What Does Physics Bias: A Comparison of Model Priors for Robot Manipulation

Jonathan Scholz [*]
Department of Computer Science
Georgia Institute of Technology
Atlanta, GA 30332
jkscholz@gatech.edu

Martin Levihn [†]
Department of Computer Science
Georgia Institute of Technology
Atlanta, GA 30332
levihn@gatech.edu

Charles L. Isbell [‡]
Georgia Institute of Technology
Atlanta, GA 30332
isbell@cc.gatech.edu

Abstract

We explore robot object manipulation as a Bayesian model-based reinforcement learning problem under a collection of different model priors. Our main contribution is to highlight the limitations of classical non-parametric regression approaches in the context of online learning, and to introduce an alternative approach based on monolithic physical inference. The primary motivation for this line of research is to incorporate physical system identification into the RL model, where it can be integrated with modern approaches to Bayesian structure learning. Overall, our results support the idea that modern physical simulation tools provide a model space with an appropriate inductive bias for manipulation problems in natural environments.

Keywords: bayesian physics reinforcement learning robotics

Acknowledgements

We are deeply indebted to the Langfeld Fund for their support of RLDM2013. LaTeX templates adapted from NIPS.

[*] http://www.cc.gatech.edu/ jscholz6/

[†] http://www.cc.gatech.edu/ mlevihn3/homepage/Welcome.html

[‡] http://www.cc.gatech.edu/ isbell/

1 Introduction

One of the most significant engineering hurdles to overcome when designing manipulation controllers for robots is getting the dynamics right. In principle, model-based Reinforcement Learning (RL) offers a solution method: attempt to learn a model from data. The learning literature offers a wide collection of probabilistic models that are both analytically tractable and domain-general. Unfortunately, domain generality typically comes at the cost of useful structure, resulting in high sample complexity. The physical world, however, is highly structured – it is governed by a set of well-known mathematical laws. In fact, the behavior of physical systems has been studied extensively by physicists and engineers for centuries, and has been distilled into a set of useful computational tools over the past several decades. Using modern software it is becoming possible to simulate a wide array of natural phenomena, including rigid and articulated bodies, fabric, and even fluids. Modern simulators thus offer a large, but structured, hypothesis-space for physical dynamics. This paper argues that by leveraging modern tools it is possible to parsimoniously model just the set of nonlinear transition functions that occur in physical domains. This drastically reduces the sample complexity compared to domain-general approaches which attempt to permit *all* possible nonlinear functions.

While physical system identification is not a new idea, our aim is to combine physics inference and Bayesian RL, where it can be integrated with the burgeoning literature on structure learning with hierarchical models. To this end we present the Bayesian Object-Oriented Markov Decision Process (BOOMDP), an MDP which uses a Bayesian-physics engine as the core transition model. We show that in the context of reinforcement learning, the performance and sample complexity of this model vastly outperforms regression-based alternatives.

2 Overview

The BOOMDP represents transition dynamics in terms of the agent's beliefs about the parameters of a physical model of the world. Like a standard Bayesian regression model, this model includes uncertainty both in the process input parameters (physical parameters) and in output noise. If F_Φ denotes a deterministic simulation function parameterized by physical parameters Φ, then the abstract model can be written as:

$$s_{t+1} = F_\Phi(s_t, a_t) + \epsilon \tag{1}$$

where $\Phi = (\phi_i)_{i=1}^N$ denotes a full assignment to the relevant physical parameters, such as masses and friction coefficients, for all objects in the world, and ϵ is zero-mean Gaussian noise.

The basic approach to computing transitions is assigning object state variables and evaluating a transition rule. We can obtain transition samples by first sampling the physical parameters, stepping the physics world for the appropriate action, and finally sampling the output noise. Overall, if $P(\Phi, \sigma|h)$ represents the agent's current model beliefs given the history h, then the generative process for sampling transitions in the BOOMDP model is:

$$\begin{aligned} \Phi, \sigma &\sim P(\Phi, \sigma|h) \\ \epsilon &\sim N(0, \sigma^2) \\ s_{t+1} &= F_\Phi(s_t, a_t) + \epsilon \end{aligned}$$

2.1 Planning

The BOOMDP planning approach follows the typical structure of model-based RL algorithms. The agent uses sampled transition (s, a, s) to construct a model of the domain, and selects actions using this model with a stochastic planning algorithm. In typical settings for which the BOOMDP is applicable, the underlying planner must be able to handle high-dimensional, continuous state spaces, as well as the limitation of only having access to *samples* of the transition model. Discretized Sparse-sampling (SS)[1] provides a solution technique that is compatible with these restrictions. SS performs forward search using the MDP model to approximate Q-values for the agent's current state. Further, it places no additional constraints on the agent, and shares a basic structure with deterministic forward-search algorithms which have been successful for large-scale object oriented planning. [1]

2.2 Inference

Unlike pure rigid-body estimation, which is linear in the known parameters, there is no straightforward estimator for a parameter set including constraints. Instead we turn to MCMC. The basic idea is to use a *physics engine* as a generative

[1] Model-based policy search is a viable alternative, but would require specifying a policy representation that places additional constraints on the space of policies that can be represented. However, policy gradient methods can offer significant advantages in complexity, and have been very successful for apprenticeship learning in robotics.

model for inferring the latent physical parameters by simulation. We wish to obtain posterior samples of the full set of physical parameters for the BOOMDP engine F_Φ:

$$P(\Phi, \sigma | D) \propto P(D|\Phi, \sigma) P(\Phi) P(\sigma) \tag{2}$$

where $\Phi = \{\phi_1, \phi_2, \ldots, \phi_k\}$ is the collection of parameters for the k objects in the domain, and σ is a scalar.

For a particular assignment to Φ, we define a Gaussian likelihood over next states:

$$L(\Phi, \sigma | D) = \prod_{t=1}^{n} P(s'_t | \Phi, s_t, a_t, \sigma) = \prod_{t=1}^{n} N(F_\Phi(s_t, a_t), \sigma^2) \tag{3}$$

Eq. 3 says that likelihood scores for proposed model parameters Φ, σ are evaluated on a Gaussian centered on the predicted next state for a BOOMDP world parameterized by Φ. The prior $P(\Phi)$ can be used to encode any prior knowledge about the parameters. In general, efficient sampling in physics-space is non-trivial, but details are omitted to save space. However at a high level, the approach requires reversible-jump [2] to handle variable-sized model-representations, and a (possibly infinite) mixture model [3] to exploit modality and reduce redundant sampling.

3 Planar Manipulation Example

In two-dimensions, object state can be represented with six parameters $s = \{x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}\}$, with $\{x, y, \theta\}$ corresponding to 2D position and orientation, and $\{\dot{x}, \dot{y}, \dot{\theta}\}$ their derivatives. Actions in this context correspond to the forces and torques used to move objects around, and can be represented with three additional parameters $a = \{f_x, f_y, \tau\}$

Overall the dynamics model therefore has the following signature:

$$f(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, f_x, f_y, \tau, d) \quad \rightarrow \quad (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) \tag{4}$$

3.1 Model Parametrization

At a minimum, a BOODMP physics model must define a core set of rigid body parameters, and can include one or more constraints. Here we focus two types of constraints that arise frequently in mobile manipulation applications: distance joints, such as a door hinge, and anisotropic friction, which can be used to model wheels. Rigid-body dynamics are parameterized by a scalar density $\{d\}$ [2].

A distance joint is a position constraint between two bodies, and can be specified with a 6-vector $J_d = \{a, b, a_x, a_y, b_x, b_y\}$ which indicates the two target objects a and b, and a position offset in each body frame. Anisotropic friction is a velocity constraint that allows separate friction coefficients in the x and y directions, typically with one significantly larger than the other. An anisotropic friction joint is defined by the 5-vector $J_w = \{x, y, \theta, \mu_x, \mu_y\}$, corresponding to the joint pose in the body frame, and the two orthogonal friction coefficients.

In summary, our dynamics model for a single body is represented by a set Ω containing the density d and zero or more constraints $j_i \in \{J_d, J_w\}$.

$$\Omega := \{d\} \cup \{J_d\}^* \cup \{J_w\}^* \tag{5}$$

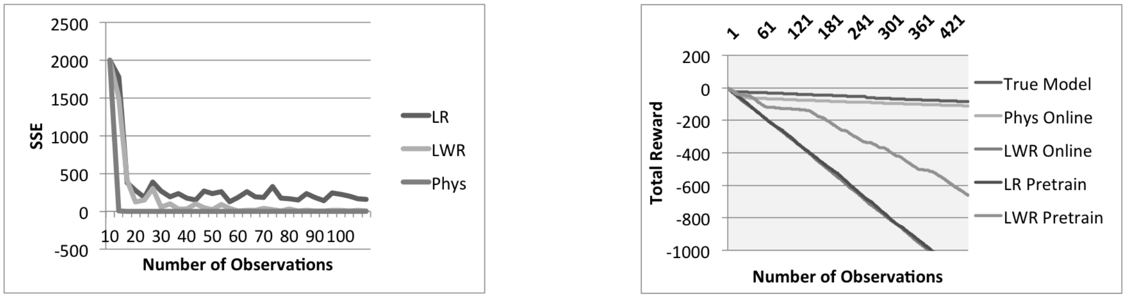
This representation is compact, but can describe a wide set of possible object behaviors. Inference over this representation can be achieved by conditioning on data of the form $s_i a_i s'_i = x_i, y_i, \theta_i, \dot{x}_i, \dot{y}_i, \dot{\theta}_i, f^x_i, f^y_i, \tau_i, x'_i, y'_i, \theta'_i, \dot{x}'_i, \dot{y}'_i, \dot{\theta}'_i$ and generating posterior samples with MCMC.

4 Experiments

We compare our physics-based model with two alternatives on either side in the bias-variance sense. Prediction accuracy is not the goal in itself, but rather the performance of a Reinforcement Learning agent equipped with each of these models for a range of manipulation tasks. The two alternative methods we investigate are based on fitting a bank of regression functions to the data matrix described in Eq. 3.1. The building block for these types of models are scalar-valued predictors of the form $f(\mathbb{R}^n) \rightarrow \mathbb{R}^1$.

The fundamental approach to building a regression model to match the signature in Eq. 4 is to stack $|s'| = 6$ independent regression models. Due to the underlying physical mechanics these output dimensions are inherently coupled, but this dependence is ignored by stacked models. Despite this shortcoming, stacked regression methods are frequently used in robotics. In the experiments below, we compare the our physics-based model with two alternatives on either side in the bias-variance sense: linear regression (LR), and Locally-Weighted Regression (LWR).

[2] Assumes uniform distribution of mass



(a) Model Error

(b) Agent Performance

Figure 1: Model quality and agent performance on Inverted Pendulum task.

4.1 Inverted Pendulum

Our first task is the standard inverted pendulum swing-up task, which presents nonlinear dynamics in a simple 4D state-space[3]. The agent is given a set of 2D forces evenly spaced around the unit circle which it can apply to the pendulum bob. Actions were chosen to result in an under-actuated system, such that the optimal policy is not greedy with respect to the reward function, and therefore requires look-ahead. Reward is defined according to the angular distance in radians of the pendulum-bob from vertical, and ranges from 0 to $-\pi$.

As illustrated in Fig. 1(a), the three models showed a predictable pattern of results for fitness over time: LR was over-biased and converged to non-zero error, LWR was under-biased (relative to Phys) and converged slowly, and Phys converged quickly.

Fig. 1(b) compares five pendulum agents equipped with variants of these three models. The first two agents are physics-based – one with access to the true model, and a second which implements our inference approach to attempt to learn it online. The third is an online agent equipped with an LWR model. Finally, we include two *Pretrain* agents which are given models initialized with 2000 *unbiased* samples from the set of reachable states. An optimal policy maintains 0 total reward, and deviations result in a negative slope.

As expected, the *True* and *LWR Pretrain* agents initially outperformed their online learning counterparts. However, within 50 samples the *Phys* agent fits an accurate model and begins to surpass *LWR Pretrain*. The poor performance of the *LWR Pretrain* agent in Fig. 1(b) can be explained by the inaccuracies in the agent's model which manifested as noise in during action selection, such that occasionally the pendulum would swing down, and it would take the agent several steps to recover.

Interestingly, the online LWR agent's policy failed to approach the quality of its pre-trained counterpart Fig. 1(b), even with well over 2000 samples. This due to these samples being biased by the agent's exploration history, which provides coverage only over the parts of the state space that have low reward. This result underscores a fundamental advantage of the generative physics approach, in that it allows the agent to make predictions about states it has not yet visited.

4.2 Shopping Cart

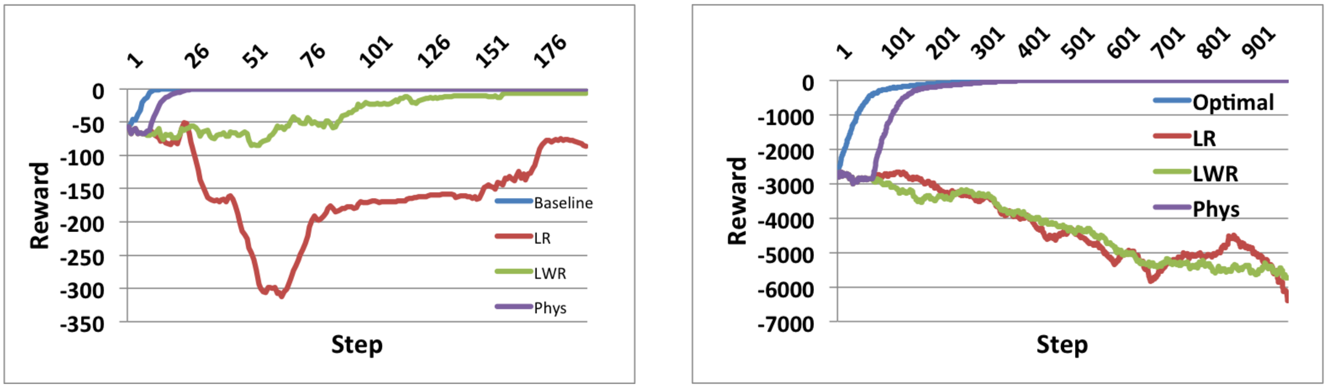
The next task involves pushing a shopping cart to a goal position, and is included to compare model performance on more complicated and practical systems. A cart is a rectangular rigid body with fixed-wheels at one end that resist lateral forces. The dynamics are again non-linear in the known forces.

As shown in Fig. 2(a), the *LWR Online* agent acquires samples from the relevant part of the state space and eventually reaches the terminal state. The *LR Online* agent also manages to push the cart in the general direction of the goal, but fails to achieve the zero-cost position and tends to oscillate about the goal position due to minor prediction errors. The *Phys* agent quickly learns the cart dynamics and achieves the best performing policy. Overall, these results suggest that LWR is a viable method for online learning of nonlinear dynamics, but that even for a single object it is significantly less efficient than estimating the cart's physical parameters.

4.3 The Living Room Task: Scalability to Multiple Bodies

The next experiment compares the three models in a task involving numerous bodies, and highlights the advantages of model generalization for large-scale problems. The task is a standard object placement task, implemented as a cost function on object position. The scene contains numerous household objects including tables, chairs, and a large piano,

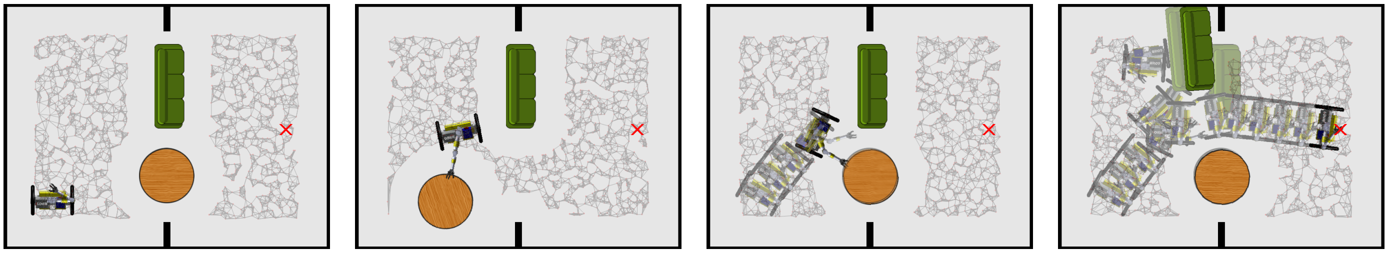
[3] angular components are unnecessary, and can result in singularities in estimating regression coefficients



(a) Low-dimensional task: Cart pushing

(b) High-dimensional task: Living Room Configuration

Figure 2: Agent performance on domains of varying size.



(a)

(b)

(c)

(d)

Figure 3: (a) Initial configuration (b) Expected outcome: table has lower mass and offers lowest-cost manipulation plan. (c) Actual behavior: table rotates in place. Robot updates beliefs to reflect high probability of revolute constraint. (d) Based on the new information the robot decides to move the couch.

which vary in mass as well as wheel parameters. The results from this experiment illustrate (a) that physical inference scales to large numbers of objects and confers significant advantages in online performance, and (b) that *both* regression approaches suffer from the task dimensionality and display poor online performance.

5 Situated Agents

In the broader context of robot manipulation, the previous tasks were somewhat artificial in that they defined actions as forces on objects directly, rather than modeling the agent explicitly. However, the practical use-case for the BOODMP model involves a robot interacting with bodies in real-time, updating its model beliefs using an object tracking system and a force-torque sensor, and planning in its full manipulator configuration space. We applied the BOOMDP model to a full implementation of a planner for a problem of this sort: Navigation Among Movable Obstacles (NAMO)[4, 5]. An example of a reasoning pattern supported by this sort of representation is depicted in Fig. 3. This result highlights the feasibility and advantages of physics-based belief representations for challenging manipulation problems in robotics.

References

[1] M. Kearns, Y. Mansour, and A. Y. Ng, “A sparse sampling algorithm for near-optimal planning in large markov decision processes,” in *International Joint Conference on Artificial Intelligence*, vol. 16, pp. 1324–1331, LAWRENCE ERLBAUM ASSOCIATES LTD, 1999.

[2] P. J. Green, “Reversible jump markov chain monte carlo computation and bayesian model determination,” *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[3] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[4] M. Stilman, K. Nishiwaki, S. Kagami, and J. Kuffner, “Planning and Executing Navigation Among Movable Obstacles,” in *IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (IROS 06)*, pp. 820 – 826, October 2006.

[5] M. Stilman and J. J. Kuffner, “Navigation among movable obstacles: Real-time reasoning in complex environments,” in *Journal of Humanoid Robotics*, pp. 322–341, 2004.