Robotics Principles and Applications

Fumiya Iida

Bio-Inspired Robotics Lab, Department of Engineering
University of Cambridge

Lecture note available at:
http://mi.eng.cam.ac.uk/Main/FI224
Outline

• Introduction
• Kinematic Motion Control
• Discussion
  – Advanced Motion Control
  – Trends and Challenges
“Rise of the Robots” is happening because:
1. Cost decrease of technologies
2. Readiness increase of technologies
3. Connected and shared technologies

The Economist, March 2014
Robotics Market and Opportunities

- Industrial manufacturing
- Cleaning
- Medical robotics
- Entertainment/edutainment
- Logistics/autonomous warehouse
- Autonomous cars
- Industrial inspection
- Surveillance and rescue
- Construction and mining
- Agriculture
- Health and elderly care
- Personal/service robots

H2020 euRobotics Multi-Annual Roadmap (2016)
Robotics Market and Opportunities

- Industrial manufacturing
- Cleaning
- Medical robotics
- Entertainment/edutainment
- Logistics/autonomous warehouse
- Autonomous cars

- Industrial inspection
- Surveillance and rescue
- Construction and mining
- Agriculture
- Health and elderly care
- Personal/service robots

H2020 euRobotics Multi-Annual Roadmap (2016)
Moravec’s Paradox

It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, but difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility.

Industrial Environment

- High predictability
- High programmability
- Algorithmic
- Top-down design
- Rigid interactions

Real-World Environment

- Low predictability
- Low programmability
- Creativity, adaptation
- Design for emergence
- Soft interactions
Outline

• Introduction
• Kinematic Motion Control
• Discussion
  – Advanced Motion Control
  – Trends and Challenges
What is Kinematic Control

**Kinematic Control:**
Specify desired robot position every time step, and calculate joint positions/angles from “Inverse Kinematics”. Position Controller then takes care of all the “physics”.

**Forward and Inverse Kinematics:**
“Forward Kinematics” specifies position/posture of a robot given its joint angles, and “Inverse Kinematics” specifies joint angles given position/posture of a robot.

Try this before any other control methods!
Chapter 2 Kinematic Control

KINEMATIC CONTROL OF ROBOT MANIPULATORS
Kinematics of Robot Manipulators

- Forward kinematics
  \[
  [\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\chi}]^T = f(\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n)
  \]

- Inverse kinematics
  \[
  [\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n]^T = f^{-1}(\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\chi})
  \]

This is what you need to control \((q_1, q_2, q_3, \ldots)\)

This is what you want to achieve \((x, y, z, \text{etc.})\)
Simple Kinematic Control

• Specify joint angles $q_1$ and $q_2$ to achieve the position of end-effector at $x$

$$x=f(q_1, q_2)$$

$$x = (l_1 \cos q_1 + l_2 \cos(q_1 + q_2))$$

$$y = (l_1 \sin q_1 + l_2 \sin(q_1 + q_2))$$

• Then think about position control of motor 1 and motor 2 to achieve the joint angles
Simple Kinematic Control

• How to determine $q_1$, $q_2$ for a desired target position of end-effector $\mathbf{x}$?

$$x = (l_1 \cos q_1 + l_2 \cos (q_1 + q_2))$$

$$y = (l_1 \sin q_1 + l_2 \sin (q_1 + q_2))$$

$$[q_1, q_2]^T = f^{-1}(\mathbf{x})$$

$q_1 = ?$

$q_2 = ?$
Where,

1. **Workspace of planar 2R arm**
   - \( WS_1 = \{ p^R_2 : |l_1 - l_2| \leq p \leq l_1 + l_2 \} \)
   - \( WS_2 = \{ p^R_1 : p \in \text{inner boundary} \} \)

2. **Robotics**
   - \( x = (l_1 \cos q_1 + l_2 \cos (q_1 + q_2)) \)
   - \( y = (l_1 \sin q_1 + l_2 \sin (q_1 + q_2)) \)

3. **Motor**
   - \( cos(q_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \)
   - \( q_2 = \pm \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \)

Given \( q_2 \),

- \( q_1 = \text{atan2}(y, x) \pm \text{atan2}(\sqrt{x^2 + y^2 - d_1^2}, d_1) \)

Where,

- \( d_1 = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1} \)
Numerical Approach to Kinematic Control

Coordinate of end-effector
\[ \mathbf{x} = [x_1, x_2, \ldots, x_n]^T \]

Coordinate of joint space
\[ \mathbf{q} = [\theta_1, \theta_2, \ldots, \theta_n]^T \]

Kinematic constraints
\[ \phi = [\phi_1(q), \phi_2(q), \ldots, \phi_n(q)]^T \]

Relationship between velocities of \( q \) and \( x \)
\[ \dot{x} = \frac{\partial \phi}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \]

\( \mathbf{J} \) is the Jacobian

or
\[ \dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \dot{x} \]
More Detailed Calculation

\[
\begin{align*}
[x, y]^T &= \phi(q_1, q_2) \\
\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{bmatrix}
\end{align*}
\]

\[
J(q) = \frac{\partial \phi}{\partial q} = \begin{bmatrix}
\frac{\partial \phi_1}{\partial q_1} & \frac{\partial \phi_1}{\partial q_2} \\
\frac{\partial \phi_2}{\partial q_1} & \frac{\partial \phi_2}{\partial q_2}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\
l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2)
\end{bmatrix}
\]

Once we obtain \(J(q)\), we can inverse it in matlab by \(J^{-1}(q) = \text{inv}(J(q))\) or, if not invertible, \(J^+(q) = \text{pinv}(J(q))\) => Pseudo-inverse

This should be enough to reach the target by running:

\[
q_{t+1} = q_t + J^{-1} (x_{\text{target}} - x_t)
\]

\[
= q_t + J^{-1} (x_{\text{target}} - f(q_t))
\]
Chapter 2 Kinematic Control

KINEMATIC CONTROL OF MOBILE ROBOTS
Kinematics of Mobile Robots

What is kinematics? Motion of system without forces, i.e. “geometry of motion”

Forward Kinematics
\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} = f(\dot{\phi}_1, \ldots, \dot{\phi}_n, \beta_1, \ldots, \beta_m, \dot{\beta}_1, \ldots, \dot{\beta}_m)
\]

Inverse Kinematics
\[
\begin{bmatrix}
\dot{\phi}_1 & \ldots & \dot{\phi}_n & \beta_1 & \ldots & \beta_m & \dot{\beta}_1 & \ldots & \dot{\beta}_m
\end{bmatrix}^T = f(\dot{x}, \dot{y}, \dot{\theta})
\]

\(\dot{\phi}\) wheel speed  \(\beta\) angle of steering wheels
Kinematics of Simple Mobile Robot

- Forward kinematics of a two wheeled robot

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} = f(l, r, \theta, \dot{\phi}_1, \dot{\phi}_2)
\]

\[
\begin{align*}
\omega_1 &= \frac{r \dot{\phi}_1}{2l} \\
\omega_2 &= \frac{-r \dot{\phi}_2}{2l} \\
x_{r1} &= \frac{1}{2} r \dot{\phi}_1 \\
x_{r2} &= \frac{1}{2} r \dot{\phi}_2
\end{align*}
\]

\[
\begin{align*}
\dot{x} &= \dot{x}_{r1} + \dot{x}_{r2} \\
\dot{y} &= 0 \\
\dot{\theta} &= \omega_1 + \omega_2
\end{align*}
\]
Kinematics of Simple Mobile Robot

![Diagram of a robot with labeled coordinates and velocities.]

**Forward kinematics (local coordinate)**

\[
\begin{align*}
\dot{x} &= \dot{x}_{r1} + \dot{x}_{r2} \\
\dot{y} &= 0 \\
\dot{\theta} &= \omega_1 + \omega_2
\end{align*}
\]

where

\[
\begin{align*}
\omega_1 &= \frac{r\varphi_1}{2l} \\
\omega_2 &= -\frac{r\varphi_2}{2l} \\
x_{r1} &= \frac{1}{2}r\varphi_1 \\
x_{r2} &= \frac{1}{2}r\varphi_2
\end{align*}
\]

**Forward kinematics (global coordinate)**

\[
\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R
\]

where

\[
\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}
\]

\[
R(\theta)^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\varphi_1}{2} + \frac{r\varphi_2}{2} \\ 0 \\ \frac{r\varphi_1}{2l} + -\frac{r\varphi_2}{2l} \end{bmatrix}
\]
Kinematics of Simple Mobile Robot

Forward kinematics
\[ \dot{\xi}_I = R(\theta)^{-1} \dot{\xi}_R \]

Inverse kinematics
\[ \dot{\xi}_R = R(\theta) \dot{\xi}_I \]

where
\[ \dot{\xi}_I = \begin{bmatrix} x \\ y \\ \dot{\theta} \end{bmatrix} \]

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

Motor Control
\[ \begin{bmatrix} \frac{r \dot{\varphi}_1}{2} + \frac{r \dot{\varphi}_2}{2} \\ 0 \\ \frac{r \dot{\varphi}_1}{2} + \frac{r \dot{\varphi}_2}{2} \end{bmatrix} \]

Target velocity
\[ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \]
Outline

• Introduction
• Kinematic Motion Control
• Discussion
  – Advanced Motion Control
  – Trends and Challenges
What is Kinematic Control

Designing your own robot:
1. Determine the kinematic chain of your robot
2. Calculate Inverse Kinematics of your robot
3. Determine target robot position/posture (can be time series)
4. Run the controller, then the robot follows the given trajectory

What can this approach NOT do?
Example Question 1

Given a three-link planar manipulator fixed on the base to the origin 0, derive the time-series trajectories of three joints $\theta_1(t)$, $\theta_2(t)$, and $\theta_3(t)$ to move the end-effector from Point A to Point B with a minimum traveling distance. $L_1 = 0.8m$, $L_2 = 0.9m$, $L_3 = 0.6m$; $X_A = 1.3m$, $X_B = -1.0m$, $X_O = 0.9m$, $a = 0.5m$, $b = 0.3m$.

Answer:
First derive the Jacobian matrix:

$$J = \begin{bmatrix} -l_1c_1 - l_2c_1 - l_3c_1 + l_3c_1, l_2c_1 - l_3c_1 + l_3c_1, -l_3c_1 + l_3c_1 \\ -l_1s_1 - l_2s_1 - l_3s_1 + l_3s_1 + l_3s_1 + l_3s_1, -l_3s_1 + l_3s_1 + l_3s_1, -l_3s_1 + l_3s_1 \end{bmatrix}$$

Then calculate $\theta_1(t)$, $\theta_2(t)$, and $\theta_3(t)$, with respect to targets (x3, y3)

Targets (x3, y3) need to be determined by considering the constraints of task-environment
What if there is a heavy object/end effector. Is Inverse-Kinematic Controller going to work? What is the limit of the payload?
Example Question 3

What if there is an unspecified object (or humans) in the environment? Any way to achieve the task without conflicting to the object?
Various Approaches of Motion Control

1. Control engineering approach
2. Sense-think-act approach (CS approach)
3. Behaviour-based approach (Bio approach)
4. Passivity-based approach (MechE approach)
5. Soft approach (Material/Chem approach)
3.1. Control Engineering Approach

- States and actions are represented without ambiguity
- Model-based approach (out-of-model is regarded as noise)
- Target trajectory is given
- Limited to simple (linear) systems

More details in Lecture 2
3.2 Sense-Think-Act Approach

- States and actions are represented without ambiguity
- Often include planning and learning (i.e. autonomously generate target trajectories)
- Sense-think-act scheme (the system always follows this order of processes)
- Also known as “Good Old Fashion AI” - GOFAI

More details in Lecture 4/5/6
Constructing World Model

What is the world model for Shakey to navigate from Point A to B?

- Perception (understanding of sensor signals)
  Obstacles, Floors, Doors, Walls, Rooms, Humans, Slippage,
- Consequences of own motions
  Move fwd/bwd, turning, stuck
- Priorities
- Changes of tasks and environment
  ...

Example: Shakey Robot (1966)

**Initial World Model**

\[
\forall x \forall y \forall z \exists w (\text{CONNECTS}(x,y) \iff \text{CONNECTS}(y,z)) \\
\text{CONNECTS(DOOR1,ROOM1,ROOM5)} \\
\text{CONNECTS(DOOR2,ROOM2,ROOM5)} \\
\text{CONNECTS(DOOR3,ROOM3,ROOM5)} \\
\text{CONNECTS(DOOR4,ROOM4,ROOM5)} \\
\text{LOCINROOM(ROBOT,ROOM4)} \\
\text{AT(BOX1,a)} \\
\text{AT(BOX2,b)} \\
\text{AT(BOX3,c)} \\
\text{AT(LIGHTSWITCH1,d)} \\
\text{ATROBOT(e)} \\
\text{TYPE(BOX1,BOX)} \\
\text{TYPE(BOX2,BOX)}
\]

**Operators**

**goto(m):** Robot goes to coordinate location m.

- **Preconditions:**
  \[
  (\text{ONFLOOR}) \land (\exists x [\text{INROOM(ROBOT,x)} \land \text{LOCINROOM(m,x)}])
  \]
  Delete list: \text{ATROBOT}(8), \text{NEXTTO(ROBOT,8)}
  Add list: \text{ATROBOT}(m)

**go\_\text{\_}m(m):** Robot goes to next item m.

- **Preconditions:**
  \[
  (\text{ONFLOOR}) \land (\exists x [\text{INROOM(ROBOT,x)} \land \text{INROOM(m,x)}] \lor (\exists x \exists y) [\text{INROOM(ROBOT,x)} \land \text{CONNECTS}(m,x,y)])
  \]
  Delete list: \text{ATROBOT}(8), \text{NEXTTO(ROBOT,8)}
3.3 Behaviour-Based Approach

- Different processes have different state/action representations
- Parallel processes can run independently
- Sensing and action processes are asynchronous

More details in Lecture 5
Principles of Parallel Processes

Sensors → perception modeling planning task execution motor control → Actuators

Reason about behaviors
Plan changes to the world
Identify objects
Monitor changes
Build maps
Explore
Wander
Avoid objects

Sensors → Actuators

Brooks, R. (1985)
Subsumption Architecture

Overall architecture

- Overall structure consists of multiple layers
- Each layer works independently without others
- Layers can suppress the sensory input or inhibit motor output as necessary
- Layers can be extended or added anytime later

Individual layer

[Brooks, 1991]
Example: Robotic Vacuum Cleaner

iRobot Roomba Cleaning Robot

- First launched in 2002
- Sold over 10 million robots worldwide
- Sensors: bumper, infra-red proximity, cliff detector, dirt sensor, optic sensor
- Price: ca. £300-600
Example: Craig Reynolds Boids

Behavior of Boids depends on external states: Every agent has a “vision” to observe external states.

5 Rules of Agent’s Motion

- **Separation**: steer to avoid crowding local flockmates
- **Alignment**: steer towards the average heading of local flockmates
- **Cohesion**: steer to move toward the average position of local flockmates
- **Obstacle Avoidance**: allows the flock to avoid obstacles by steering away from approaching objects.
- **Goal Seeking**: drives the flock in the direction of a target/goal.

Goal seeking
3.4 Passivity-Based Architecture

- Mechanical dynamics dominate behaviors of the systems
- Behaviors are often not fully controllable (i.e. underactuated systems)
- Often no state and no action are clearly defined in the controller

More details in Lecture 3
Example: Passive Dynamic Walker,

No Motor
No Controller

Knee-lock mechanism

Pendulum dynamics

Slope

Ground friction

Passive Dynamic Walker, Cornell University

[Collins, et al. 2005]
3.5 Soft Robotics Approach

Functions are distributed in materials

- Controller: Central nervous systems
- Soft Functional Materials: Deformable Cells
- Task-environment: Ecological niche

- No clear distinction between control, body, and environment
- All functions are results of physical/material processes
- Continuum deformable body

[No explicit state]
State space can change

Continuum and deformable body

Universal Gripper

Physically Self-Adapting to Unknown Objects

Elastic bag
(radius = 4.3 [cm], thickness = 0.3 [mm])

Inside: granular material
(ground coffee)

Take home messages

• Robot revolution is happening now but not all problems can be solved immediately

• Inverse-Kinematic Motion Control is the most powerful tool of robot applications. Try this first.

• Various other approaches exist for other challenges.
Thank you!

For publications, video, pictures:

Fumiya Iida
Bio-Inspired Robotics Laboratory
Dept. Engineering, Univ. of Cambridge

Email: fi224@cam.ac.uk
URL: http://divf.eng.cam.ac.uk/birl/