

---

# Reducing Model Bias in Reinforcement Learning

---

**Marc Peter Deisenroth**  
Department of Computer Science & Engineering  
University of Washington

**Carl Edward Rasmussen**  
Department of Engineering  
University of Cambridge

## Abstract

Model bias is one of the main reasons why reinforcement learning (RL) algorithms often need so many trials to successfully learn a task. Model bias has been known for decades, but no general solution to this problem has yet been proposed. We shed some light on the challenges of learning models from data and propose learning probabilistic models to reduce model bias by faithfully incorporating the model’s uncertainty into planning and policy learning.

Consider the problem of autonomously learning control tasks from scratch, i.e., without using specific prior knowledge such as a known dynamics model and/or demonstrations by a “teacher”. Solving these scenarios using model-free RL, often requires many interaction with the system, which can be impractical or infeasible in robotic systems, for instance. Model-based RL uses information from trials (which are used to build a model of the system) more efficiently than model-free learning; the policy, however, is always optimized in light of the model, not the true environment (*model bias*). Researchers often shy away from model-based methods since they rely on an “accurate” dynamics model, which is difficult to obtain. A key challenge in model-based RL is therefore to determine a faithful representation of the surrounding world, in particular, when we have only few interactions with the system

Typically, to learn a dynamics model, a deterministic function is fitted using least squares/maximum likelihood/evidence maximization, see Fig. 1, left. The figure also illustrates the problem with deterministic functions: During predictions, they claim full confidence everywhere—independent of whether states have been encountered before or not. Using a deterministic function for policy evaluation/planning leads to arbitrary results, but not to learning a successful policy in the absence of expert knowledge. This is an example of the effects of model bias and why model-based RL from scratch can be “daunting” [8].

To reduce model bias, we require a probabilistic model to faithfully describe the uncertainty of the model in regions of the state space that have not been encountered before, see Fig. 1, right. Moreover, we require this *model uncertainty* to be taken into account during policy evaluation. We implement the probabilistic model using a Gaussian process (GP), which also allows for closed-form approximate inference for propagating uncertainty over longer horizons. Using the GP, model uncertainty is explicitly incorporated into long-term planning.

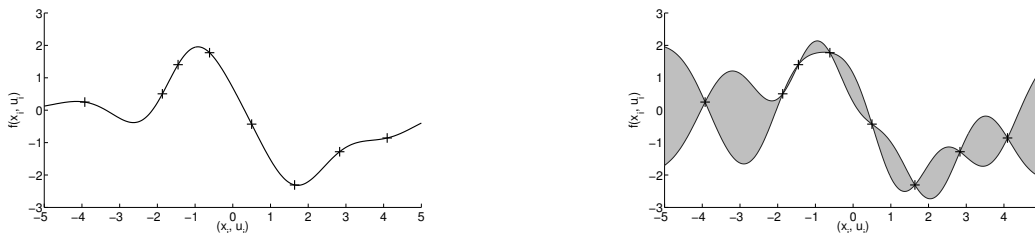


Figure 1: Deterministic model (left) and probabilistic model (right) with training targets (crosses).

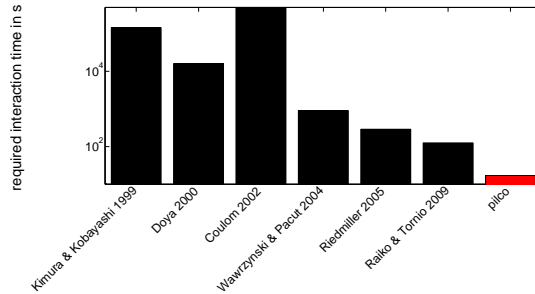


Figure 2: Learning efficiency when learning without expert knowledge (cart-pole problem).

GPs have been used before to learn models of robotic systems, see [5, 4], for instance. However, in these methods either expert knowledge was given in form of a teacher, by demonstrations, or a vague dynamics model, or the GP models were not used for long-term planning/policy evaluation.

**Results** We considered learning a controller for swinging up and balancing a pendulum attached to a cart (cart-pole problem), a common RL benchmark problem. A typical experiment consisted of the following steps: 1) collect initial training data by applying random controls, 2) train dynamics model, 3) learn policy for given model (policy search), 4) apply policy to real system, collect more data. We then repeated steps 2)–4) ten times. We first trained a radial basis function network. For each data point in the training set, we used a basis function centered at this data point. The shared widths of the basis functions were trained using evidence maximization (avoid overfitting). Although the state-control space was only five dimensional, this deterministic model never led to a successful controller. This unsuccessful learning was due to the fact that when extrapolating away from the training data, the model’s confidence in its predictions made the target state unreachable.

We then trained a probabilistic GP model using evidence maximization. During policy evaluation and policy learning, we explicitly took model uncertainty into account. Our method, which we call PILCO (probabilistic inference and learning for control), reliably learned the task in only a few trials. The only difference from the previous deterministic model is the incorporation of model uncertainty into planning and decision making: When PILCO’s predictions left the current training set, the predictive variance blew up—many things were plausible including reaching the target state. When applying the learned policy to the system, either these regions were encountered, which confirmed the predictions, or they were not encountered, which reduced the probability of reaching the target state. In the subsequent policy evaluation, these trajectories became unfavorable. Fig. 2 summarizes PILCO’s success in terms of learning efficiency in the context of state-of-the-art RL methods.

## References

- [1] R. Coulom. *Reinforcement Learning Using Neural Networks, with Applications to Motor Control*. PhD thesis, Institut National Polytechnique de Grenoble, 2002.
- [2] K. Doya. Reinforcement Learning in Continuous Time and Space. *Neural Computation*, 12(1):219–245, 2000.
- [3] H. Kimura and S. Kobayashi. Efficient Non-Linear Control by Combining Q-learning with Local Linear Controllers. In *ICML 1999*, pp. 210–219.
- [4] J. Ko, D. J. Klein, D. Fox, and D. Haehnel. Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp. In *ICRA 2007*, pp. 742–747.
- [5] D. Nguyen-Tuong, M. Seeger, and J. Peters. Local Gaussian Process Regression for Real Time Online Model Learning. In *NIPS 2009*, pp. 1193–1200.
- [6] T. Raiko and M. Tornio. Variational Bayesian Learning of Nonlinear Hidden State-Space Models for Model Predictive Control. *Neurocomputing*, 72(16–18):3702–3712, 2009.
- [7] M. Riedmiller. Neural Fitted  $Q$  Iteration—First Experiences with a Data Efficient Neural Reinforcement Learning Method. In *ECML 2005*.
- [8] S. Schaal. Learning From Demonstration. In *NIPS 1997*, pp. 1040–1046.
- [9] P. Wawrzynski and A. Pacut. Model-free off-policy Reinforcement Learning in Continuous Environment. In *IJCNN 2004*, pp. 1091–1096.