Babeș-Bolyai University
Faculty of Mathematics and
Computer Science

Intelligent Models for Robotic Behavior, Decision Making and Environment Interaction

Summary of the PhD thesis

Scientific supervisor:
Prof. Dr. Horia F. Pop
Dr. Istenes Zoltán

PhD student:
Hunor Sándor Jakab

Cluj Napoca
2012

Table of contents of the abstract

Table of contents of the thesis

Keywords: robotics, reinforcement learning, machine learning, non-parametric methods.

1 Introduction

Robotics related research has gained a lot of popularity in the last decade due to a significant increase in automation in different industrial and commercial applications by means of robots. One of the most important goals of robotics is to develop intelligent robotic agents capable of evolving autonomous decision making mechanisms and learning new skills. Our work focuses on robotic motion control, which we consider to be a crucial prerequisite for any robot in order to operate in an uncertain and unstructured environment.

Although much progress has been made in this area of research, the acquisition of otherwise basic abilities like walking or grasping with multi-degrees of freedom robots, sill cannot be fully accomplished by means of autonomous learning. Researchers in neuroscience have shown evidence [Kawato, 1999] that biological systems handle the problem of motor control based on estimated internal world models, which are effective in predicting the outcome and long-term consequences of executing motor commands.

To accomplish motor control in a similar fashion for robotic agents both mathematical and algorithmic tools need to be employed for the development of internal models and decision making mechanisms. An important characteristic of the learning problems treated in this research is the unsupervised nature of learning, the need for autonomous knowledge acquisition through data acquired during environment interaction. Therefore our proposed methods and algorithms are based on machine learning methods from the framework of reinforcement learning. Applying reinforcement learning algorithms for the control of realistic robotic systems, however proves to be challenging even in simple settings, where the robot has only a small number of possible states and actions. These difficulties can be attributed mainly to the exponential increase of the search-space volume with the increase in degrees of freedom of the controllable robotic system and the high uncertainty which is the result of physical imperfections of robot components. The number of experiments that can be performed on a physical robot is also limited, which is a consequence of both physical and temporal constraints and this alone makes it impossible to rely on exhaustive search algorithms. The goal and motivation of this thesis is to address some of the fundamental problems of developing intelligent learning models in robotic control, by introducing novel reinforcement learning methods specifically designed for the robotic control domain. The main problem areas that we have put special emphasis on, are the treatment of continuous state-action spaces, treatment of uncertainty, on-line(real time) operation, sample efficiency and the inference of structural properties of the learning system from data gathered during environment interaction. For the purpose of validation we make use of a series of realistically simulated robotic control tasks with continuous state-action spaces, noisy state transitions multiple degrees of freedom.

2 Reinforcement learning for robotic control

Robotic locomotion is accomplished by sending control signals to actuators based on an action selection policy. Motor control policies define an action selection mechanism which enables the robot to generate motion sequences for the execution of certain tasks with a predefined goal. Formally a control policy ($\mathcal{CP}$) is represented as a parameterized function $a \leftarrow \pi_\theta(s,t)$, which is a mapping from a state s to an executable action a. In case of a real life robot a state is represented by a vector which can contain kinematic, dynamic and sensory informations whereas the actions are for example a set of torque values for the different joints. The learning of an optimal control policy is accomplished by interacting with the environment and adjusting the decision making mechanism based on sensory information received after the execution of certain actions. Autonomous robotic learning also imposes some constraints on the range of tools that can be used for its accomplishment. Opposed to supervised learning, we have to choose methods from within the framework of reinforcement learning($\mathcal{RL}$) where no predefined training data is available in form of optimal actions. The robot only knows how to measure the utility of an action, and it is supposed to learn a good control policy by interacting with it's environment and maximizing the received feedback based on the utility measure, also called *Reward*. In the majority of control tasks, however maximizing the immediate reward is insufficient for optimality, therefore delayed rewards need also be taken into account. In other words, the agent has to learn which of its actions are the most optimal, based on rewards that can take place arbitrarily far in the future. A common assumption is that a description of the environment and robot or a state-variable at a given time t contains all the relevant information to make decisions, called the Markov property. As a consequence the conditional distribution of future states depends only on the current state [Papoulis, 1991]. Problems of sequential decision making with the Markov property are mathematically modelled by a Markov Decision Process($\mathcal{MDP}$). They provide the mathematical foundations for the earliest solutions of the sequential decision making problem, and have played a central role in the development of $\mathcal{RL}$.

Definition 1. *A continuous Markov Decision Process is a quadruple* $M(S, A, P, R)$ *where the following notations are used:* S *is the set of states,* A *is the set of actions,* $P : S \times A \times S \to [0, 1]$, *written as* $P(s'|s, a)$ *is the probability of arriving in state* s' *when taking action* a *in state* s. P *is called the transition probability or the model of the environment.* $R : S \times A \to \mathbb{R}$, $r = R(s, a)$ *denotes the* reward function.

As an abstraction of the decision making mechanism driving the robot, we define a *parameterized policy* $\pi : S \times A \to [0, 1]$ in form of a probability distribution over state-action space, a mapping from each state $s \in S$ and action $a \in A$ through the conditional probability $\pi(a|s)$ of taking action a while being in state s. Non-deterministic policies are obtained by adding a Gaussian *exploratory noise* with covariance $\boldsymbol{\Sigma}_{\text{ex}} = \boldsymbol{\sigma}_{\text{ex}} I$ to a deterministic controller function $f : S \to R$.

$$\begin{aligned} \pi_\theta(\mathbf{a}|\mathbf{s}) &= f(s, \theta_c) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{\sigma}_\epsilon I) \\ &= \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \exp\left(-\frac{(a - f_{\theta_c}(s))^2}{\sigma_\epsilon^2}\right) \end{aligned} \tag{.1}$$

where $\pi_\theta(\cdot, s)$ is the parameterized controller with parameter vector $\theta = \begin{bmatrix}\theta_c^T & \sigma_\epsilon\end{bmatrix}^T$, $\boldsymbol{\epsilon}$ is the zero mean Gaussian noise term with standard deviation $\boldsymbol{\sigma}_\epsilon$ and θ_c is the reduced parameter vector of the controller. The controller function is central for movement generation and throughout the scientific literature it has been defined in many different ways: trajectory generating controllers like spline-based trajectory generators, dynamic motion primitives, and direct torque controllers like the cerebellar model articulation controller (CMAC). The solution of a sequential decision making control problem is *the optimal policy* π^* maximizing the expected return :

$$\pi^* = \pi_{\theta^*} = \underset{\pi}{\operatorname{argmax}}\left(J^\pi\right), \quad \text{with} \qquad J^\pi = E_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right] \tag{.2}$$

Where J^π is the objective function of the optimization problem, an expectation of the discounted reward; the expectation being computed *for the policy* π_θ and $r_1, r_2, \ldots$ are instantaneous rewards. During environment interaction the data gathered through sensory measurements is used to build up models of optimality upon which action-selection mechanisms can be built. These models are called value functions, they can be associated to either state or state-action space:

$$Q^{\pi}(s,a) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r_{t} | s_0 = s, a_0 = a\right] \tag{.3}$$

Classical reinforcement learning methods are all based on the estimation of value functions which serve as an intermediate step between experience accumulation and policy generation. Bellman's equations play a central role in this regard.

$$Q_{\pi}(s,a) = \sum_{s' \in S} P(s'|s,a) \sum_{a' \in A} \pi(a'|s') \left(R(s,a) + \gamma Q_{\pi}(s',a')\right)$$

A large number of algorithms have been developed based on the recursive relation between value functions and the contractive properties of the Bellman equations.Their application to robotic control however is limited since exact tabular representations become impossible for continuous state-action spaces and the use of parametric function approximation is unstable.

Another policy learning approach, the family of policy gradient algorithms is more suited for the robotic control domain. It optimizes a parameterized policy π_{θ} by stochastic gradient ascent, where the objective function is the same as before, and the gradient is estimated from experience gathered through environment interaction.

$$\nabla_{\theta} J(\theta) = E_{\tau}\left[\sum_{t=0}^{H-1} \nabla_{\theta} \log \pi(a_t|s_t) R(\tau)\right] \qquad \tau = \{(s_1,a_1,r_1),\ldots,(s_H,a_H,r_H)\} \tag{.4}$$

Where τ is a trajectory i.e. a sequence of state-action pairs visited during an experiment of H steps, and the rewards received. When formulated in this form, the gradient can be approximated by Monte Carlo integration, by taking the average over a number of controller output histories [Williams, 1992]. Extending these algorithms to continuous state-action spaces is done mainly by using function approximation to represent value functions, however this raises convergence problems and can introduce bias in the estimates. In this work we make extensive use of non-parametric function approximation, namely Gaussian process regression, to represent knowledge and uncertainty throughout the learning process. Our novel robotic control methods are all based on building up an internal model of optimality by means of a $\mathcal{GP}$ and using it efficiently to direct the environment interaction process and learn optimal control policies.

3 Non-parametric approximation of value functions for robotic control with proper uncertainty treatment

Building models of optimality in form of value functions is an important element of all reinforcement learning problems. To cope with the problem of handling continuous state-actions spaces and high degree of uncertainty in robotic control, we analyzed the use of non-parametric function approximation specially emphasizing the benefits and drawbacks of using Gaussian processes for the purpose of approximating state and state-action value functions [Jakab and Csató, 2010]. We model the value function by placing a Gaussian prior directly in the space of functions. During environment interaction we obtain a sequence of n state-action pairs and corresponding rewards in form of a trajectory: $\tau = [(s_1, a_1, r_1), \ldots, (s_H, a_H, r_H)]$. After performing m trajectories we obtain a training set $\mathcal{D} = [(x_1, r_1) \ldots (x_n, r_n)]$, ,where $n = mH$. Using the state-action pairs as training points $x_t \overset{\text{def}}{=} (s_t, a_t) \in \mathcal{D}$ and the corresponding cumulative returns $\mathrm{Ret}(x_t) = \mathrm{Ret}(s_t, a_t) = \sum_{i=t}^{H} \gamma^{i-t} R(s_t, a_t)$ as training targets we obtain a fully probabilistic model in form of a Gaussian posterior for the value function:

$$\begin{aligned} Q_{\mathcal{GP}}|\mathcal{D}, x_{n+1} &\sim \mathcal{N}\left(\mu_{n+1}, \sigma^2_{n+1}\right) \\ \mu_{n+1} = \mathbf{k}_{n+1}\boldsymbol{\alpha}_n \qquad & \sigma^2_{n+1} = k_q\left(x_{n+1}, x_{n+1}\right) - \mathbf{k}_{n+1}\mathbf{C}_n\mathbf{k}^T_{n+1}, \end{aligned} \tag{.5}$$

where $\boldsymbol{\alpha}_n$ and $\mathbf{C}_n$ are the parameters of the $\mathcal{GP}$ – for details see [Rasmussen and Williams, 2006] – with the following form

$$\boldsymbol{\alpha}_n = [\mathbf{K}^n_q + \boldsymbol{\Sigma}_n]^{-1}\hat{\mathbf{Q}}, \quad \mathbf{C}_n = [\mathbf{K}^n_q + \boldsymbol{\Sigma}_n]^{-1}. \tag{.6}$$

The set of training points (in this case $\mathcal{D}$) based on which the $\mathcal{GP}$ parameters $\boldsymbol{\alpha}$ and $\mathbf{C}$ are calculated is called the basis vector set. Since the mean function μ can be set to zero without losing generality, the only *unspecified* element of the Gaussian process is the covariance function. Choosing an appropriate covariance function and tuning the hyper-parameters to fit the problem at hand are crucial for the achievement of good approximation accuracy. To conform to the sequential nature of robotic control experiments, we apply an on-line version of the Gaussian process regression algorithm, where the parameters $\boldsymbol{\alpha}$ and $\mathbf{C}$ are updated each time a new experience-data arrives. The on-line updates enable us to run the value function approximation parallel to environment interaction.

Claim 1. *Using a Gaussian process approximated value function the gradient variance can be significantly reduced in policy gradient algorithms which leads to faster convergence and better peak-performance. It also enhances performance in approximate temporal difference learning for robotic control. [Jakab et al., 2011] [Jakab and Csató, 2010].*

To calculate the policy gradient from data acquired through a number of experiments we apply the likelihood ratio trick and replace the Monte Carlo samples of cumulative rewards by a combination of k-step discounted rewards and Gaussian process approximated action-values. The expression for the gradient becomes:

$$\nabla_\theta J(\theta) = \left\langle \sum_{t=0}^{H-1} \nabla_\theta \log \pi(a_t|s_t) \left(\sum_{i=0}^{k-1} \gamma^k R(s_{t+k}, a_{t+k}) + \mathbf{k}_{x_{t+k}} \left[\mathbf{K}_q + \mathbf{\Sigma}\right]^{-1} \hat{\mathbf{Q}} \right) \right\rangle_\tau \quad (.7)$$

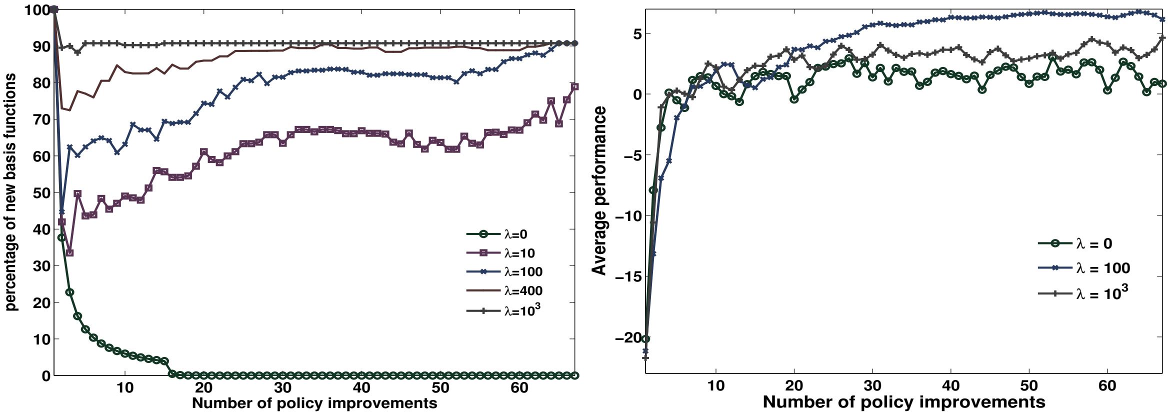
We study the behaviour of temporal difference algorithms with Gaussian process function approximation. Let us consider an episode τ consisting of $\{(s_t, a_t)\}_{t=\overline{1,H}}$ state-action pairs. The data from the visited state-action pairs and the immediate rewards serve as a basis for the generation of our $\mathcal{GP}$ training data. As training points we use the unchanged state-action pairs, and for training labels we use a combination of immediate rewards and estimated Q-values according to the following formula:

$$Q^m(s_t, a_t) = \sum_{i=0}^{m-1} \gamma^i R(s_{t+i}, a_{t+i}) + \gamma^m \max_a Q_{\text{pred}}(s_{t+m}, a_{t+m}), \quad (.8)$$

where $R(s_t, a_t)$ denotes the reward observed at time t, $\gamma \in (0, 1]$ is a discount factor, and Q_{pred} is the approximated value of the action-value function using $\mathcal{GP}$ inference at step $t + m$. Further, we combine the Q-learning update rule and the on-line Gaussian process update to obtain the update expressions for the $\mathcal{GP}$ action-value function approximator:

$$\begin{aligned} q^{n+1} &= \frac{\hat{Q}^1(s_t, a_t) - Q_{\text{pred}}(s_t, a_t)}{\sigma_0^2 + \sigma_{n+1}^2} \\ &= \frac{\alpha \left(R(s_t, a_t) + \gamma \max_a Q_{\text{pred}}(s_{t+1}, a) - Q_{\text{pred}}(s_t, a_t)\right)}{\sigma_0^2 + \sigma_{n+1}^2}. \end{aligned}$$

Using the above expression to incorporate temporal difference errors into future predictions and into the expanded covariance matrix, corresponds to the stochastic averaging that takes place in tabular Q-learning.



(a)

(b)

Figure .1: Effect of λ on (a) performance evolution (b) average percentage of out-of date basis functions present in the action-value function approximation

Due to the stochasticity of real-world learning systems the performance of learning algorithms can be largely affected by noisy measurements and external disturbances. The combination of fully probabilistically estimated value-functions with existing gradient-based policy search and value function based temporal difference learning provides significant performance improvement for robotic control problems.

4 Efficient sample reuse and improved stability in non-parametric value function approximation

Due to the non-parametric nature of the value function approximation scheme that we used above, the computational complexity of our methods increases with the number of the training data. To overcome this problem, we present a Kullback Leibler distance-based sparsification mechanism which decreases the computational cost of the Gaussian process approximation and opens up further possibilities for improving sample efficiency. A new training data-point can be expressed as the linear combination of the previous inputs in feature space and an additional error term:

$$\phi_{n+1} = \gamma_{n+1}\phi_{res} + \sum_{i=1}^{n} \hat{e}_{n+1}(i)\boldsymbol{\phi}_i \tag{.9}$$

analytical form for the projection coordinates and the residual:

$$\hat{e}_{n+1} = [\mathbf{K}_q^n]^{-1}\mathbf{k}_{n+1}, \quad \gamma_{n+1} = k_q(x_{n+1}, x_{n+1}) - k_{n+1}^T \hat{e}_{n+1} \tag{.10}$$

where $\mathbf{K}_q^n$ is the kernel Gram matrix for the first n data-points, and k_{n+1} is the empirical kernel map:

$$k_{n+1} = \begin{bmatrix} k_q(x_1, x_{n+1}) & \dots & k_q(x_n, x_{n+1}) \end{bmatrix}^T \tag{.11}$$

When deciding upon the addition of a new data points to the basis vector set, we have to verify if γ is within a tolerance threshold. This ensures that only data-points from significantly different regions of the state-action space are added to the basis vector set.

Claim 2. *We introduce a novel sparsification mechanism [Jakab and Csató, 2011], which allows us to avoid the re-estimation of the value function after a policy change occurs. It improves the stability of the estimated value function and recycles old training data. We provide evidence that it improves the approximation accuracy and the sample efficiency of the resulting reinforcement learning algorithms. Moreover it enables on-line updates for policy iteration algorithms and eliminates the fluctuation of the approximated value function, encountered when using parametric approximation architectures.*

The main idea of our contribution is that we assign a time variable to every data point in our $\mathcal{BV}$ set which signifies at which stage of the learning process the data point has been added to the basis vector set.

$$D = \{(x_1, y_1), \dots x_n, y_n\} \rightarrow \{(x_1, y_1, t_1), \dots (x_n, y_n, t_n)\} \tag{.12}$$

Moreover, we introduce a modified KL-distance based scoring mechanism by adding a term which penalizes basis vectors that have been introduced in early stages of the learning process and those that do not contribute significantly to the posterior mean of the Gaussian process. Whenever a new data point is being processed which needs to be included in the basis vector set but the maximum number of basis vectors has been reached we compute a modified score $\varepsilon'(\cdot)$ for each element:

$$\varepsilon'(i) = (1-\lambda)\frac{\alpha^2(i)}{q(i) + c(i)} + \lambda g\left(t(i)\right) \tag{.13}$$

, where the $\lambda \in [0, 1]$ term from eq. (.13) serves as a trade off factor between loss of information and accuracy of representation, c is a scaling constant. We replace the lowest scoring data point from the basis vector set with our new measurement. Here $g(\cdot)$ is a function of the time variable assigned to each basis vector. Since we want to favour the removal of out-of date basis vectors, this function needs to be monotonically increasing. In our experiments we used exponential and logit functions of the form:

$$g(t_i) = c \exp\left(t_i - \min_i(t_i)\right) \quad g(t_i) = c \log\left(\frac{t_i/\max(t_i)}{1 - t_i/\max(t_i)}\right) \quad i = 1 \dots n \tag{.14}$$

Figure .1(a) shows the composition composition of the training data set for different value of the trade-off parameter λ. Putting higher emphasis on information content lowers the percentage of up-to date measurements. The effect of λ also has a major impact on the peak performance of the resulting algorithms as shown on Figure .1(b).

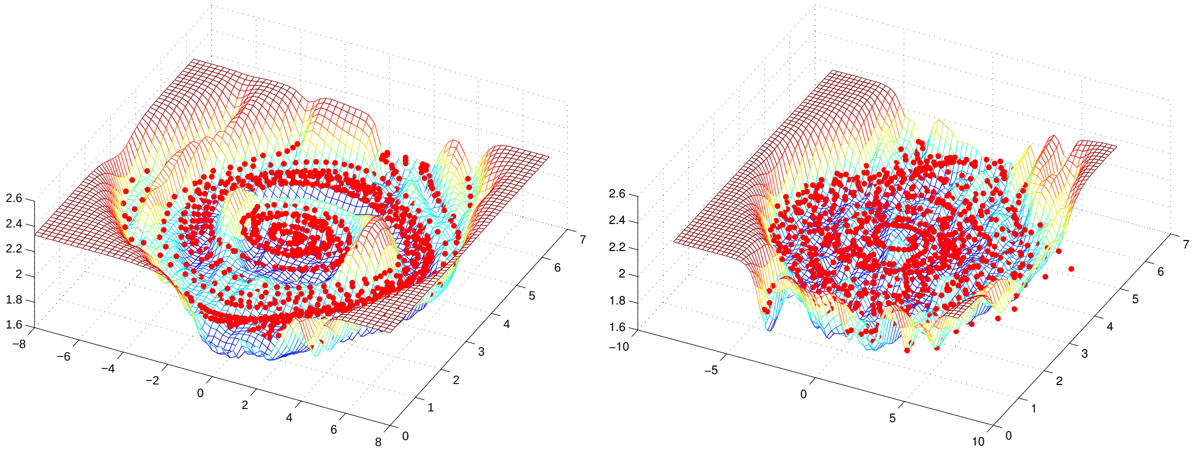
5 Intelligent exploration strategies

The third problem area investigated in the thesis is that of ***efficient balancing between exploration and exploitation***. The already elaborated fully probabilistic models provided by the non-parametric estimation of value functions can be made use of, to accomplish good exploration strategies [Jakab, 2010]. Searching for optimal control policies is done by executing previously untried actions and observing the changes that they induce in the environment. In Chapter 4 of the thesis we present two intelligent exploratory mechanisms which is are based on the predicted values and model confidence of the estimated value functions. $\mathcal{GP}$ model of the state-action value function in (.6).

Claim 3. *We developed a model confidence based adaptive exploratory mechanism for both policy gradient and value based algorithms. We achieved the adaptive behaviour by replacing the* fixed *exploratory noise of the Guassian policy* (.1) *with noise proportional to the model confidence of the $\mathcal{GP}$ approximated value function in a given region. As a result, exploratory actions target regions of the search space with high uncertainty, leading to a more in-depth exploration.*

The obtained stochastic action selection policy with adaptive exploratory noise variance takes the following form:

$$\begin{aligned} \pi_\theta &= f(s,\theta) + \mathcal{N}(0, \sigma^2_{\mathcal{GP}} I) \\ \sigma^2_{\mathcal{GP}} &= \lambda\left(k_q(x^*, x^*) - \mathbf{k}^* \mathbf{C}_n \mathbf{k}^{*T}\right) \end{aligned} \qquad \text{with} \quad x^* = \{s, f(s,\theta)\}, \tag{.15}$$



where $x^* \overset{\text{def}}{=} (s, f(s, \theta))$, k_q is the covariance function and $\mathbf{C}_n$ is the parameter of the $\mathcal{GP}$ after processing n data-points. As learning progresses, the predictive variance will decrease with the arrival of new data-points, leading to a gradual decrease in exploration, and stabilization of the value function.

Claim 4. *We developed a novel stochastic action-selection policy which balances between greedy action selection and following the path described by the deterministic controller* $f(s, \theta)$. *The new policy enables us to guide the exploration to regions of the state-action space which have higher estimated values and at the same time keeping the generated distribution of state-action pairs close to the path described by the deterministic controller. It can be considered as a transition between on and off-policy learning [Jakab and Csató, 2011],[Jakab and Csató, 2012b].*

Through the use of a gibbs distributed stochastic policy where the energies are functions of a Gaussian process estimated action-value function, we can restrict exploration to relevant regions of the search space. This leads to a decrease in the number of necessary environment interactions and faster convergence. We propose a stochastic policy $\pi(a|s)$ in form of a Boltzmann distribution[Neal, 2010] over actions from the neighbourhood of $f_\theta(s)$:

$$\pi(a|s) = \frac{e^{\beta E(s,a)}}{Z(\beta)}, \quad \text{where } Z(\beta) = \int da\ e^{\beta E(s,a)} \tag{.16}$$

The term $Z(\theta)$ is a normalizing constant and β is the inverse temperature. The Gibbs distribution has its origins in statistical mechanics and is related to the modelling of the particle velocities in gases as a function of temperature. The temperature β^{-1}

determines how significantly the energy differences affect the selection probabilities of the different actions.

To include the deterministic controller f_θ in the action selection, we construct the energy function $E(s, a)$ such that only actions neighbouring $f_\theta(s)$ have significant selection probability. At the same time we want to assign higher probability to actions that – in the current state s_t – have higher estimated Q-values. The energy function has the following form:

$$E(s, a) \quad = \quad Q_{\mathcal{GP}}(s, a) \cdot \exp\left[-\frac{\| a - f_\theta(s) \|^2}{2\sigma_e^2}\right]$$

It is composed of the $\mathcal{GP}$-estimated Q-value $Q_{\mathcal{GP}}(s, a)$ for the state-action pair (s, a) and a Gaussian on the action space to limit the selection to the neighbourhood of the controller output $f_\theta(s)$. The variance parameter σ_e can either be fixed, or made dependent on the $\mathcal{GP}$ predictive variance. Combining the above defined energy function with eq. (.16) we get the following expression for the Gibbs distribution based stochastic action selection policy:

$$\pi(a|s) = \frac{\exp\left(\beta Q_{\mathcal{GP}}(s, a) \cdot \exp\left[-\frac{\|a - f_\theta(s)\|^2}{2\sigma_e^2}\right]\right)}{Z(\beta)} \tag{.17}$$

$$Z(\beta) = \int da \exp\left(\beta Q_{\mathcal{GP}}(s, a) \cdot \exp\left[-\frac{\| a - f_\theta(s) \|^2}{2\sigma_e^2}\right]\right)$$

Figure .2 illustrates the obtained stochastic action selection mechanism. As it is shown in [Jakab and Csató, 2011],[Jakab and Csató, 2012b] model confidence-based search magnitude and direction guidance can improve the performance of control policy learning significantly and enables the targeting of exploratory actions to important regions of the state-action space.

6 Manifold-based on-line learning from structural information

There are many robotic control tasks where the corresponding action-value function is discontinuous in some regions of the space, and this discontinuity has great influence on the algorithms performance. One of the major drawbacks of using $\mathcal{GP}$ action-value function approximation with traditional stationary kernel functions is

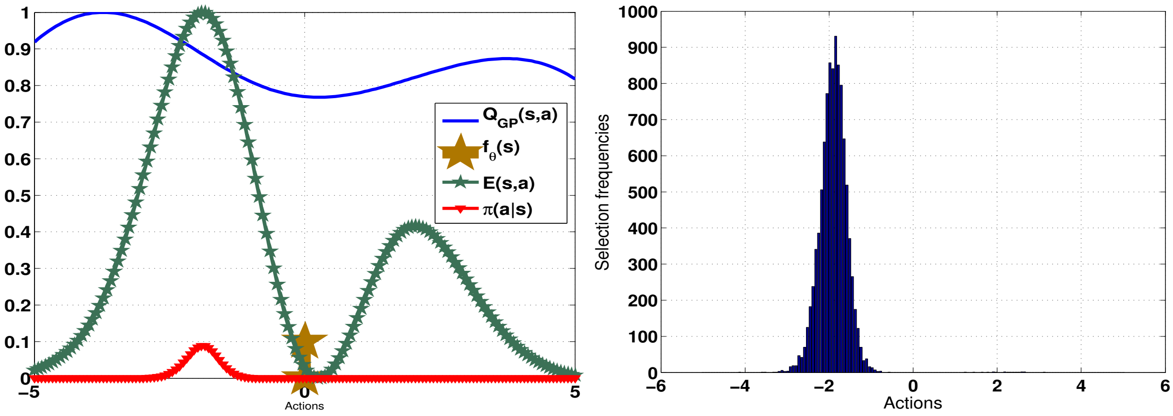


Figure .2: The Gibbs action selection policy with temperature set to 1 (a)estimated Q-values , (b) selection frequencies

that the accurate representation of discontinuities in case of discontinuous $\mathcal{VF}$s is not possible. Most notably $\mathcal{RL}$, algorithms that use a greedy action-selection policy based on the value function suffer from this phenomenon.

Stationary kernel functions essentially operate on distances between data-points and are invariant to translations in the input space. A value approximation at a certain data-point is a locally weighted average of training targets, where the weights are dependent on the Euclidean distances between training and test points. In contrast by temporal difference value function evaluation each update is based on the expected values of states that lie on the agents traversed trajectory.

Claim 5. *We present a modality to increase the accuracy of our $\mathcal{GP}$ action-value function estimates and to achieve a more similar working mechanism to temporal difference learning by introducing a new class of kernel functions that operate on a graph structure induced by the Markov decision process underlying the $\mathcal{RL}$ problem. We construct a graph structure from the state-action sequences visited during experiments, conditioned on their addition to the basis vector set in the $\mathcal{GP}$ approximation, and define a new distance measure in form of shortest paths on the graph [Jakab, 2011b],[Jakab and Csató, 2012a].*

Learning in case of a robotic agent always needs to be performed on-line and generally the parameters of the learning system are not known. The arrival of data from sensory measurements, contains valuable information about the state-transition dynamics and configuration space of the agent. The aim is at extracting and using this information by building up a representative graph structure and defining new kernel functions on it. In Chapter 5 of the thesis, based on [Jakab, 2011b] and [Jakab and Csató, 2012a] we present the construction of the Markov decision process induced graph structure parallel to the on-line approximation of the state-action value function with a Gaussian process. Nodes of the graph are represented by visited state-action pairs, the addition of new nodes is conditioned on the on-line updates of the Gaussian process. We show that this construction mechanism leads to sparse, connected graphs which represent a crude approximation to the system dynamics.

Let G denote a sparse graph structure induced by the $\mathcal{MDP}$ which we will define as follows: $G(V, E)$ is a sparse graph with vertices V, and edges E. The graph that has n nodes where $n = |BV|$ is the number of basis vectors present in the $\mathcal{GP}$ value function approximator. The connection between these nodes are initialized parallel to the addition of each basis vector to the BV set of the $\mathcal{GP}$.

Using the $\mathcal{GP}$ basis vectors as nodes in our graph construction makes sure that the graph structure remains sparse and the nodes are placed in important regions of the state space. The construction of the $\mathcal{MDP}$ induced graph structure during the learning process proceeds as follows: If x_t is added to the basis vectors, it is also added to the graph and connect it to the existing graph as follows:

$$E_{x_t,x_i} = \begin{cases} \|x_i - x_t\|^2 & \text{if} \quad \exp\left(-\|x_i - x_t\|^2\right) > \gamma \quad \gamma \in [0,1] \\ 0 & \text{otherwise} \end{cases} \quad i = 1 \dots n \qquad (.18)$$

The threshold value γ limits the number of neighbours of a node x_t. Based on the graph structure $G(V, E)$ a new type of kernel function can be built which uses as a distance measure the shortest path between two data-points.

$$k_{sp}(x, x') = A \exp\left(-\frac{\mathcal{SP}(x, x')^2}{2\sigma_{sp}}\right) \qquad (.19)$$

where the amplitude of the Q-function A and σ_{sp} are hyper-parameters to the system (we set these values to 1). The definition of the shortest path exists only between data-points that are present in the $\mathcal{GP}$ basis vector set. In a continuous state-action space visiting the same state-action pair twice has very low probability, therefore we have to define our shortest path measure between two points as the distance between the two basis-vectors that are the closest to the data-points plus the distance of the data-points from these Basis vectors. The problem is that not all inputs are represented in the graph and we employ two methods for shortest path computation

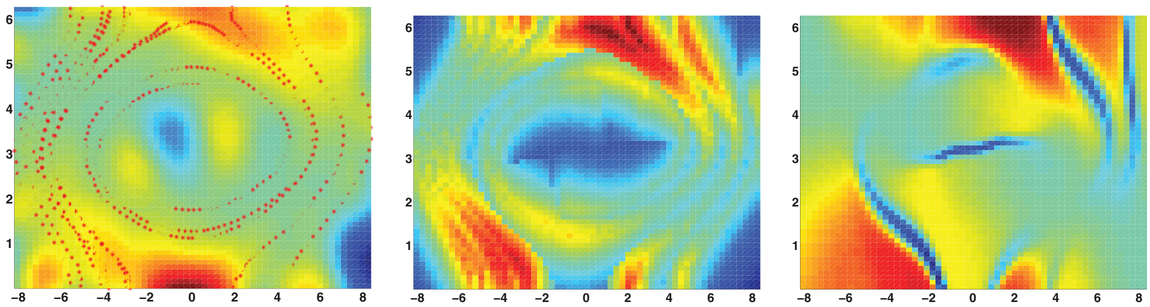


Figure .3: (a) Euclidean distance , (b) minimum shortest path eq. (.21) , (c) interpolated shortest path eq. (.20)

between an input and a basis vector x_j:

$$\mathcal{SP}(x^*, x_j) \overset{(1)}{=} \|x^* - x_i\|^2 + \mathbf{P}_{i,j} \quad \text{where } x_i = \underset{x_\ell \in BV}{\operatorname{argmin}} \|x^* - x_\ell\|^2 \tag{.20}$$

$$\overset{(2)}{=} \mathbf{k}_{x*}^T \mathbf{P} \boldsymbol{e}_j = \sum_{i=1}^{n} k(x^*, x_i)\mathbf{P}_{i,j} \tag{.21}$$

where $\mathbf{P}$ stores the lengths of the shortest paths between basis vectors x_i and x_j, and $\boldsymbol{e}_j$ is the j-th unit vector of length n. The first method uses only the closest node to the new input x^* to obtain the shortest path to x_j, whereas the second performs an averaging over all existing nodes in the BV set. The weighted averaging is necessary in some cases to avoid sudden inconsistencies in the obtained Q-function. As seen in Figure .3, the standard $\mathcal{GP}$ approximation smooths out the value estimates across points. These value functions correspond to a fixed sub-optimal policy on the inverted pendulum control task. The policy was deliberately set up in such a way as to provide close to optimal actions only when the pendulum approaches the target region with a fairly low speed. The resulting discontinuities in the estimated value function are clearly visible on both shortest path approximations, however the interpolated version has a greater generalization potential.

7 Conclusions

The goal of this research was to develop new methods for autonomous learning of control policies in robotic agents within the framework of reinforcement learning . As a result we developed a set of methods for extending $\mathcal{RL}$ algorithms to continuous state-action space robotic control problems and we have shown that different variants of non-parametric function approximation techniques with proper uncertainty treatment can be successfully used for this purpose. We investigated the possibility of using Gaussian processes in reinforcement learning in the role of state and state-action value function approximators in conjunction with both value-based and direct policy search algorithms. Value functions represent the long-term optimality of being in a specific state or state-action pair, and form the basis of many learning algorithms. Although the basic set-up is not new, we consider our approach a novel one: we approximated the value functions corresponding to an action-selection policy with the help of a Gaussian process where for inputs we used state-action pairs and for targets we used discounted cumulative returns obtained along sample trajectories through environment interaction. With the help of the $\mathcal{GP}$ posterior distribution we were able to obtain both point-estimates of the underlying value function and confidence bounds. We extended the original $\mathcal{GP}$ inference algorithm with an on-line version since in practice any robotic control problem requires on-line treatment .

The fully probabilistic model of the value function can be used in different ways to facilitate the learning of action selection policies. First we investigated its usefulness in the family of approximate temporal-difference learning methods where we applied the $\mathcal{GP}$ approximation for Q-learning in order to extend it to continuous state spaces. We exploited the on-line nature of our Gaussian process regression method to perform bootstrapping and avoid inconsistent updates of the value function. Although theoretical convergence could not be guaranteed our experiments show that in practice the algorithm achieves convergence and better performance than its discretization-based counterparts.

We also employed the $\mathcal{GPR}$ approximation in conjunction with the *reinforce* family of policy gradient algorithms to reduce the variance of the estimated gradients. Experiments were performed on simulated robot control tasks. Results show that the $\mathcal{GPR}$ approximation of the action-value function does not lead to worse performance. On the contrary, it provides better policies or faster convergence. We have also introduced a new way of improving sample reuse efficiency and maintaining continuity between gradient update steps in $\mathcal{GPR}$ value function approximation and policy gradient algorithms. We presented a mechanism which makes it possible to restrict the size of the $\mathcal{GP}$ thereby making it computationally less demanding. The sparsification mechanism allowed the development of a method that facilitates the removal of out-of-date basis vectors from the basis vector set also called pruning. To achieve efficient pruning we introduced a measure composed from the Kullback Leibler distance between the original and a constrained $\mathcal{GP}$ and a time dependent term. For evaluation we compared our method with Williams' *reinforce* algorithm which is known to suffer from high gradient variance. Experiments on simulated robotic control tasks show that maintaining continuity in value function approximation leads to better long-term performance and more efficient variance reduction. To further improve the performance of learning algorithms for robotic control we presented two new modalities for adjusting different characteristics of the exploration in policy gradient methods with the help of Gaussian process action-value function approximation. By using these methods the search for an optimal policy can be restricted to certain regions of the state-action space and better performance can be achieved. The presented methods can also be viewed as a transition between off-policy and on-policy learning, which opens up further interesting research directions. There are many $\mathcal{RL}$ learning tasks where the corresponding action-value function is discontinuous in some regions of the space, and this discontinuity has great influence on the algorithms performance. Traditional stationary kernel functions essentially operate on distances between data-points and are invariant to translations in the input space. we presented a modality to increase the accuracy of our $\mathcal{GP}$ action-value function estimates and to achieve a more similar working mechanism to $\mathcal{TD}$ by introducing a new class of kernel functions that operate on a graph structure induced by the Markov decision process underlying the $\mathcal{RL}$ problem. Using sparse on-line Gaussian process regression the nodes and edges of the graph structure are allocated during on-line learning parallel with the inclusion of new measurements to the basis vector set. This results in a more compact and efficient graph structure and more accurate value function estimates. We tested the approximation accuracy on simulated robotic control tasks the pole balancing and the swinging Atwood's machine. Results show that incorporating structural information into the approximation process improves the quality of the value function estimates and reduces their variance.

The algorithms and methods discussed in this work address some of the major problems of achieving robotic control by means of autonomous learning. The whole problem domain, however covers a much larger number of topics, which due to the size of this work could not be treated here. As part of our future work we plan to test our introduced methods on real-life robotic agents, improve their computational efficiency and further investigate possibilities of exploiting the fully probabilistic nature of Gaussian process regression in the context of robotic control policy learning.

8 Publications related to the thesis

Published papers:

1. H. Jakab. Geodesic distance based kernel construction for Gaussian process value function approximation. *KEPT-2011:Knowledge Engineering Principles and Techniques International Conference, Selected Papers.*, 2011c. ISSN 2067-1180

2. H. Jakab. Geodesic distance based kernel construction for Gaussian process value function approximation. *Studia Universitatis Babes-Bolyai Series Informatica*, 61(3):51–57, 2011b. ISSN 1224-869

3. H. Jakab and L. Csató. Improving Gaussian process value function approximation in policy gradient algorithms. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 221–228. Springer, 2011. ISBN 978-3-642-21737-1

4. H. Jakab. Controlling the swinging atwood's machine using reinforcement learning. *Müszaki tudományos füzetek: XVI. FMTÜ international scientific conference*, pages 141–145, 2011a. ISSN 2067 - 6808

5. H. Jakab, B. Bócsi, and L. Csató. Non-parametric value function approximation in robotics. In H. F. Pop, editor, *MACS2010: The 8th Joint Conference on Mathematics and Computer Science*, volume Selected Papers, pages 235–248. Györ:NOVADAT, 2011. ISBN 978-963-9056-38-1

6. H. Jakab. Guided exploration in policy gradient algorithms using Gaussian process function approximation. In *volume of extended abstracts CSCS2010, Conference of PhD Students in Computer Science*, 2010

7. H. Jakab and L. Csató. Using Gaussian processes for variance reduction in policy gradient algorithms. In A. Egri-Nagy, E. Kovács, G. Kovásznai, G. Kusper, and T. Tómács, editors, *ICAI2010: Proceedings of the 8th International Conference on Applied Informatics*, volume 1, pages 55–63, Eger, Hungary, 2010. BVB. ISBN 978-963-989-72-3

8. H. Jakab and L. Csató. Q-learning and policy gradient methods. *Studia Universitatis Babes-Bolyai Series Informatica*, 59:175–179, 2009. ISSN 1224-869

Papers under publication:

- H. Jakab and L. Csató. Manifold-based non-parametric learning of action-value functions. In *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium., 2012a (Accepted on 31.01.2012)
- H. Jakab and L. Csató. Reinforcement learning with guided policy search using Gaussian processes. In *International Joint Conference on Neural Networks (IJCNN)*, 2012b (Accepted on 21.02.2012)

Submitted papers:

- H. Jakab and L. Csató. Guided exploration in direct policy search with Gaussian processes. *Acta Cybernetica*, Under Review, 2011

Bibliography of the thesis

J. S. Albus. A new approach to manipulator control: The cerebellar model articulation controller. *Journal of Dynamic Systems, Measurement, and Control*, pages 220–227, 1975.

A. Antos, R. Munos, and C. Szepesvári. Fitted Q-iteration in continuous action-space mdps. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *NIPS*. MIT Press, 2007.

K. E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, New York, 1978.

J. A. D. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the International Conference on Robotics and Automation 2001*. IEEE, May 2001.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 30–37. Morgan Kaufmann, 1995.

L. Baird and A. Moore. Gradient descent for general reinforcement learning. In *In Advances in Neural Information Processing Systems 11*, pages 968–974. MIT Press, 1998.

A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artif. Intell.*, 72(1-2):81–138, 1995. ISSN 0004-3702.

D. J. Benbrahim H. and F. J. Real-time learning: A ball on a beam. In *Proceedings of the international joint conference on neural networks*, volume Proceedings of the international joint conference on neural networks, pages 92–103, 1992.

D. A. Berry and B. Fristedt. Bandit problems: Sequential allocation of experiments. In *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1985.

D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific,

1996. ISBN 1886529108.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121:49–107, August 2000. ISSN 0004-3702.

J. Boyan and A. Moore. Generalization in reinforcement learning: Safely approximating the value function. In G. Tesauro, D. Touretzky, and T. Lee, editors, *Neural Information Processing Systems 7*, pages 369–376, Cambridge, MA, 1995. The MIT Press.

J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2-3):233–246, 2002.

S. J. Bradtke, A. G. Barto, and P. Kaelbling. Linear least-squares algorithms for temporal difference learning. In *Machine Learning*, pages 22–33, 1996.

P. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1):24–32, Mar. 1996.

L. Csató. *Gaussian Processes – Iterative Sparse Approximation*. PhD thesis, Neural Computing Research Group, March 2002.

L. Csató and M. Opper. Sparse representation for Gaussian process models. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *NIPS*, volume 13, pages 444–450. The MIT Press, 2001.

L. Csató and M. Opper. Sparse on-line Gaussian Processes. *Neural Computation*, 14 (3):641–669, 2002.

L. Csató, E. Fokoué, M. Opper, B. Schottky, and O. Winther. Efficient approaches to Gaussian process classification. In *NIPS*, volume 12, pages 251–257. The MIT Press, 2000.

E. E. Dar and Y. Mansour. Learning rates for q-learning. *Journal of Machine Learning Research*, 5:1–25, 2003.

R. Dearden, N. Friedman, and S. Russell. Bayesian q-learning. In *In AAAI/IAAI*, pages 761–768. AAAI Press, 1998.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, June 2011.

M. P. Deisenroth, C. E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009. ISSN 0925-2312. doi:

http://dx.doi.org/10.1016/j.neucom.2008.12.019.

G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng. Learning cpg sensory feedback with policy gradient for biped locomotion for a full-body humanoid. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, pages 1267–1273. AAAI Press, 2005. ISBN 1-57735-236-x.

Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: The Gaussian process approach to temporal difference learning. In *Proc. of the 20th International Conference on Machine Learning*, pages 154–161, 2003.

Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 201–208, New York, NY, USA, 2005. ACM. doi: http://doi.acm.org/10.1145/1102351.1102377.

Y. Engel, P. Szabo, and D. Volkinshtein. Learning to control an octopus arm with gaussian process temporal difference methods. *Advances in Neural Information Processing Systems*, 14:347–354, 2006.

D. Ernst, P. Geurts, L. Wehenkel, and L. Littman. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

J. Frey. Introduction to stochastic search and optimization: Estimation, simulation, and control. james c. spall. *Journal of the American Statistical Association*, 99:1204–1205, 2004.

C. Gearhart. Genetic programming as policy search in markov decision processes. *Genetic Algorithms and Genetic Programming at Stanford*, page 61?67, 2003.

M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *NIPS '07: Advances in Neural Information Processing Systems 19*, pages 457–464, Cambridge, MA, 2007. MIT Press.

G. Gordon. Stable function approximation in dynamic programming. In *Proceedings of IMCL '95*, 1995.

E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *J. Mach. Learn. Res.*, 5:1471–1530, December 2004. ISSN 1532-4435.

H. Hachiya, T. Akiyama, M. Sugiayma, and J. Peters. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Netw.*, 22:1399–1410, December 2009. ISSN 0893-6080. doi: 10.1016/j.neunet.2009.01.002.

K. Harbick, J. F. Montgomery, and G. S. Sukhatme. Planar spline trajectory following for an autonomous helicopter. In *in IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 237–242, 2001.

M. E. Harmon and S. S. Harmon. Reinforcement learning: A tutorial., 1997.

P. Hennig. Optimal reinforcement learning for gaussian systems. Technical Report arXiv:1106.0800, Max Planck Institute for Intelligent Systems Department of Empirical Inference, Spemannstrasse 38, 72070 Tübingen, Germany, Jun 2011.

T. Herbert. *Modeling and Control of Robot Manipulators, Lorenzo Sciavicco and BrunoSiciliano*, volume 21. Kluwer Academic Publishers, Hingham, MA, USA, January 1998. doi: 10.1023/A:1007979428654.

G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita. Evolving robust gaits with aibo, 2000.

A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems 15*, pages 1547–1554. MIT Press, 2002a.

J. A. Ijspeert, J. Nakanishi, and S. Schaal. movement imitation with nonlinear dynamical systems in humanoid robots. In *international conference on robotics and automation (icra2002)*, 2002b.

H. Jakab. A frame based motion system for the aibo four legged robotic agent. Master's thesis, Computer Science Department, Babes-Bolyai University, 2008.

H. Jakab. Guided exploration in policy gradient algorithms using Gaussian process function approximation. In *volume of extended abstracts CSCS2010, Conference of PhD Students in Computer Science*, 2010.

H. Jakab. Controlling the swinging atwood's machine using reinforcement learning. *Müszaki tudományos füzetek: XVI. FMTÜ international scientific conference*, pages 141–145, 2011a. ISSN 2067 - 6808.

H. Jakab. Geodesic distance based kernel construction for Gaussian process value function approximation. *Studia Universitatis Babes-Bolyai Series Informatica*, 61(3): 51–57, 2011b. ISSN 1224-869.

H. Jakab. Geodesic distance based kernel construction for Gaussian process value function approximation. *KEPT-2011:Knowledge Engineering Principles and Techniques International Conference, Selected Papers.*, 2011c. ISSN 2067-1180.

H. Jakab and L. Csató. Q-learning and policy gradient methods. *Studia Universitatis Babes-Bolyai Series Informatica*, 59:175–179, 2009. ISSN 1224-869.

H. Jakab and L. Csató. Using Gaussian processes for variance reduction in policy gradient algorithms. In A. Egri-Nagy, E. Kovács, G. Kovásznai, G. Kusper, and T. Tómács, editors, *ICAI2010: Proceedings of the 8th International Conference on Applied Informatics*, volume 1, pages 55–63, Eger, Hungary, 2010. BVB. ISBN 978-963-989-72-3.

H. Jakab and L. Csató. Guided exploration in direct policy search with Gaussian processes. *Acta Cybernetica*, Under Review, 2011.

H. Jakab and L. Csató. Improving Gaussian process value function approximation in policy gradient algorithms. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 221–228. Springer, 2011. ISBN 978-3-642-21737-1.

H. Jakab and L. Csató. Manifold-based non-parametric learning of action-value functions. In *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium., 2012a.

H. Jakab and L. Csató. Reinforcement learning with guided policy search using Gaussian processes. In *International Joint Conference on Neural Networks (IJCNN)*, 2012b.

H. Jakab, B. Bócsi, and L. Csató. Non-parametric value function approximation in robotics. In H. F. Pop, editor, *MACS2010: The 8th Joint Conference on Mathematics and Computer Science*, volume Selected Papers, pages 235–248. Györ:NOVADAT, 2011. ISBN 978-963-9056-38-1.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

A. Kanarachos, M. Sfantsikopoulos, and P. Vionis. A splines?based control method for robot manipulators. *Robotica*, 7(03):213–221, 1989. doi: 10.1017/S026357470000607X.

M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727, 1999.

M. S. Kim and W. Uther. Automatic gait optimisation for quadruped robots. In *In Australasian Conference on Robotics and Automation*, 2003.

H. Kimura and S. Kobayashi. Reinforcement learning for continuous action using stochastic gradient ascent. *Intelligent Autonomous Systems (IAS-5)*, pages 288–295, 1998.

N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *in Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2619–2624, 2004.

K. J. Kyriakopoulos and G. N. Saridis. Minimum jerk for trajectory planning and control. *Robotica*, 12:109–113, 1994.

M. G. Lagoudakis and R. Parr. Model-free least squares policy iteration. Technical report, Advances in Neural Information Processing Systems, 2001.

M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *J. Mach. Learn. Res.*, 4: 1107–1149, December 2003. ISSN 1532-4435.

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules*1. *Advances in Applied Mathematics*, 6:4–22, 1985. doi: 10.1016/0196-8858(85)90002-8.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.

H. Maei, C. Szepesvari, S. Bhatnagar, D. Precup, D. Silver, and R. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. *Advances in Neural Information Processing Systems NIPS*, 22:1204–1212, 2009.

H. Miyamoto, F. Gandolfo, H. Gomi, S. Schaal, Y. Koike, O. Rieka, E. Nakano, Y. Wada, and M. Kawato. a kendama learning robot based on a dynamic optimization principle. In *preceedings of the international conference on neural information processing*, pages 938–942, 1996a.

H. Miyamoto, S. Schaal, F. Gandolfo, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. a kendama learning robot based on bi-directional theory. *Neural Networks*, 8(8):1281–1302, 1996b.

A. Moore. Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces. In L. Birnbaum and G. Collins, editors, *Machine Learning: Proceedings of the Eighth International Conference*, 340 Pine Street, 6th Fl., San Francisco, CA 94104, June 1991. Morgan Kaufmann.

A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Mach. Learn.*, 13(1):103–130, 1993. ISSN 0885-6125. doi: http://dx.doi.org/10.1023/A:1022635613229.

A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Mach. Learn.*, 21:199–233, December 1995. ISSN 0885-6125. doi: 10.1023/A:1022656217772.

D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:241–276, 1999.

J. Morimoto and K. Doya. Robust reinforcement learning. *Neural Comput.*, 17:335–359, February 2005. ISSN 0899-7667. doi: 10.1162/0899766053011528.

Y. Nakamura, T. Mori, and S. Ishii. Natural policy gradient reinforcement learning for a cpg control of a biped robot. In *PPSN*, pages 972–981, 2004.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996. ISBN 0387947248.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Press, 2010.

P. Olivier, P. J.P., S. C., S. S., and W. J.A. Swinging atwood's machine: Experimental and numerical results, and a theoretical study. *Physica D*, 239:1067–1081, 2010.

A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.

J. Peng and R. J. Williams. Efficient learning and planning within the dyna framework. *Adapt. Behav.*, 1(4):437–454, 1993. ISSN 1059-7123. doi: http://dx.doi.org/10.1177/105971239300100403.

J. Peters and S. Schaal. Policy gradient methods for robotics. In *IROS*, pages 2219–2225. IEEE, 2006.

J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 745–750, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008a.

J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71:1180–1190, March 2008b. ISSN 0925-2312.

D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th International Conference on Machine Learning*, pages 417–424, 2001.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.

C. E. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In L. K. S. Thrun, S. and B. Schölkopf, editors, *NIPS 2003*, pages 751–759. MIT Press, 2004.

C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

M. Riedmiller. Neural fitted q iteration ? first experiences with a data efficient neural reinforcement learning method. In *In 16th European Conference on Machine Learning*, pages 317–328. Springer, 2005.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

T. Rückstieß, M. Felder, and J. Schmidhuber. State-dependent exploration for policy gradient methods. In *ECML PKDD '08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, pages 234–249, Berlin, Heidelberg, 2008. Springer-Verlag.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

A. Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *ICML*, pages 298–305, 1993.

D. F. Sebastian Thrun, Wolfram Burgard. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005.

M. R. Shaker, S. Yue, and T. Duckett. Vision-based reinforcement learning using approximate policy iteration. In *14th International Conference on Advanced Robotics (ICAR)*, 2009.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

R. Smith. Open dynamics engine v0.5 user guide, 2006.

M. Sugiyama, H. Hachiya, C. Towell, and S. Vijayakumar. Geodesic gaussian kernels for value function approximation. *Auton. Robots*, 25:287–304, October 2008. ISSN 0929-5593.

M. Sugiyama, H. Hachiya, H. Kashima, and T. Morimura. Least absolute policy iteration for robust value function approximation. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, pages 699–704, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

R. S. Sutton. First results with dyna, an integrated architecture for learning, planning and reacting. *Neural networks for control*, pages 179–189, 1990.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *NIPS*, pages 1057–1063, 1999.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1.

C. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2011.

I. Szita and A. Lörincz. Optimistic initialization and greediness lead to polynomial time learning in factored mdps. In *ICML*, page 126, 2009.

J. N. Tsitsiklis and B. V. Roy. An analysis of temporal-difference learning with function approximation. Technical report, IEEE Transactions on Automatic Control, 1997.

C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, London, 1989.

C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber. Solving deep memory pomdps with recurrent policy gradients. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2007.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520, 1996.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

X. Xu, T. Xie, D. Hu, and X. Lu. Kernel least-squares temporal difference learning. *International Journal of Information Technology*, pages 55–63, 2005.

P. Zhang, X. Xu, C. Liu, and Q. Yuan. Reinforcement learning control of a real mobile robot using approximate policy iteration. In *Proceedings of the 6th International Symposium on Neural Networks: Advances in Neural Networks - Part III*, ISNN 2009, pages 278–288, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-01512-0.