Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 24 (2018) 15–20

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

4th International Conference on System-Integrated Intelligence

# A Reinforcement Learning Strategy for the Swing-Up of the Double Pendulum on a Cart

Michael Hesse[a,*], Julia Timmermann[a], Eyke Hüllermeier[b], Ansgar Trächtler[a]

[a]*Heinz Nixdorf Institute, Control Engineering and Mechatronics Group, Universität Paderborn, Fürstenallee 11, 33104 Paderborn, Germany*
[b]*Heinz Nixdorf Institute, Intelligent Systems Group, Universität Paderborn, Warburger Straße 100, 33098 Paderborn, Germany*

**Abstract**

The effective control design of a dynamical system traditionally relies on a high level of system understanding, usually expressed in terms of an exact physical model. In contrast to this, reinforcement learning adopts a data-driven approach and constructs an optimal control strategy by interacting with the underlying system. To keep the wear of real-world systems as low as possible, the learning process should be short. In our research, we used the state-of-the-art reinforcement learning method PILCO to design a feedback control strategy for the swing-up of the double pendulum on a cart with remarkably few test iterations at the test bench. PILCO stands for "probabilistic inference for learning control" and requires only few expert knowledge for learning. To achieve the swing-up of a double pendulum on a cart to its upper unstable equilibrium position, we introduce additional state restrictions to PILCO, so that the limited cart distance can be taken into account. Thanks to these measures, we were able to learn the swing up at the real test bench for the first time and in only 27 learning iterations.



*Keywords:* Reinforcement Learning; PILCO; double pendulum; experimental validation

## 1. Introduction

In contrast to the established paradigm of *model-based* control design, which requires a general understanding of the physical behavior of a technical system, *reinforcement Learning* (RL) algorithms [7] seek to learn an optimal control strategy through interaction with the system. Thus, RL is essentially data-driven and requires little prior knowledge. As such, it can be seen as a promising alternative that we are investigating for reasonably complementing more classical approaches in the field of control. In this paper, we demonstrate the potential of RL by learning the control of the double pendulum on a cart—for the first time also showing the swing-up on the real test bench.

* Corresponding author. Tel.: +49 5251 60-6228 ; fax: +49 5251 6297.
*E-mail address:* michael.hesse@hni.upb.de

2351-9789 
10.1016/j.promfg.2018.06.004

Our approach is based on the PILCO (Probabilistic Inference for Learning Control) algorithm [2], which is able to deal with continuous state and action spaces in a very data-efficient manner, and which requires rather few interactions on the test bench. It belongs to the class of model-based RL methods and uses a Gaussian process (GP) as a non-parametric approximation of the system. This type of model-based learning is able to extract valuable information very efficiently. Since PILCO adopts a probabilistic approach, it is able to suitably represent uncertain or unknown system behavior prevailing at the beginning of the learning process.

The double pendulum on a cart is an underactuated system with deterministic chaotic behavior. For the swing-up and the stabilization of the pendulum from the lower to the unstable upper equilibrium position, the two-degree-of-freedom method is a standard approach, and optimal swing-up trajectories can be determined, e. g. by optimal control [10], [4]. To make the problem amenable to RL, we have extended PILCO to include state constraints, so that, for example, the travel restriction of the linear motor on the test bench is reliably adhered to. Thanks to these extensions of the learning process, we were able to learn the swing-up of the double pendulum into the upper equilibrium position with only 27 learning iterations on the test bench. To the best of our knowledge, this is the first practical implementation of this maneuver through learning methods, and clearly demonstrates the efficiency of our implementation.

In the next section, we give an introduction to PILCO and describe our extensions in more detail. Then, the double pendulum on a cart and the application of PILCO to this practical example are discussed in Section 3. The results of the practical implementation and a corresponding analysis are presented in Section 4, followed by a conclusion and an outlook on future work in Section 5.

## 2. Probabilistic Inference for Learning Control: PILCO and Its Extension

In this section, we briefly recall the PILCO framework [2] and extend it by state-space constraints.

### 2.1. Problem formulation

We consider a discrete dynamical system of the form $x_{t+1} = f(x_t, u_t) + w_t$, $t \in \mathbb{N}_0$, with $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ defines the normal distribution of the initial state and white stationary Gaussian (iid) noise $w_t \sim \mathcal{N}(0, \Sigma_w)$, states $x_t \in \mathbb{R}^n$ and controls $u_t \in \mathbb{R}^p$. The transition dynamics $f$ is assumed to be unknown (and can not or should not be determined on the basis of physical laws). Our objective is to find a deterministic control policy $\pi$ with $u = \pi(x; \theta)$ that transfers the dynamical system to an equilibrium state and minimizes the extended expected long-term cost

$$J(\theta) = \sum_{t=0}^{T} \mathbb{E}[c(x_t)] - \lambda \sqrt{\mathrm{var}[c(x_t)]}, \qquad J(\theta) \in \mathbb{R} \tag{1}$$

for $T$ time steps, where $\mathbb{E}$ is the expected value operator, $c(x_t)$ the immediate cost of being in state $x_t$ at time step $t$ (cf. Section 2.3). The control policy $\pi$ is characterized by parameters that are summarized in the vector $\theta \in \mathbb{R}^{N_c}$.

The structure of our objective function in (1) is motivated by Bayesian optimization [9], especially the "optimism in the face of uncertainty" principle as implemented by the upper confidence bound (UCB) method. Via the weighting factor $\lambda \geq 0$, we are able to balance exploration and exploitation, which is important to avoid local minima: the larger $\lambda$, the stronger the exploration.

### 2.2. Gaussian process dynamics model and controller

Since we do not have access to the system dynamics $f$, a (posterior) distribution over plausible models of the dynamics is required. PILCO uses Gaussian processes [8] to infer this posterior distribution from currently available observations $(x_{t,i}, u_{t,i}, x_{t+1,i}), t = 0, \ldots, T-1, i = 1, \ldots, N_o$, which are directly measured from the real system and might be corrupted due to noise. The dynamic GP model is trained based on these observations (evidence maximization [3])

and is used within the simulation to make one-step predictions

$$p(x_{t+1}) = \Phi[p(x_t, u_t)], \quad p(x_t, u_t) = \mathcal{N}\left(\begin{bmatrix}\mu_{x_t}\\ \mu_{u_t}\end{bmatrix}, \begin{bmatrix}\Sigma_{x_t} & \Sigma_{x_t,u_t}\\ \Sigma_{x_t,u_t}^T & \Sigma_{u_t}\end{bmatrix}\right). \tag{2}$$

Thereby $\Phi$ maps the joint distribution of $p(x_t, u_t)$ to the distribution of the successor state $p(x_{t+1})$, which is a Gaussian approximation of the true distribution (moment matching [2]). For the GP model, the prior mean function is set to zero, essentially corresponding to no prior knowledge about the underlying dynamics function, and the squared exponential (SE) covariance function with additive noise is used. Due to this approach, we are able to perform long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$ from the initial state distribution $p(x_0)$ by cascading one-step ahead predictions via our learned dynamical GP model.

For the control policy $u_t = \pi(x_t; \theta) \in \mathbb{R}^p$, PILCO uses $p$ deterministic GPs, where the variance of the posterior is set to zero. Therefore, it uniquely maps the current state onto a specific input vector. The GP controller depends on several parameters (pseudo inputs: $M = [m_1, \ldots, m_{N_b}] \in \mathbb{R}^{n \times N_b}$, corresponding pseudo targets $l_i \in \mathbb{R}^{N_b}$ and diagonal weighting matrices $\Lambda_i^{-1} \in \mathbb{R}^{n \times n}, i = 1, \ldots, p$). This results in a dimension $N_c = nN_b + pN_b + pn$ of the optimization vector $\theta$ of the objective function (1). Thanks to an additional transformation of the input vector, it is also possible to consider input constraints, such that $|u| \leq u_{max}$ can be ensured.

### 2.3. Cost functions

The immediate cost function $c(x_t)$ in (1) can be built from different sub cost functions, such that $c(x_t) = \sum_i c_i(x_t)$. The original version of PILCO considers only one particular cost function, namely a saturated quadratic function with

$$c_q(x_t) = 1 - \exp\left(-\tfrac{1}{2}(x_t - x^*)^T W (x_t - x^*)\right) \in [0, 1], \tag{3}$$

where $x^*$ is the desired goal state and $W$ is an a priori defined weighting matrix. Using this single cost function, the swing-up and balancing of a double pendulum on a cart with an infinitely long track in simulation could be realized [1]. However, in order to apply PILCO to a real double pendulum on a cart, we have to take state space constraints into account, such as a limited track length. For that reason we introduce another cost function, the double hinge function

$$c_h(x^i) = \max\left(0,\ -a\,(x^i - b_1),\ a\,(x^i - b_2)\right) = \begin{cases} -a(x^i - b_1) & \text{for } x^i < b_1 \\ 0 & \text{for } b_1 \leq x^i \leq b_2 \\ a(x^i - b_2) & \text{for } x^i > b_2 \end{cases}, \qquad c_h(x^i) \in [0, \infty), \tag{4}$$

which depends on a single state variable $x^i \in \mathbb{R}$ and $a > 0, b_1 \leq b_2$. Leaving the region of feasible states $[b_1, b_2]$ results in additional costs via (4). Using a penalty term in this context is motivated by the interior-point optimization method. The parameter $a$ determines how strict the constraints are.

### 2.4. PILCO algorithm

Algorithm 1 summarizes the PILCO framework, which is a data-efficient reinforcement learning method for complex control tasks. Lines 1 and 2 describe the initialization phase, where the controller parameters are set randomly and a random control sequence is applied to the system. During this first interaction with the system, measurements are taken in parallel. After that, PILCO alternates between learning the system dynamics based on the currently avaible measurement data (line 4) and optimizing the controller parameters with regard to the current learned model (line 5).

In order to minimize the objective function (1) PILCO uses gradient-based optimizers, such as BFGS (Broyden-Fletcher-Goldfarb-Shanno [6]), which highly benefit from the fact that the gradient of the long term costs $\nabla J(\theta) = [\partial J(\theta)/\partial\theta_1, \ldots, \partial J(\theta)/\partial\theta_{N_c}]^T$ can be determined analytically (see [2] for details). Note that the controller optimization phase does not need any interaction with the real system, but is solely based on the learnt dynamical GP model. After the optimization, the current optimal control strategy is applied to the system, whereby additional measurement data is generated (line 6). If the control task was not fulfilled in the current iteration, the new data is added to the existing dataset and the dynamical GP model is re-trained on the basis of this enlarged dataset. The goal is to achieve a fast convergence of the GP model to the real system dynamics based on as few data as possible.

**Algorithm 1:** PILCO

1 Select random controller parameter $\theta$
2 Apply a random control sequence to the system and collect first measurement data
3 **repeat**
4 Learn the system dynamics $f$ by means of GP $\Phi$ based on the existing measurement data
5 Determine the optimal control policy $u = \pi(x_t; \theta^*)$ for the currently learnt model by minimizing $J(\theta)$
6 Apply the control strategy $\pi(x_t; \theta^*)$ to the system and collect further measurement data, which are added to the existing measurement data
7 **until** *Control task fulfilled*

## 3. Experimental Setup

In this section, we present the modeling and experimental setup of the double pendulum on a cart. The physical model is used to test the learning algorithm in an adequate simulation environment. Furthermore, we describe the parameter settings according to the modeled GPs and cost functions.

### *3.1. Modeling and experimental setup of the double pendulum on a cart*

To illustrate the presented approach, we apply it to a double pendulum on a cart test bench. A picture of our test bench is shown on the left side of Fig. 1. The system under consideration constitutes a serial double pendulum pivoted on a high-performance linear motor. The linear motor has a maximal travel of $\pm 0.6\ m$, a maximal permitted velocity of $6\ m/s$, and a maximal permitted acceleration of $100\ m/s^2$. We use the abstract scheme on the right side of Fig. 1 to represent the essential dynamics of the system. The sketched model has three degrees of freedom: two angles between the pendulums and the vertical axis, denoted by $\varphi_1$ and $\varphi_2$, and the position of the cart denoted by $y$. We use the Lagrange formalism to derive the differential equations. This derivation and the explicit mechanical parameters corresponding to the test bench can be found in [10]. A partial feedback linearization makes it possible to reduce the system dynamics of the pendulum. The new input is the acceleration of the motor $u = \ddot{y}$ and its realization is ensured by an underlaying fast velocity controller. The simplified physical model of the double pendulum on a cart is used to test the learning method based on simulation before applying it to the real system. The simulation study showed promising results, so that we were confident that the learning process will also succeed on the real system.

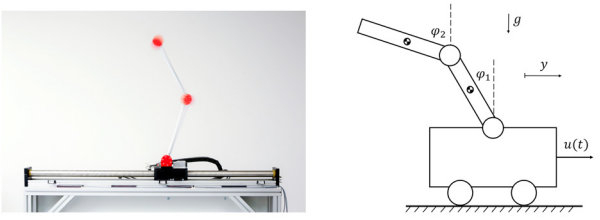


Fig. 1. Test bench of the double pendulum on a cart (left) and the representative physical model (right).

### 3.2. Parameter settings

The application of PILCO to the double pendulum on a cart test bench requires the specification of the parameters in the cost functions listed in Table 1. To this end, we leverage our expert knowledge about the underlying control task. According to Section 2, we build the immediate cost function $c(x)$ from different sub cost functions $c_1, \ldots, c_6$. In total, the cost function is composed of two saturated quadratic $(c_1, c_2)$ and four double hinge $(c_3, \ldots, c_6)$ cost functions, depending on the states $\varphi_1, \varphi_2, y, \dot{y}, \dot{\varphi}_1, \dot{\varphi}_2$ and additional quantities $\sin(\varphi_1), \cos(\varphi_1), \sin(\varphi_2), \cos(\varphi_2)$, where the angles of the pendulum are mapped to the cartesian coordinates. Hence the goal of the maneuver can be described by $\cos(\varphi_1) = \cos(\varphi_2) = 1$, which has the advantage of taking the ambiguity of the upper equilibrium position into account. We used two squared quadratic cost function, where the latter, $c_2$ has high entries of the weighting matrix to ensure that the gradient of (1) does not vanish during the optimization. The parameters of the hinge functions are set so as to enforce a gap to the actual state constraints, e. g. $y_{max} = 0.6\,m > b_{2,3} = 0.5\,m$. To facilitate the learning process, the input constraint is set to $|u| \leq 30\,m/s^2$, so that the control input is appropriately limited for the optimization. Another adjustment is made with regard to the prediction time $T$. We start with $T_0 = 3\,s$ and increase the prediction time after each trial by 25% if the actual immediate costs are below 0.25 for the last 0.5 $s$ (see Fig. 2, red bar). This approach helps the controller to learn the swing-up maneuver first, and to stabilize the system in the upper equilibrium point afterward. We implicitly assume that the controller can stabilize the pendulum for an infinitely long period of time once it achieved a stabilization for 6 $s$.

Table 1. Parameters for the double pendulum on a cart experiment

| sub cost function | dependent on | parametrization |
|---|---|---|
| $c_1$ (saturated quadratic) | $y$ $\sin(\varphi_1)$ $\cos(\varphi_1)$ $\sin(\varphi_2)$ $\cos(\varphi_2)$ | $W_1 = 0.75^{-2} C^T C$, $x_1^* = [0, 0, 1, 0, 1]$ |
| $c_2$ (saturated quadratic) | $y$ $\sin(\varphi_1)$ $\cos(\varphi_1)$ $\sin(\varphi_2)$ $\cos(\varphi_2)$ | $W_2 = 9W_1$, $x_2^* = x_1^*$ |
| $c_3$ (double hinge) | $y$ | $a_3 = 10$, $b_{1,3} = -0.5$, $b_{2,3} = 0.5$ |
| $c_4$ (double hinge) | $\dot{y}$ | $a_4 = 1$, $b_{1,4} = -3.5$, $b_{2,4} = 3.5$ |
| $c_5$ (double hinge) | $\varphi_1$ | $a_5 = 1$, $b_{1,5} = -8\pi$, $b_{2,5} = 8\pi$ |
| $c_6$ (double hinge) | $\varphi_2$ | $a_6 = 1$, $b_{1,6} = -8\pi$, $b_{2,6} = 8\pi$ |
| time discretization $\Delta t$ : 0.05$s$ | number of controller basis functions $N_b$ : 200 | exploration factor $\lambda$ : 0.1 $C = \begin{bmatrix} 1 & -0.5 & 0 & -0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \end{bmatrix}$ |

## 4. Experimental Validation and Analysis of Results

We applied PILCO with the parameter settings and adjustments described above to a real double pendulum on a cart. This section is dedicated to the presentation and analysis of the results obtained from this experiment. Fig. 2 shows the state and cost trajectories of the $10^{th}$ and $27^{th}$ (last) learning iteration.

The first iterations are characterized by a high uncertainty (standard deviation) regarding the predictions. This is due to the initial inaccuracy of the dynamic GP model, and leads to a violation of the state restrictions in the first few iterations. After some more interaction with the system, a fundamental avoidance strategy is developed by the controller. In the $10^{th}$ iteration, the target (upper equilibrium) state is approached for the first time at 1.8 $s$. This is an essential step in the learning process, because till then the target state had to be extrapolated on the basis of the existing data, which mainly contain observations around the lower equilibrium state. After the $10^{th}$ iteration, the found swing-up trajectory is maintained and refined. Moreover, the prediction horizon is gradually increased to 6 $s$, so that the stabilization of the upper equilibrium state is learned as well. Fig. 2 (right) shows the final result after the last learning step. The remaining uncertainty between 1.8 $s$ and 3.8 $s$ of the cart's position prediction can be explained by the transition from swing-up to balancing, as the highly accelerated masses have to be adequately slowed down.

Overall, we needed 27 iterations, which is close to the 23 iterations reported in [1], where PILCO was applied to a simulation model of the double pendulum on a cart. The main differences are as follows: (i) We applied PILCO to a real test bench; (ii) our system input is the cart acceleration and not the actuator force acting on the cart; (iii) the maximum cart displacement of the final trajectory in [1] amounts to 1.4 $m$ and in our experiment to 0.4 $m$, since the

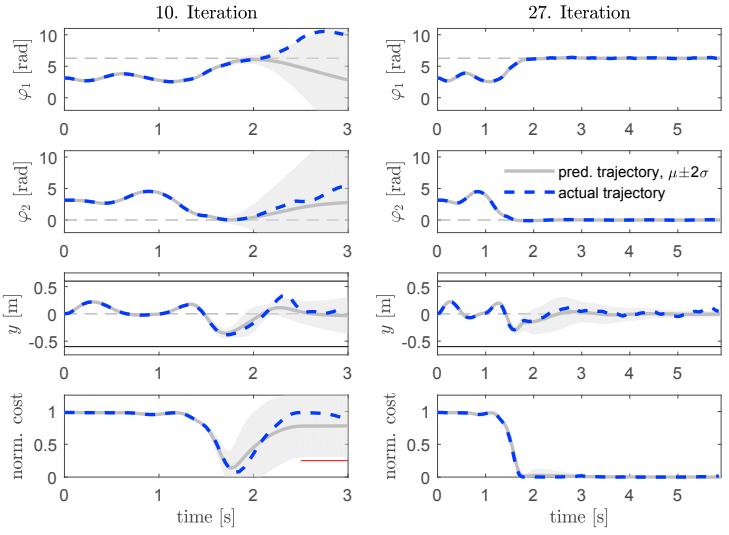


Fig. 2. Application of PILCO to the double pendulum on a cart test bench. Presentation of the 10. and 27. iteration of the learning process.

cart limitation is taken into account during the learning process. A video of the experimental validation is available at https://www.youtube.com/watch?v=N-yrQu9zuOI.

## 5. Conslusion and Future Work

We were able to demonstrate the swing-up of the double pendulum into the upper unstable rest position at the test bench within a few learning iterations by modifying and applying the PILCO algorithm. In the future, we intend to explore Deep-PILCO [5] to investigate whether the better scaling properties of Bayesian neural networks are also advantageous for the comparatively high dimensional state vector of the double pendulum on a cart. Moreover, we want to elaborate on the advantages of combining data-driven methods such as PILCO with established model-based approaches of control theory in situations where at least partial knowledge about the technical system is available.

## References

[1] Deisenroth, M.P., 2010. Efficient reinforcement learning using Gaussian processes: Zugl.: Karlsruhe, KIT, Diss., 2009. volume 9 of *Karlsruhe series on intelligent sensor-actuator-systems*. KIT Scientific Publishing and Technische Informationsbibliothek u. Universitätsbibliothek, Karlsruhe and Hannover.

[2] Deisenroth, M.P., Fox, D., Rasmussen, C.E., 2015. Gaussian Processes for Data-Efficient Learning in Robotics and Control. IEEE transactions on pattern analysis and machine intelligence 37, 408–423.

[3] Deisenroth, M.P.b., Turner, R.D.b., Huber, M.F.b., Hanebeck, U.D.b., Rasmussen, C.E.b., 2012. Robust Filtering and Smoothing with Gaussian Processes. IEEE Transactions on Automatic Control 57, 1865–1871.

[4] Flaßkamp, K., Timmermann, J., Ober-Blöbaum, S., Trächtler, A., 2014. Control strategies on stable manifolds for energy-efficient swing-ups of double pendula. International Journal of Control 87, 1886–1905.

[5] Gal, Yarin and McAllister, Rowan and Rasmussen, Carl Edward, 2016. Improving PILCO with Bayesian neural network dynamics models. Data-Efficient Machine Learning workshop, ICML .

[6] Nocedal, J., Wright, S. (Eds.), 2005. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. 2., ed. ed., Springer New York and Springer Bln, Berlin and Berlin.

[7] Richard S. Sutton and Andrew G. Barto, 1998. Reinforcement Learning : An Introduction. MIT Press.

[8] Seeger, M., 2004. Gaussian Processes for Machine Learning. International Journal of Neural Systems 14, 69–104.

[9] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N., 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proceedings of the IEEE 104, 148–175.

[10] Timmermann, J., Khatab, S., Ober-Blöbaum, S., Trächtler, A., 2011. Discrete Mechanics and Optimal Control and its Application to a Double Pendulum on a Cart. IFAC Proceedings Volumes 44, 10199–10206.