Physics-Based Reinforcement Learning for Mobile Manipulation

PhD Dissertation Defense
Jonathan Scholz
August 17, 2015

Committee:
Dr. Charles Isbell (IC, Georgia Institute of Technology)
Dr. Andrea Thomaz (IC, Georgia Institute of Technology)
Dr. Henrik Christensen (IC, Georgia Institute of Technology)
Dr. Magnus Egerstedt (ECE, Georgia Institute of Technology)
Dr. Michael Littman (CS, Brown University)
Thesis Vision

Robots operating in human environments

Learning behaviors from experience
Good Ideas

• Self-guided exploration

• Shaping behavior through feedback
Reinforcement Learning

Agent, State $s_t$ ➔ Action $a_t$ ➔ Next State $s_{t+1}$ ➔ Reward $r_t$ ➔ World

- **Pacman states:** all positions of pacman, ghosts, food, & pellets
- **Pacman actions:** {N,S,E,W}
- **Pacman rewards:** -1 per step, +10 food, -500 die, +500 win,...
RL+Robotics: Prior work

- Humanoid Walking
  (Peters, Schaal, Vijayakumar 2003)

- Acrobatic Helicopter
  (Ng et al. 2003)

- Ball-in-Cup
  (Kolber & Peters 2009)

- PILCO: Cart-Pole
  (Diesenroth et al. 2011)

- PILCO: Block-Stacking
  (Diesenroth et al. 2011)
Domestic Manipulation Tasks

Task: Stand on Other Platform
Standard Humanoid Locomotion Planner

Not achieved using RL methods
Limitations

Task

Sensorimotor

Feedback

Decisions
Sensorimotor Level

What we see

What the robot sees

Pixel Intensities

Joint Currents

2,073,618 Dimensional Time-Series!
Physical Object Level

What we see

What the robot sees

It is hard to imagine a truly intelligent agent that does not conceive of the world in terms of objects and their properties and relations to other objects.

- Leslie Kaelbling, 2001

9 Dimensional Time-Series
## Method Comparison

<table>
<thead>
<tr>
<th>Reinforcement Learning</th>
<th>Policy</th>
<th>Model</th>
<th>Horizon</th>
<th>Problem Space</th>
<th>Algorithm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ball in Cup</td>
<td>DMP</td>
<td>None</td>
<td>Short</td>
<td>Robot-Space</td>
<td>EM w/ Natural Gradient</td>
<td>Autonomous</td>
</tr>
<tr>
<td>Humanoid Walking</td>
<td>RBF</td>
<td>None</td>
<td>Short</td>
<td>Robot-Space</td>
<td>LSTD w/ Natural Gradient</td>
<td>Autonomous</td>
</tr>
<tr>
<td>Helicopter</td>
<td>Neural-Net</td>
<td>Locally Weighted Regression</td>
<td>Short</td>
<td>Robot-Space</td>
<td>Hill-Climbing w/ Monte-Carlo Eval.</td>
<td>LfD</td>
</tr>
<tr>
<td>PILCO Cart-Pole</td>
<td>RBF</td>
<td>Gaussian Process</td>
<td>Short</td>
<td>Robot-Space</td>
<td>CG/L-BFGS</td>
<td>Autonomous</td>
</tr>
<tr>
<td>PILCO Block-Stacking</td>
<td>Linear</td>
<td>Gaussian Process</td>
<td>Short</td>
<td>Mixed</td>
<td>CG/L-BFGS</td>
<td>Autonomous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robotics</th>
<th>Policy</th>
<th>Model</th>
<th>Horizon</th>
<th>Problem Space</th>
<th>Algorithm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cart Pushing</td>
<td>Jacobian PD-Control, Primitives</td>
<td>Geometric (Projected 2D)</td>
<td>Long</td>
<td>Task-Space</td>
<td>A* Variant</td>
<td>N/A</td>
</tr>
<tr>
<td>PR2 Towels</td>
<td>Scripted Grasp and Manipulation Prim.</td>
<td>Implicit Cloth Physics</td>
<td>Long</td>
<td>Task-Space</td>
<td>State Machine</td>
<td>N/A</td>
</tr>
<tr>
<td>PR2 Socks</td>
<td>Scripted Grasp and Manipulation Prim.</td>
<td>Implicit Cloth Physics</td>
<td>Long</td>
<td>Task-Space</td>
<td>State Machine</td>
<td>N/A</td>
</tr>
<tr>
<td>HRP-2 NAMO</td>
<td>ZMP Walking, Jacobian PD Grasp</td>
<td>2D Rigid Body Physics</td>
<td>Long</td>
<td>Task-Space</td>
<td>A* Variant</td>
<td>N/A</td>
</tr>
<tr>
<td>PR2 NAMO</td>
<td>RRT Path Planner, PD Control</td>
<td>2/3D Geometric Simulation</td>
<td>Long</td>
<td>Mixed</td>
<td>Hierarchical Backchaining</td>
<td>N/A</td>
</tr>
<tr>
<td>HRP-2 MacGyver</td>
<td>ZMP Walking, Scripted Primitives</td>
<td>3D Physical Simulation</td>
<td>Long</td>
<td>Mixed</td>
<td>A* Variant</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Physics-based Reinforcement Learning is a feasible and efficient method for autonomous robot manipulation, and enables adaptive behavior in natural environments.

Thesis statement
Thesis intuition

Reinforcement Learning

Robotics

Representational Gap
1. Whole-Body Manipulation
   ➡ Utilizes full robot capabilities
2. Multi-Object Planning
   ➡ Challenging domestic robotics tasks
3. Stochastic Planning
   ➡ Handles model uncertainty
4. Model Learning
   ➡ Adapt to novel environments
5. Dynamic Constraints
   ➡ Capture rigid body behavior
Outline

1. Physics-based Reinforcement Learning

2. Navigation Among Movable Objects with Uncertain Dynamics

3. Online Learning for Navigation Among Movable Objects
Outline

1. Physics-based Reinforcement Learning
2. Navigation Among Movable Objects with Uncertain Dynamics
3. Online Learning for Navigation Among Movable Objects
Efficiency in Model-Based RL

Models let you simulate experience

Online Learning Curves

Faster model learning = faster policy learning
Object-Oriented MDPs

**State:**
\[ s = \bigcup_{i=1}^{n} o_i.\text{attributes} \]

**Condition:**
\[ c = \bigcup_{k=1}^{R} r_k(o_i, o_j) \quad \forall i \neq j \]

Attributes
- \( x \)
- \( y \)
- \( \text{inTaxi} \)

Classes
- Person
- Taxi
- Wall
- Destination

Relations
- \( \text{in(person, taxi)} \)
- \( \text{contact}_N(o_1, o_2) \)
- \( \text{contact}_S(o_1, o_2) \)
- \( \text{contact}_E(o_1, o_2) \)
- \( \text{contact}_W(o_1, o_2) \)

---

**Key Idea:**
- \( S_1 \neq S_2 \neq S_3 \)
- \( C_1 \neq C_2 = C_3 \)
OO-MDP limitations

Discrete state-space

Deterministic boolean effects

No plant dynamics

Diuk et al. (ICML 2007)
Object-Oriented Regression

1) Define oriented collision predicates

\[
\begin{align*}
\text{contact}_{0=0}(o_i) & \quad \text{True} \\
\text{contact}_{0=20}(o_i) & \quad \text{True} \\
\text{contact}_{0=180}(o_i) & \quad \text{False}
\end{align*}
\]

Applied to entire scene

C=[001100…01010]

(240 dim, n=12)

2) Use regression model for each condition

\[
s' = \begin{bmatrix}
\dot{x}^{t+1} \\
\ldots \\
\dot{\theta}^{t+1} \\
\dot{\gamma}^{t+1}
\end{bmatrix} = \begin{bmatrix}
f(x_1, \ldots, f_x, \ldots; \beta_{x_1}) + \epsilon \\
\ldots \\
f(x_1, \ldots, f_x, \ldots; \beta_{\theta_{n}}) + \epsilon
\end{bmatrix}
\]

Need 6n+4 of these per condition

\[
f(\mathbb{R}^{6n+4}) \rightarrow \mathbb{R}^1
\]

used locally-weighted regression

\[
\beta_i^* = ((\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W y_i)^T \\
s_i' = \tilde{X}^T \beta_i^*
\]
OO-LWR limitations

Must *discretize* the collision space

\[ \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12} \]

\[ |C| = n^{2^p} \]

C: number of conditions
n: number of objects
P: number of collision sectors

Limits data per condition

Need exponential number of predicates!

\[
\begin{bmatrix}
  s_1 & a_1 & s'_1 \\
  s_2 & a_2 & s'_2 \\
  \vdots & \vdots & \vdots \\
  s_T & a_T & s'_T \\
\end{bmatrix}
\]

\[
\begin{array}{c|c|c|c|c|c}
  c_1 & c_2 & c_3 & c_4 & \cdots \\
\end{array}
\]
Dynamic physical inferences
Static physical inferences

Will it fall?  Which color is heavier?
Physical Understanding in Humans

Battaglia et al. 2013
The Intuitive Physicist

- Physics-engine with random variables
- Works by *sampling* model beliefs and *simulating* to generate predictions

Battaglia et al. 2013
Learning Kinematic and Dynamic Constraints

<table>
<thead>
<tr>
<th>Learning Kinematic and Dynamic Constraints</th>
<th>Whole-Body Manipulation</th>
<th>Multi-Object Planning</th>
<th>Stochastic Planning</th>
<th>Model Learning</th>
<th>Dynamic Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

• Domain: Planar object manipulation (arbitrary)

• Objective: Learn compact model

• Learning approach:
  • Stochastic physics engine model representation
  • MCMC estimator for unknown physical parameters
A Physics–Based approach

**Model-Based RL Loop**

- **Idea:** Use physics-engine as model representation
- **Technical contribution:** Formalize model learning problem and develop inference method
Physics-Based Reinforcement Learning

Physics API is Learning Target

Place prior on API parameters

Estimate posterior from data

Physics API is

Learning Target

Place prior on API

parameters

Estimate posterior

from data

\[ f(s, a) \]

\[ f(s, a; \theta) \sim P(s_0 | s, a; \theta) \]

\[ L(h) = P(s_0 | s, a, \theta) \frac{P(h | \Phi) P(\Phi)}{P(h | \Phi) P(\Phi)} \]

\[ P(\Phi | h) \propto P(h | \Phi) P(\Phi) \]

\[ \Phi = \text{body} \{ \text{mass, inertia, joint.wheel, joint.hinge} \} \]
Model Parameters

Three classes of parameters

- Rigid-body parameters
- Anisotropic friction constraints
- Distance constraints
Rigid body parameters

\[(m, r, \mu_c)\]

**Mass**
- For computing *accelerations*
- We assume uniform density

**Restitution**
- For computing *perpendicular* contact forces

**Friction**
- For computing *tangential* contact forces
  - Proportional to normal-force (Coulomb friction)
Anisotropic friction

\[(w_x, w_y, w_\theta, \mu_x, \mu_y)\]

**Pose parameters**
For placing constraint on a body

**Friction Coefficients**
The orthogonal friction components

\[(-0.3, 0, 0, 0.1, 0.8)\]

Object-frame

---

\(q(t) = \ln(P(h|t)P(t)) (7)\)

\[\sum_{t=1}^{n} \Pi(s_0 t f(s, a; \tilde{a}))(8)\]

\[P_{accept}(t|t_1) = \min(\mu_1, q(t))\]

Random \(\pi(s)\)

rigid \((m, r, \mu_c)\)

wheel \((w_x, w_y, w_\theta, \mu_x, \mu_y)\)

distance \((i_a, i_b, a_x, a_y, b_x, b_y)\)

All params \(i := (m, r, \mu_c, w_x, w_y, w_\theta, \mu_x, \mu_y, i_a, i_b, a_x, a_y, b_x, b_y)\)
Distance constraint

\[(i_a, i_b, a_x, a_y, b_x, b_y)\]

**Body indices**
Indicates two bodies to anchor the constraint

**Position A**
Anchor offset on the first body

**Position B**
Anchor offset on the second body

---

Inverted pendulum

- Body indices: \(i_a, i_b\)
- Position A: \(a_x, a_y\)
- Position B: \(b_x, b_y\)
- Anchor offset: \(l\)

All parameters: 
\[i := (m, r, \mu_c, w_x, w_y, \theta, \mu_x, \mu_y, i_a, i_b, a_x, a_y, b_x, b_y)\]
Overall model space

14 physical parameters per object

\[ \phi_i := (m, r, \mu_c, w_x, w_y, w_\theta, \mu_x, \mu_y, i_a, i_b, a_x, a_y, b_x, b_y) \]

Scenes have many objects
Gathering data

Touch or Grasp Object → Apply forces → Track object → Compute equivalent force in body-frame

Data = State Trajectory & Applied forces

Locked Caster

Unlocked Caster
Learning Parameters

Bayesian Inference: \( \Phi \) is a generative model of \( h \)

\[
P(\Phi|h) \propto P(h|\Phi)P(\Phi)
\]

Posterior: will reflect robot’s updated beliefs after observing \( h \)

Likelihood: should prefer accurate predictions

Prior: should support only legal values
Learning Parameters

L² penalty on sample trajectories

\[
\ln P(h|\Phi, \sigma) = -\sum_{t=1}^{n} \left( (s'_t - f(s_t, a_t; \tilde{\Phi}))^2 \right)
\]

Posterior samples generated by MCMC

\[
P_{accept}(\Phi_t|\Phi_{t-1}) = \min \left( 1, \frac{q(\Phi_t)}{q(\Phi_{t-1})} \right)
\]

\[
q(\Phi) = \ln (P(h|\Phi)P(\Phi))
\]

\{w_x, w_y, 0, 0.1, 0.8\}

Error:

Action

Hypothesis
Online Performance

Shopping Cart MDP

Apartment MDP

Reward

Step

ICML 2014
Learning on Steel Cart

Prior Prediction

Updated Beliefs

Gathers Data

ICRA 2014
Contributions

- Successfully applied to real robot
- Compact physics-based model
- Formally learning from manipulation data

- Allows autonomous learning on arbitrary objects
- Very data-efficient vs. OO-LWR
- One-shot learning on tables and cart
Limitations

Not demonstrated in real online task

Up Next!
Outline

1. Physics-based Reinforcement Learning
2. Navigation Among Movable Objects with Uncertain Dynamics
3. Online Learning for Navigation Among Movable Objects
Navigation Among Movable Objects with Uncertain Dynamics

Objective:

• Create collision-free path to goal

Challenge:

• Which objects are relevant?
A Well-Studied Problem


...
Curse of Dimensionality

- State space: configuration of robot and every object

- Assuming resolution $n$:

  $C_r = \{x, y, \theta\} \rightarrow |C_r| = O(n^3)$

  $C_{o_i} = \{x, y, \theta\} \rightarrow |C_{o_i}| = O(n^3)$

  $\implies C = C_r \times \prod_{i=0}^{k} C_{o_i} = O(n^{3(k+1)})$

- Search is exponential in the number of objects
Model Uncertainty

- Low-level uncertainty: how does this object move?

- High-level uncertainty: can I use this object?
NAMO MDP

- Natural state abstraction: *free-space regions*
- Corresponding action abstraction: *clear opening to neighboring free-space*
## Solving a NAMO-MDP

### Dynamics

#### Physics

\[ P(f_{t+1} = j | f_t = i) = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,j} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ f_{i,1} & f_{i,2} & \cdots & f_{i,j} \end{bmatrix} \]

### Planning

#### Monte-Carlo Tree Search

#### Value Iteration

\[ V^*_t(s) \leftarrow \max_a \left[ R(s, a) + \gamma \sum_{s'} P(s'|s, a)V^*_t(s') \right] \]
Scalability

- Algorithm typically linear in $|O|$.
- Adjacency graph directly controls the complexity of Value-Iteration.

![Graph showing scalability](image)

- Only 8 States!
Highlighted Behavior

- Robot adapts to stationary object:
  - Initially selects table
  - Updates model and switches to couch

- Limitations:
  - Manipulation planning in joint-space (KD-RRT)
  - Model “learning”: thresholded movable flag
Contributions

Value-based Planning

Combines reward and model uncertainty

Compact high-level dynamics

Solves object relevance problem
Limitations

Stationary Belief Distributions

No online *model* learning

Implemented in Simulation

 Doesn’t use realistic control stack
Outline

1. Physics-based Reinforcement Learning
2. Navigation Among Movable Objects with Uncertain Dynamics
3. Online Learning for Navigation Among Movable Objects
Platform

GoToFreeSpace($O_i,F_j$)
- GetGraspPoint
- NavigateToPoint
- Grasp
- ClearObstacle

(v$_{obj}$, k)

Body Controller

(v$_l$, v$_r$)

Base Controller

(p$_{base}$) x k

Manipulation Controller

(p$_{obj}$) x k

(v$_m$) x 16

(p$_{obj}$)

(p$_{base}$)

(v$_{obj}$, k)

Wheels

IMU

Vision

FT

Modules

System Control

Motor Control

Sensorimotor

Sensory

Motor
Whole-Body Manipulation

Free Parameter: $v_o$ (3-dim)

Manipulation Controller

- $v_o$: Desired Object Velocity
- $v_b$: Robot Body Trajectory
- $v_m$: Manipulation Controller
Grasp Sampling

Free Parameter: $g_\theta$ (1-dim)

- Sample grasp angle:
  
  $g_\theta \sim P(\theta | \Phi)$

- Compute projected boundary point

- Reject if not reachable
## Manipulation Policies

\[ P(\pi | \Phi) \] Planner operates over \( v_o \) and \( g_\theta \)

<table>
<thead>
<tr>
<th></th>
<th>Linear Velocity ((v_x, v_y))</th>
<th>Angular Velocity ((v_\theta))</th>
<th>Grasp Points ((g_\theta))</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static</strong></td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Unconstrained</strong></td>
<td>( \pm v_x ) and ( \pm v_y )</td>
<td>( \pm v_\theta ) at object center</td>
<td>Uniform</td>
<td></td>
</tr>
<tr>
<td><strong>Anisotropic</strong></td>
<td>( \pm v_u ) along unconstrained axis ((u = \arg\min_{x,y}(\mu_x, \mu_y)).)</td>
<td>( \pm v_\theta ) at constraint anchor</td>
<td>Faces perpendicular to unconstrained axis.</td>
<td></td>
</tr>
<tr>
<td><strong>Fixed-Point</strong></td>
<td>None</td>
<td>( \pm v_\theta ) at constraint anchor</td>
<td>Top and bottom faces opposite to constraint</td>
<td></td>
</tr>
</tbody>
</table>
NAMO with Static Constraint

Task: Navigate to Goal
NAMO with Non-Static Constraint
Learning Via Contact
Summary and Contributions

Physics Engine + Reinforcement Learning + $\epsilon$ = Adaptive Manipulation

1. Physics-Based Reinforcement Learning
   - Data-efficient model learning for multi-body dynamics

2. NAMO-MDP
   - Fast decision-theoretic solution for NAMO with uncertain dynamics

3. Online Learning for NAMO
   - First NAMO planner which adapts to novel objects
   - First collision-learning result in mobile manipulation
Publications


Special thanks to my committee, and Mike Stilman