Curious iLQR: Resolving Uncertainty in Model-based RL

Sarah Bechtle [1] **Akshara Rai** [2] **Yixin Lin** [2] **Ludovic Righetti** [1 3] **Franziska Meier** [2]

Abstract

Curiosity as a means to explore during reinforcement learning problems has recently become very popular. However, very little progress has been made in utilizing curiosity for learning control. In this work, we propose a model-based reinforcement learning (MBRL) framework that combines Bayesian modeling of the system dynamics with *curious iLQR*, a risk-seeking iterative LQR approach. During trajectory optimization the *curious iLQR* attempts to minimize both the task-dependent cost and the uncertainty in the dynamics model. We scale this approach to perform reaching tasks on 7-DoF manipulators, to perform both simulation and real robot reaching experiments. Our experiments consistently show that MBRL with *curious iLQR* more easily overcomes bad initial dynamics models and reaches desired joint configurations more reliably and with less system rollouts.

1. Introduction

Curiosity has repeatedly been recognized as a fundamental building block of human behaviour (Loewenstein, 1994). Some researchers go as far as considering curiosity essential for the development of autonomous behaviour in humans (White, 1959). The concept of intrinsically motivated, curious, behavior has also been explored within the reinforcement learning literature from various angles. For example, a first attempt towards intrinsically motivated agents consisted in rewarding agents to minimize prediction errors of sensory events (Barto, 2004; Singh et al., 2004; 2010). This initial work on curiosity-driven agents was designed for low-dimensional and discrete state-and-action spaces. Recently, curiosity as a means to better explore was also investigated for high-dimensional continuous state spaces (Bellemare et al., 2016; Pathak et al., 2017). Most of this work, including recent efforts towards curiosity driven robot learning (Laversanne-Finot et al., 2018; Tanneberg et al., 2019), has defined curiosity as a function of model prediction error. In this work, we take inspiration from Kagan (Kagan, 1972), who define curiosity as motivation to resolve uncertainty in the environment. Following this definition, our goal is to develop a model-based reinforcement learning (MBRL) algorithm that is aware of its model uncertainty, and optimizes action sequences to not only maximize a reward but also to reduce model uncertainty. MBRL comes with a lot of promise for sample-efficient learning on real robots (Atkeson & Santamaria, 1997). However, one of the key challenges centers around making use of a bad model to generate useful data to improve the model and eventually achieve the task. We believe, that curiosity can help with this challenge. Our MBRL algorithm uses a Gaussian process (Williams & Rasmussen, 2006) to represent the dynamics and a *curious* version of the iterative Linear Quadratic Regulator (iLQR)(Tassa et al., 2014) for trajectory optimization, as summarized in Figure 1. In a nutshell, our curious iLQR aims at optimizing trajectories that minimize the cost as well as the uncertainty in the dynamics model. The contributions of this work are as follows: We combine curious iLQR with Gaussian process regression into a model-based RL loop that learns a model of the system dynamics from scratch, while trying to achieve a task. We demonstrate that such optimal control algorithms can be scaled to seven Degree of Freedom (DoF) manipulation platform and show that curiosity helps to achieve the task faster, and more reliably,

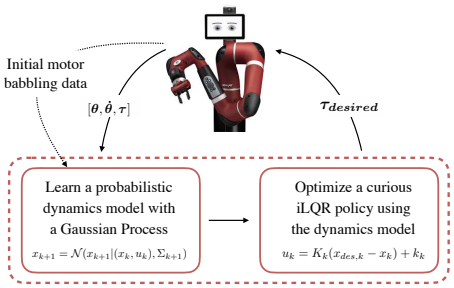

Figure 1. An overview of our approach for model-based reinforcement learning. We start with motor babbling data to initialize the dynamics model, followed by an iterative loop of learning the model and updating an iLQR policy.

[1]Max Planck Institute for Intelligent Systems [2]Facebook AI Research [3]New York University. Correspondence to: Sarah Bechtle <sbechtle@tuebingen.mpg.de>.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 36^{th} International Conference on Machine Learning, Long Beach, California, USA, 2019.

when starting from a bad model of the dynamics.

2. MBRL via Curious iLQR

In this section we present how we can incorporate curious behaviour into a robot's learning control loop and how it can affect it's capability of achieving a task. Rooted in the literature (Kagan, 1972), we believe that by seeking out uncertainties, a robot is able to solve a reinforcement learning task faster and therefore gain higher rewards sooner compared to a non curious robot. We design an approach that explores while optimizing a trajectory. The exploration should reduce uncertainty about the robots dynamics model, while staying close to the task trajectory. More concretely, having high uncertainty in the model means high variance in an optimized state sequence. In our work we want to seek out actions that reduce this variance.

2.1. Background: risk sensitive iLQR

To include stochastic processes when optimizing a trajectory, it is necessary to consider a nonlinear optimal control problem where the system dynamics are defined by the following stochastic differential equation

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \mathbf{f}\left(\mathbf{x_k}, \mathbf{u_k}\right)\Delta t + \mathbf{g}\left(\mathbf{x_k}, \mathbf{u_k}\right)\Delta\omega \quad (1)$$

where $\mathbf{f}$ represents the dynamics of the system and $\mathbf{g}$ the stochasticity of the problem. $\Delta\omega$ is a Brownian motion with zero mean and covariance $(\mathbf{\Sigma}\cdot\Delta t)$. Following the idea of (Jacobson, 1973) to include higher order momenta of the cost function, the objective function takes the form of an exponential transformation of the performance criteria J:

$$J = \min_{\pi} \mathbb{E}\left\{\exp[\sigma\mathcal{J}\left(\pi\right)]\right\} \quad (2)$$

where $\mathcal{J}(\pi)$ is the performance index, which is a random variable, and a functional of the policy π. $\sigma \in \mathbb{R}$ accounts for the sensitivity of the cost to higher order moments (variance, skewness, etc). Notably from (Farshidian & Buchli, 2015), the cost is

$$\frac{1}{\sigma}\log(J) = \mathbb{E}(\mathcal{J}^*) + \frac{\sigma}{2}\mathrm{var}(\mathcal{J}^*) + \frac{\sigma^2}{6}\mathrm{sk}(\mathcal{J}^*) + \cdots \quad (3)$$

where var and sk stand for variance and skewness and $\mathcal{J}^*$ is the optimal task cost.

2.1.1. Algorithm derivation

The algorithm begins with a nominal state and control input trajectory $\mathbf{x^n}$ and $\mathbf{u^n}$. The dynamics are linearized and the cost is quadratized along $\mathbf{u_k^n}$, $\mathbf{x_k^n}$ in terms of state and control deviations, together with the following quadratic approximation of the value function Ψ:

$$\Psi(\delta\mathbf{x_k}, \mathbf{k}) = \frac{1}{2}\delta\mathbf{x_k^T}\mathbf{S_k}\delta\mathbf{x} + \delta\mathbf{x_k^T}\mathbf{s_k} + s_k \quad (4)$$

and solving for the optimal control, we get $\delta\mathbf{u_k} = \mathbf{k_k} + \mathbf{K_k}\delta\mathbf{x_k}$, $\mathbf{k_k} = -\mathbf{H_k^{-1}}\mathbf{g_k}$ and $\mathbf{K_k} = -\mathbf{H_k^{-1}}\mathbf{G_k}$ where $\mathbf{H_k}$, $\mathbf{g_k}$, $\mathbf{G_k}$ are given by

$$\begin{aligned}\mathbf{H_k} &= \mathbf{R_k} + \mathbf{B_k^T}\mathbf{S_k}\mathbf{B_k} + \sigma\mathbf{B_k^T}\mathbf{S^T}\mathbf{C}\mathbf{\Sigma}\mathbf{C^T}\mathbf{S_k}\mathbf{B_k}\\ \mathbf{g_k} &= \mathbf{r_k} + \mathbf{B_k^T}\mathbf{s_k} + \sigma\mathbf{B_k^T}\mathbf{S_k^T}\mathbf{C}\mathbf{\Sigma}\mathbf{C^T}\mathbf{s_k}\\ \mathbf{G_k} &= \mathbf{P_k^T} + \mathbf{B_k^T}\mathbf{S_k}\mathbf{A_k} + \sigma\mathbf{B_k^T}\mathbf{S_k^T}\mathbf{C}\mathbf{\Sigma}\mathbf{C^T}\mathbf{S_k}\mathbf{A_k}\end{aligned} \quad (5)$$

the corresponding backward recursions are

$$\begin{aligned}\mathbf{s_k} = \mathbf{q_k} + \mathbf{A_k^T}\mathbf{s_{k+1}} + \mathbf{G_k^T}\mathbf{k_k} + \mathbf{K_k^T}\mathbf{H_k}\mathbf{k_k} +\\ \sigma\mathbf{A_k^T}\mathbf{S_{k+1}^T}\mathbf{C}\mathbf{\Sigma}\mathbf{C^T}\mathbf{s_{k+1}}\end{aligned} \quad (6)$$

$$\begin{aligned}\mathbf{S_k} = \mathbf{Q_k} + \mathbf{A_k^T}\mathbf{S_{k+1}}\mathbf{A_k} + \mathbf{K_k^T}\mathbf{H_k}\mathbf{K_k} + \mathbf{G_k^T}\mathbf{K_k} +\\ \mathbf{K_k^T}\mathbf{G_k} + \sigma\mathbf{A_k^T}\mathbf{S_{k+1}^T}\mathbf{C}\mathbf{\Sigma}\mathbf{C^T}\mathbf{S_{k+1}}\mathbf{A_k}\end{aligned} \quad (7)$$

With $\sigma = 0$ the recursions revert to the usual Ricatti recursions for iLQR (Tassa et al., 2014).

2.2. Curious iLQR via model uncertainty

We now describe the details of our instantiation of a curious iLQR approach. The optimizer presented in Section 2.1 works with the assumption that a known dynamics model is disturbed by a stochastic noise process. In our work, we do not assume this but learn and improve the dynamics model as we are trying to achieve a task. The quality of our current model affects the quality of the solution an optimizer can find. This can lead to MBRL loops getting stuck in 'local optima' or converging very slowly. Our hypothesis is that, by trying to explore uncertainties in the model, our MBRL loop can escape these local minima and find better solutions faster. On a high level our optimizer reasons about uncertainties in the probabilistic dynamics model to find solutions to a nonlinear optimal control problem. The algorithm alternates between improving the probabilistic dynamics of the robot model from the recently unrolled trajectory, optimizing the trajectory using the current model, and rolling out the current locally optimal policy on the system. See Figure 1 for an overview of the approach.

2.2.1. Learning a Bayesian dynamics model

Because of their intuitive uncertainty behavior, the Bayesian dynamics model is implemented as a Gaussian Process (GP). Let $\mathbf{x_k}$ denote the state of the system at time step k, where $\mathbf{x_k} = [\boldsymbol{\theta_k}, \dot{\boldsymbol{\theta}}_\mathbf{k}]$ and $\boldsymbol{\theta_k}$, $\dot{\boldsymbol{\theta}}_\mathbf{k}$ are joint positions and velocities respectively. Furthermore, let $\mathbf{u_k}$ denote commanded torques $\mathbf{u_k} = [\tau_1, \tau_2, ..., \tau_N]$ where $N = \text{DoF}$. Then tuples of states and actions $(\mathbf{x_k}, \mathbf{u_k}) \in \mathbb{R}^{F+A}$ are used as training inputs and $\ddot{\boldsymbol{\theta}}_\mathbf{k+1} \in \mathbb{R}^F$, joint accelerations at the next time step, as training outputs. Once trained, the GP delivers one step predictions of the form

$$p(\ddot{\boldsymbol{\theta}}_\mathbf{k+1}|\mathbf{x_t}, \mathbf{u_t}) = \mathcal{N}(\ddot{\boldsymbol{\theta}}_\mathbf{k+1}|\mathbf{f}\left(\mathbf{x_k}, \mathbf{u_k}\right)\Delta t, \mathbf{\Sigma_{k+1}}) \quad (8)$$

where $\mathbf{f}$ is the mean vector and $\mathbf{\Sigma_{k+1}}$ the covariance matrix of the predictive distribution evaluated at $(\mathbf{x_k}, \mathbf{u_k})$. The GP outputs the acceleration at the next time step $\ddot{\boldsymbol{\theta}}_{\mathbf{t+1}}$ which is numerically integrated to velocity $\ddot{\boldsymbol{\theta}}_{\mathbf{k+1}}\Delta t + \dot{\boldsymbol{\theta}}_{\mathbf{k}} = \dot{\boldsymbol{\theta}}_{\mathbf{k+1}}$ and position $\dot{\boldsymbol{\theta}}_{\mathbf{k+1}}\Delta t + \boldsymbol{\theta}_{\mathbf{k}} = \boldsymbol{\theta}_{\mathbf{k+1}}$.

2.2.2. Seeking out uncertainties

When referring back to equation (1), our system dynamics can be defined as

$$\mathbf{x_{k+1}} \sim \mathcal{N}(\mathbf{x_{k+1}}|\mathbf{h}\left(\mathbf{x_k}, \mathbf{u_k}\right), \mathbf{\Sigma}) \tag{9}$$

where $\mathbf{x_{k+1}}$ is a random variable, $\mathbf{h}$ is the mean prediction of the GP integrated to the future state $\mathbf{x_{k+1}}$. Σ is the covariance matrix of the posterior distribution of the GP predictions. When linearising our dynamics around a nominal trajectory

$$\delta\mathbf{x_{k+1}} = \mathbf{A_k}\delta\mathbf{x_k} + \mathbf{B_k}\delta\mathbf{u_k} + \mathbf{C_k}\omega_k \tag{10}$$

we have $\mathbf{A_k} = \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{x_k}}$, $\mathbf{B_k} = \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{u_k}}$ and $\omega_k \sim \mathcal{N}(\omega_k|0, \mathbf{\Sigma_{k+1}})$, where $\mathbf{A_k}$ and $\mathbf{B_k}$ are the analytical gradients of the GP prediction at each time step. $\mathbf{C_k}$ represents how the uncertainty propagates through the system. For our formulation, the covariance matrix is of particular interest, as it provides the information that is essential for our algorithm to explore. The uncertainty of the Gaussian process model when evaluating the posterior distribution at $(\mathbf{x_k}, \mathbf{u_k})$ is the notion of uncertainty that can be used in order to encourage the agent to choose actions such that the uncertainty of the model will be resolved in the future. It is important to note now that $\mathbf{\Sigma_{k+1}}$ in equations (5) and (6) represents the uncertainty of the model when predicting $\mathbf{x_{k+1}}$ from $(\mathbf{x_k}, \mathbf{u_k})$. As can be seen from equations (6) and (7), with a negative σ value, the cost explicitly favors for the uncertainty in the dynamics. As a result, the agent is encouraged to select actions that include some level of uncertainty while still trying to reduce the task specific error. With $\sigma = 0$ the agent will ignore any uncertainty in the environment and therefore not explore around the trajectory that should be optimized. In this case the optimizer ignores any higher order statistics of the cost function and only resolves the optimal control problem by minimizing the expectation over the cost.

3. Illustration: Curious iLQR

In this section, we show some toy examples that illustrate the advantages of using curiosity – the motivation to resolve uncertainty – as an exploration tool. The objectives of these toy experiments are twofold (1) they should provide an analysis of whether it is possible to escape local minima with curiosity, that a locally optimal optimizer, can potentially get stuck in (Tassa et al., 2014). And (2) if by trying to resolve the uncertainty in the model, the agent explores parts of the state space that are less known but contain relevant information for solving potential future tasks and is therefore able to transfer easier. In the following, and throughout the paper, we will refer to the agent that tries to resolve the uncertainty in its environment as curious and the one that is not following the uncertainty as normal.

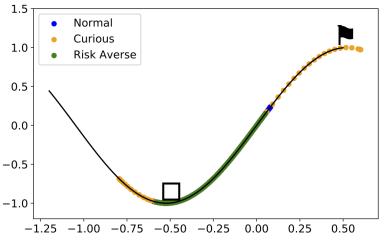

Figure 2. Explored states of the environment in the mountain car problem by risk-averse, normal and curious iLQR.

3.1. Mountain Car Example

In this experiment, we analyze the benefit of including model uncertainty during trajectory planning to find a globally optimal solution, instead of converging to a local optimal solution, when using a locally optimal feedback controller. The mountain car problem (Moore, 1990) consists of an under-actuated car that should be parked at the top of a steep hill when starting from the valley. The car cannot simply accelerate to drive up the hill and must gain momentum by driving up the opposite hill. This problem is representative for how an optimizer, that finds locally optimal solutions, is not able to compute a trajectory to actually solve the task. Locally the best action would always be to drive up the goal hill and not drive away from the goal, to leverage potential energy, and finally reach the desired goal position. We learn a probabilistic model of the system dynamics and a policy as presented in Section 2. Fig. 2 shows that the behaviour of driving away from the goal is only observed in the curious agent, and thus this agent is the only one receiving a reward by the end of the episode.

3.2. Learning model faster by following its own uncertainty

The second experimental platform is the OpenAI gym Reacher environment (Brockman et al., 2016), a two degrees of freedom arm attached at the center of the scene. The goal of the task is to reach a target placed in the environment. In the experiments presented below, actions were optimized as described in section 2. The intuition behind this experiment is that, if an agent is driven to resolve uncertainty in its model, by exploring less known states, a better model of the system dynamics can be learned and therefore used to optimize a control sequence more reliably. Our hypothesis is that the model learned by the curious agent is more certain by the end of learning. As it was

driven to resolve uncertainties in the environment we expect it to be more robust when solving the reaching task for new targets. In Figure 3 the end-effector position in euclidean space shows the behaviour of the 2D arm of the curious and normal agent. Each row depicts a different target position. The curious agent tries to resolve the uncertainty within the model. For the first target position the normal agent seemingly reaches the target after the second learning iteration, the curious agent only manages to reach the target during the third iteration. Interestingly, the exploration of the curious agent leverages the arm to reach the second target position immediately and continues to reach it consistently thereafter.

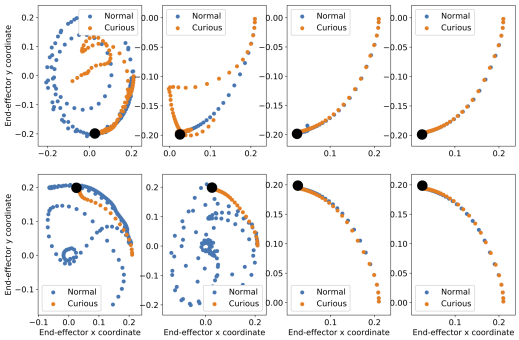

Figure 3. End-effector position of curious and not curious agent for 4 learning iterations on 2 different targets. The targets are represented by the black dots.

4. Experiments on high-dimensional problems

Lastly the goal of this work is to perform experiments on torque-controlled manipulator. Our first experimental platform is a 7 degrees of freedom Kuka iiwa7 simulation in the PyBullet physics simulator (Pybullet). The goal of these experiments is to reach a desired target joint configuration. The dynamics are learned with Gaussian process regression. We believe that curiosity (on average) helps the optimizer to find better solutions faster, because it helps escape local optima within the MBRL loop. Curiosity also means we explore more and because of that we learn more about the dynamics which helps when using the model for new targets. In simulation, we performed three different kinds of experiments that we will describe in detail below. Throughout all of the simulation experiments the trajectory was $0.625s$ long at a sampling rate of 240 Hz. Motor babbling was performed at the beginning for $0.5s$ by commanding sine-trajectories in each joint.

4.1. Reaching task from scratch

During the first set of experiments, we wanted to compare the performance of both controllers when learning to reach a given target configuration from scratch. For a given target joint configuration, 20 MBRL with the curious iLQR loops for 3 different target joint configurations were performed. For each target we run 10 trials, where the variability between trials is caused by the random initial torque trajectory, and by small perturbations to the starting configuration. Figure 4 visualizes the mean and variance of the end-effector trajectories across the 10 trials on the first target. We observe that MBRL with the curious iLQR reaches the target end-effector position, by the end of learning, more consistently then with normal iLQR. Most notably is the variance of the final trajectories when using MBRL with regular iLQR. This suggests, that at least for a few trials MBRL with normal iLQR failed to converge to the target within 20 learning iterations.

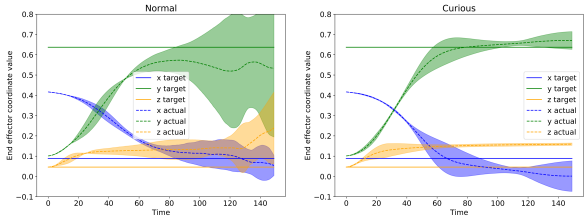

Figure 4. Final performance in end-effector space after learning to reach one target position using MBRL with (left) normal, (right) curious iLQR. Solid lines represent targets, dashed lines actual trajectories in x, y, z. The mean and the standard deviation are computed across 10 different trials.

4.2. Optimizing towards new targets after model learning

To confirm the hypothesis that the models learned by MBRL with curious iLQR generalize better, because they have explored the state space better, we decided to evaluate the learned dynamics models on a second set of experiments in which the robot tries to reach new, unseen targets. Thus, in this experiment with take the GP models learned during experiment 1 in Section 4.1 and use them to optimize trajectories to reach new targets that were not seen during training of the model. The results are shown in Figure 5, where three randomly chosen targets where set and the trajectory was optimized with regular iLQR. Note, that here we use regular iLQR to optimize for the trajectory so that we can better compare the models learned with/without curiosity in the previous set of experiments. The results are averaged across 10 trials. Here, the 10 trials correspond to using one of the 10 dynamics models at the end of Experiment 1 in Section 4.1. For each trial, the initial torque trajectory was initialized randomly, and the initial joint configuration slightly perturbed. The mean trajectories and standard deviation of the end-effector trajectories across the 10 trials are shown. We see that MBRL with curious iLQR results in a model that performs better when presented with a new target. The new targets are reached more reliably and pre-

cisely. Here, we chose the new target location to be close to the initial target (from Experiment 1 in Section 4.1) because we assume that the MBRL runs in the previous Section only learned reliable dynamics models locally.

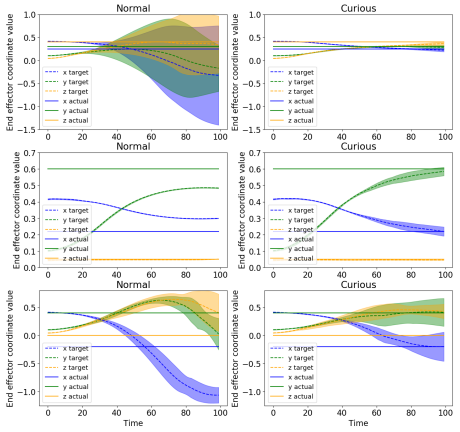

Figure 5. Optimizing to reach new targets with regular iLQR after dynamics models were learned. 3 different targets (one per row) are evaluated and the final end-effector trajectories presented. The mean and the standard deviation of the optimized trajectories are computed across the 10 models learned via MBRL with normal iLQR (left col) and MBRL with curious iLQR (right col).

5. Real hardware experiments

The experimental platform for our hardware experiments is the Sawyer Robot (Sawyer), again a 7 DoF manipulator. The purpose of the experiments was to demonstrate the applicability and the benefits of our algorithm on real hardware. We perform reaching experiments for 4 different target locations. Each experiment is started from scratch with no prior data. The results are summarized in Table 1 and show the number of learning iterations needed in order to reach the target together with the precision in end-effector space. If the target was reached with a precision of below

Target	**Learning Iterations**							**Distance to Target**							
	Curious				**Normal**			**Curious**				**Normal**			
1	6	2	3	**3.67**	8	3	8	**7.0**	0.05	0.09	0.09	**0.07**	0.37	0.08	0.18	**0.21**
2	3	4	4	**3.67**	8	3	5	**5.3**	0.05	0.09	0.09	**0.09**	0.20	0.08	0.09	**0.12**
3	6	4	3	**4.33**	8	8	8	**8.0**	0.09	0.09	0.09	**0.09**	0.17	0.16	0.11	**0.15**
4	3	2	2	**2.33**	3	3	3	**3.0**	0.04	0.07	0.07	**0.06**	0.04	0.08	0.05	**0.06**
				3.5				**5.9**				**0.07**				**0.14**

Table 1. Results on a reaching task on the Sawyer robot. The experiment for each target was repeated three times, the number of learning iterations and the final end-effector distance to the target is reported.

10 cm, we would consider the task as achieved. Running the experiment on hardware was a lengthy process, as the GP training and the rollout would happen iteratively and GP training time increases with growing amount of data. For this reason we decided to terminate our experiments after eight iterations and consider the last end-effector position. We repeated each experiment three times to demonstrate the repeatability of our method as we expected measurement noise to affect solutions. From the table we can see that MBRL with curious iLQR would reach a target on average after 3.5 iterations with a average precision of 7 cm compared to MBRL with regular iLQR that needed 5.9 iterations (often not ever reaching the target after eight iterations with the desired precision), with a precision of 14cm on average. As in simulation, similar to Experiment 4.2 we wanted to evaluate the quality of the learned models on new target positions. The results are summarized in Table 2 and are similar to what we observe in simulation, the models learned with curiosity, when used to optimize for new targets can achieve higher precision then when using the models learned without curiosity.

	Reaching Precision (m)	
Target	**Curious**	**Normal**
1	0.20	0.67
2	0.26	0.61
3	0.25	1.06
4	0.24	0.67
5	0.37	0.49
	0.26	**0.7**

Table 2. Results on optimizing to reach a new target not seen during training. The distance of the endeffector to the target is reported in meters.

6. Conclusion and future work

In this work, we presented a model-based reinforcement learning algorithm that uses an optimal control framework to trade-off between optimizing for a task specific cost, and exploring around a locally optimal trajectory. Our algorithm explicitly rewards actions that seek out uncertainties in our model, by incorporating them into the cost. By doing so, we are able to learn a model of the dynamics that achieves the task faster than MBRL with standard iLQR, and also transfers well to other tasks. We present experiments on Kuka iiwa7 arm in simulation, and a Sawyer robot on hardware. In both sets of experiments, our approach not only learns to achieve the specified task faster, but also generalizes to new tasks and initial conditions. All this points towards the conclusion that resolving dynamics uncertainty during model-based reinforcement learning is indeed a powerful tool. As (Loewenstein, 1994) states, curiosity is a superficial affection, it can arise, diverge and end promptly. We were able to observe similar behaviour as well in our experiments. Once the task was achieved, the robot would explore and deviate slightly from the task, and go back to exploiting once its fairly certain about the dynamics. In the future, we would like to explore this direction by considering how to maintain exploration strategies and potentially reuse them later, especially on new tasks. This could be helpful if the robot is still certain about a task, even though the environment or task has changed. Furthermore, we noticed that, with increasing amount of high dimensional data, the scalability of the Gaussian processes was reaching its limits. Therefore we

would want to investigate the potential of Bayesian neural networks in our MBRL framework in the future.

References

Atkeson, C. G. and Santamaria, J. C. A comparison of direct and model-based reinforcement learning. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pp. 3557–3564. IEEE, 1997.

Barto, A. G. Intrinsically motivated learning of hierarchical collections of skills. *International Conference on Developmental Learning and Epigenetic Robotic*, pp. 112–119, 2004. doi: 10.1.1.117.6436.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Farshidian, F. and Buchli, J. Risk sensitive, nonlinear optimal control: Iterative linear exponential-quadratic optimal control with gaussian noise. *arXiv preprint arXiv:1512.07173*, 2015.

Jacobson, D. H. Optimal Stochastic Linear Systems with Exponential Performance Criteria and Their Relation to Deterministic Differential Games. *IEEE Transaction on Automatic Control*, 18, 1973.

Kagan, J. Motives and development. *Journal of Personality and Social Psychology*, 22(1):51–66, 1972.

Laversanne-Finot, A., Péré, A., and Oudeyer, P.-Y. Curiosity driven exploration of learned disentangled goal spaces. *arXiv preprint arXiv:1807.01521*, 2018.

Loewenstein, G. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75, 1994.

Moore, A. Efficient memory-based learning for robot control. *PhD thesis, University of Cambridge*, 1990.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-Driven Exploration by Self-Supervised Prediction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July: 488–489, 2017. ISSN 21607516. doi: 10.1109/CVPRW.2017.70.

Pybullet. URL http://pybullet.org/.

Sawyer. URL https://www.rethinkrobotics.com/sawyer/.

Singh, S., Barto, A., and Chentanez, N. Intrinsically motivated reinforcement learning. *18th Annual Conference on Neural Information Processing Systems (NIPS)*, 2004. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2051031.

Singh, S., Lewis, R. L., Barto, A. G., and Sorg, J. Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2051031.

Tanneberg, D., Peters, J., and Rueckert, E. Intrinsic motivation and mental replay enable efficient online adaptation in stochastic recurrent networks. *Neural Networks*, 109: 67–80, 2019.

Tassa, Y., Mansard, N., and Todorov, E. Control-limited differential dynamic programming. *IEEE International Conference on Robotics and Automation, ICRA*, 2014.

White, R. W. Motivation reconsidered: The concept of competence. *Psychological review*, 66(5):297, 1959.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.