DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch091

Self-Organisation of Generic Policies in Reinforcement Learning

Simón C. Smith and J. Michael Herrmann
Institute of Perception, Action and Behaviour, School of Informatics
The University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, U.K.
artificialsimon@ed.ac.uk, michael.herrmann@ed.ac.uk

Abstract

We propose the use of an exploratory self-organised policy to initialise the parameters of the function approximation in the reinforcement learning policy based on the value function of the exploratory probe in a low-dimensional task. For a high-dimensional problems we exploit the property of the exploratory behaviour to establish a coordination among the degrees of freedom of a robot without any explicit knowledge of the configuration of the robot or the environment. The approach is illustrated by a learning tasks in a six-legged robot. Results show that the initialisation based on the exploratory value function improve the learning speed in the low-dimensional task and that some correlation towards a higher reward can be acquired in the high-dimensional task.

Introduction

Reinforcement learning aims at solving dynamical optimisation problems which may be formulated in terms of discrete or continuous variables (Sutton, 1988; Sutton and Barto, 1998; Doya, 2000). It is based on an utility function and/or the construction of a control policy such that optimal performance can be reached asymptotically under certain conditions. Particularly, in continuous time and space the use of function approximation is imperative to match the complexity of the problem. Various techniques have been proposed in order to approximate the relevant functions, e.g. kernel-based methods (Xu et al., 2007; He et al., 2011), normalised Gaussian networks (Sato and Ishii, 2000; Doya, 2000), Fourier basis function (Konidaris et al., 2008) and echo state networks (Jaeger, 2001; Szita et al., 2006).

The initialisation of the parameters of the function approximator becomes non-trivial when — as in most robotic tasks — a high learning speed is required. Ideally, the initialisation should be such that the learning trajectory can follow the gradient without being trapped in undesired optima. Often, the approximator is initialised by small random values, which in some cases aids the exploration of the state and action space. Other approaches assign optimistic value to every position of the state space which also provides an initial incentive to explore until values in a more realistic range are found. Often, however, these value will not be close to the true expected future rewards, because the optimistic values decay linearly while sufficiently exploration takes usually longer, such that, at least in the more complex problems, more flexible exploration strategies are worth being considered.

We propose a self-organising exploration mode that will discover coherent behaviour in a robot in an autonomous learning stage before reward signals are used or are available. This behaviour will be used here to pre-shape the parametric representation of the policy in an actor-critic reinforcement learning scheme. The exploration method produces an on-policy estimate of the value function, such that its value can be a good indicator of how well the robot will perform in a specific task later if only the promising actions of the policy are used. The exploration will in particular introduce a bias that can reduce the complexity of the problem by using information that was inexpensively obtained earlier. In our robotic application this means that the robot preferentially guided to regions in the state space where controllability and predictability of the dynamics is high.

Nevertheless, function approximation does not easily generalise to high dimensions unless independence or hierarchical structures can be assumed. In robotic problems as well as in biological examples, however, such assumptions are rarely justified, i.e. often the exploitable structure is not explicitly known. As the main contribution of this study, we propose a combination of homeokinetic and reinforcement learning which uses for high-dimensional reinforcement learning tasks a combination of autonomous exploration with a reward-weighted extraction of information.

In the case that the exploitable structure is known in advance, a similar effect has been shown before (Martius and Herrmann, 2011). In this study, it was shown for a track-like robot (“armband”) that the learning time can decrease even for an increase of the mechanical complexity of the robot if the complexity of the control problem was relatively low. The reason for this observation was in addition to the built-in interaction structure that the robot was less likely to self-obstruct in the high-dimensional case. The speed-up saturated at a few tens of dimensions and the remaining learning time was low due to the homogeneity of the robot's configuration. Here we will study a more complex problem, namely a hexapod with twelve degrees of freedom which require a measure of coordination for ambulation or navigation.

The ambivalence of training and self-organisation reflects an important principle in biological learning. Although in many cases the external event distribution is sufficient to drive learning successfully, there are often intrinsic mechanisms available as a more or less equally successful fall-back option which the organism can rely on when the environment deviates from the evolutionarily anticipated standard. We will not discuss, how the organisms deal with the unfortunate latter case, but with the potential benefits of a preparation before environmental reward signals are available or while they are not yet critical such as in play in a protected environment. In addition to this consideration and particularly in robotic applications, the prior-learning scheme can add naturalness to the movements and simplify the search space when the purposeful movements are to be learned subsequently.

The early-learning algorithm relies on a self-organising control paradigm (Martius et al., 2007). This controller creates coherent exploratory behaviour by maximising the predictability of the robot action at the same time that it tries to maximise the sensitivity of each motor command. In order to propagate the best actions from the exploratory mode to reinforcement learning, we let the value function to be learned by the critic while the actor is fixed to the exploratory policy. The on-policy property of the actor-critic algorithm, i.e. the fact that the value function is learned based on the actual policy, makes this method suitable to asses the performance of the exploratory regime. In a first low-dimensional experiment the exploration policy is propagated directly to the actor's policy when the value function is positive. For negative values of the value function we propagate the opposite actions. In reinforcement learning the value function invokes the beneficial actions but it gives us little information about where to explore next if its value is not sufficient for the task. Another experiment is realised with a high-dimensional case. In order to overcome the curse of dimensionality, a closed-loop controller is learned which can function similar to a central pattern generator (CPG), where its coordination factors are shaped following the instantaneous reward.

We present a comparison of our approach with a standard version of continuous reinforcement learning (Doya, 2000) in low-dimensionality, whilst in high-dimensionality the direct reward is used to propagate the correlation between the degrees of freedom. The reward of the tasks is the horizontal speed of a six-legged robot.

Reinforcement learning in continuous domains

For continuous reinforcement learning (Doya, 2000), we will have to adjust the weights $\boldsymbol{w}^A$ that determine the output $\boldsymbol{u}$ of a controller U which is given by

$$\boldsymbol{u}_t = U_t\left(\boldsymbol{x}_t\right) = s\left(A\left(\boldsymbol{x}_t;\boldsymbol{w}^A\right) + \sigma\boldsymbol{n}_t\right), \tag{1}$$

where s is usually a sigmoidal or an identity output function, $\boldsymbol{n}$ is a probing input signal of strength σ and

$$A\left(\boldsymbol{x}_t;\boldsymbol{w}^A\right) = \frac{1}{N(\boldsymbol{x}_t)}\sum_i w_i^A \exp\left(-\frac{\|\boldsymbol{x}_t-\boldsymbol{\mu}_i\|^2}{2\boldsymbol{\rho}_i}\right) \tag{2}$$

represents the approximator function with parameter $\boldsymbol{w}^A$ of the actor's policy. The values of $\boldsymbol{\rho}_i$ and $\boldsymbol{\mu}_i$ represent the size and centre of a basis function, here are assumed to be fixed. The factor $N\left(\boldsymbol{x}_t\right) = \sum_i \exp\left(-\frac{\|\boldsymbol{x}_t-\boldsymbol{\mu}_i\|^2}{2\boldsymbol{\rho}_i}\right)$ normalises the output. The parameters $\boldsymbol{w}^A$ are updated according to

$$\dot{w_i}^A = \varepsilon_A\delta_t\boldsymbol{n}_t\frac{\partial A\left(\boldsymbol{x}_t;\boldsymbol{w}^A\right)}{\partial w_i^A}, \tag{3}$$

where ε_A is the actor's learning rate. The last term in Eq. 3 can be obtained directly from the explicit form of the policy in Eq. 2. The essential part of the learning rule includes the correlation of the probing input $\boldsymbol{n}$ and the delta error,

$$\delta_t = r_t - \frac{1}{\tau}V_t + \dot{V}_t, \tag{4}$$

where r_t is the instant reward at time t, τ is the time constant for discounting future rewards and the utility function V is approximated by another parametrised function which is updated based on the approximation of the critic by the relation

$$\dot{V}_t \cong \left(V_t - V_{t-\Delta t}\right)/\Delta t$$

which can be obtained from Eq. 4. The update of the parameter w_i^V of V follows the gradient descent with respect to δ,

$$\dot{w}_i^V = \varepsilon_V\delta\left(t\right)\frac{\partial V\left(\boldsymbol{x}_{t-\Delta t};\boldsymbol{w}^V\right)}{\partial w_i}, \tag{5}$$

with ε_V learning rate.

The alternatives for choosing the probing signal of the robot control in Eq. 1 range from the use of noise (Gullapalli, 1990) to high-frequency oscillatory modulations of the motor command (Wiener, 1948). Our experiments (Smith and Herrmann, 2012) confirm that the type of the probe does not matter in low-dimensional problems. The dynamics of the correlation among the degrees of freedom of the controlled system becomes crucial for robots with many degrees of freedom, such that the choice of the probing stimulus becomes non-trivial. In high-dimensional problems it is not possible to test all actions in all states infinitely often as it would be required in discrete reinforcement learning algorithms. Also for continuous algorithms orienting the exploration to promising directions is essential. We propose to use an approach in the present context that has previously developed in a different setting (Martius, 2010).

Learning in motor space

As exploration signal we propose the exploratory controller

$$\boldsymbol{y}_t = K\left(\boldsymbol{x}_t\right) = g\left(C\boldsymbol{x}_t + \boldsymbol{c}\right). \qquad (6)$$

This controller receives the current sensory input vector $\boldsymbol{x}_t \in \mathbb{R}^n$ and determines the direction of exploration in dependence on the multidimensional parameters $C \in \mathbb{R}^{m\times n}$ and $\boldsymbol{c} \in \mathbb{R}^m$ and the nonlinear function g, where $\boldsymbol{y}_t \in \mathbb{R}^m$. In order to adapt the parameters C and $\boldsymbol{c}$, the new sensory inputs are compared with a prediction $\hat{\boldsymbol{x}_t} \in \mathbb{R}^n$ by a world model M based on previous inputs or outputs. For simplicity, we use a linear predictor that uses only the motor commands from Eq. 6 and receives thus information about previous inputs only indirectly,

$$\hat{\boldsymbol{x}}_{t+1} = M\left(\boldsymbol{y}_t\right) = D\boldsymbol{y}_t + \boldsymbol{d}, \qquad (7)$$

where $D \in \mathbb{R}^{n\times m}$ and $\boldsymbol{d} \in \mathbb{R}^n$.

The comparison of the corresponding sensory input $\boldsymbol{x}_{t+1}$ and its estimate by the internal model $\hat{\boldsymbol{x}}_{t+1}$ results in the prediction error $\boldsymbol{\xi}_{t+1} = \hat{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_{t+1}$ which is a vector in the perceptual space where $\boldsymbol{\xi}_t \in \mathbb{R}^n$.

In order to formulate a learning rule for the exploratory controller of Eq. 6, we will follow the procedure in (Martius, 2010) and express the error in the motor space which can be achieved by defining a transformed error $\boldsymbol{\eta}_t \in \mathbb{R}^m$ via

$$M\left(\boldsymbol{y}_t\right) + \boldsymbol{\xi}_{t+1} = M\left(\boldsymbol{y}_t + \boldsymbol{\eta}_t\right). \qquad (8)$$

Because $M\left(\boldsymbol{y}_t\right) + \boldsymbol{\xi}_{t+1} = \boldsymbol{x}_{t+1}$, the motor error $\boldsymbol{\eta}_t$ can be interpreted as the control correction required to compensate the inaccuracy of the model M. The vector $\boldsymbol{\eta}_t$ is a retrospective error that can be determined only after the event of receiving the new stimulus $\boldsymbol{x}_{t+1}$. Nevertheless, minimisation of $\boldsymbol{\eta}$ is a relevant goal for the adaptation of the system. The definition in Eq. 8 is implicit and may be empty which calls for the use of a regularised inverse of M to explicitly obtain an approximation of $\boldsymbol{\eta}$. Practically, Eq. 8 is transformed into a motor level error exploiting the assumed linearity of the model in Eq. 7,

$$\boldsymbol{\eta}_t = M'^{+}\boldsymbol{\xi}_{t+1}, \qquad (9)$$

where M'^{+} is the pseudo-inverse of the derivative of the model in Eq. 7, i.e. the pseudoinverse of D. In analogy to (Der et al., 2002) this defines a homeokinetic error function in the motor space

$$E_t = \boldsymbol{\eta}_t^\top \left(J_t J_t^\top\right)^{-1} \boldsymbol{\eta}_t \qquad (10)$$

where J is the Jacobian of the sensorimotor loop, see below. We are going to perform a gradient descent with respect to this error function in order to adapt the parameters of the controller defined in Eq. 6.

To calculate the Jacobian, we use the derivatives $M'_y = D$ and $K'_x = g' \circ C$, with $\circ$ defined as element-wise multiplication, such that we find from $J_t = g'_t \circ C_t D_t = g'_t \circ R_t$, with $R_t \in \mathbb{R}^{m\times n}$ and $R_t = C_t D_t$. This gives rise to the following formulation of the shift $\boldsymbol{\nu}$, i.e. the change in motor command that would have been required to correctly predict the following motor command, namely

$$\boldsymbol{\nu}_{t-1} = J_t^{-1}\boldsymbol{\eta}_t.$$

While the interpretation of $\boldsymbol{\eta}$ (Eq. 9) as retrospective error connects sensor and motor space, we have here a connection between the two points in time within the motor space that reflects the dynamical properties of the full sensorimotor loop. The error function in Eq. 10 becomes thus simply

$$E_t = \boldsymbol{\nu}_{t-1}^\top \boldsymbol{\nu}_{t-1}$$

which lead to a convenient update rule of the controller matrix C. Omitting the time indices we find

$$\begin{aligned}\frac{1}{\varepsilon_C}\Delta C = -\frac{\partial E}{\partial C} &= -2\boldsymbol{\nu}^\top \frac{\partial \boldsymbol{\nu}}{\partial C} \\ &= 2\boldsymbol{\nu}^\top J^{-1}\frac{\partial J}{\partial C} J^{-1}\eta - 2\boldsymbol{\nu}^\top J^{-1}\frac{\partial \boldsymbol{\eta}}{\partial C}\end{aligned}$$

using the rule $\frac{\partial Y^{-1}}{\partial X} = -Y^{-1}\frac{\partial Y}{\partial X}Y^{-1}$. The derivative $\frac{\partial \boldsymbol{\eta}}{\partial C}$ cannot be determined, because we have no information of the dependence of the prediction error on the controller parameters, therefore we set $\frac{\partial \boldsymbol{\eta}}{\partial C} = 0$ and are left with

$$\frac{1}{\varepsilon_C}\Delta C = 2\boldsymbol{\nu}^\top J^{-1}\frac{\partial J}{\partial C} J^{-1}\boldsymbol{\eta} = 2\boldsymbol{\nu}^\top J^{-1}\frac{\partial J}{\partial C}\boldsymbol{\nu}$$

where

$$\frac{\partial J_t}{\partial C} = \frac{\partial}{\partial C}\left(\frac{\partial D}{\partial \boldsymbol{x}} + g'_t \circ C_t\right) D_t.$$

We may ignore the effect of the controller on the sensitivity of the actor in the reinforcement learning component, i.e. set $\frac{\partial}{\partial C}\frac{\partial D}{\partial \boldsymbol{x}} = 0$. We may also assume that the details of the actor are not specified by the reward but will follow essentially the homeokinetic control. In this case the term $\frac{\partial}{\partial C}\frac{\partial D}{\partial \boldsymbol{x}}$ is parallel to the remainder and the resulting numerical factor can be absorbed into the learning rate. We have thus arrived at essentially the same learning rule as in (Martius, 2010),

$$\frac{1}{\varepsilon_C}\Delta C = \boldsymbol{\chi}\left(D\boldsymbol{\nu}\right)^\top - \boldsymbol{\chi}^\top\frac{\partial g'^{-1} \circ \boldsymbol{\eta}}{\partial C},$$

where ε_C is a learning rate and $\boldsymbol{\chi} \in \mathbb{R}^m$ as $\boldsymbol{\chi} = R^{-1^\top}\boldsymbol{\nu}$.

Inserting the correct time indexes we obtain

$$\begin{aligned}\frac{1}{\varepsilon_C}\Delta C_t = {} & \boldsymbol{\chi}_{t-1}(D_t\boldsymbol{\nu}_{t-2})^\top \\ & - 2(\boldsymbol{\chi}_{t-1} \circ g_{t-2} \circ (g'_{t-2})^{-1} \circ \boldsymbol{\eta}_{t-1})\boldsymbol{x}_{t-2}^\top, \qquad (11)\end{aligned}$$

with $\boldsymbol{\chi}_{t-1} = \left(R_t^\top\right)^{-1}\boldsymbol{\nu}_{t-2}$. The update rule for $\boldsymbol{c}$ can be found similarly,

$$\frac{1}{\varepsilon_C}\Delta\boldsymbol{c}_t = -2\left(\boldsymbol{\chi}_{t-1}\circ g_{t-2}\circ\left(g'_{t-2}\right)^{-1}\circ\boldsymbol{\eta}_{t-1}\right). \qquad (12)$$

Direct learning of the actor

The exploratory controller presented above can provide a variety of coherent behaviours based solely on the interaction of the agent and its environment. We propose that such behaviours can be used to shape the action space for an actor-critic reinforcement learning problem by shaping the structure of the search space.

Initially, the agent is controlled by the homeokinetic controller (Eqs. 6, 11 and 12) giving the motor signal to the agent and also shaping the action space. Following homeokinetic motor command in Eq. 6, the implementation of the actor in reinforcement learning shown in Eq. 2 with learning rule from Eq. 3, the difference between the actual motor command and the actor's output $\boldsymbol{e}_t^A \in \mathbb{R}^m$, $\boldsymbol{e}_t^A = \boldsymbol{y}_t - A(\boldsymbol{x}_t;\boldsymbol{w}^A)$, gives rise to the objective function, $E_t = \frac{1}{2} \parallel \boldsymbol{e}_t^A \parallel^2$. The weights $\boldsymbol{w}^A$ of the actor's approximator are updated with a gradient descent algorithm, $\dot{w}_i^A = -\varepsilon_H \frac{\partial E}{\partial w_i}$ with ε_H the learning rate.

After the adaptation, the function approximator has stabilised, the reinforcement learning algorithm is activated and the policy calculated from the actor following Eq. 1, and the actor's parameters are updated based on Eq. 3 using random noise as probing signal.

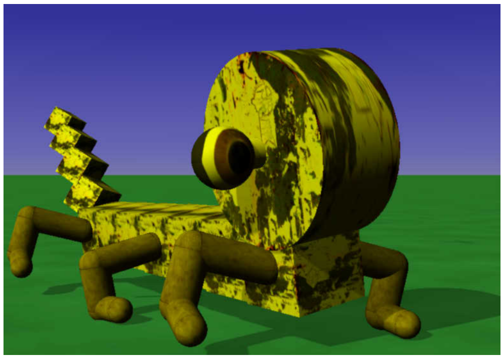

Figure 1: Simulated robot in the lpzrobots simulation environment. The design of the robot is inspired by M. C. Escher's lithograph Wentelteefje (1951).

Self-organisation for parameters initialisation

In order to test the described approach, we will study a low-dimensional control problem for a simulated six-legged robot, see Fig. 1. Switching from the initialisation mode to reinforcement learning is triggered by the amplitude of the error in the approximation which is required to be below a threshold for a certain time. The propagated values from the self-organised policy to the initialisation of the actor's policy is directly translated when the value function has a positive value as this part of the policy already shows the suitability to perform the task. When the value function is negative we propagate the opposite values of the self-organised policy. The justification for this is that the reward only tells what is a beneficial (we want to propagate) and what is not (we only know that this behaviour should not be propagated), since we have all the action state to choose from (except from the actual not beneficial behaviour) we assume that the opposite of the actual command is a better guess than a random action. This will carry the coherence found by the probing but in the opposite direction in our servo motor robot. To illustrate this point we present a toy example where the reward is directly related to the y-axis position of one leg of the robot. In Fig. 2a, the random initialisation of the reinforcement learning policy can be seen, in Fig. 2b, the shape of the exploration signal, in the Fig. 2c, the value function of derived from the exploration policy and in Fig. 2d, the propagated values from the exploration signal to the initial conditions of the actor's policy. The values of the exploration signal that have positive (blue) value function are propagated directly while the exploration signal with negative (red) value function is inverse propagated.

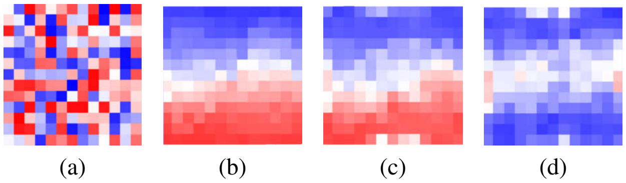

Figure 2: Shape of the approximation of the actor. The x-axis of the position of the leg is represented in the horizontal axis, the y-axis is represented in the vertical axis. For (a), (b) and (d) the colour represents a motor command with blue values closer to 1 and red values closer to -1; for (c) the colour represents the value function with same range. Figure (a) represents the random initialisation, (b) is learned from the homeokinetic controller, (c) is the value function and (d) the initialised actor's policy based on the exploratory signal following the value function. It can be seen how the propagated values from (b) to (d) depend on (c).

In the low-dimensional set-up the task of the robot will be to walk forwards as fast as possible and the rewards will be directly proportional to the absolute value of the velocity of the centre of mass of the robot. A virtual leg is trained and will form a CPG whose motor signal is transmitted to the rest of the limbs either as an in-phase or as and anti-phase signal. While the random initialisation in of the two degrees of freedom may lead to local minima or slow convergence, a smoother function is brought about due to the training with the homeokinetic controller, This may allow for a faster learning once the information about the task is available by the reward signal.

The results shown in Fig. 3 demonstrate that the homeokinetic learning indeed improves the performance in the learning task.

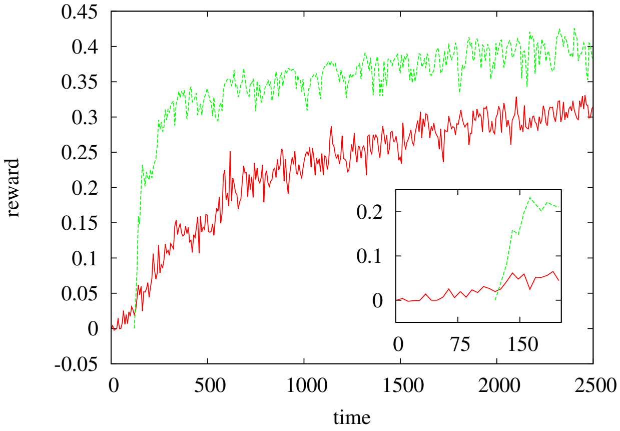

Figure 3: Results of reinforcement learning with random initialisation (red continuous line) of the parameters and with parameters shaped by homeokinetic controller (green dashed line). The six-legged robot receives reward based on horizontal speed. For the green dashed curve the first 600 seconds (shown as the first 120 point of averaged speed over 5 seconds) are used to pre-train the robot, i.e. no reward is available, which can be seen in detail in the inset. With the homeokinetic pre-training, the time to achieve the highest velocity achievable by reinforcement learning is significantly decreased. If the two systems continue to learn for much longer times, our experiments show that both arrive at a very similar reward level.

The basis for the comparison is the reinforcement learning initialised with random weights in the function approximator. An increment in the learning speed can be noticed as a result of the exploratory learning due to homeokinetic control.

Reward-weighted correlation

A more flexible method will be discussed in the following as a generalisation of the previous approach. At the same time we generalise the variants in (Martius and Herrmann, 2012) by including the reward signal in the extraction of the interaction structure. We consider the correlation between sensory inputs and motor commands,

$$W_{ij} = \frac{\langle (\boldsymbol{x}_{i,t} - \langle \boldsymbol{x}_i \rangle)(\boldsymbol{y}_{j,t} - \langle \boldsymbol{y}_j \rangle) \rangle}{\sqrt{\langle (\boldsymbol{x}_{i,t} - \langle \boldsymbol{x}_i \rangle)^2 \rangle \langle (\boldsymbol{y}_{j,t} - \langle \boldsymbol{y}_j \rangle)^2 \rangle}}, \tag{13}$$

where $\langle \cdot \rangle$ denotes a sliding temporal average with time constant τ_W. Eq. 14 can be transformed into a reward-related quantity by an appropriate weighting based on the rectified reward signal $r^{[+]}$.

$$W^{r^{[+]}}_{ij,t} = \frac{\langle r^{[+]}_t (\boldsymbol{x}_{i,t} - \langle \boldsymbol{x}_i \rangle)(\boldsymbol{y}_{j,t} - \langle \boldsymbol{y}_j \rangle) \rangle}{\sqrt{\langle (r^{[+]}_t - \langle r^{[+]} \rangle)^2 \rangle \langle (\boldsymbol{x}_{i,t} - \langle \boldsymbol{x}_i \rangle)^2 \rangle \langle (\boldsymbol{y}_{j,t} - \langle \boldsymbol{y}_j \rangle)^2 \rangle}}, \tag{14}$$

As before the reward signal r is determined by the forward speed of the centre of mass of the robot. The factor $r^{[+]}$ equals r for positive forward speed and is zero if the robot is actually moving backwards. In this way only those sensorimotor couplings that directly contribute to the reward enter the average. The control weights are a smoothed version of the result of Eq. 15.

$$\bar{W}_{ij,t+1} = \varepsilon_W W^{r^{[+]}}_{ij,t} + (1 - \varepsilon_W)\bar{W}_{ij,t}. \tag{15}$$

where $\varepsilon_W < 1$ is the adaptation rate.

Learning gait patterns in a hexapod

In the high-dimensional task, instead of learning with a classic reinforcement learning approach we try to discover the correlation between the different degrees of freedom based on the instantaneous reward. The exploration is produced by the homeokinetic controller and the learning rule is based on Eq. 16. In the following experiment we compare the results obtained from the behaviour of the robot for differently obtained controller matrices in a close-loop setting. Closed-loop feedback control is realised by a controller output related to the current input via $\mathbf{y}_t = H\mathbf{x}_t$, where $\mathbf{x}_t$ and $\mathbf{y}_t$ are the input sensors and output motor commands vectors respectively. In the first case served as a baseline, the matrix H is obtained from a hand-crafted CPG matrix that was designed to control the robot in a smooth and highly rewarded fashion. The CPG matrix can be used to perform an open-loop control of the robot, but by a minor phase shift it functions also in closed-loop. In the second case the feedback matrix was learned from the correlations observed in the first case (Eq. 14), i.e. without taking the reward in consideration, while in the third case the matrix was learned by the robot while exploring based on a homeokinetic controller. The three matrices are shown in Fig. 4, where the sensor inputs are presented as rows and the motor commands as columns. All matrices are scaled appropriately such that resting state becomes unstable and the legs of the robot start to move.

In order to characterise the behaviour generated in these cases we show in Fig. 5 the behaviour of one leg of the robot with its x and y position. Fig. 6 is added to show the phase relations between the two degrees of freedom for one leg. The CPG-style matrix produces the desired behaviour for the legs with a tripod gait and maximising the use of the leg state space. The second matrix inherits from the behaviour of the first case a consistent relation between the degrees of freedom, however, the matrix is blurred due to hardware-induced deviation from the ideal interaction matrix. In some cases white boxes representing the absence of a correlation,

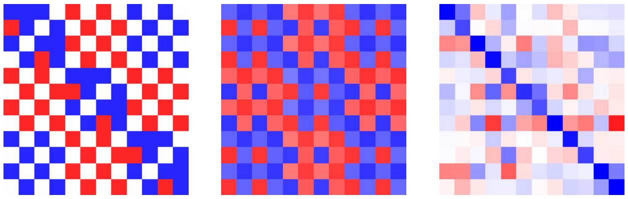

Figure 4: The matrix on the left contains the coefficients similar to a coupled CPG, which is sufficient to perform a tripod gait. With the small changes this matrix can also be used as a closed-loop controller. The matrix on the middle has been learned by the system actuating over a close-loop disregarding the reward. On the right is the re-estimated matrix that was obtained from Eq. 15 while the robot was exploring with the self-organised probing signal. In Fig. 9, the reward obtained by the left and middle matrices can be seen.

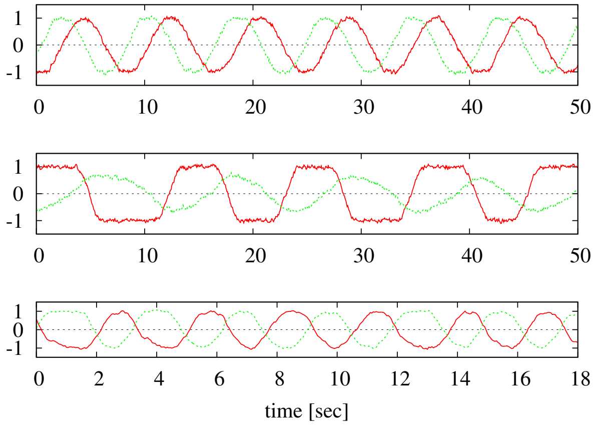

Figure 5: Considering a single leg controlled by a designed coupling matrix (top) we observe a phase shift between the horizontal (red continuous line) and vertical (green dashed line) actuation pattern. The movement of the leg (middle) does not follow the trajectory precisely, but keeps a similar phase shift. Using the present approach (bottom, Eq. 16) the movement pattern becomes more smooth which may point to a reduced energy consumption, but the phase shift has increased, the speed of the robot (as implied by the guidance matrix $W^{r^{[+]}}$ Eq. 15) being in the same range, see Fig. 8

while in other cases the correlations values appear blurred, see the centre image in Fig. 4.

The third matrix is learned based on the reward (Eq. 16) while exploring using the homeokinetic adaptation rule. Correlations in one leg can be seen in the bottom graphic of Fig. 5, this behaviour has been learned within a small time of exploration and the rotational displacement of the leg can be seen in the blue line in Fig. 6. The contact in the ground is less and the exploited state space is smaller, although this still generates a behaviour that is positive towards reward.

The relationship between degrees of freedom from different legs is illustrated by Fig. 7, where we show the relation between a front leg with the lateral middle leg in the horizontal direction. The tripod gait generated by the CPG -the upper and middle graph of the figure- shows the expected phase relations. The same relation is observable in the learned matrix outlining an incipient tripod gait. This generation of the later behaviour is not influenced by the designed matrix which is shown here only for comparison.

The result of the longer experiment can be seen in Fig. 8 where the averaged reward of the tripod gait produced by a designed matrix, the homeokinetic controller, and the learned from reward matrix has been collected for 30 minutes of closed-loop exploitation. The reward of the CPG-style matrix is consistent and positive for all the experiment as expected. The homeokinetic reward is small and also consistent in time, since this controller is not promoted to follow any specific action other than explore coherently. The bigger amplitude of the reward in the learned approach is interpreted as the robot behaving with a bigger variety of actions that tends to maximise the reward but still not completely removing all the actions that leads to a negative reward behaviour. The relations shown in Figs. 5, 6 and 7 holds for some of the degrees of freedom but not for all of them. The final behaviour of the robot produces movement to the front and to the side as well. It can be seen that the homeokinetic exploration produces a small quantity of reward so the captured behaviours are an average of good but still not maximum rewarded actions. Note that the learned matrices have been normalised and multiplied by a factor in order to make the robot responsive in the closed-loop mode, this is required as the acquired results tend to be not big enough given the averaging nature of the approach.

Discussion

We should note that the effect of the discovered structure may not always be beneficial for the robot by itself, but the potential misguidance can be diminished by a manipulation of the value function. As the shape of the robot's body distinguishes one of the directions of movement, there is also a bias in the exploration towards the forward direction. If the goal was instead to move backwards, then our algorithm would fail to provide a direct advantage, but the propagation of the opposite policy values of the policy may still provide a better starting point than random initialisation. Nevertheless, our results confirm that even if the exploration does not directly bring about a coherent behaviour that will receive high reward, it can still induce an acceleration of the learning of the task.

Obviously, also the learning scheme based on Eq. 15 will not be effective for all systems and that more complex relations between reward and sensorimotor coupling than studied here are clearly possible, but it is not the goal to im-

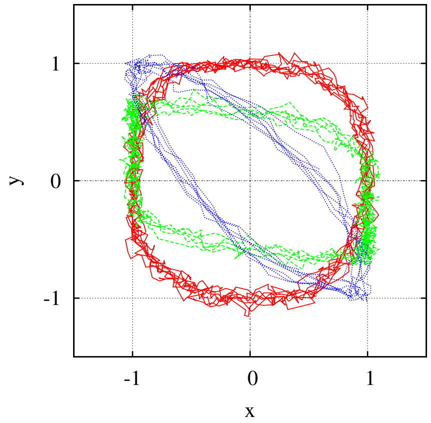

Figure 6: Configuration space representation of the trajectories from Fig. 5. The red continuous line represents the CPG-style matrix, the green dashed line is for the behaviour learned by following the first case without reward and the blue-dotted line represents the rotation learned by the system following Eq. 16.

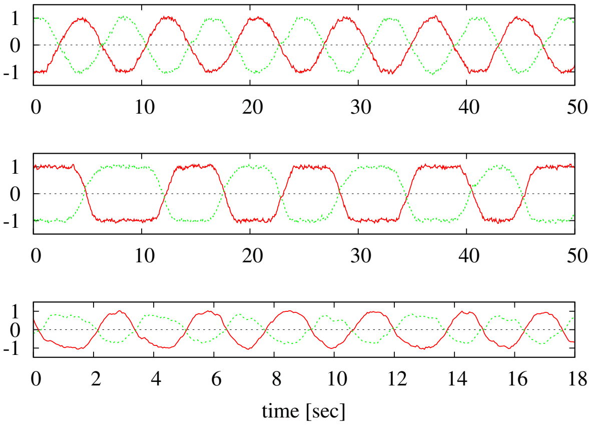

Figure 7: In tripod gait, opposite legs follow an antiphase movement (red continuous line represents front-right leg and green dashed line represents middle-left leg horizontal position), which can be enforced by a designed matrix (top). A similar pattern is discovered by the system following the designed matrix (middle). For the present approach (bottom, Eq. 16) the movement pattern is similar but does not reproduce the same trajectory for all expected DoF as in the designed matrix.

pose these relations precisely, but rather to introduce a bias into the self-organising system such that any deviations between the true sensorimotor couplings and the relation that is implied by the guidance matrix $W^{r^{[+]}}$ are resolved by the exploratory behaviour of the self-organising controller. We should remark, however, that a substantial deviation between guidance and realisable behavioural modes may compromise the efficiency although usually not the effectivity of the control.

The use of the rectified reward signal in Eq. 15 avoids a critical step in the low-dimensional case that was considered in the first part. If the reward signal is negative, taking the opposite action might not always be beneficial or even possible. We have, thus implicitly assumed in the first part, that the opposite action is meaningful and in the second part that all positive rewards are actually relevant for the task which is not required in many other algorithms were only difference of reward signals enter.

Conclusions

We have studied an exploratory self-organised mechanism for discovering promising initialisations for a parametrised policy and to establish coordination among the controllable degrees of freedom of a robot. We used a homeokinetic controller that is based on sensible and predictable exploration and does not require explicit knowledge of the robot's configuration or the environment. The approach is illustrated by a low and a high dimensional tasks implemented in a six-legged robot. The results imply that the initialisation of the parameters for the function approximation by a self-organised approach improves the learning of the proposed tasks and that in high-dimensional set-up the correlation between degree of freedom can be acquired to improve the long-term reward.

Acknowledgements

S.C.S. has been funded by *The Advanced Human Capital Program of the National Commission for Scientific and Technological Research (CONICYT) of the Republic of Chile*. J.M.H. was partially funded by the BMBF in the framework of the *Bernstein Focus for Neurotechnology Göttingen*, grant number 01GQ0811.

References

Der, R., Herrmann, J. M., and Liebscher, R. (2002). Homeokinetic approach to autonomous learning in mobile robots. *VDI-Berichte*, 1679:301–306.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245.

Gullapalli, V. (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks*, 3:671–692.

He, H.-G., Hu, D., and Xu, X. (2011). Efficient reinforcement learning using recursive least-squares methods. *arXiv preprint arXiv:1106.0707*.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum

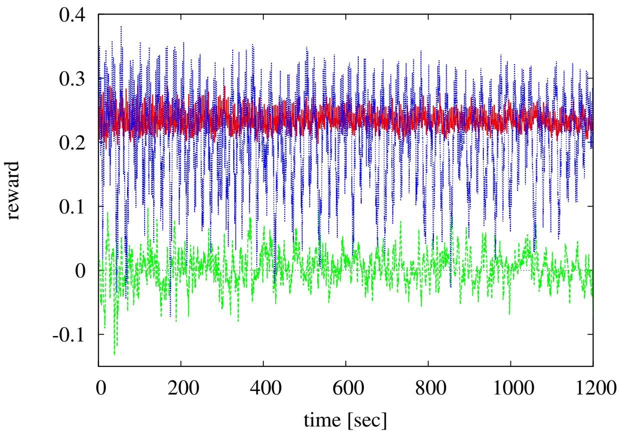

Figure 8: Results for the high dimensional case for a test run of 20 minutes. The reward is derived from the forward velocity of the hexapod, fluctuations correspond to steps. The red continuous line is the tripod gait produced by the designed matrix, the green dashed line is for behaviour obtained within the exploration mode, and the blue dotted line gives the results for a learned matrix that was sampled during homeokinetic exploration for 20 minutes.

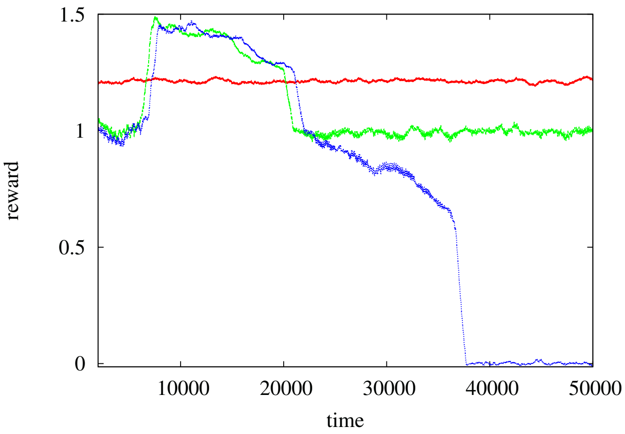

Figure 9: Performance comparison of the reward obtained by the designed matrix (left image in Fig. 4) in an open-loop setting (red continuous line) and the learned correlation matrix (middle image in Fig. 4). The green dashed line and the blue dotted one learn from the first matrix for about 1000 steps, after near 10000 steps they are combined with the designed matrix to produce the motor commands $\boldsymbol{y}_t = (1-\gamma)P\boldsymbol{x}_t + \gamma H\boldsymbol{x}_t$ where P and H are the designed and learned matrices respectively, and $0 \leq \gamma \leq 1$ is the factor that allows different combination of the two terms. Initially, small values of γ are proven which favour reward. Then γ is incrementally increased until step 20000. A decay on reward can be seen. After this point the green dashed line continues in a closed loop ($\gamma = 1$) without learning, and the blue dotted line continues in a closed loop but learning. After some steps the error accumulate and the robot stops walking.

note. *Bonn, Germany: German National Research Center for Information Technology, GMD Technical Report*, 148.

Konidaris, G., Osentoski, S., and Thomas, P. S. (2008). Value function approximation in reinforcement learning using the fourier basis. *Computer Science Department Faculty Publication Series*, page 101.

Martius, G. (2010). *Goal-oriented control of self-organizing behavior in autonomous robots*. PhD thesis, Göttingen University.

Martius, G. and Herrmann, J. M. (2011). Tipping the scales: Guidance and intrinsically motivated behavior. In *Proc. of Europ. Conf. on Artificial Life*, pages 766–775.

Martius, G. and Herrmann, J. M. (2012). Variants of guided self-organization for robot control. *Theory in Biosciences*, pages 1–9.

Martius, G., Herrmann, J. M., and Der, R. (2007). Guided self-organisation for autonomous robot development. In *Europ. Conf. on Artificial Life*, pages 766–775.

Sato, M.-A. and Ishii, S. (2000). On-line EM algorithm for the normalized Gaussian network. *Neural Computation*, 12(2):407–432.

Smith, S. C. and Herrmann, J. M. (2012). Homeokinetic reinforcement learning. In *Partially Supervised Learning*, pages 82–91. Springer.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA. A Bradford Book.

Szita, I., Gyenes, V., and Lőrincz, A. (2006). Reinforcement learning with echo state networks. In *Artificial Neural Networks (Proc. ICANN 2006)*, pages 830–839. Springer.

Wiener, N. (1948). *Cybernetics or Control and Communication in the Animal and the Machine*. Hermann Editions, Paris.

Xu, X., Hu, D., and Lu, X. (2007). Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 18(4):973–992.