Introduction to ROBOTICS

Midterm Summary

Dr. John (Jizhong) Xiao
Department of Electrical Engineering
City College of New York
jxiao@ccny.cuny.edu

Robotics

• What are Robots?
  – Machines with sensing, intelligence and mobility (NSF)

• Why use Robots?
  – Perform 4A tasks in 4D environments
  4A: Automation, Augmentation, Assistance, Autonomous
  4D: Dangerous, Dirty, Dull, Difficult

Course Coverage

• Robot Manipulator
  – Kinematics
  – Dynamics
  – Control

• Mobile Robot
  – Kinematics/Control
  – Sensing and Sensors
  – Motion planning
  – Mapping/Localization

Robot Manipulator

Homogeneous Transformation

Homogeneous Transformation Matrix

\[ \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{P}_{3 \times 1} \\ 0 & 1 \end{bmatrix} \]

• Composite Homogeneous Transformation Matrix

• Rules:
  – Transformation (rotation/translation) w.r.t. (X,Y,Z) (OLD FRAME), using pre-multiplication
  – Transformation (rotation/translation) w.r.t. (U,V,W) (NEW FRAME), using post-multiplication

Composite Rotation Matrix

• A sequence of finite rotations
  – matrix multiplications do not commute
  – rules:
    • if rotating coordinate O-U-V-W is rotating about principal axis of OXYZ frame, then **pre-multiply** the previous (resultant) rotation matrix with an appropriate basic rotation matrix
    • if rotating coordinate OUVW is rotating about its own principal axes, then **post-multiply** the previous (resultant) rotation matrix with an appropriate basic rotation matrix
**Homogeneous Representation**

- A frame in space (Geometric Interpretation)

\[
F = \begin{bmatrix} R_{i03} & P_{i03} \\ 0 & 1 \end{bmatrix}
\]

**Interpretation (Denavit-Hartenberg Convention)**

- When necessary, Euler angle representation

- Transformation matrices of adjacent joints

- Assign D-H coordinates frames

**Manipulator Kinematics**

- Steps to derive kinematics model:
  - Assign D-H coordinates frames
  - Find link parameters
  - Transformation matrices of adjacent joints

\[
F_i = \begin{bmatrix} P_i & R_i \\ 0 & 1 \end{bmatrix}
\]

- Calculate kinematics model

  - chain product of successive coordinate transformation matrices

\[
F_n = F_n F_{n-1} \cdots F_2 F_1
\]

- When necessary, Euler angle representation

**Denavit-Hartenberg Convention**

- Number the joints from 1 to n starting with the base and ending with the end-effector.

- Establish the base coordinate system. Establish a right-handed orthonormal coordinate system \((X_0, Y_0, Z_0)\) at the supporting base with \(X_0\) axis lying in the plane of motion of joint 1.

- Establish joint axis. Align the \(Z_i\) with the axis of motion (rotary or sliding) of joint \(i+1\).

- Establish origin of the \(i^{th}\) coordinate system. Locate the origin of the \(i^{th}\) coordinate at the intersection of \(Z_i\), or at the intersection of common normal between the \(Z_i\) and \(Z_{i-1}\) axes when they are parallel.

- Establish \(X_i\) axis. Establish \(X_i = 
\]

**Link Parameters**

- Distance from \(O_3\) to \(Z_{i-1}\) about \(X_{i-1}\)

- Distance from intersection of \(Z_{i-1}\) and \(X_i\) along \(X_i\)
Example: Puma 560

\[
q(t_0) = q_0, \quad q(t_1) = q_1, \quad q(t_2) = q_2, \quad q(t_f) = q_f
\]

Jacobian Matrix Revisit

Forward Kinematics

\[
T^n = \begin{bmatrix} n & s & a & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

\[
\begin{array}{ccc}
\frac{\partial x}{\partial s} & \frac{\partial x}{\partial a} & \frac{\partial x}{\partial \rho} \\
\frac{\partial y}{\partial s} & \frac{\partial y}{\partial a} & \frac{\partial y}{\partial \rho} \\
\frac{\partial z}{\partial s} & \frac{\partial z}{\partial a} & \frac{\partial z}{\partial \rho} \\
\end{array}
\]

\[
\begin{array}{ccc}
\frac{\partial h_1}{\partial s} & \frac{\partial h_1}{\partial a} & \frac{\partial h_1}{\partial \rho} \\
\frac{\partial h_2}{\partial s} & \frac{\partial h_2}{\partial a} & \frac{\partial h_2}{\partial \rho} \\
\end{array}
\]

\[
J = \left[ \begin{array}{ccc}
\frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial \phi} & \frac{\partial h_1}{\partial \psi} \\
\frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial \phi} & \frac{\partial h_2}{\partial \psi} \\
\end{array} \right]_{n \times 3}
\]

Trajectory Planning

- Motion Planning:
  - Path planning
    - Geometric path
    - Issues: obstacle avoidance, shortest path
  - Trajectory planning,
    - "interpolate" or "approximate" the desired path by a class of polynomial functions and generates a sequence of time-based "control set points" for the control of the manipulator from the initial configuration to its destination.

- n-th order polynomial, must satisfy 14 conditions,
- 13-th order polynomial
  \[ a_0 x^3 + \cdots + a_1 x + a_2 = 0 \]
- 4-3-4 trajectory
  \[ h_0(t) = a_0 t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4 \]
  \[ 10+3+1, 5 unknown \]
  \[ h_1(t) = a_0 t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4 \]
  \[ 11+4, 4 unknown \]
  \[ h_2(t) = a_0 t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4 \]
  \[ 12+5, 5 unknown \]
- 3-5-3 trajectory
Manipulator Dynamics

Joint torques \( \tau \) and Robot motion, i.e. position velocity,

- Lagrange-Euler Formulation

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \]

- Lagrange function is defined

\[ L = K - P \]

- \( K \): Total kinetic energy of robot
- \( P \): Total potential energy of robot
- \( \dot{q}_i \): Joint variable of \( i \)-th joint
- \( \dot{q}_i \): First time derivative of \( \dot{q}_i \)
- \( \tau_i \): Generalized force (torque) at \( i \)-th joint

\[ \tau = \sum_{i=1}^{n} \tau_i \]

\[ \text{Driving torque applied on each link} \]

Example

Example: 1-link robot with point mass (m) concentrated at the end of the arm.

Set up coordinate frame as in the figure

\[ r_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \]

According to physical meaning:

\[ K = \frac{1}{2} m \dot{\theta}_1^2 \]

\[ P = 9.8 \text{ m/s}^2 \cdot 5 \theta_1 \]

\[ L = K - P \]

\[ \tau = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = \dot{\theta}_1 m \ddot{\theta}_1 - 9.8 \text{ m/s}^2 \cdot C \theta_1 \]

Robot Motion Control

- Joint level PID control

- Each joint is a servo-mechanism
- Adopted widely in industrial robot
- Neglect dynamic behavior of whole arm
- Degraded control performance especially in high speed
- Performance depends on configuration

Joint Level Controller

- Computed torque method

- Robot system:

\[ \dot{q} \quad \dot{q} + H(q, \dot{q}) + C(q) - r \]

\[ \tau = D(q) \dot{q} + h(q, \dot{q}) + k(q^2 - \dot{q}) + H(q, \dot{q}) + C(q) \]

\[ \ddot{q} + k(q^2 - \dot{q}) + k(q^2 - \dot{q}) = 0 \]

Error dynamics:

\[ \dddot{q} + k(q^2 - \dot{q}) + k(q^2 - \dot{q}) = 0 \]

Advantage: compensated for the dynamic effects

Condition: robot dynamic model is known exactly

Manipulator Dynamics

- Dynamics Model of n-link Arm

\[ \tau = D(q) \ddot{q} + h(q, \dot{q}) + C(q) \]

\[ P = \begin{bmatrix} D_{11} & \cdots & D_{1n} \\ \vdots & \ddots & \vdots \\ D_{n1} & \cdots & D_{nn} \end{bmatrix} \]

The Acceleration-related Inertia matrix term, Symmetric

\[ h(q, \dot{q}) = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} \]

The Coriolis and Centrifugal terms

\[ C(q) = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \]

The Gravity terms

\[ \tau = \tau_1 \]

Driving torque applied on each link

Example: 1-link robot with point mass (m) concentrated at the end of the arm.

Set up coordinate frame as in the figure

\[ p = \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \]

\[ q = \begin{bmatrix} r \\ \theta \end{bmatrix} \]

\[ P = -m g r \cos \theta = -m g (T_1 \dot{\theta} \dot{r}) \]

\[ g = (g_x, g_y, g_z) \]

\[ \dot{r} = 9.8 \text{ m/s}^2 \]

\[ g \text{ : gravity row vector expressed in base frame} \]

- Potential energy of link \( i \)

\[ P_i = -m g T_i \dot{r} \]

\[ T_i : \text{Center of mass w.r.t. base frame} \]

\[ r_i : \text{Center of mass w.r.t. i-th frame} \]

- Potential energy of a robot arm

\[ P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} [-m g (T_i \dot{r})] \]

Function of \( \dot{q}_i \)
Robot Motion Control

Error dynamics: \[ \ddot{e} + k_e \dot{e} + k_v e = 0 \]

Define states: \[ x_i = e \Rightarrow \dot{x}_i = \dot{x}_i - k_v x_i \]

In matrix form: \[ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_v & -k_v \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \Delta X \]

Characteristic equation: \[ |X - \lambda I| = \lambda^2 + k_v \lambda + k_v = 0 \]

The eigenvalues of A matrix are: \( \lambda_1, \lambda_2 \) such that \( \lambda_1 > 0 \), \( \lambda_2 > 0 \)
Condition for stability: have negative real parts and the selection of \( k_v > 0 \)

Task Level Controller

Nonlinear Feedback Controller:

\[ \tau = D(q)J^{-1}(U - \dot{J}q) + H(q, \dot{q}) + C(q) \]

Then the linearized dynamic model:

\[ D(q)J^{-1}\ddot{Y} = D(q)J^{-1}U \Rightarrow \ddot{Y} = U \]

Linear Controller: \[ U = \ddot{Y} + k_v (\ddot{Y} - \dot{Y}) + k_e (\dot{Y} - \dot{Y}) \]

Error dynamic equation: \[ \ddot{e} + k_e \dot{e} + k_v e = 0 \]

Midterm Exam

- Study lecture notes
- Understand homework and examples
- Have clear concept

- 2-hour exam
- close book, close notes
- But you can bring one-page cheat sheet

Thank you!

Next class: Nov. 1 (Tue): Midterm Exam
Time: 6:45-8:45