A Stochastic Optimization Approach for Energy-Efficient Robotic Manipulator Simulations

Seminars on Undergraduate Mathematics in Montreal 2024

Laurence Liang

January 6, 2024

Reference Source

- Most equations in the following slides are from *Modern Robotics* by K. M. Lynch and M. C. Park.
- *Introduction to Autonomous Robotics* by N. Correll is also used.

Motivation to Study Robots

- "Robotics is about turning ideas into action. Somehow, robots turn abstract goals into physical action: sending power to motors, monitoring motions, and guiding things towards the goal. Every human can perform this trick, but it is nonetheless so intriguing that it has captivated philosophers and scientists, including Descartes and many others"
  - Matthew T. Mason, Foreword for *Modern Robotics* by Lynch and Park
- Tasks that are mundane or occur in extreme environments could be performed by robots.
- Studying robots is hard, but it turns out that a set of mathematics is relatively good at producing working and reproducible results.

Foundational Ideas for Kinematics

- Each point P (dimension 0) can be represented by 2 or 3 spatial dimensions
  - i.e. $P = (x = 0, y = 1, z = 0.5)$
- We use position vectors to calculate joint positions
- A positional vector can be written as a function of time $\vec{x}(t)$
  - $\dot{\vec{x}}(t)$ is velocity
  - $\ddot{\vec{x}}(t)$ is acceleration
  - $\dddot{\vec{x}}(t)$ is jerk

Foundational Ideas for Dynamics

- Newton's laws of motion
  - Newton's Second Law: $\vec{F} = m\vec{a}$

Foundational Definitions of Robotic Arms

- **Link**: typically a bar between two joints
- **Joint**: the connector between each linkage. Joint types include revolute (R) or prismatic (P).
- **Degress of freedom (DoF)**: the number of ways in which a joint can move about (or along) an axis.

Forward Kinematics

- We use Denavit-Hartenberg (DH) notation to standardize calculations.
- Consider the 4×4 T matrix:

$$T_{n-1,n} = \begin{bmatrix} R & \vec{p} \\ 0 & 1 \end{bmatrix} \tag{1}$$

  where R is a 3×3 rotational matrix, $\vec{p}$ is a position vector between joint $n-1$ and joint n.
- The rotation matrix is based on the SO(3) Lie group.
- We can further write:

$$T_{0n} = T_{01} T_{12} ... T_{n-1,n} \tag{2}$$

  which describes the final position of the last joint (the end effector).

Inverse Kinematics

- Inverse kinematics problems can typically be solved analytically for a low number of degrees of freedom.
- Robotic manipulators with greater complexity usually require numerical solutions, as shown below:

$$g(\theta_d) = x_d - f(\theta_d) = 0 \tag{3}$$

$$\Delta\theta = J^{\dagger}(\theta_0)(x_d - f(\theta_0)) \tag{4}$$

$$e = x_d - f(\theta_i) \tag{5}$$

While $||e|| > e$:

$$\theta_{i+1} = \theta_i + J^{\dagger}(\theta_i)e \tag{6}$$

Forces or Torque on Each Joint

- **Lagrangian mechanics** can be used in some cases (i.e. low degrees of freedom) to obtain the generalized forces f (force or torque) at each joint:

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \tag{7}$$

$$f = \frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \tag{8}$$

Foundational Ideas for Manipulation

- Idea: Consider what physics-derived principle you want to grasp an object.

Trajectory Planning

- We define a path as $\theta(s)$, where $0 \leq s \leq 1$.
- We can also define s as $s(t)$.
- $\dot{\theta}(s)$ is velocity at time t, and $\ddot{\theta}(s)$ is acceleration at time t.
- We can then use **polynomial interpolation** to have the trajectory pass through a set of points in 2D or 3D space.
- We can also identify time-optimal trajectories within the motor (actuator)'s working limits, as θ is dependent on t.

Motion Planning

- Defines the motion (the joint angles or positions) of the robot arm in order to follow the desired path.
- One approach involves numerically solving the joint states at every small timestep.
- How can we deal with error accumulation over time? Control theory is one tested approach!

Control Theory

- The fundamental idea is to adjust the "state" of the robot by a small amount, based on the robot's error:

$$\theta_{i+1} = \theta_i - Ke \tag{9}$$

where K is a coefficient in $\mathbb{R}$ and e is the measured error.

- We can design more robust controllers, such as the Proportional-Integral-Derivative (PID) controller:

$$\tau = K_p\theta_e + K_i \int \theta_e(t)dt + K_d\dot{\theta} - e \tag{10}$$

where coefficients $K_p, K_i, K_d \in \mathbb{R}$

Identifying Energy-Efficient Trajectories

- Robots with high degrees of freedom face many possible trajectories, and it is not easy identifying the set of robot motions that require the least amount of energy expended.
- Existing approaches include:
  - Shortest path traversals
  - Sampling algorithms
  - Rapidly Exploring Random Tree Star (RRT*)
  - Covariant Hamiltonian Optimization for Motion Planning (CHOMP)
  - Stochastic Trajectory Optimization for Motion Planning (STOMP)
- Finding an energy optimal motion path becomes difficult with high degrees of freedom and highly complex desired trajectories.

A Proposed Stochastic Approach for Minimizing Expended Energy

1. Define a starting point P_{start} and an end point P_{end}.
2. Calculate the joint angles for an initial position, based on a randomized initial guess with weights W_I
3. Define a smooth path $\vec{L}_n$ from P_{start} to P_{end} based on a randomized initial set of weights W_P.
4. Define E_E as the total energy expended during this trajectory
5. Calculate the gradient ∇E_E in terms of W, which is the set of W_I and W_P combined.
6. By iteratively updating W_I and W_P, we might be able to converge to a local minima.

The Problem Statement and Desirable Properties

- Problem Statement
  - For what values of W will $E_E(W)$ be at a local minima?
- Desirable Properties
  - Possible convergence even with high dimensionality
  - Intuitive to visualize and understand
  - Customizable initial conditions due to W

An Overview of Some Mathematical Terms

- Defining the path trajectory L_n

$$\vec{L_n} = \vec{L_0} + \frac{n}{K}\vec{t} + w_n\vec{n_t} \tag{11}$$

$$\vec{L_n}(w_n) = (x_n, y_n) \tag{12}$$

$n, K, w_n \in \mathbb{R}$
$\vec{L_n}, \vec{L_0}, \vec{t}, \vec{n_t} \in \mathbb{R}^2$

  - We can apply polynomial interpolation to define a set of adjacent spatial points.

- Calculating energy expended:

$$E_E(W) = \sum_{n=0}^{n} m_n g h_n + \frac{1}{2} m_n V_n^2 \tag{13}$$

with step size of n as small as possible.

- Performing stochastic gradient descent (SGD):

$$W_{i+1} = W_i - \alpha \nabla E_E(W_i) \tag{14}$$

where $\alpha \in \mathbb{R}_+^*$ and α very small by definition.

Limitations and Unanswered Questions

- Limitations
  - High computational cost to perform SGD
  - Assume no guarantees to converge to the global minima
  - Does not take time-dependent operations, collision detection or obstacles into account.
- Unanswered Questions
  - How can we compute the gradient of $E_E(W)$ in terms of W?
  - How would robots behave in simulators or physical reality, given the solutions from this approach?
  - Does this method provide any advantages over existing methods?
  - How robust are the generated trajectories, given a PID controller and error accumulation?
  - How can this method deal with singularities?

General Questions

- Do analytical solutions exist to identify the most energy efficient trajectory and motion of a robotic manipulator with many degrees of freedom?
- Are there conditions that can define the existence and uniqueness of the most energy-efficient solution?
- Are there conditions that guarantee finding the global minima for the most energy-efficient solution?

Imagine the Possibilities

Figure: Perseverance Rover. Image credit: NASA.

Thank You

Contact: laurence.liang@mail.mcgill.ca

Supplementary Slides

2-Linkage Planar Arm

$$T_{01} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$T_{12} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$T_{23} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$T_{03} = T_{01}\,T_{12}\,T_{23} \tag{18}$$

2-Linkage Planar Arm

$$\begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_1c_2c_3-c_3s_1s_2-c_2s_1s_3-c_1s_2s_3 & -c_2c_3s_1-c_1c_3s_2-c_1c_2s_3+s_1s_2s_3 & 0 & L_1c_1+L_2c_1c_2-L_2s_1s_2 \\ c_2c_3s_1+c_1c_3s_2+c_1c_2s_3-s_1s_2s_3 & c_1c_2c_3-c_3s_1s_2-c_2s_1s_3-c_1s_2s_3 & 0 & L_1s_1+L_2c_2s_1+L_2c_1s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure: $T_{03} = T_{01} T_{12} T_{23}$

$$p_x = L_1 \cos\theta_1 + L_2 \cos\theta_1 \cos\theta_2 - L_2 \sin\theta_1 \sin\theta_2 \tag{19}$$

$$p_y = L_1 \sin\theta_1 + L_2 \cos\theta_2 \sin\theta_1 + L_2 \cos\theta_1 \sin\theta_2 \tag{20}$$

$$\frac{\partial p_x}{\partial \theta_1} = -L_1 \sin\theta_1 - L_2 \sin\theta_1 \cos\theta_2 - L_2 \cos\theta_1 \sin\theta_2 \tag{21}$$

$$\frac{\partial p_x}{\partial \theta_2} = -L_2 \cos\theta_1 \sin\theta_2 - L_2 \sin\theta_1 \cos\theta_2 \tag{22}$$

$$\frac{\partial p_y}{\partial \theta_1} = L_1 \cos\theta_1 + L_2 \cos\theta_2 \cos\theta_1 + L_2 \sin\theta_1 \sin\theta_2 \tag{23}$$

$$\frac{\partial p_y}{\partial \theta_2} = -L_2 \sin\theta_2 \sin\theta_1 - L_2 \cos\theta_1 \cos\theta_2 \tag{24}$$

Stochastic Gradient Descent

$$L(h_1, h_2, h_3) = ? \tag{25}$$

$$\vec{L_0} = (x_0, y_0) \tag{26}$$

$$\vec{L_n} = \vec{L_0} + \frac{n}{K}\vec{t} + w_n\vec{n_t} \tag{27}$$

$n, K, w_n \in \mathbb{R}$
$\vec{L_n}, \vec{L_0}, \vec{t}, \vec{n_t} \in \mathbb{R}^2$

$$\vec{L_n}(w_n) = (x_n, y_n) \tag{28}$$

Some Additional Resources (after SUMM 2024)

- https://www.cs.cmu.edu/ venkatrn/papers/icra14.pdf