Lecture 12: Trajectory Formation

Contents:

• Based on Heuristic Optimisation
• ZMP based walking
• Optimization Criterion Based
  • Minimum Distance, Time
  • Minimum Acceleration Change, Torque Change
  • Minimum End Point Variance
• Multiple Model Learning
• Using Motor Redundancies Efficiently
Trajectory Planning Phases

- **Trajectory generation**
  - Involves computation of the best trajectory for the object

- **Force Distribution**
  - Involves determining the force distribution between different actuators (a.k.a. resolving actuator redundancy)

Some of the approaches solve the trajectory generation and force distribution problems separately in two phases.

It has been argued that solving the two issues simultaneously (as a global optimization problem) is superior in many cases.

**Kinematics**: refers to geometrical and time-based properties of motion; the variables of interest are positions (e.g. joint angles or hand Cartesian coordinates) and their corresponding velocities, accelerations and higher derivatives.

**Dynamics**: refers to the forces required to produce motion and is therefore intimately linked to the properties of the object such as it’s mass, inertia and stiffness.
Question: how to generate good trajectories?

- (Very) simple options:
  - Provide a few desired positions (angles) over time (controller with step functions)
  - Provide smooth hand-tuned trajectories (e.g. with spline fitting)
  - Use a sinusoidal controller

- These are open-loop solutions, i.e. no feedback to the trajectory planner
Simply provide a new desired angle every so often:

Controller with step function

Desired angle

Actual angle

DOF $i$

DOF $j$

Problems:
- very saccadic motion,
- depends strongly on PID gains
Hand-tuned trajectories

- Provide a trajectory as a vector
- Hand-tuned trajectory: usually only give a few via points
- Use of spline-fitting to interpolate

**DOF \( i \)**

**Problems:**
- How to be sure that the locomotion is stable?
- Need for mathematical tools to prove stability, c.f. ZMP control in a few slides
• Desired angle for each DOF: \[ \theta_i = \theta_i^0 + A_i \sin(\nu_i t + \varphi_i) \]

• Problem: finding, for each DOF \( i \) suitable \( \theta_i^0, A_i, \nu_i \) and \( \varphi_i \)

Open-loop sinus-based controller
Walking based on Trajectory Methods

- **Main idea:** design walking kinematic trajectories, and use the dynamic equations to test and prove that locomotion is stable.
- Trajectories are designed by trial-and-error, or from human recordings.
Zero Moment Point Approach

• Method for proving that a trajectory is stable

• *Zero Moment Point (ZMP)*: point on the ground about which the net moment of the inertial forces and the gravity forces has no component along the horizontal planes (a.k.a. center of pressure, CoP).

• ZMP is different from the projection of the center of mass (CoM) on the ground

• ZMP ~ projection on the ground of the point around which the robot is rotating
**ZMP Approach**

\[ \mathbf{p}_{ZMP} \text{ is such that:} \]

\[ \mathbf{r}_i = \mathbf{p}_i - \mathbf{p}_{ZMP} \]

\[
\sum_{i=1}^{N} \left( \mathbf{r}_i \times m_i \mathbf{a}_i + \mathbf{I}_i \mathbf{a}_i + \mathbf{\omega}_i \times \mathbf{I}_i \mathbf{\omega}_i - \mathbf{r}_i \times m_i \mathbf{g} \right) = (0,0,\ast)^T
\]

- \( N \): number of links
- \( p_i \): position of link \( i \)
- \( m_i \): mass of link \( i \)
- \( \mathbf{a}_i \): external acceleration
- \( \mathbf{I}_i \): moment of inertia (matrix)
- \( \mathbf{a}_i \): angular acceleration
- \( \mathbf{\omega}_i \): angular velocity
- \( \mathbf{g} \): gravity

Two independent equations, Two unknowns: \( p_{ZMP,x} \) \( p_{ZMP,y} \)
Locomotion is **stable** if the ZMP remains within the *foot-print polygons*.

Foot-print polygon
Honda, SONY and some HRP robots use ZMP control
ZMP Approach

Most used method:
1. Human motion capture for getting trajectories,
2. Modify trajectories such that locomotion is stable according to the ZMP criterion
3. Add online stabilization to deal with perturbations.

Example of online stabilization:
• Use of hip actuators to manipulate the ZMP
Some Simple Cost Functions

- **Shortest Distance**

- **Minimum Acceleration**

- **Minimum Time (Bang-Bang Control)**
Minimum Jerk Trajectory Planning

- Proposed by Flash & Hogan (1985)
- Optimization Criterion minimizes the jerk in the trajectory

\[ C_J = \frac{1}{2} \int_0^T \left( \left( \frac{d^3 x}{dt^3} \right)^2 + \left( \frac{d^3 y}{dt^3} \right)^2 \right) dt \]

For movement in x-y plane

- The minimum-jerk solution can be written as:

\[
\begin{align*}
x(t) &= x_0 + (x_0 - x_f)(15\hat{t}^4 - 6\hat{t}^5 - 10\hat{t}^3) \\
y(t) &= y_0 + (y_0 - y_f)(15\hat{t}^4 - 6\hat{t}^5 - 10\hat{t}^3)
\end{align*}
\]

where \( \hat{t} = t / t_f \) and \((x_0, y_0)\) are initial coordinates at \( t = 0 \).

- Depends only on the kinematics of the task and is independent of the physical structure or dynamics of the plant
- Predicts bell shaped velocity profiles
Minimum Torque Change Planning

- Proposed by Uno, Kawato & Suzuki (1989)
- Optimization Criterion minimizes the change of torque

\[ C_T = \frac{1}{2} \int_0^T \left( \left( \frac{d \tau_1}{dt} \right)^2 + \left( \frac{d \tau_2}{dt} \right)^2 \right) dt \]

- The Min. Jerk and Min. Torque change cost functions are closely related since acceleration is proportional to torque at zero speed.
- No Analytical solution possible for Min. Torque change criterion but iterative solution is possible.

- Like Min. Jerk, predicts bell shaped velocity profiles.
- But also predicts that the form of the trajectory should vary across the arm’s workspace.

For two joint arm or robot
Min. Jerk vs Min. Torque Change

One way of resolving how humans plan their movement is by setting up an experiment which can distinguish between the kinematic vs dynamic plans.
Min. Jerk vs Min. Torque Change (II)

Most studies suggest that trajectories are planned in visually-based kinematic coordinates.

No perturbations at the start & end.

Perturbation

Hypotheses

Kinematic coordinates

Dynamic coordinates

Predictions

Adaptation

No adaptation
Minimum Endpoint Variance Planning

Proposed by Harris & Wolpert (1998)
Also called TOPS (Trajectory Optimization in the Presence of Signal dependent noise)

Basic Theory:

- Single physiological assumption that neural signals are corrupted by noise whose variance increases with the size of the control signal.
- In the presence of such noise, the shape of the trajectory is selected to minimize the variance of the final end-point position.
- Biologically more plausible since we do have access to end point errors as opposed to complex optimization processes (min. jerk and min. torque change integrated over the entire movement) that other optimization criterion suggest the brain has to solve.

Testing the Internal Model Learning Hypothesis
Using the force manipulandum, one can create an interesting experiment in which you modify the dynamics of your arm movement in only the x-y plane.

Humans are very adept at learning these changed dynamic fields and adapt at a relatively short time scale.

When we remove these effects, after effects of learning can be felt for some time before re-adaptation, providing evidence that we learn internal models and use them in a predictive feedforward fashion.
Multiple Model Hypothesis
Recursive Bayes Estimation

\[ P(y | x) = \frac{P(x | y)P(y)}{P(x)} \]

If we explicitly index the individual experiences by \( n \), i.e., \( \mathcal{D}^n = \{x_1, \ldots, x_n\} \),

Using ...

\[ p(D | \theta) = \prod_{k=1}^{n} p(x_k | \theta) \]
\[ p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\int p(D | \theta)p(\theta)d\theta} \]

We get ...

\[ p(D^n | \theta) = p(x_n | \theta)p(D^{n-1} | \theta) \]
\[ p(\theta | D^n) = \frac{p(x_n | \theta)p(\theta | D^{n-1})}{\int p(x_n | \theta)p(\theta | D^{n-1})d\theta} \]

where \( p(\theta | D^0) = p(\theta) \)
Multiple model hypothesis (II)

Multiple Paired Forward-Inverse Models (MPFIM)

Resolving Motor Redundancies

Inverse Kinematics

\[(\Delta x) \rightarrow (\Delta \theta) \quad f_{\text{eye+head}} : \mathbb{R}^4 \rightarrow \mathbb{R}^7\]

\[x = \begin{pmatrix} x_R & x_L & y_R & y_L \end{pmatrix}^T\]

\[\theta = \begin{pmatrix} \theta_{\text{Head}} & 1 & \theta_{\text{Head}} & 2 & \theta_{\text{EyeLP}} & 3 & \theta_{\text{EyeLT}} & \theta_{\text{EyeRP}} & \theta_{\text{EyeRT}} \end{pmatrix}^T\]

Retinal Displacement

Eye & Head displacements
Resolved Motion Rate Control with locality in joint positions

\[ \Delta x = J(\theta) \Delta \theta \]

Integrate over small increments (where linearity holds) to get complete trajectory

Based on collected training data forward kinematics in velocity space was almost completely linear, irrespective of joint position

Hence, in this case, we can use pseudo-inversion with the constant Jacobian!!

Null Space Manipulation

\textbf{Given} \dot{x} = J \dot{\theta}, \textit{what is} \dot{\theta}?

\dot{\theta} = J^\# \dot{x} + (I - J^\# J) k_{null}

\begin{align*}
J^\# &= J^T (JJ^T)^{-1} & \text{: Pseudoinverse} \\
k_{null,i} &= -\frac{\partial L_{opt}}{\partial \theta_{current,i}} \\
L_{opt} &= \min \frac{1}{2} \sum w_i (\theta_{current,i} - \theta_{default,i})^2 & \text{Optimization criterion}
\end{align*}

Optimization based on gradient descent in Null space

\text{Inertial Weighting for each joint}

Lecture 12: RLSC - Prof. Sethu Vijayakumar