Loco-manipulation

Jane Li

Assistant Professor
Mechanical Engineering Department, Robotic Engineering Program
Worcester Polytechnic Institute
Quiz (10 pts)

- (6 pts) A 2-DOF manipulator arm is attached to a mobile base with non-holonomic constraints. How does the mobile base affect the manipulability when the 2-DOF is at its singularity configuration?

- (2 pts) What is loco-manipulation affordance?

- (2 pts) How to extract loco-manipulation affordance given a RGB+D camera?
Manipulability of 2-DOF arm

Ellipsoids of manipulability

Measure of manipulability
Manipulability of mobile manipulator
Manipulability of mobile manipulator

- Ellipsoids of manipulability
- Measure of manipulability
Affordance of loco-manipulation

- Loco-manipulation affordance
  - Actions that involve the whole body for stabilization, locomotion or manipulation

- Affordance validation
  - Assign whole-body affordance to environmental primitives, based on their shape, orientation and extent
  - Use perception feedback to validate the affordance hypotheses
  - Execute the task
Typical loco-manipulation tasks
Affordance extraction

Registered point cloud → Segmentation → Geometric primitives → Affordance hypothesis
Optional assignment
Optional assignment

• Student talk on “trajectory optimization”
  • If you need to make up for your low-score/late submission assignment
  • So far, 7 students signed up in total
  • Three lectures + additional section on the day of course review

• Reference:
  • http://www.matthewpeterkelly.com/tutorials/trajectoryOptimization
Optional assignment

- **Wednesday, April 4**
  - Samruddhi Kadam  spkadam@wpi.edu

- **Friday, April 6**
  - Nalin Raut  nraut@wpi.edu
  - Abhilasha Rathod  arathod@wpi.edu
  - Nathaniel Goldfarb

- **Wednesday, April 11**
  - Max Merlin – lecture with Gunnar on high-level motion planning
  - Guled Elmi  ggelmi@wpi.edu
  - Gaurav Vikhe  gsvikhe@wpi.edu
Literature review student talk

- 4/13/2018
  - Bimanual team, Swarm team

- 4/18/2018
  - High-level planning
Project presentation

- 4/25/2018
  - Mobile team, Bimannual team, High-level planning
  - Surgical robot (Sam)

- 4/27/2018
  - pHRI team
Loco-Manipulation
Overview

• Loco-manipulation
  • Affordance

• Loco-manipulation motion planning
  • Motion Primitives

• Motion skill transferring from humans to humanoid robots
  • Inverse optimal control
Complex loco-manipulation can be composed using **parametrized control laws** (i.e., motion primitives)

Simultaneous execution of motion primitives may cause **instability**
Whole-body motion planning

- Whole-body planning
  - High dimensional, numerically intractable problems
  - Multi-contacts, many constraints

- Pseudo-inverse
  - Prioritized tasks and constraints
  - Project secondary tasks to the null space of pseudo-inverse Jacobian

- Sampling-based strategy
  - Sample and search the solution in C-space
  - How to address tasks constraints?
Sampling-based planning with Constraints

Arm with three joints

Heavy Object (Torque Constraint)

Sliding Surfaces (Pose Constraint)
Joint 1
Joint 2
Joint 3
Configuration Space (C-space)
Rapidly-exploring Random Tree (RRT)
Rejection Sampling and Pose Constraints

C-space

Full Dimensional

Lower Dimensional
Projection Sampling

• Sample on any manifold or dimension
Constrained BiDirectional RRT (CBiRRT)
Projection Sampling

- Gradient descent on distance metric to reach manifold
More fluent and efficient motions
P-search*

- Derived from RRT*

- Similar to informed RRT
  - Use the information available from the primitives design to structure a sampling space with desirable properties
- **RRT**
  - Extends the nearest vertex towards the sample

- **RRG**
  - Extends all vertices returned by the Near procedure (if first was success).
• Similar to RRG, except for "rewiring" the tree as better paths are discovered.

• After rewiring the cost has to be propagated along the leaves
RRT*

Similar to RRG

Rewire and propagate cost
Limitation of RRT*

- RRT* is asymptotically optimal everywhere
- Not necessary for single-query planning
- Improvement
  - Limit the search to the **sub-problem** that would have a better solution
  - How to define the space of sub-problem?
The sub-problem can be defined as “search in a n-dimensional ellipse” where to draw the new sample.
heuristic sampling domain

\[ X_f = \left\{ x \in X \mid \|x_{start} - x\|_2 + \|x - x_{goal}\|_2 \leq c_{best} \right\} \]
Informed RRT*
Informed RRT*

By *directly* sampling the ellipse, we focus the search to only the states that have the possibility of improving the solution.
Definition 1: We define a generic motion primitive $\pi$ as a 6-tuple $\pi(q, \chi, \sigma, T, \xi, C)$ with

- $q \in \mathbb{Q}$: the parameters that characterize the primitive;
- $\chi$: the image space of the primitive that corresponds to the image space of the output function of the dynamical system;
- $\sigma: X \times \mathbb{Q} \to \chi$: the steering function of the primitive that is a set-valued function based on the system dynamics from the primitive space to the image space; it can be a map on $(0, 1)^d$, with $d \geq 2$;
- $T \in \mathbb{R}_{\geq 0}$: the duration of the execution of the primitive;
- $\xi = \rho(t, y), \rho: \mathbb{R}_{\geq 0} \times \chi \to \Xi = \{0, 1\}$: a trigger that enables the execution of the primitive, where $t$ is the time variable;
- $C: \mathbb{R}_{\geq 0} \times X \times \mathbb{Q} \to \mathbb{R}$: the cost function associated with the primitive.

Example – Locomotion primitives

- $q_L = \emptyset$,
- $\chi_L \ni [x, y, \theta, v]^T$,
- $\sigma_L$, an optimization routine, applied on a simplified dynamics, minimizes the time variable $t$ subject to state and control constraints and returns the robot desired trajectory,
- $C_L = t$,
- $T_L$, is the duration of execution of the steering function $\sigma_L$,
- $\xi_L = 0$ until a sample laying in $\chi_L$ is added to $T$. 

RBE 550 – Motion Planning – Instructor: Jane Li, Mechanical Engineering Department & Robotic Engineering Program - WPI 

4/4/2018 37
Motion primitives as parameterized actions

Definition 1: We define a generic motion primitive \( \pi \) as a 6-tuple \( \pi(q, \chi, \sigma, T, \xi, C) \) with:

- \( q \in Q \): the parameters that characterize the primitive;
- \( \chi \): the image space of the primitive that corresponds to the image space of the output function of the dynamical system;
- \( \sigma : X \times Q \rightarrow \chi \): the steering function of the primitive that is a set-valued function based on the system dynamics from the primitive space to the image space; it can be a map on \((0, 1)^d\), with \( d \geq 2 \);
- \( T \in \mathbb{R}_{\geq 0} \): the duration of the execution of the primitive;
- \( \xi = \rho(t, y) \), \( \rho : \mathbb{R}_{\geq 0} \times \chi \rightarrow \Xi = \{0, 1\} \): a trigger that enables the execution of the primitive, where \( t \) is the time variable;
- \( C : \mathbb{R}_{\geq 0} \times X \times Q \rightarrow \mathbb{R} \): the cost function associated with the primitive.

Example – manipulation primitives

- \( q_M = o \), where \( o \) is the object pose,
- \( \chi_M \ni [x \ y \ \tau]^T \),
- \( \sigma_M \), the inverse kinematics of the robotic arm, giving the joints desired values corresponding to a certain value of \( o \) and \( \tau \),
- \( C_M = t \),
- \( T_M \), is the duration of execution of the steering function \( \sigma_M \),
- \( \xi_M = 1 \) when \( \|o - \tau\| \leq \delta \), with \( \tau \) the pose of the robot and \( \delta > 0 \), otherwise it is 0.
Main idea

• Main idea
  • Use the motion primitives for a subsystems as local planner in classical sample based planning algorithms to obtain a plan for the whole system

• Basic assumption
  • A motion primitive has an associated control law that stabilize the subsystem it belongs to, while the control of other sub-systems are null (i.e., generate steady motion)

• Check for feasibility
  • e.g. using ZMP-condition for humanoid robots
P-Search* algorithm

Algorithm 1: $\mathcal{T} \leftarrow \text{P-Search}^*(z_I, z_G)$

Data: $\mathcal{P} = (V, E), z_I, z_G$

Result: $\mathcal{T} \text{ a tree whose vertices are points } z \in \chi$

Given two vertices $z_i, z_j \in \chi_k$ an edge $(z_i, z_j)$ is an instantiation of the primitive $\pi_k \in V$ that steers $z_i$ toward $z_j$ in $\chi_k$.

1. $\mathcal{T} \leftarrow \text{InsertNode}(\emptyset, z_I, \mathcal{T})$;
2. $z_{\text{new}} = z_I$;
3. for $i = 1 \text{ to } N$ do
4. \hspace{1em} $\mathcal{P}_A \leftarrow \text{ActivePrimitives}(z_{\text{new}})$;
5. \hspace{1em} $\chi_i \leftarrow \text{SamplePrimitive}(\mathcal{P}_A)$;
6. \hspace{1em} $z_{\text{rand}} \leftarrow \text{Sample}(\chi_i)$;
7. \hspace{1em} $(z_{\text{new}}, \mathcal{T}) \leftarrow \text{LocalRRT}^*(\chi_i, z_{\text{rand}}, \mathcal{T})$;
8. return $\mathcal{T}$;

Motion primitive available given the current states

Pick up one motion primitives (e.g. choose manipulation primitives)

Specify the motion primitive (e.g., sample a set of joint angles for manipulation motion)
P-Search* algorithm

Algorithm 2: \((z_{\text{new}}, \mathcal{T}) \leftarrow \text{LocalRRT}^*(\chi_k, z, \mathcal{T})\)

1. \(z_{\text{nearest}} \leftarrow \text{Nearest}(\chi_k, z, \mathcal{T})\);
2. \((z_{\text{new}}, x_{\text{new}}) \leftarrow \text{steer}(z_{\text{nearest}}, z)\);
3. \text{if Unfeasible}(x_{\text{new}}) \text{ then}
   \text{ return } (-, \mathcal{T});
4. \text{if ObstacleFree}(x_{\text{new}}) \text{ then}
   \text{ if } \ Z_{\text{near}} \leftarrow \text{Near}(z_{\text{new}}, \mathcal{T})\;
   \text{ z_{\text{min}} } \leftarrow \text{ChooseParent}(\mathcal{T}, Z_{\text{near}}, z_{\text{nearest}}, z_{\text{new}});
   \mathcal{T} \leftarrow \text{InsertNode}(z_{\text{min}}, z_{\text{new}}, \mathcal{T});
   \mathcal{T} \leftarrow \text{Rewire}(\mathcal{T}, Z_{\text{near}}, z_{\text{min}}, z_{\text{new}});
   \text{ return } (z_{\text{new}}, \mathcal{T});
5. \text{ else}
6. \text{ return } (-, \mathcal{T});

Check feasibility (e.g., ZMP-condition)

Rewiring
Experiment

No-zero walking speed
Transfer human walking motion to humanoids [2]
Inverse optimal control

human motion

data

inverse optimal control
(with human dynamical model)

weights

objective
(for humans)

transfer rules

robot motion

trajectories

optimal control
(with robot dynamical model)

objective
(for robot, usually robot specific)
Optimality criteria

• Actuation and energy consumption
  • Minimize actuation in the stance foot, swing foot, torque, hip torque of the swing foot, angular momentum in x and y direction, vertical center of mass oscillations, absolute swing foot velocity

• Motion fitting error
  • Minimize planar distance between foot position at touch down and capture point, periodicity gap in center of mass velocities

• Others
  • Minimize overall single support duration, absolute swing foot velocity at touch down
Demonstration

- Link for demo video:


Extra credit homework – evaluation form