written as the combination of a scalar value $s$ and a vector value $v$

$$q = \langle s, v \rangle$$

where

$$s = q_0$$
$$v = [q_1, q_2, q_3]$$
Unit Quaternions

• For convenience, we will use only unit length quaternions, as they will be sufficient for our purposes and make things a little easier.

\[ |q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \]

• These correspond to the set of vectors that form the ‘surface’ of a 4D hypersphere of radius 1.
• The ‘surface’ is actually a 3D volume in 4D space, but it can sometimes be visualized as an extension to the concept of a 2D surface on a 3D sphere.
Quaternions as Rotations

- A quaternion can represent a rotation by an angle $\theta$ around a unit axis $\mathbf{a}$:

$$
\mathbf{q} = \begin{bmatrix}
\cos \frac{\theta}{2} & a_x \sin \frac{\theta}{2} & a_y \sin \frac{\theta}{2} & a_z \sin \frac{\theta}{2}
\end{bmatrix}
$$

or

$$
\mathbf{q} = \left\langle \cos \frac{\theta}{2}, \mathbf{a} \sin \frac{\theta}{2} \right\rangle
$$

- If $\mathbf{a}$ is unit length, then $\mathbf{q}$ will be also
Quaternions as Rotations

\[ |q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \]

\[ = \sqrt{\cos^2 \frac{\theta}{2} + a_x^2 \sin^2 \frac{\theta}{2} + a_y^2 \sin^2 \frac{\theta}{2} + a_z^2 \sin^2 \frac{\theta}{2}} \]

\[ = \sqrt{\cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} \left( a_x^2 + a_y^2 + a_z^2 \right)} \]

\[ = \sqrt{\cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} |a|^2} = \sqrt{\cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2}} \]

\[ = \sqrt{1} = 1 \]