6.141:

Robotics systems and science Lecture 14: Forward and Inverse Kinematics

> Lecture Notes Prepared by Daniela Rus EECS/MIT Spring 2011

Reading: Chapter3, Craig: Robotics

http://courses.csail.mit.edu/6.141/
Challenge: Build a Shelter on Mars

What is coming up in RSS

- Last time: grasping
- Today in class: grasping, FK, IK
- Today in lab: Lab 7, Lab 6 presentations
- Monday: Arthur Licata on Robot Ethics

Point Contact with Friction

• Consider a point contact exerting force at some angle θ to the surface normal. What happens?



 $\theta_{\rm crit} = \tan^{-1} \mu$

Surface

• Produces a

of force directions

Grasp Synthesis with Friction

Pick f1 and valid green direction

f1

f2

Intersect with edge to get f2

Grasp Analysis With Friction Consider forces **f**₁, **f**₂ at frictional contacts **p**₁, **p**₂



When can **f**₁, **f**₂ oppose one another without sliding? Each force must Point **p**₁ (resp. **p**₂) must

Grasp Synthesis With Friction

Choose a *compatible* pair of edges **e**₁, **e**₂ Intuition? Using what data? How to choose?



Grasp Synthesis (regions)

- f2 placement has error ε
- f2 can point to any force in pink region



Grasp Synthesis (regions)

But if we put f1 in the pink region, which points in the blue region can point to it?

f2

Grasp Synthesis (friction)

- 2 Finger Forces have to be within friction cones to stick
- 2 Finger Forces have to point at each other
- So...
- We need to find 2 edges with overlapping friction cones

Grasp Synthesis With Friction Choose target region for contact point p_1 Determine feasible target region for contact p_2 Orient and scale f_1 , f_2 so as to cancel along $\overline{p_1p_2}$



Example: 6.141 robot



Arm Control to Reach

- Mechanism design
- Forward kinematics
- Inverse kinematics



Kinematic Mechanisms

Link: rigid body Joint: constraint on two links Kinematic mechanism: links and joints





Cylindrical 2 freedoms



Prismatic 1 freedom





The Planar 3-R manipulator

- Planar kinematic chain
- All joints are revolute





Kinematic transformations

- Direct kinematics
 - Joint coordinates to end effector coordinates
 - Sensors are located at the joints. DK algorithm is used to figure out where the robot is in 3-D space.
 - Robot "thinks" in joint coordinates. Programmer/ engineer thinks in "world coordinates" or end effector coordinates.
- Inverse kinematics
 - End effector coordinates to joint coordinates
 - Given a desired position and orientation of the EE, we want to be able to get the robot to move to the desired goal. IK algorithm used to obtain the joint coordinates.
 - Essential for control.

Inverse kinematics has multiple solutions



Which is the correct robot pose ?

Kinematics Summary



Robot kinematic calculations deal with the relationship between joint positions and an external fixed Cartesian coordinate frame.

Dynamics, force, momentum etc. are not considered.

Forward and Inverse Kinematics

So far, have cast computations in Cartesian space

But manipulators controlled in configuration space:

REFERENCE POSE (x, y, ϕ)

Link 3 (EE

 θ_2

 θ_1

Link 2

Link 1

 θ_3

X

- Rigid *links* constrained by *joints*
- For now, focus on joint values
- Example 3-link mechanism:
 - Joint coordinates θ_1 , θ_2 , θ_3 Y
 - Link lengths L_1 , L_2 , L_3
- End effector coordinates
 - "Reference pose" described L_1 by x, y, and ϕ (w.r.t. vertical)
- How can we relate EE to configuration variables?

Forward Kinematics

- Given mechanism description and joint values, express end effector pose in Cartesian coordinates
 - Example: two-link arm with one sliding, one rotating joint
- Configuration variables:
 - Joint coordinates d_{r} θ
 - Link lengths (both 1)
- End effector coordinates
 "Reference point" (x, y)
- Challenge: express as $x = x (d, \theta) =$ $y = y (d, \theta) =$



Inverse Kinematics

- Given end effector pose in Cartesian coordinates, identify the joint values that yield specified pose
- Challenge: solve for joint values in terms of pose $\theta = \theta (x, y)$ REFERENCE POSE (x, y) d = d(x, y)Hints: α ')² X

Why is IK difficult?

- Nonlinear
 - Revolute joints → inverse trigonometry
- Multi-valued



- Often multiple solutions for a single Cartesian pose
- Discontinuities and singularities
 - Can lose one or more DOFs in some configurations
- Possibly over-constrained (no exact solution)
 - Use of approximation and iterative algorithms
- Dynamics
 - In reality, want to apply forces and torques (while respecting physical constraints), not just move arm!

Direct kinematics



Transform joint coordinates to end effector coordinates

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$$\phi = (\theta_1 + \theta_2 + \theta_3)$$

Inverse kinematics



A Trigonometric Digression

$$\sin \theta = -\sin(-\theta) = -\cos(\theta + \frac{\pi}{2}) = \cos(\theta - \frac{\pi}{2})$$

 $\cos \theta = \cos(-\theta) = \sin(\theta + \frac{\pi}{2}) = -\sin(\theta - \frac{\pi}{2})$

$$\cos^2 \theta + \sin^2 \theta = 1$$
$$\sin \theta = \pm \sqrt{1 - \cos^2 \theta}$$

 $\sin(\alpha + \beta) = \sin \alpha \ \cos \beta + \cos \alpha \ \sin \beta$ $\cos(\alpha + \beta) = \cos \alpha \ \cos \beta - \sin \alpha \ \sin \beta$ $\sin(\alpha - \beta) = \sin \alpha \ \cos \beta - \cos \alpha \ \sin \beta$ $\sin \theta = 2t / (1 + t^2)$ $\cos(\alpha - \beta) = \cos \alpha \ \cos \beta + \sin \alpha \ \sin \beta$ Arctangents By definition $\tan \theta = \frac{\sin \theta}{\cos \theta}$, but $\tan has a period of \pi$, whereas sin and $\cos have a period of 2\pi$. If it is known that $x = r \cos \theta$, $y = r \sin \theta$, r > 0then recover with $\theta = \arctan\left(\frac{y}{r}\right) = \arctan\left(\frac{r \sin \theta}{r \cos \theta}\right)$

But, this only gives solution in first and fourth quadrants. The signs of x and y actually uniquely determine the quadrant, so use: $\theta \leftarrow \operatorname{atan2}(y, x)$

and then
$$\tan \theta = \frac{y}{x}$$
, $\sin \theta = \frac{y}{\sqrt{x^2 + y^2}}$, $\cos \theta = \frac{x}{\sqrt{x^2 + y^2}}$

Forward Kinematics

- Traverse kinematic tree and propagate transformations downward
 - Use stack
 - Compose parent transformation with child's
 - Pop stack when leaf is reached
- High DOF models are tedious to control this way



Planar Example



Example: 3DOF Revolute Arm



Three revolute joints. Write s_1 and c_1 for $\sin \theta_1$ and $\cos \theta_1$ respectively.

Downward elbow angle is $\theta_{3-2} = \theta_3 - \theta_2$ write:

$$s_{3-2} = \sin \theta_{3-2} = \sin(\theta_3 - \theta_2)$$
$$c_{3-2} = \cos \theta_{3-2} = \cos(\theta_3 - \theta_2)$$

Forward Kinematics







 $x = [l_2 c_2 + l_3 c_{3-2}]c_1$

b.

 $r = \sqrt{x^2 + y^2} = l_2 c_2 + l_3 c_{3-2}$

 $y = [l_2c_2 + l_3c_{3-2}]s_1$ $z = l_1 + l_2s_2 - l_3s_{3-2}$

$$h = z - l_1 = l_2 s_2 - l_3 s_{3-2}$$

Solution for Planar Example

$$x^{2} + y^{2} = a_{1}^{2} + a_{2}^{2} - 2a_{1}a_{2}\cos(\pi - \theta_{2})$$

$$\cos\theta_{2} = \frac{x^{2} + y^{2} - a_{1}^{2} - a_{2}^{2}}{2a_{1}a_{2}}$$
for greater accuracy
$$\tan^{2}\frac{\theta_{2}}{2} = \frac{1 - \cos\theta}{1 + \cos\theta} = \frac{2a_{1}a_{2} - x^{2} - y^{2} + a_{1}^{2} + a_{2}^{2}}{2a_{1}a_{2} + x^{2} + y^{2} - a_{1}^{2} - a_{2}^{2}}$$

$$= \frac{\left(a_{1}^{2} + a_{2}^{2}\right)^{2} - \left(x^{2} + y^{2}\right)}{\left(x^{2} + y^{2}\right) - \left(a_{1}^{2} - a_{2}^{2}\right)^{2}}$$

$$\theta_{2} = \pm 2\tan^{-1}\sqrt{\frac{\left(a_{1}^{2} + a_{2}^{2}\right)^{2} - \left(x^{2} + y^{2}\right)}{\left(x^{2} + y^{2}\right) - \left(a_{1}^{2} - a_{2}^{2}\right)^{2}}}$$



Inverse Kinematics of 3DOF Manipulator



$$r = \sqrt{x^2 + y^2} = l_2 c_2 + l_3 c_{3-2}$$

$$h = z - l_1 = l_2 s_2 - l_3 s_{3-2}$$

Solving for the Elbow

Can choose an "up elbow" solution with $s_3 = +\sqrt{1-c_3^2}$

or a "down elbow" solution with
$$s_3 = -\sqrt{1 - c_3^2}$$

where $c_3 = \frac{x^2 + y^2 + (z - l_1)^2 - l_2^2 - l_3^2}{2l_2l_3}$

In either case can then recover θ_3 with

 $\theta_3 = \operatorname{atan2}(s_3, c_3)$



$$r = l_{2}c_{2} + l_{3}c_{3-2} = l_{2}c_{2} + l_{3}c_{3}c_{2} + l_{3}s_{3}s_{2} = k_{1}c_{2} + k_{2}s_{2}$$
where
$$k_{1} = l_{2} + l_{3}c_{3}$$

$$h = l_{2}s_{2} - l_{3}s_{3-2} = l_{2}s_{2} - l_{3}s_{3}c_{2} + l_{3}c_{3}s_{2} = k_{1}s_{2} - k_{2}c_{2}$$

$$k_{2} = l_{3}s_{3}$$

$$d = +\sqrt{k_{1}^{2} + k_{2}^{2}}$$
and so
$$k_{1} = d\cos\gamma$$

$$k_{1} = d\cos\gamma$$
substitute in...
$$k_{2} = d\sin\gamma$$



$$\theta_2 - \gamma = \operatorname{atan2}(\frac{h}{d}, \frac{r}{d}) = \operatorname{atan2}(h, r) \text{ since } d > 0$$

Final Inverse Kinematics

$$\begin{aligned} \theta_{1} &\leftarrow \operatorname{atan2}(y, x) \\ c_{3} &\leftarrow \frac{x^{2} + y^{2} + (z - l_{1})^{2} - l_{2}^{2} - l_{3}^{2}}{2l_{2}l_{3}} \\ s_{3} &\leftarrow +\sqrt{1 - c_{3}^{2}} \\ \theta_{3} &\leftarrow \operatorname{atan2}(s_{3}, c_{3}) \\ \theta_{2} &\leftarrow \operatorname{atan2}(z - l_{1}, \sqrt{x^{2} + y^{2}}) - \operatorname{atan2}(l_{3}s_{3}, l_{2} + l_{3}c_{3}) \end{aligned}$$

Putting it All Together: Grasping

- Input workspace, obstacles, and manipuland:
 - Determine a feasible grasp (set of contact points)
 - Use IK to solve for target end-effector pose in c-space
 - Plan a collision-free reach to the computed pose
 - Control end-effector along desired trajectory



What have we swept under the rug?

- Sensing
 - Shape, pose of target object, accessibility of surfaces
 - Classification of material type from sensor data
 - Freespace through which grasping action will occur
- Prior knowledge
 - Estimate of μ, mass, moments given material type
 - Internal, articulated, even active degrees of freedom
- Uncertainty & compliance
 - Tolerate noise inherent in sensing and actuation
 - Ensure that slight sensing, actuation errors won't cause damage
 - Handle soft fingers making contact over a finite area (not a point)
- Dynamics
 - All of the above factors may be changing in real time

Straight-line motion



Clockwise rotation



Counter-clockwise rotation



Robust translation



What if we do not know where the center of mass is?

Robust translation



Push and sense: if clockwise rotation, move right if counterclockwise rotation move left

Moving Parts

Decentralized Control and Planning for Manufacturing

A. Bolger, M. Schwager, D. Stein, L. White, S-K. Yun, D. Rus CSAIL, MIT

Mini Manipulator



Grasping and manipulation summary

- Reaching: forward and inverse kinematics
- Grasping: analysis and synthesis of closure grasps
- Manipulation: prehensile and nonprehensile