Lecture 11: (23/02/15)

Reaching and Grasping

• Reference Frames
• Configuration space
• Reaching
• Grasping

Michael Herrmann

michael.herrmann@ed.ac.uk, phone: 0131 6 517177, Informatics Forum 1.42
Robot arms

- Typically constructed with rigid *links* between movable one d.o.f. *joints*
- Joints typically
  - *rotary* (revolute) or *prismatic* (linear)
Kinematic pairs

Lower pairs
- Rotary (hinge, 1 axis)
- Prismatic (linear)
- Cylindrical (rotary + linear)
- Spherical (3 rotary)
- Planar (2 linear + rotary)

Higher pairs (different curvature): rolling bearing, cam joint, …

Wrapping pair: belt, chain, …

Linkages: Crankshaft-piston, …

Kinematic chains
Classification by geometry

Variants

Robot arms (manipulators)

Classification by topology (kinematic chain):

- (left) Serial chain
- Parallel manipulator (below)
Robot arm: End effectors (at Tool Center Point at top of kinematic chain)

- Simple push or sweep
- Gripper – different shape, size or strength
- Hook, pins, needles (e.g. for handling textiles)
- Vacuum cup, magnetic
- Adhesion by contact (e.g. glue)
- Tools for specific purposes (drills, welding torch, spray head, scalpel, scoop,…)
- Hand for variety of purposes
Issues in choosing actuators

- Load (e.g. torque to overcome inertia of arm)
- Speed (fast enough but not too fast)
- Accuracy (will it move to where you want?)
- Resolution (can you specify exactly where?)
- Repeatability (will it do this every time?)
- Reliability (mean time between failures)
- Compliance (how does stiffness change?)
- Power consumption (how to feed it)
- Energy supply & its weight
- Geometry (linear vs. rotary) and other trade-offs between physical design and ability to control
Prehension

Goal: Understand ideal robot mechanisms for reaching, grasping and manipulation

Robot positions and configurations as a function of control parameters – kinematics

Need to know:

• Representing mechanism geometry
• Standard configurations
• Degrees of freedom
• Grippers and graspability conditions
Vectors & Points in 3D

Point \( \vec{p} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = (a, b, c)^T \)
Vector \( \vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = (v_x, v_y, v_z)^T \)

3D Coordinate Systems usually defined as left handed (right handed reverses the +Z direction)
Local Reference Frames

\( \vec{\rho} \) can be expressed by

- Local (translated) coordinates \((d, e, f)\)
- Global (untranslated) coordinates \((a + d, b + e, c + f)\)

Local frame could also have further sub-local frames
Translations

Move $\vec{p}$ to $\vec{q}$:

$$\vec{q} = \vec{p} + \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$
Rotations

Rotate about Z axis

A lot of conventions
Here: $\theta$ positive is anti-clockwise when looking along $+Z$

- $\vec{p}$ in local (rotated) coordinates is $(a, b, c)^T$
- $\vec{p}$ in global (unrotated) coordinates is
  $$(a \cos(\theta) - b \sin(\theta), a \sin(\theta) + b \cos(\theta), c)^T$$
Rotation Matrix Representation I

\[ \tilde{p} = (a, b, c)^T \longrightarrow (a \times \cos(\theta) - b \times \sin(\theta), a \times \sin(\theta) + b \times \cos(\theta), c)^T \]

\[ R_z(\theta_z) \tilde{p} = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \]

Much more compact and clearer!
Rotation Matrix Representation II

\[ R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \]

\[ R_y(\theta_y) = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \]
Other Rotation Representations

All equivalent but different parameters:
• Yaw, pitch, roll
• Azimuth, elevation, twist
  (slant, tilt, twist)
• Axis + angle
→ Quaternions
• N.B.: in mammals:
Full Rotation Specification

• Need 3 angles for arbitrary 3D rotation
• Lock & key example (axis and direction of key bit)
• Rotation angles \((\alpha, \beta, \gamma)\):

\[
R(\alpha, \beta, \gamma)p = R_z(\gamma)R_y(\beta)R_x(\alpha)p
\]

• Warning: rotation order by convention but must be used consistently:

\[
R_z(\gamma)R_y(\beta)R_x(\alpha) \neq R_x(\alpha)R_y(\beta)R_z(\gamma)
\]
Full Transformation Specification

Each connection has a new local coordinate system

Need to specify 6 degrees of freedom = 3 rotation + 3 translation

\[ \text{transform}(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z) = \text{trf}(\theta, t) \]
Kinematic Chains in 2D

$p$ is:

in $C_2$: \[ \begin{pmatrix} d_2 \\ 0 \end{pmatrix} \]

in $C_1$: \[ R(\theta_2) \begin{pmatrix} d_2 \\ 0 \end{pmatrix} + \begin{pmatrix} d_1 \\ 0 \end{pmatrix} \]

in $C_0$: \[ R(\theta_1)[R(\theta_2) \begin{pmatrix} d_2 \\ 0 \end{pmatrix} + \begin{pmatrix} d_1 \\ 0 \end{pmatrix}] + \begin{pmatrix} d_0 \\ 0 \end{pmatrix} \]
Homogeneous Coordinates I

Messy when more than 2 links, as in robot
So: pack rotation and translation into

Homogeneous coordinate matrix

Extend points with a 1 from 3-vector to 4-vector
Extend vectors with a 0 from 3-vector to 4-vector
Pack rotation and translation into 4x4 matrix:

\[
H_1 = \begin{bmatrix}
R(\vec{\theta}_1) & \vec{t}_1 \\
\vec{0} & 1
\end{bmatrix}
\]

3 rotation parameters: \( \vec{\theta}_1 \)
3 translation parameters: \( \vec{t}_1 \)
Homogeneous matrices

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha & 0 \\
0 & \sin \alpha & \cos \alpha & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \quad \begin{pmatrix}
1 & 0 & 0 & t_x \\
0 & 1 & 0 & t_y \\
0 & 0 & 1 & t_z \\
0 & 0 & 0 & 1
\end{pmatrix} \quad \begin{pmatrix}
s_x & 0 & 0 & 0 \\
0 & s_y & 0 & 0 \\
0 & 0 & s_z & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

scaling

\[
\begin{pmatrix}
\cos \beta & 0 & \sin \beta & 0 \\
0 & 1 & 0 & 0 \\
-\sin \beta & 0 & \cos \beta & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

translation

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

projection

\[
\begin{pmatrix}
\cos \gamma & -\sin \gamma & 0 & 0 \\
\sin \gamma & \cos \gamma & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

rrotations

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{d} & 0
\end{pmatrix}
\]
perspective

view plane z=0, center of projection at (0,0,-d)

Isometries also represented as 6D vector: \( \text{trf}(\alpha, \beta, \gamma, l_x, l_y, l_z) \)
Homogeneous coordinates II

\[ \vec{p}^* = \begin{pmatrix} \vec{p} \\ 1 \end{pmatrix} \]

In \( C_2 \): \( p^* \)

In \( C_1 \): \( H_2 p^* \)

In \( C_0 \): \( H_1 H_2 p^* \)

Longer chains for robot arms (e.g. 6 links):

\[ H_1 H_2 H_3 H_4 H_5 H_6 p^* \]
Joint geometry

Linear (prismatic) joint: slides
parametrize one translation direction per joint
e.g. Sliding in the $x$ direction

$$\text{trf}(0,0,0,\lambda,0,0)$$

Hinge (revolute) joint: rotates
parametrize one rotation angle per joint
e.g. Rotation about $x$-axis

$$\text{trf}(\theta,0,0,0,0,0,0)$$
Configuration Space I

Alternative representation to scene coordinates

Number of joints = $J$

$J$-dimensional space

Binary encoding: 0 for invalid pose, 1 for free space

Real-valued encoding:

“distance” from goal configuration

Point in C.S. = configuration in real space

Curve in C.S. = motion path in real space
Configuration Space II

ALLOWABLE CONFIGURATIONS

CURRENT POSE

α

β
Dynamics: Trajectory Planning

What is the shortest path in configuration space? At what speed the path is traversed?

- Cost to go (energy or wear)
- Time to goal
- Least perturbations (predictability)
- Maximal smoothness
- Minimal intervention
Forward and Inverse Kinematics

Forward:
Given joint angles, find gripper position
Easy for sequential joints in robot arm:
just multiply matrices

Inverse:
Given desired gripper position, find joint angles
Hard for sequential joints – geometric reasoning
Sequential & Parallel Mechanisms

Simplified into 2D

Serial manipulator
vary: $\theta_1, \theta_2, \theta_3$

Parallel manipulator
vary: $d_1, d_2, d_3$
<table>
<thead>
<tr>
<th></th>
<th>Serial</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Forward</strong></td>
<td>Easy (just multiply matrices)</td>
<td>Hard (messy robot specific geometry)</td>
</tr>
<tr>
<td>(param-&gt;position)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Inverse</strong></td>
<td>Hard (messy robot specific geometry)</td>
<td>Easy (just read off lengths from gripper position)</td>
</tr>
<tr>
<td>(position-&gt;param)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
N.B.: Biological motor control

Equilibrium point hypothesis

• Extensors and flexors controlled by different pathways:
• Force-field experiments: violation of equifinality at variations of velocity-dependent load (Hinder & Milner 03)

Solution: Internal dynamics models (Shadmehr & Mussa-Ivaldi 94)
Specifying Robot Positions

1. Actuator level: specify voltages that generate required joint angles.
2. Joint level: specify joint angles and let system calculate voltages.
3. Effector level: specify tool position and let system compute joint angles.
4. Task level: specify the required task and let the system compute the sequence of tool positions.

Most robot programming is at levels 2 and 3.
Grippers and Grasping

- Gripper: special tool for general part manipulation
- Fingers/gripper: 2, 4, 5
- Joints/finger: 1, 2, 3

Your hand: 4 fingers * 4 DoF + thumb * 5 DoF + wrist * 6 DoF = 27 DoF (22 controllable DoF).
Barrett Technology Hand

2 parallel fingers (spread uniformly)
1 opposable finger
DoF: 4 fingers (2 finger joints bend uniformly)
Shadow Dextrous Robot Hand

http://www.shadowrobot.com/hand/
High-Speed Robotic Hand

Skillful Manipulation Based on High-speed Sensory Motor Fusion
Finger Contact Geometry

- Coefficient of friction at fingertip: $\mu \in [0,1]$
- Surface normal: Direction perpendicular to surface: $\vec{n}$
- Friction cone: Angles within $\alpha = \cos^{-1}(\mu)$ about surface normal
- Force direction: $\vec{f}$ direction in which finger pushed
- No-slip condition: $\vec{f} \cdot \vec{n} \geq \mu$
Force Closure

Need balanced forces or else object twists

2 fingers – forces oppose: \( \vec{f}_1 + \vec{f}_2 = 0 \)

3 fingers – forces meet at point: \( f_1 + f_2 + f_3 = 0 \)

Force closure: point where forces meet lies within 3 friction cones otherwise object slips
Other Grasping Criteria

Some heuristics for a good grasp:

- Contact points form nearly equilateral triangle
- Contact points make a big triangle
- Force focus point near Center of Mass (CoM)
Grasp Algorithm

1. Isolate boundary
2. Locate large enough smooth graspable sections
3. Compute surface normals
4. Pick triples of grasp points
5. Evaluate for closure & select by heuristics
6. Evaluate for reachability and collisions
7. Compute force directions and amount
8. Plan approach and finger closing strategy
9. Contact surface & apply grasping force
10. Lift (& hope)
Kinematics Summary

1. Need vector & matrix form for robot geometry
2. Geometry of joints & joint parameters
3. Forward & inverse kinematics
4. Degrees of freedom
5. Grippers & grasping conditions