Direct Control of an Active Tactile Sensor using Echo State Networks

André Frank Krause, Bettina Bläsing, Volker Dürr and Thomas Schack

Abstract Tactile sensors (antennae) play an important role in the animal kingdom. They are also very useful as sensors in robotic scenarios, where vision systems may fail. Active tactile movements increase the sampling performance. Here we directly control movements of the antenna of a simulated hexapod using an echo state network (ESN). ESNs can store multiple motor patterns as attractors in a single network and generate novel patterns by combining and blending already learned patterns using bifurcation inputs. *Index Terms* – Active Tactile Sensors, Motor Learning, Neuronal Networks

André Frank Krause e-mail: andre_frank.krause@uni-bielefeld.de · Bettina Bläsing e-mail: bettina.blaesing@uni-bielefeld.de · Thomas Schack e-mail: thomas.schack@uni-bielefeld.de
Neurocognition and Action - Biomechanics Research Group, Bielefeld University, Germany

Volker Dürr e-mail: volker.duerr@uni-bielefeld.de
Dept. for Biological Cybernetics, University of Bielefeld, Faculty of Biology, PO Box 100131, 33501 Bielefeld, Germany

all authors:
Cognitive Interaction Technology, Center of Excellence, University of Bielefeld, 33615 Bielefeld, Germany

1 Introduction

Animals and humans are autonomous mobile systems of prime performance, partly due to their highly adaptive locomotor behaviour, partly due to their sensory abilities that allow for rapid, parallel object recognition and scene analysis. In animals, near-range sensing, particularly the active tactile sense is often of great importance: many insects actively move their antennae (feelers) and use them for orientation, obstacle localisation, pattern recognition and even communication [1]; mammals like cats or rats use active whisker movements to detect and scan objects in the vicinity of the body. Active tactile sensors offer some advantages over vision based sensors [2]: The tactile sense is independent of light conditions, it works at day and night. It is also independent of the surface properties of objects (colour, texture, reflectivity) that may be very difficult for vision, radar or ultrasound based systems. Furthermore, the 3d spatial contact position with an object is immediately available due to the known geometry of the sensor. No computationally expensive stereovision algorithms are required. A drawback of tactile sensors might be lower information density about a scanned object and the danger of mechanical damage to both the sensor and the object. Insect-like tactile sensors have been pioneered by Kaneko and co-workers, who used either vibration signals [3] or bending forces [4], both measured at the base of a flexible beam, to determine contact distance. Recently an actively movable tactile sensor inspired by a stick insect antenna was co-developed by Fraunhofer IFF Magdeburg [5] and the University of Bielefeld [6] with a single acceleration sensor located at the tip of the probe. It was shown to be remarkably sensitive in object material classification. An efficient movement pattern that maximises obstacle detection performance while minimising energy consumption for such an active tactile sensor is a circular movement of the sensor probe [7]. Stick insect antennae perform elliptical exploratory movement patterns relative to the head until they encounter an obstacle. After the first contact, the movement switches to a pattern with smaller amplitude and higher cycle frequency [8].

A straightforward way to learn motor patterns is to store them in the dynamics of recurrent neuronal networks [9]. The network implements a forward model, that predicts the sensor informations for the next time step [10]. In [9] it is argued, that this distributed storage of multiple patterns in a single network gives good generalisation compared to local, modular neural network schemes [11][12]. In [13] it was shown that it is also possible to not only combine already stored motor patterns into new ones, but to establish an implicit functional hierarchy by using leaky integrator neurons with different time constants in a single network. This network can then generate and learn sequences over stored motor patterns and combine them to form new complex behaviours. Tani [9][14][13] uses backpropagation through time (BBTT, [15]), that is computationally complex and rather biologically implausible. Echo State Networks (ESNs [16]) [17] are a new kind of reccurent neuronal networks that are very easy and fast to train compared to classic, gradient based training methods (backpropagation through time [15], real time recurrent learning [18]). Gradient based learning methods suffer from bifurcations that are often encountered during training. Bifurcations apruptly change the dynamic behaviour of

a network, rendering gradient information invalid [19]. Additionally it was mathematically shown that it is very difficult to learn long term correlations because of vanishing or exploding gradients [20]. The general idea behind ESNs is to have a large, fixed random reservoir of recurrently and sparsely connected neurons. Only a linear readout layer that taps this reservoir needs to be trained. The reservoir transforms usually low-dimensional, but temporally correlated input signals into a rich feature vector of the reservoir's internal activation dynamics. This is similiar to a linear finite impulse response filter or Wiener filter [21], that reads out a tap delay line with a linear combiner. Here, the delay line acts as a preprocessor that constructs a sufficiently large state space from the input time series, such that the temporal dependencies become implicit. Batch training involves collecting the internal states of the reservoir neurons and applying fast linear regression methods to calculate the output layer. More biologically plausible online learning methods for ESNs exist, for example the recursive least squares algorithm [17] or backpropagation decorrelation (BPDC [22]). BPDC was successfully applied in a developmental learning scenario with the humanoid robot ASIMO [23] and for learning an inverse model of the industrial PA-10 robot arm [24]. Here, we use an ESN to implement a forward model that actively moves the tactile sensor / antenna of a simulated hexapod walker.

2 Simulations

A basic, discrete-time, sigmoid-unit ESN was implemented in C++ using the expression template matrix library Eigen2. The state update equations used are:

$$\begin{aligned} \mathbf{y}(n) &= \mathbf{W}^{out}\mathbf{x}(n) \\ \mathbf{x}(n+1) &= \tanh(\mathbf{W}^{res}x(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{back}\mathbf{y}(n) + \nu(n)) \end{aligned} \tag{1}$$

where $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$ are the activations of the input, reservoir and output neurons, respectively. $\nu(n)$ adds a small amount of uniformly distributed noise to the activation values of the reservoir neurons. This tend to stabilize solutions especially in models using output feedback for cyclic attractor learning [25]. $\mathbf{W}^{in}$, $\mathbf{W}^{res}$, $\mathbf{W}^{out}$ and $\mathbf{W}^{back}$ are the input, reservoir, output and backprojection weight matrices. All matrices are sparse and randomly initialised and stay fixed, except for $\mathbf{W}^{out}$. The weights of the linear output layer are learned using offline batch training. During training, the network is driven with the input and teacher data and internal reservoir activations are collected (state harvesting). The teacher data is forced into the network via the backprojection weights (teacher forcing). After collecting internal states for all training data, The output weights are directly calculated using ridge regression. Ridge regression uses the Wiener-Hopf solution $\mathbf{W}^{out} = \mathbf{R}^{-1}\mathbf{P}$ and adds a regularization term (Tikhonov regularization):

$$\mathbf{W}^{out} = (\mathbf{R} + \alpha^2\mathbf{I})^{-1}\mathbf{P} \tag{2}$$

where α is a small number, $\mathbf{I}$ is the identity matrix, $\mathbf{R} = \mathbf{S}'\mathbf{S}$ is the correlation matrix of the reservoir states and $\mathbf{P} = \mathbf{S}'\mathbf{D}$ is the cross-correlation matrix of the states and the desired outputs. Ridge regression leads to more stable solutions and smaller output weights, compared to ESN training using the Moore-Penrose pseudoinverse.

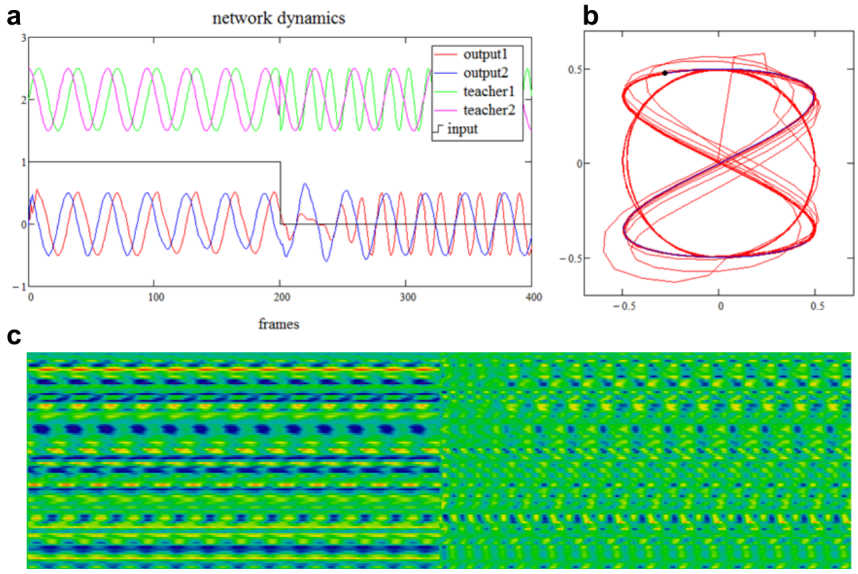

Fig. 1 Dynamic behaviour of an ESN with a single input trained on a circular (input value = 1) and figure-eight pattern (input = 0). a) After 200 time-steps, the input value is switched from one to zero. The network smoothly changes its behaviour to the figure-eight pattern. b) 2d trajectories of the network output. c) Colour coded internal activations of the reservoir neurons. Lines indicate individual neuroids, columns indicate time.

Table 1 ESN structure parameters. 200 reservoir neurons, 2 inputs and 2 outputs used. Direct input to output connections and bias inputs were not used. Sparseness is the fraction of connections with non-zero weigths. Synaptic weights were randomly initialised in the range [-strength strength]. $\alpha = 0.01$, $\nu = 0.001$.

from	to	Sparseness	Strength
Input	Reservoir	1.0	0.6
Reservoir	Reservoir	0.1	0.4
Output	Reservoir	1.0	0.8

Dynamic Behaviour. The input-, reservoir- and backprojection weight matrices were sparsely initialised with uniformly distributed random values. See table 1 for the network parameters used. In a first simulation, a simple ESN with one input, two outputs, no bias inputs and no input-to-output connections was trained on the

circle (input value = 1) and figure-eight pattern (input value = 0) (see table 3). Fig. 1 shows the dynamic behaviour of this network. Abruptly switching the input from one to zero smoothly moves the network from the circular pattern attractor state to the figure-eight pattern. In Fig. 2 the input parameter space is explored within the range -1.0 to 1.2. Interestingly, an input value of -1 evokes the same circular pattern as for an input value of one. This symmetric effect is mentioned in [19]. Increasing the input value further causes gradual morphing from a circular to an elliptical and later to the figure-eight pattern. Increasing the input above 1 or below -1 drives the network into a fixpoint attractor state.

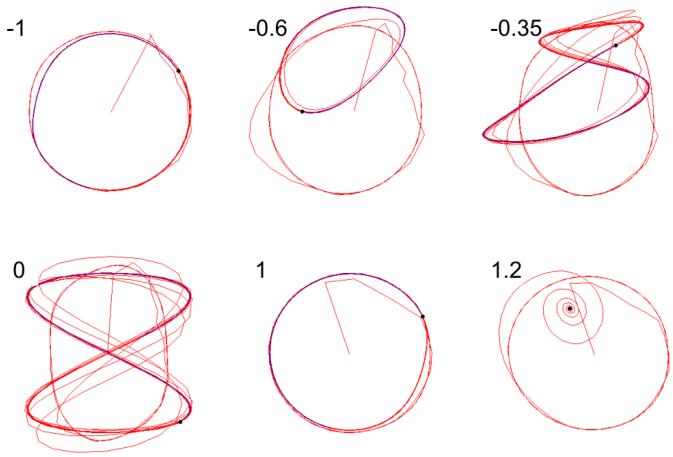

Fig. 2 Shifting the dynamics of the network by gradually changing the input value from -1 to 1.2. Because of symmetry effects [19], the circular pattern reappears with input value -1. Increasing the input to the network causes a slow morphing between the two learned patterns, allowing to generate new patterns that were not explicitly trained. The network keeps stable with no chaotic regions until it converges to a fixpoint at input value 1.2.

Training Success. In a second simulation, the training success and the transition behaviour between two patterns after switching the input values was analysed. Instead of a single input, this time 2 binary encoded inputs were used. The first pattern was learned with an input vector $\begin{pmatrix} 1 & 0 \end{pmatrix}$ and the second with $\begin{pmatrix} 0 & 1 \end{pmatrix}$. This avoids symmetry effects as shown in simulation 1 and reduces the training error. ESN parameters used are shown in table 1. The training error was defined as the smallest Euclidean Distance between the training pattern and a time-shifted version of the network output ($\pm$200 time-steps). The selected time-shift corresponded to the largest cross-correlation coefficient. The error of a single network was averaged over n=50 runs. Input patterns were presented in random order and all network activations were randomly intialized in a range of $\pm$0.4 before each run. N=500 networks were trained, resulting in a median error of 0.029 (0.6% deviation relative to the circle pattern radius of 0.5). 60% of the trained networks had an error below 1%.

Table 2 ESN structure parameters. 300 reservoir neurons, 4 inputs and 2 outputs were used in the multiple pattern storage task. $\alpha = 0.001$, $\nu = 0.001$

from	to	Sparseness	Strength
Input	Reservoir	1.0	0.8
Reservoir	Reservoir	0.1	0.2
Output	Reservoir	1.0	1.2

Transition Smoothness. After switching between patterns, the network might become unstable and show chaotic behavior. Relating to the Minimum Jerk Theory [26], the smoothness of the movement after the switch can be quantified as a function of jerk, which is the time derivative of acceleration. The jerk was calculated for 100 timesteps after the switch. The mean jerk for 500 networks averaging over 50 runs for each net was 0.024 with a standard deviation of 0.003. This is just slightly larger than the averaged jerk of both training patterns (0.0196). The transition behaviour was sufficiently stable for all 500 networks, the maximum jerk found was 0.038. For comparison, mean jerk of purely random, untrained networks was 41.6 with a SD of 40.713.

Learning Capacity. A larger ESN was trained with four different patterns, see table 3. ESN parameters used are shown in table 2. Fig. 3 shows that it is possible to store multiple motor patterns distributed in a single network. Nonetheless it requires manual parameter fine-tuning to get stable attractors that are close to the training data.

Table 3 Training patterns used for the ESN experiments.

Pattern	Parameters
Circle	$0.5\,(\,\sin(0.2n)\ \cos(0.2n)\,)$
Eight	$0.5\,(\,\sin(0.4n)\ \sin(0.2n)\,)$
Rectangle	$0.5\,(\tanh(2\sin(0.2n))\ \tanh(2\cos(0.2n))\,)$
Star	$0.2\,(\text{atanh}(0.98\sin(0.2n))\ \text{atanh}(0.98\cos(0.2n))\,)$

Motor Control. Stick insects continuously move their antennae during walking using a wide, exploratory movement pattern. If the antennaes detect obstacles, the antennal movements immediatly change to a sampling pattern [8]. This switching behaviour was modeled using an ESN and a simulated hexapod walker with antennae. The simulation was implemented in C++ using the Open Dynamics Engine (ODE). The joints of the antenna were steered using a p-controller and constraint-based angular velocity control (hinge joint angular motor in ODE). Due to the dynamic nature of the system and the p-controller, actual joint angles always lag some frames behind the desired values and have a slightly smaller amplitude. The network thus has to learn to predict new motor commands based on the proprioreceptive input from the antennal joints. In a first step, training data was created by sinusoidal

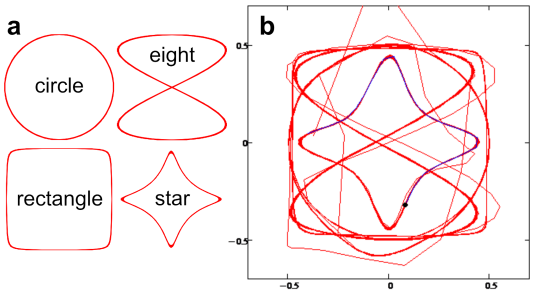

Fig. 3 An ESN with 4 binary inputs, two outputs and 300 reservoir neurons was trained to store multiple patterns. a) shows the 4 patterns used as training data (for parameters see table 3. b) shows the network dynamics. in the generation phase, the network was started with random network activations and simulated for 4000 time steps. the input value changed every 1000 time steps.

modulation of the antennal joints and simultaneously recording actual joint angles in the simulation, see Fig. 4. An ESN was then trained on the collected data and put into the loop (identical network parameters as in the initial experiment, see table 1). If the left antenna encountered contacts with obstacles, the input values to the ESN were switched, causing a transition from a wide, exploratory movement pattern to the figure-eight pattern. After some decay time, the inputs were switched back. Fig. 5 shows a behaviour sequence of an hexapod walker with an ESN-controlled left antenna. Obstacle contact causes a pre-programmed obstacle avoidance reflex by turning the insect away from the object.

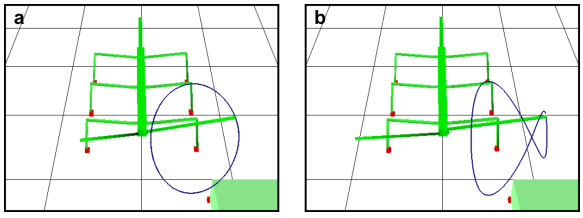

Fig. 4 An ESN was trained to generate a circular and a figure eight pattern with the tip of the left antenna. Inputs to the net are the current joint angles and the pattern selection inputs; outputs are the target joint angles for the p-controller steering the joints of the active tactile sensor.

3 Discussion and Outlook

First basic experiments with ESNs have shown that they can be used for direct motor control of simple articulated limbs. The ESN implements a forward model that predicts sensory and motor commands for the next time step. It is also possible to gen-

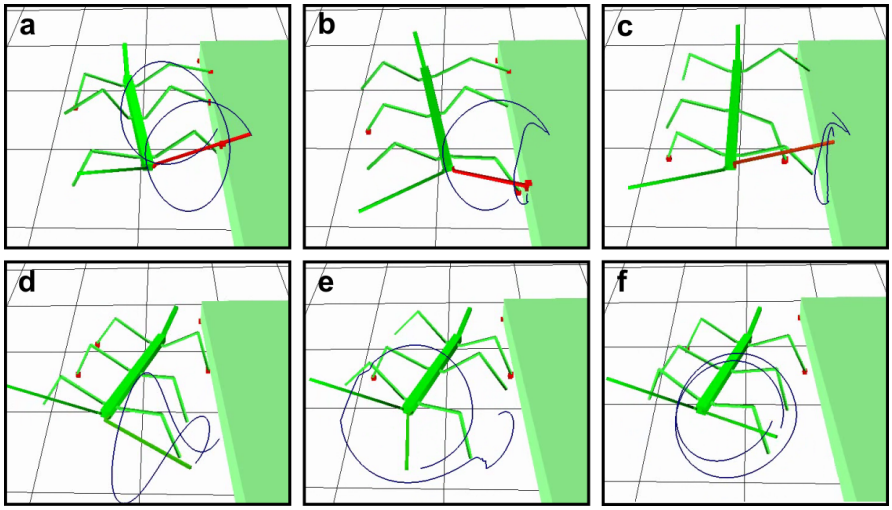

Fig. 5 Sequence of images showing antennal behaviour before, during and after contacts with an obstacle. From left to right: a) first contact of the left antenna with the obstacle. The antenna still performs a circluar, exploratory movement. b) the contact information causes a switch in behaviour from the circular to the figure-eight pattern. c) Second and third contact: the hexapod walker starts to turn away from the obstacle. d) The figure-eight pattern continues for a while until the contact information decays. e-f) After that, the behaviour switches back to the exploratory circular movement. A video can be downloaded at: `http://www.andre-krause.net/publications/hcrs09_s1.avi`

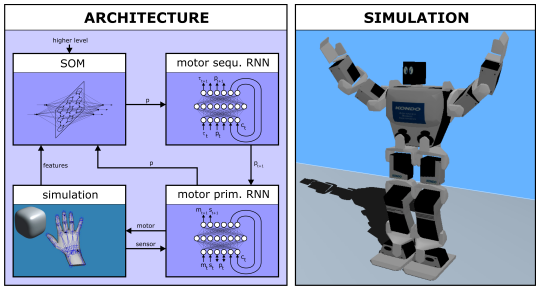

Fig. 6 A proposed architecture for learning and generation of complex movements. Hierarchically coupled ESNs are controlled using a hierarchical self organizing map, that implements basic action concepts. Image: ©Webots [27]

erate new, not explicitly trained patterns by shifting the network dynamics through additional bifurcation inputs. This was already demonstrated by [9] via parametric bias inputs for a variant of Elman type networks. If exploited properly, this dynamic feature of ESN networks makes it possible to generate and interpolate numerous motor patterns from a few, well choosen basic motor patterns. ESNs can also store multiple motor patterns in a single network, although it is important to fine-tune all network parameters to succeed. Pretraining of the reservoir using Intrinsic Plasticity

[28] can help to make the training process more robust. ESN parameters could also be automatically fine-tuned in a reinforcement scenario using new, very fast and powerful black box optimisation algorithms [29] [30]. ESNs seem to be suitable for a planned hierarchical architecture for learning and control of long, complex movement sequences, as illustrated in Fig. 6. ESNs scale well to a high number of training patterns and motor outputs [31]. A more complex simulation - for example of a humanoid robot - might reveal possible limitations of direct, attractor-based motor pattern learning. The future goal is to couple two or more ESNs hierarchically or even embed an implicit hierarchy into a single network using neurons with random time constants similiar to [13]. On top of that, a hierarchical self-organizing map (HSOM) can implement cognitive structures of basic action concepts [32] [33] and provide input and reference values to the ESNs. The HSOM can integrate perceptual features of the environment, proprioceptive sensory data of the robot body and higher level commands (intention, affordance) to select a proper motor program. Cluster structures learned in the HSOM might then be compared to cognitive structures that can be measured in human long term memory using a psychological method called SDA-M [32].

References

1. Staudacher, E., Gebhardt, M. J., and Dürr, V. (2005) Antennal movements and mechanoreception: neurobiology of active tactile sensors, *Adv. Insect Physiol.* **32**, 49 – 205.
2. Dürr, V. and Krause, A. (2001) The stick insect antenna as a biological paragon for an actively moved tactile probe for obstacle detection., In K. Berns and R. Dillmann, (ed.), Climbing and walking robots - from biology to industrial applications, Proceeding of Fourth International Conference Climbing and Walking Robots (CLAWAR 2001), : Professional Engineering Publishing, Bury St. Edmunds, London pp. 87–96.
3. Ueno, N., Svinin, M., and Kaneko, M. (1998) Dynamic contact sensing by flexible beam, *Mechatronics, IEEE/ASME Transactions on* **3**, 254–264.
4. Kaneko, M., Kanayma, N., and Tsuji, T. (1998) Active antenna for contact sensing, *IEEE Transactions on Robotics and Automation* **14**, 278–291.
5. Lange, O. and Reimann, B. Vorrichtung und Verfahren zur Erfassung von Hindernissen. German Patent 102005005230, (2005).
6. Dürr, V., Krause, A. F., Neitzel, M., Lange, O., and Reimann, B. Oktober 2007 Bionic tactile sensor for near-range search, localisation and material classification., In Karsten Berns and Tobias Luksch, (ed.), Autonome Mobile Systeme 2007, 20. Fachgespräch Kaiserslautern, : Heidelberg: Springer pp. 240 – 246.
7. Krause, A. F. and Dürr, V. (2004) Tactile efficiency of insect antennae with two hinge joints, *Biological Cybernetics* **91**, 168–181.
8. Krause, A. F., Schütz, C., and Dürr, V. (2007) Active tactile sampling patterns during insect walking and climbing., In Proc. Göttingen Neurobiol. Conf. 31. : .
9. Tani, J., Itob, M., and Sugitaa, Y. (2004) Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb, *Neural Networks* **17**, 1273 – 1289.
10. Webb, B. (2004) Neural mechanisms for prediction: do insects have forward models?, *Trends in Neuroscience* **27**, 278–282.
11. Haruno, M., Wolpert, D. M., and Kawato, M. (2001) Mosaic model for sensorimotor learning and control, *Neural Computation* **13(10)**, 2201–2220.

12. Tani, J., , and Nolfi, S. (1999) Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems, *Neural Networks* **12**, 1131–1141.
13. Yamashita, Y. and Tani, J. (2008) Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment, *PLoS Computational Biology* **4 (11)**, –.
14. Tani, J. (2007) On the interactions between top-down anticipation and bottom-up regression, *Frontiers in Neurorobotics* **1**, 2.
15. Werbos, P. (1990) Backpropagation through time: what it does and how to do it, In Proceedings of the IEEE volume **78(10)**, : pp. 1550–1560.
16. Jäger, H. and Haas, H. (2004) Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science* **304**, 78 – 80.
17. Jaeger, H. (2002) Adaptive nonlinear system identification with echo state networks, In S. Becker, S. Thrun, and K. Obermayer, (ed.), Advances in Neural Information Processing Systems, volume **15**, : MIT Press, Cambridge, MA pp. 593–600.
18. Williams, R. J. and Zipser, D. (1989) A learning algorithm for continually running fully recurrent neural networks, *Neural Computation* **1(2)**, 270–280.
19. Jaeger, H. Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "'echo state network"' approach, Technical Report GMD Report 159 German National Research Center for Information Technology (2002).
20. Hochreiter, S., Bengio, Y., P.Frasconi, and Schmidhuber, J. (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, In J. F. Kolen S. C. Kremer, (ed.), A Field Guide to Dynamical Recurrent Neural Networks, IEEE Press.
21. Wiener, N. (1949) Extrapolation, interpolation, and smoothing of stationary time series with engineering applications, Cambridge, Technology Press of Massachusetts Institute of Technology and New York, Wiley, .
22. Steil, J. J. (2004) Backpropagation - decorrelation: online recurrent learning with o(n) complexity, In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on volume **2**, : pp. 843–848.
23. Rolf, M., Steil, J. J., and Gienger, M. (2009) Efficient exploration and learning of whole body kinematics, In IEEE 8th International Conference On Development And Learning : .
24. Reinhart, R. F. and Steil, J. J. (2009) Attractor-based computation with reservoirs for online learning of inverse kinematics, In European Symposium on Artificial Neural Networks (ESANN) – Advances in Computational Intelligence and Learning : .
25. Jaeger, H., Lukosevicius, M., Popovici, D., and Siewert, U. (2007) Optimization and applications of echo state networks with leaky integrator neurons, *Neural Networks* **20(3)**, 335–352.
26. Hogan, N. (1984) An organizing principle for a class of voluntary movements, *Journal of Neuroscience* **4**, 2745–2754.
27. Webots http://www.cyberbotics.com, Commercial Mobile Robot Simulation Software.
28. Steil, J. J. (2007) Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning, *Neural Networks* **20(3)**, 353–364.
29. Kramer, O. (2009) Fast blackbox optimization: Iterated local search and the strategy of powell., In The 2009 International Conference on Genetic and Evolutionary Methods (GEM'09) : in press.
30. Vrugt, J., Robinson, B., and Hyman, J. (2008) Self-adaptive multimethod search for global optimization in real-parameter spaces., *Evolutionary Computation, IEEE Transactions on* **13(2)**, 243–259.
31. Jäger, H. Generating exponentially many periodic attractors with linearly growing echo state networks, technical report 3 IUB (2006).
32. Schack, T. and Mechsner, F. (2006) Representation of motor skills in human long-term memory., *Neuroscience Letters* **391**, 77–81.
33. Krause, A. F., Bläsing, B., and Schack, T. (2009) Modellierung kognitiver Strukturen mit hierarchischen selbstorganisierenden Karten, In I Pfeffer and D Alfermann, (ed.), 41. Jahrestagung der Arbeitsgemeinschaft für Sportpsychologie (asp), volume **188**, : Czwalina Verlag Hamburg p. 91.