Self-Exploration of the Stumpy Robot with Predictive Information Maximization

Georg Martius[1⋆], Luisa Jahn[2,3], Helmut Hauser[3], and Verena V. Hafner[2]

[1] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany
martius@mis.mpg.de
[2] Humboldt-Universität zu Berlin, Institut für Informatik, Berlin, Germany
[3] University of Zurich, Artificial Intelligence Lab, Zurich, Switzerland

Abstract. One of the long-term goals of artificial life research is to create autonomous, self-motivated, and intelligent animats. We study an intrinsic motivation system for behavioral self-exploration based on the maximization of the predictive information using the Stumpy robot, which is the first evaluation of the algorithm on a real robot. The control is organized in a closed-loop fashion with a reactive controller that is subject to fast synaptic dynamics. Even though the available sensors of the robot produce very noisy and peaky signals, the self-exploration algorithm was successful and various emerging behaviors were observed.

Keywords: self-exploration, intrinsic motivation, robot control, information theory, dynamical systems, learning

1 Introduction

One of the long term goals of artificial life research is to create autonomous, self-motivated, and intelligent animats. It has been repeatedly argued, e.g., in [19], that one of the prerequisites for a successful interaction of such complex agents with their environments is the exploitation of their embodiment. In other words, the agent has to acquire knowledge on the impact of its actions on its sensory information and the environment. Developmental robotics, aiming at mechanisms for creating a mind in an embodied agent through a development process, formulates the following additional requirements [11] for the learning system: not task specific, environmental openness, raw information processing, online learning and a continual learning ring/hierarchy.

In this context different artificial intrinsic motivation approaches have been proposed. For example, there exist frameworks based on learning progress [10, 18, 21] and novelty [9], on the reinforcement learning framework, or based on homeokinesis [4], predictive information maximization [12] and empowerment [8] as gradient methods on information theoretic or dynamical systems quantities.

This paper uses predictive information maximization (Pimax) [12], which has been previously successfully applied in simulation, but is here, for the first time, applied to a real robot, more specifically to the robot Stumpy [7]. Predictive

⋆ GM was supported by a grant of the DFG (SPP 1527).

information is the past-future mutual information and measures how much information (in Shannon sense) can be used from the past of a time-series to predict the future. It is different from the bare prediction quality as it also requires the information content itself to be high. This avoids the "dark room problem" [6], i.e., doing nothing in a dark room is best predictable. Consequently, if one would optimize the prediction quality, the agent would not depart from this situation. The PIMAX approach, however, differs by yielding active and coordinated behaviors from scratch in a short amount of time (a few minutes of interaction) and is thus particularly suitable for real robots where the possible interaction time is very limited. In order to estimate the predictive information locally, an internal model is required. This technique is widely used in robotics, e. g. to perform mental simulation [3, 20], and it is also believed to play an important role in human motor planning [22].

In terms of the above mentioned requirements for a developmental program our approach satisfies all but the learning hierarchy by being not task specific, environmentally open, operating on raw sensor information in an online learning fashion. Challenges we address are the from-scratch formation of sensorimotor coordinations leading to smooth behavior, the autonomous selection of sensor information for a particular behavioral mode and the coping with morphological changes.

The Stumpy robot used for the experiments was designed to comply with the principles of cheap design and ecological balance as described in Iida et al. [7], also see Pfeifer and Bongard [19] for a comprehensive overview. The basic design idea was to demonstrate the concept of embodiment, i.e., to show that complex behavior can emerge even from a simple structure due to its physical interaction with the environment. In its original form, the robot was controlled in an open-loop manner by an operator using a joystick. Despite its remarkably simple design, a range of interesting stable locomotion behaviors were demonstrated.

For our experiments we equipped Stumpy with additional sensors and created an adaptive closed-loop system enabling the robot to self-explore its behavior space. The added acceleration sensors provide signals that are very noisy and dominated by shock events. Nevertheless, the implemented approach was successful and was able to generate a variety of smooth locomotion behaviors. Section 2 will give an overview on the Stumpy robot followed by the description of the control algorithm in Section 3. In Section 4, the experimental results are presented.

2 Stumpy, the Pendulum Driven Rocking Robot

The robot Stumpy was first introduced in Iida et al. [7]. The mechanical design of the latest version of Stumpy (developed at the AI-Lab in Zurich [1]) is depicted in Fig 1(a). It has only two joints, which are actuated by servo motors.

The robot is a simple metallic beam structure. Note that due to its design the robot exhibits compliance to a certain extent by the torsion of the beams, which allows for the emergence of dynamic behavior. In addition, rubber blocks are

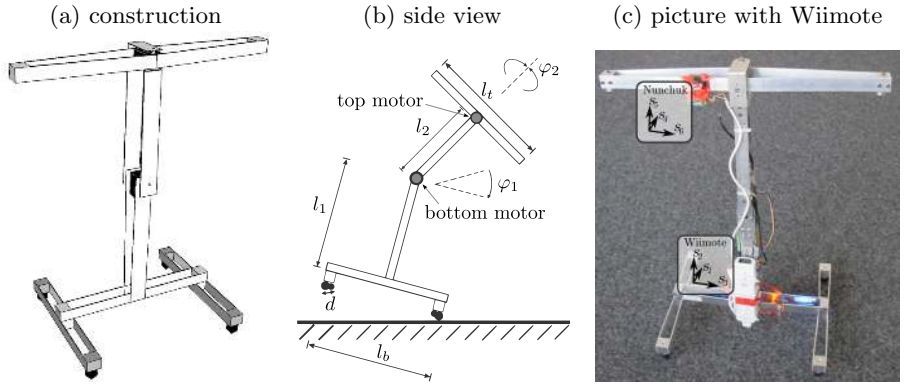

Fig. 1. The Stumpy. **(a,b)** Schematic construction: $d = 25\,\text{cm}$, $l_b = 30\,\text{cm}$, $l_1 = 25\,\text{cm}$, $l_2 = 20\,\text{cm}$, $l_t = 42\,\text{cm}$, $\varphi_1 = \pm 90°$, $\varphi_2 = \pm 35°$. **(c)** Picture of the robot.

attached to the feet to absorb impact shock. While originally the robot was controlled by a human via a joystick in an open-loop fashion, for this work, Stumpy has been equipped with sensors in order to obtain a closed feedback loop that is autonomously driven by the PIMAX algorithm. For simplicity, we use the commercially available controllers of the gaming console Wii by Nintendo [17] called Wiimote and Nunchuk. Both measure the acceleration in all three dimensions in the range of $\pm 3\,\text{g}$ and $\pm 2\,\text{g}$, respectively. The placement of the sensors is depicted in Fig 1(c). The Wiimote can send the data via a Bluetooth connection to the controlling computer [5] at a frequency of approximately 13–25 Hz. The control signals are also sent to the robot via Bluetooth.

The locomotion of Stumpy is achieved by exploiting its inverse pendulum dynamics. By rotating the whole upper part of the body (using the bottom motor) left and right, enough momentum is eventually created to lift one side from the floor, to alternate between left and right on the spot. If the upper horizontal beam is moved as well (rotation of the top motor), it can perform forward and backward movements or turns. Human operators have been exploring many different modes by varying the parameter of the open-loop controller, which are frequency and amplitude of a sinus wave for both DoFs and a phase-shift between them. In contrast to that, in our case the robot is controlled by a reactive controller that uses only the available noisy sensors and it explores the robot's behavioral capabilities in an autonomous, intrinsically driven process.

3 Predictive Information Maximizing Controller

The controller we use for our experiments is the predictive information (PI) maximizing controller (PIMAX) introduced in Martius et al. [12].

We consider the sensor values as a stochastic process S_t with real-valued realizations $s_t \in \mathbb{R}^n$. The PI [2] measures the mutual information between past and future of a time series. Intuitively, the PI corresponds to how much information

can be used from the past to predict the future. The rational to use the PI of the sensor stream as intrinsic motivation is that its maximization leads to a high variance in the sensor values, while keeping a temporal structure. We consider the simplified one-step PI

$$I(S_t; S_{t-1}) = \left\langle \ln \frac{p(s_t, s_{t-1})}{p(s_{t-1})p(s_t)} \right\rangle = H(S_t) - H(S_t|S_{t-1}) \tag{1}$$

where $H(\cdot)$ denotes the Shannon entropy. In order to work with non-stationary processes, which is characteristic for this case, we consider a time-local version called TiPI. Furthermore, to turn this formula into an operational algorithm, we formulate it in the form of a dynamical system.

The stochastic process S can be decomposed as

$$s_t = \phi\left(s_{t-1}\right) + \xi_t = VK(s_{t-1}) + b + \xi_t \tag{2}$$

into the deterministic model ϕ and a stochastic component ξ_t, also called prediction error. The matrix V and the vector b represent the parametrization of the predictor, which are adapted online by a supervised gradient procedure to minimize the prediction error $\xi^\top \xi$ as $\Delta V = \eta_\phi \xi a^\top$ and $\Delta b = \eta_\phi \xi$. The learning rate has been set to $\eta_\phi = 0.05$, which allows a fast adaptation process. The reactive controller K producing the actions (motor values) is realized as a neural network

$$a_t = K\left(s_t\right) = \tanh\left(Cs_t + h\right) \tag{3}$$

with weight matrix C and bias vector h (tanh is understood component-wise). The entire setup is illustrated in Fig 2.

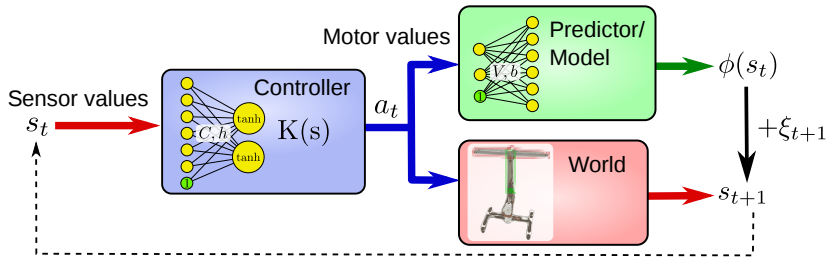

Fig. 2. Sensorimotor loop with controller, predictor and world.

Let us now return to information theory. The TiPI is given by the mutual information conditioned on a fixed sensor state experienced at the beginning of a moving time window of τ steps $I^\tau\left(S_t; S_{t-1}\right) := I\left(S_t; S_{t-1}|S_{t-\tau} = s_{t-\tau}\right)$, here we use the simplest case $\tau = 2$. With a coordinate transformation relative to the start of the time window $\delta s_{t'} = s_{t'} - s_{t-\tau}$ for $t - \tau \leq t' < t$ we can approximate the TiPI assuming Gaussian noise by

$$I^\tau\left(S_t : S_{t-1}\right) = \frac{1}{2} \ln |\Sigma_t| - \frac{1}{2} \ln |D_t| \tag{4}$$

where $\Sigma = \left\langle \delta s \delta s^\top \right\rangle$ is the covariance matrix of δS and $D = \left\langle \xi \xi^\top \right\rangle$ is the covariance matrix of the noise. Sampled covariance matrices tend to be very noisy. However, for calculating the gradient below we can make use of an explicit expression for the Σ using the model ϕ. This is done by approximating $\delta s_{t'}$ in terms of the Jacobian L (state dependent) as $\delta s_{t'} = L\,(s_{t'-1})\,\delta s_{t'-1} + \xi_{t'}$ where $\delta s_{t-\tau} = 0$. In our case the Jacobian matrix is given by $L = VG'\,(z)\,C$, where $z = Cs + h$ and $G'(z) = \mathrm{diag}[\tanh'(z_1), \ldots, \tanh'(z_m)]$.

The controller parameters (C,h) are adapted to increase the TiPI using gradient ascent, i. e. $\Delta C \propto \frac{\partial I^\tau}{\partial C}$, which yields the following simple update rules:

$$\frac{1}{\varepsilon}\Delta C_{ij} = \delta\mu_i \delta s_j - \gamma_i a_i s_j\,, \quad \text{and} \quad \frac{1}{\varepsilon}\Delta h_i = -\gamma_i a_i\,, \tag{5}$$

where all variables are time dependent and are at time t, except δs, which is at time $t-1$. The vector $\delta\mu \in \mathbb{R}^m$ is defined as

$$\delta\mu_t = G' V^\top \Sigma^{-1} \delta s_t \tag{6}$$

and the channel specific learning rates are given by $\gamma_i = 2\,(C\delta s_{t-1})_i\,\delta\mu_i$. The learning rate was set to $\varepsilon = 0.1$. It is interesting to note that the covariance matrix of the noise (D) cancels and does not enter the update formulas. The inverse of the covariance matrix Σ in Eq (6) is sampled with an exponential moving average with 100 timesteps. In practice, we found that it may even be replaced by the identity matrix (which is not done here). For the derivation and more details we refer to [12].

4 Experiments

Several experiments have been conducted applying the PIMAX algorithm to the Stumpy robot. Note that the behavior can only emerge through the connection of sensors and motors via (Eq 3). This connection, however, is constantly changing during exploration to locally maximize the TiPI. This results in an intricate interplay between physical dynamics and parameter dynamics. We will proceed by analyzing the sensor data and then present the self-exploration process.

4.1 Evaluating sensors

First, we evaluated the sensors by controlling the robot in an open-loop fashion as described in [7] in a forward locomotion. Both motor signals followed a sine wave at 1.7 Hz with a certain phase-shift between the bottom and top motor. The corresponding sensor readings, however, are very noisy and, on first glance, do not seem to reflect the harmonic control signal (see Fig 3).

However, performing a wavelet transformation of the sensor time series allows us to identify two sensors (s_3, s_4) that show a major oscillatory component at 1.7 Hz (see Fig 4). However, to demonstrate that the algorithm is able to find its most valid sensor information on its own, we provided all sensors ($s_1, \ldots, s_6$) to the controller. Although this seems to be a very challenging task, as we will see, our self-organizing control approach is able to overcome these difficulties.

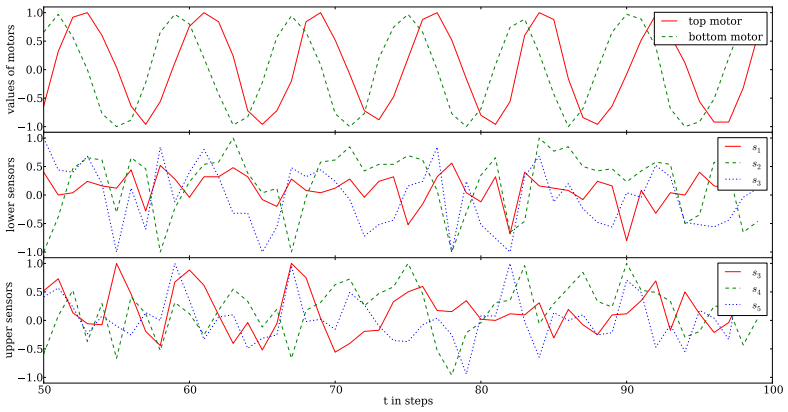

Fig. 3. Motor and sensor values for a forward motion. Top: motor values following a sine wave of 1.7 Hz (nominal angle normalized to [-1,1]). Center and bottom: sensor values from lower and upper acceleration sensors, see Fig 1(c). The harmonic control signal cannot be identified from the very noisy sensor readings. Update frequency was 14 Hz.

4.2 Behavioral Self-Exploration

For the rest of the paper the robot is controlled by the reactive controller (Eq 3) with the learning dynamics given in Eq (5). Initialized with a "do nothing" controller (i.e., all synaptic weights set to 0) and a randomly initialized forward model, the robot starts to gently move after some initial time. The entropy term in the PI (Eq 1) drives the system to activity, which initially leads to a progressive noise amplification. As soon as the movements becomes large enough to cause a definite effect on the sensor values, the forward model is able to capture these correspondences and the movements become more coherent and related to the body/environment. A rocking behavior quickly emerges, which develops into different types of locomotion and swinging behavior. The evolution

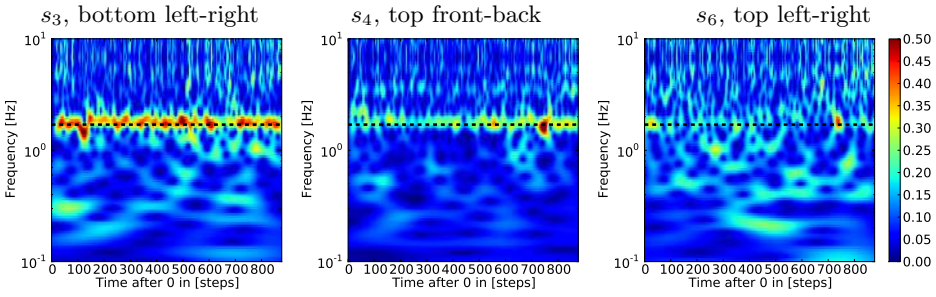

Fig. 4. Wavelet transform of sensor values for forward motion with open loop control. The prominent frequencies are clearly visible at 1.7 Hz for the sensors s_3 and s_4. Even though the robot is making periodical movements, the other sensors do not show a clear major frequency (s_1,s_2,s_5 are less pronounced than s_6). Update frequency: 14 Hz.

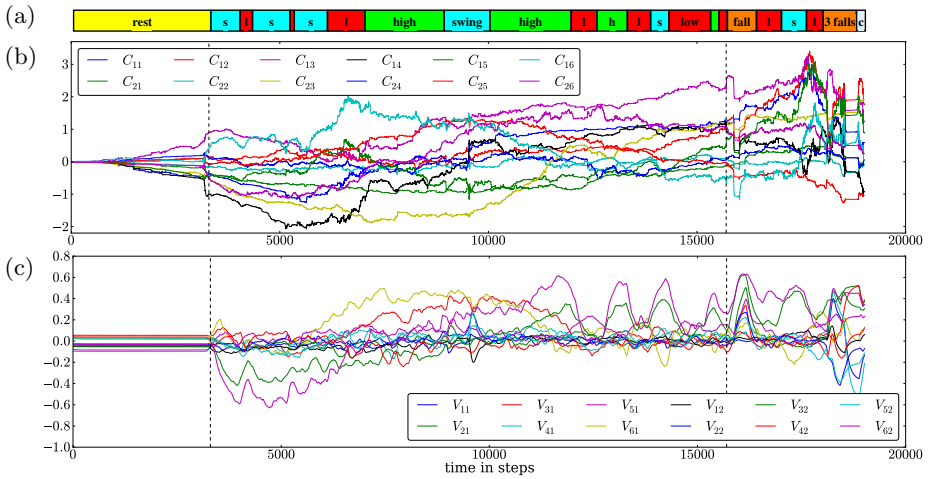

Fig. 5. Time evolution of the weights of the controller and the forward model. **(a)** Movements classified visually into: rest, **s**winging, **l**ow, **h**igh, falling, and **c**rawling see text. **(b)** Entries of the controller matrix C over time. The weights adapt constantly without making big jumps. In this way the controller is able to produce various motions, see (a). At about 15700 and 18000 the robot tips over. **(c)** Weights of the forward model (matrix V). Update frequency was 23 Hz.

of the controller parameters during an typical run is displayed in Fig 5(b). The elements of the matrix C (Eq 3) change constantly, which is expected from the algorithm. The gradient on the TiPI is never zero as long as there is a non-zero prediction error. This is counter-intuitive, but can be understood by the fact that the landscape (the TiPI) changes with the behavior. Thus, all values change during the whole experiment in a more or less smooth fashion. The robot starts to move at approximatelly step 3000. Then it rather quickly enters a swinging motion followed by a slow, but steady sweep through several behaviors, see Fig 5(a). For simplicity the behaviors are grouped into the following types: "low" movement: robot is either locomoting with low amplitude or it is trying to excite a new mode (feet are on the ground); "high" movements: cause locomotion with high amplitude; "swinging" means rocking at the spot with high amplitude and balancing with the top to not fall; "falling": which is due to swinging to high; and "crawling" means locomotion in laying position. The disturbance at step 15700 is introduced by Stumpy falling over. Immediately afterwards it has been manually lifted. After step 18000, the robot fell another three times and, finally, was left in this state, where it produced, suddenly confronted with a completely different body/environment relationship, remarkably fast a crawling behavior.

There is no obvious relationship between the controller parameters and the observed behavior (Fig 5(a,b)). Apparently, different parameter configurations can lead to similar behavior. The forward model has a more defined structure, at least later in the experiment, see Fig 5(c). During the swinging and high motions,

Fig. 6. Different behaviors emerged from the control of the Stumpy with the PIMAX. During the swinging motion, the robot swings to either side and then, just before falling, moves the upper body towards the center. Further behaviors include fast locomotion with low amplitude, and swinging and turning to either side. Videos are available [16].

the values V_{32} and V_{62} (reflecting the correspondence between motor actions and left-right accelerations) raise in value. For low movements, the model collapses, because of no or little defined responses in the sensors.

In order to give an impression of the behavior, Fig 6 shows a series of frames for different movements. To get a quantitative characterization of the behavior, wavelet transformations of the motor and sensor values have been carried out and are presented in Fig 7. It can be seen how the main frequency changes with the behavior being lower for high and swinging movements and higher for low movements. Note that these frequencies are still lower than the ones induced by the forward movement in the open-loop setup (operator and joystick). It remains for future work to evaluate whether these movements are closer to the Eigenfrequency of the system and, thus, more energy efficient.

A variety of different behaviors have been generated including transitions between them. So far we have analyzed a single run to identify some key features of the behavioral self-exploration process. In order to see which effect the initialization has on the performance, we ran multiple runs and found consistently similar results. The exact order and timing of individual movements were different, but typically all types of movements have been generated.

4.3 Changing the Morphology

Since the PIMAX algorithm does not have any information about the robot under control (except the number of sensors and motors), changing the morphology of the robot should make no difference to the algorithm. It explores and exploits its given embodiment. In order to demonstrate this remarkable capability, we first put the Stumpy robot intentionally horizontally on the ground, as it already happened accidentally at the end of the exemplary run. The PIMAX algorithm achieves a crawling movement after a few seconds. The result can be seen in a video [16].

A modification of the morphology of the robot was carried out by putting Stumpy into a Chinese cooking pot, also called wok. The wok was modified on the bottom to make a smooth rotating movement possible and it was equipped with a heavy weight to prevent the construction from falling over. When the algorithm started to work, an emerging rotation motion of the wok was observed, see Fig 8. The motion was very steady and smooth, also due to the fact that the sensor readings were much smoother than compared to the previous experiments,

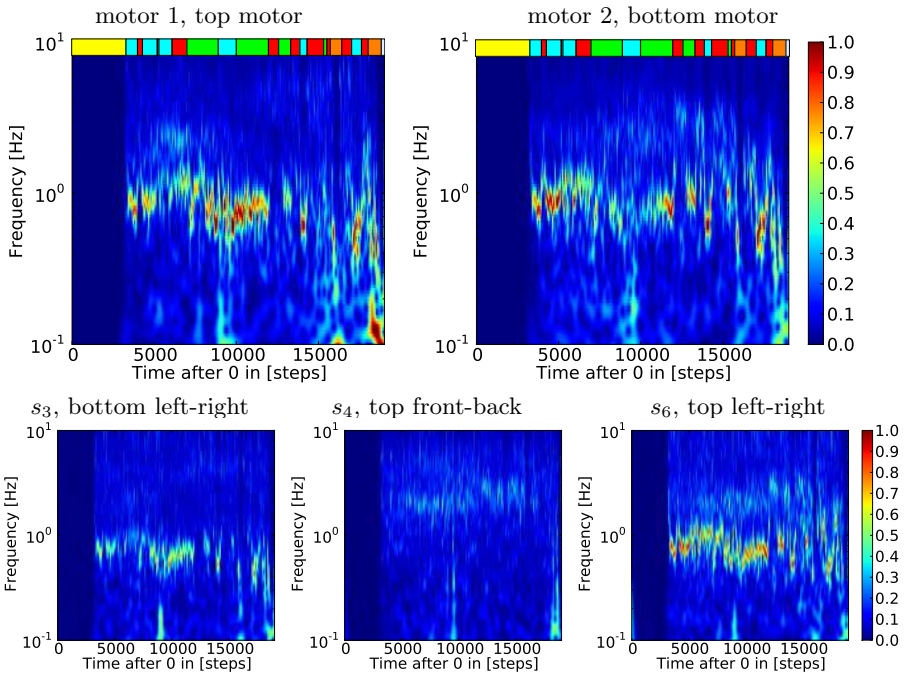

Fig. 7. Wavelet transform of the sensor and motor values for the exemplary experiment. It is clearly visible that the prominent frequency fluctuates around 0.5 to 1.5 Hz which corresponds to various motions. Also both motors behave in an individual way which leads to a high variance in motions and transitions between motions. Only the sensors s_3, s_4, and s_6 show a characteristic footprint of the motions (s_1, s_2, s_5 not shown). The sensor s_4 (top front-back) shows a faint trace typically at twice the frequency.

where the robot had to deal with strong impacts. This can also be observed in the collected sensory data and the corresponding wavelet transformation in Fig 8. In this experiment the robot started to rotate already after about 70 seconds (step 1600). After 150 seconds (step 3250) the robot was manually stopped. It took the algorithm only a short amount of time to reenter the rotating motion (25 sec).

5 Discussion

We report on the control of the robot Stumpy with the PIMAX algorithm to obtain a self-organized behavioral exploration. Even though there is no specific goal, just the generic drive to locally maximize the predictive information of the sensor stream, the algorithm generates a variety of active behaviors that exploit the given embodiment. When the robot is upright, different movements emerge including various types of locomotion, turning, and swinging. When the robot lays on the ground, a crawling behavior is generated, and when the robot is

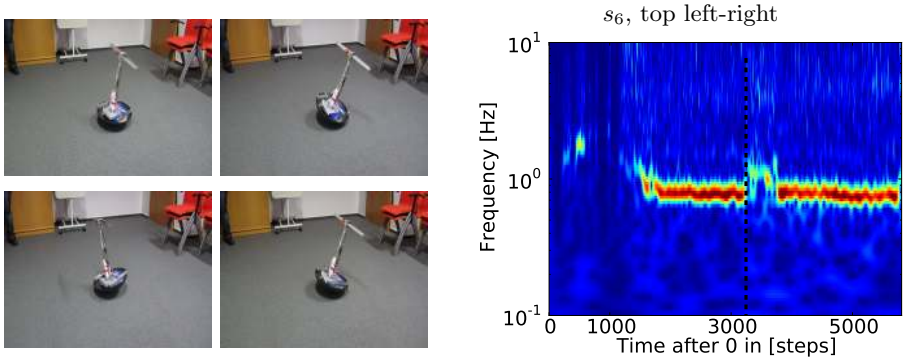

Fig. 8. Stumpy rotating with an attached wok. Left: frames from the behavior. Right: wavelet transform of the sensor s_6, ($s_{1,3,4}$ have a similar spectrum). The behavior is very smooth and steady at a frequency of ≈ 0.8 Hz. The robot was stopped manually at step 3250 (black dashed line) and recovered itself, for videos see Martius et al. [16].

placed in a wok (Chinese cooking pot) it starts to excite a stable rotation movement. The performance of the system is even more astonishing given the available sensor quality. The control of the robot is based on two 3-axis acceleration sensors in a reactive manner (no pattern generator nor recurrent connections, but fast synaptic dynamics). The acceleration sensor values are dominated by shock events and seem to be unusable for a smooth control at first glance. However, as it was demonstrated, the PIMAX algorithm organizes the sensorimotor loop in such a way that smooth behaviors are generated. Due to its fast adaptation mechanism, it can quickly react to changing responses of the physical system, e. g. due to a different behaviors mode. In this way it amplifies latent modes, such as swinging at intrinsic frequencies or the rotation of the robot with the wok. At the same time it can cope with drastically different situations, e.g. when the robot tips over.

On a higher level of learning, the found behaviors can be potentially memorized as primitives [14] such that they do not need to be rediscovered every time. As demonstrated (see Fig 5) the algorithm generates different behavioral modes that are persistent for some time and then transition to other modes. Each of these can be captured as a primitive behavior either by storing the controller parameters or by training a separate control module [14]. If goal directed behaviors are to be achieved then these primitives can be used as actions in a reinforcement learning setup. Alternatively, the self-organization process itself can be guided with various methods, see [13] for an overview, which have been shown to be particularly powerful in high-dimensional systems [15].

Acknowledgments

The authors thank Ralf Der for helpful discussions and the idea with the wok and Fumiya Iida for providing material on the Stumpy. LJ thanks Rolf Pfeifer and the AI-Lab team for their hospitality during her stay in Zurich and in particular Max Lungarella for helping with the robot.

References

[1] Artificial Intelligence Laboratory, Zurich: (2013), http://www.ifi.uzh.ch/ailab.html

[2] Bialek, W., Nemenman, I., Tishby, N.: Predictability, complexity and learning. Neural Computation 13(11), 2409–2463 (2001)

[3] Bongard, J.C., Zykov, V., Lipson, H.: Resilient machines through continuous self-modeling. Science 314, 1118 –1121 (2006)

[4] Der, R., Martius, G.: The Playful Machine - Theoretical Foundation and Practical Realization of Self-Organizing Robots. Springer (2012)

[5] Donnie Smith: Cwiid: Linux Nintendo Wiimote interface library (2014), http://abstrakraft.org/cwiid

[6] Friston, K., Thornton, C., Clark, A.: Free-energy minimization and the dark room problem. Frontiers in Psychology 3(130) (2012)

[7] Iida, F., Dravid, R., Paul, C.: Design and control of a pendulum driven hopping robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. vol. 3, pp. 2141 – 2146 (2002)

[8] Klyubin, A.S., Polani, D., Nehaniv, C.L.: Empowerment: a universal agent-centric measure of control. In: Evolutionary Computation. pp. 128–135 (2005)

[9] Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. In: Proc. Intl. Conf. on Artificial Life (ALIFE XI). p. 329. MIT Press, Cambridge, MA (2008)

[10] Luciw, M., Kompella, V., Kazerounian, S., Schmidhuber, J.: An intrinsic value system for developing multiple invariant representations with incremental slowness learning. Frontiers in Neurorobotics 7(9) (2013)

[11] Lungarella, M., Metta, G., Pfeifer, R., Sandini, G.: Developmental robotics: a survey. Connection Science 15(4), 151–190 (2003)

[12] Martius, G., Der, R., Ay, N.: Information driven self-organization of complex robotic behaviors. PLoS ONE 8(5), e63400 (2013)

[13] Martius, G., Der, R., Herrmann, J.M.: Robot learning by guided self-organization. In: Prokopenko, M. (ed.) Guided Self-Organization: Inception. Springer (2014)

[14] Martius, G., Fiedler, K., Herrmann, J.M.: Structure from Behavior in Autonomous Agents. In: Proc. IEEE IROS 2008. pp. 858 – 862 (2008)

[15] Martius, G., Herrmann, J.M.: Tipping the scales: Guidance and intrinsically motivated behavior. In: Advances in Artificial Life, pp. 506–513. MIT Press (2011)

[16] Martius, G., Jahn, L., Hauser, H., Hafner, V.V.: Supplementary materials (2014), http://playfulmachines.com/Stumpy2014

[17] Nintendo: Wii official website (released 2006), http://www.nintendo.com/wii

[18] Oudeyer, P.Y., Kaplan, F., Hafner, V.V.: Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation 11(2), 265–286 (2007)

[19] Pfeifer, R., Bongard, J.C.: How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books). The MIT Press (2006)

[20] Schillaci, G., Hafner, V.V., Lara, B.: Coupled inverse-forward models for action execution leading to tool-use in a humanoid robot. In: Proc. of 7th Intl. Conf. on Human-Robot Interaction (HRI '12). pp. 231–232. ACM (2012)

[21] Schmidhuber, J.: Curious model-building control systems. In: In Proc. Intl. Joint Conf. on Neural Networks, Singapore. pp. 1458–1463. IEEE (1991)

[22] Wolpert, D.M., Miall, R.C., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Sciences 2, 338 – 347 (1998)