EU – FP7

AMARSi
Adaptive Modular Architectures for Rich Motor Skills

ICT-248311

D 5.2

March 2012 (24 months)

Technical report on Hierarchical Reservoir Computing architectures

Francis wyffels, Tim Waegeman, Benjamin Schrauwen – Ghent University
Herbert Jaeger – Jacobs University

Due date of deliverable	1st April 2012
Actual submission date	28th March 2012
Lead Partner	Ghent University
Revision	Final
Dissemination level	Public

Description from Technical Annex

Hierarchical Reservoir Computing can cope with dynamical concepts at several hierarchical timescales and levels of representation, which can be used for unified learning of motor con- trollers and their sequencing. This approach is complementary to an approach to learn hierar- chical classifiers with Reservoir Computing, in the EU project ORGANIC, where UGent and Jac participate, such that a combination can finally lead to a unified action-perception archi- tecture. JAC will taylor its fundamental hierarchical RC learning architectures for unsuper- vised and supervised training of multiscale data toward motor behavior. All feature/concept units are temporal and some may have external (actuator-like) effects. UGent will contribute hierarchical Reservoir Computing networks, based on the combination with unsupervised learning rules such as slow feature analysis, deep belief networks, ... The goal is an architec- ture that can be driven by low-level sensory data and that is able to generate a downward flow of expectation.

Abstract

One approach for building architectures (of which an overview was given in D.6.1) in AMARSi is to use reservoir computing. Here, untrained (or unsupervised trained) recurrent neural networks are used for motion control by learning simple readouts on the dynamic representation generated by the dynamic RNN system. Although single reservoirs are able to generate rich and tunable control patterns (as demonstrated in D.4.1), to allow composition of motion or high-level control, these modules need to be built in an architecture.

An active research area in reservoir computing is to build hierarchical reservoir systems. The main reason for this is that reservoirs basically are band-pass systems and can only represent information in a limited frequency band. If information at both fast and slow timescales needs to be integrated, a natural approach is to build a hierarchical system where each layer operates at a different time scale. The big challenge in these hierarchies is how to learn intermediate representations that link the various layers, and especially how bottom-up and top-down information flows need to be organized.

We believe that these hierarchical reservoir computing systems are good candidates to build (at least part of) architectures required in AMARSi for rich motor control.

In this short deliverable we give an overview of and references to current approaches in hierarchical reservoir computing, several of which have been investigated on speech and handwriting recognition problems in the sister EU project ORGANIC (http://reservoir-computing.org/organic). Many of these hierarchical systems can be used to not only generate dynamical feature hierarchies, but are also able to learn a hierarchy of pattern controller, of special interest to the AMARSi project.

Overview of hierarchical reservoir computing architectures in AMARSi

In this overview section we discuss briefly a number of hierarchical reservoir computing based architectures which are developed and evaluated in the AMARSi project. Several of those have previously been proposed and used for speech and handwriting recognition in the ORGANIC project, but will be used as building block for motor control hierarchies in AMARSi.

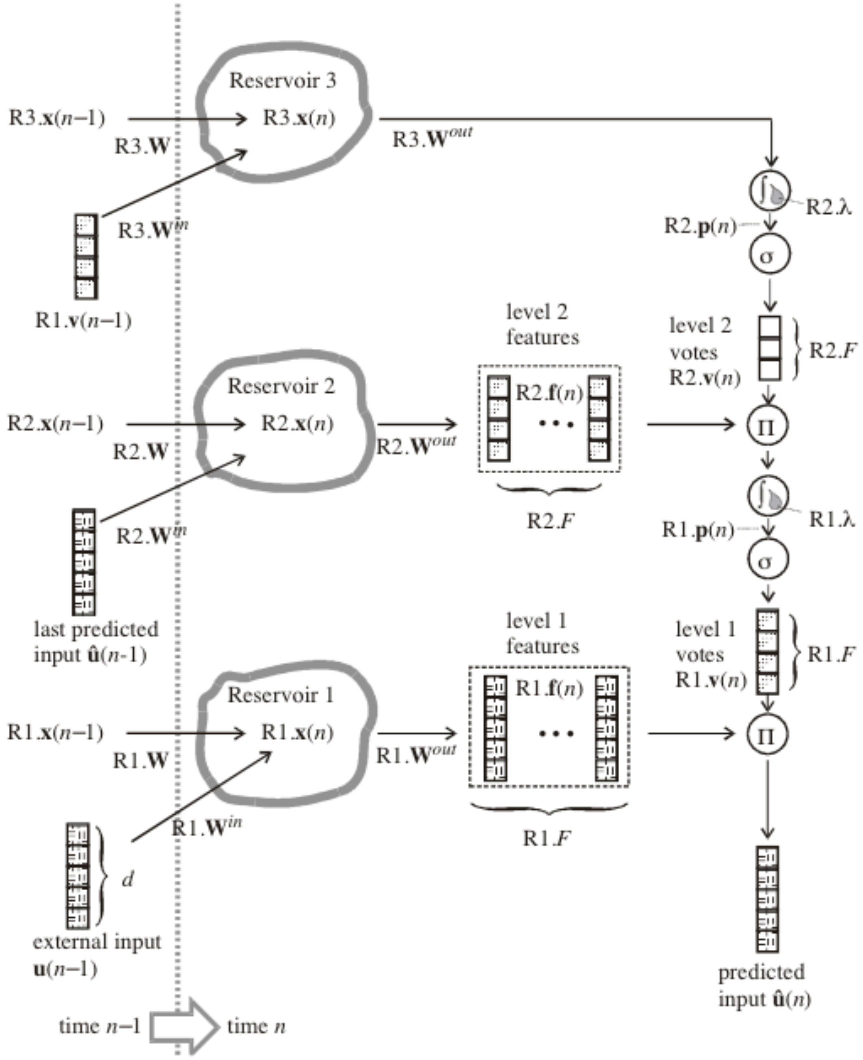
Hierarchical Echo State Networks (JAC)

Although in biology hierarchical information processing is considered natural, in machine learning no system can be found that (1) can deal with high-dimensional temporal data, (2) is able to discover its own dynamical features and (3) uses a computationally efficient, online adaptive learning algorithm. In (Jaeger 2007) a hierarchical learning architecture was proposed that incorporates the three desired features mentioned above. This hierarchy can discover dynamical features in a high-dimensional stationary stochastic time series in such a way that each a feature itself is a time series, the dynamical features are organized hierarchically and the original time series can be reconstructed from all the dynamical features.

In summary, on the lowest level a time series can be decomposed into a set of features and votes. The combination of features and votes should be such that the original signal can be reconstructed. The votes of the lowest level are obtained by the second layer and consists of the combination of second layer features and second layer votes. As previously, the second layer votes are derived by a third layer etc... A schematic overview of the proposed hierarchy can be seen in the figure on the next page.

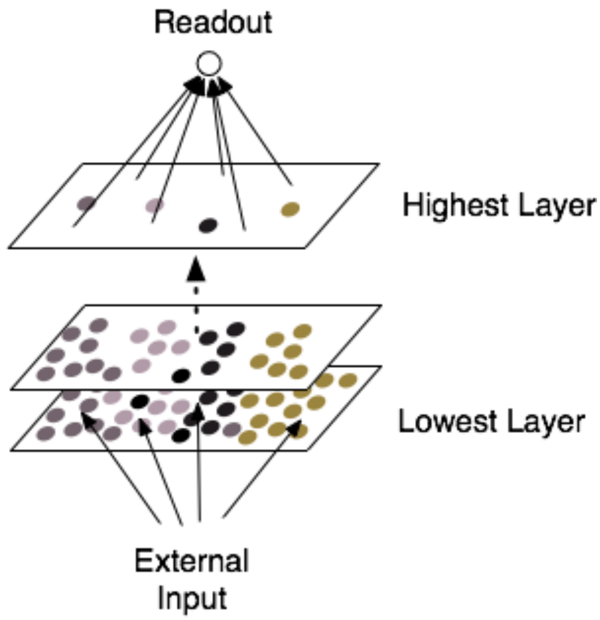
Learning within the hierarchy occurs by changing the output weights of the reservoir systems of each level. This is done on each time step by using stochastic gradient descent. Therefore the gradient is taken with respect to the squared prediction error (= difference between original external input and predicted input).

A formal description of the hierarchy and its learning process can be found in (Jaeger 2007). A downside of this architecture is that it depends on backpropagation-based stochastic gradient descent through several layers of gating, which makes training the hierarchy in real world noisy datasets hard.



Self-Organized Neural Hierarchies (JAC)

A hierarchy of Recurrent Neural Networks (reservoirs) can be envisioned as a layered structure of reservoirs where the lowest level is driven by an external input, and each higher level is excited by the activation of its lower neighbouring layer:

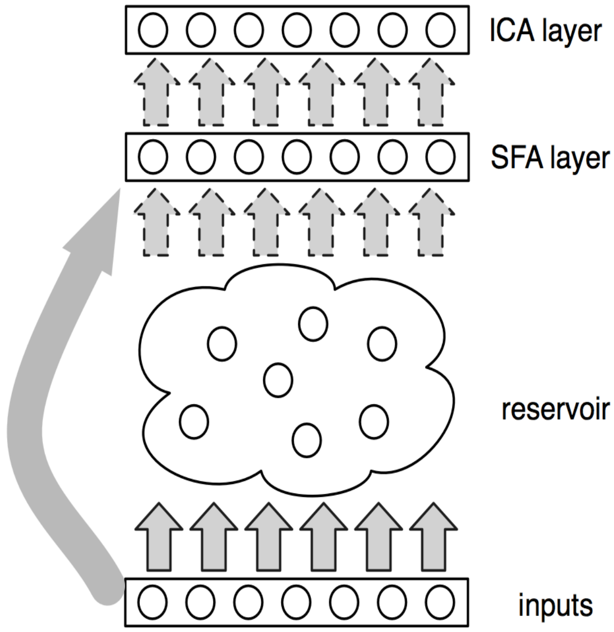


The internal reservoir weights are trained layer-wise in an unsupervised way by applying the recurrent self-organizing maps (recurrent SOMs) algorithm which is the classical SOM algorithm but generalized from the conventional static pattern domain to temporal patterns. For this, the reservoirs need to be constructed from radial basis function (RBF) units. The readout on the top layer of the hierarchy, is trained supervised by using linear regression.

The idea behind this hierarchy is to achieve a sparse and localised representation of temporal invariances in the driving input at the readout layer. Thus making it easier to identify which representation is invariant. From bottom to top, the representation becomes gradually more sparse and localized. Tested configurations of this architecture can be found in (Lukosevicius 2010 and Lukosevicius 2012)

RC-SFA architecture for unsupervised learning of a spatial representation of a robot environment (UGent)

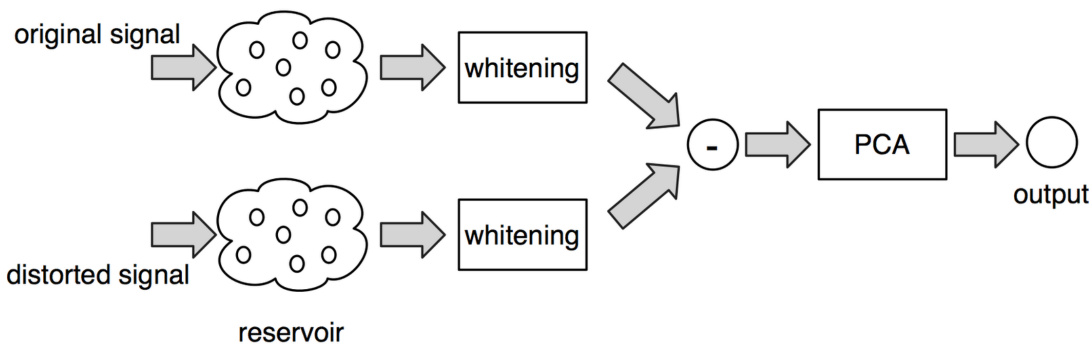
Experiments with rats in open fields have shown the existence of hippocampal place cells. These cells form an implicit spatial representation of an animal's environment (Moser et al., 2008). Using a constrained environment for rats it is shown that hippocampal CA1 cells and entorhinal cortex cells code for spatial information on a way dependent on the rat's path or behavior. This assumes that if the entorhinal cortex and the hippocampus are related to path planning, these structure should reflect where the animal intends to go (Frank et al., 2000). This can be mimicked by using a three level layered hierarchy which consists of a reservoir system, a slow feature analysis layer and an independent component analysis layer:



Sensory information is mapped into a high-dimensional non-linear dynamical space. Because of the dynamical nature of the reservoir system, the entire hierarchy can rely on both current input changes as changes of the past. From the reservoir system slow varying signals are derived in an unsupervised way using Slow Feature Analysis (Wiskott and Sejnowski, 2002). This has the advantage that slow varying properties, such as environmental changes, can be tracked from (often) fast varying sensor information. From the obtained slow varying signals, a linear combination is taken using by the Independent Component Analysis layer. This learns in an unsupervised way a sparse coding of the slow varying signals, resulting in units that are activated only for a specific position in the environment. Or in other words, place cells can be derived from them. A more detailed description including results on robots can be found in (Antonelo and Schrauwen, 2012). This building block would allow to build hierarchical reservoir system by stacking these modules.

Spatial SFA convolutional reservoir hierarchies (JAC)

Signal processing of complex real world data is a difficult task for computational intelligence due to the many distorting variations in the data. In (Sakenas 2010) a method for learning temporal features which are invariant to the distortions in the input data was proposed. In the proposed approach a distorted signal together with its original example is first expanded by using a random non-linear dynamical system. Secondly the states are normalized by a whitening step such that all resulting signals has a unit variance in all directions. Thirdly, the difference between the whitened states obtained from the distorted signal and the original signal is taken. Finally, principal component analysis is applied on this whitened state difference The procedure is illustrated in the following Figure:



The core idea of the proposed technique to find linear projections from a reservoir such that the reservoir states vary minimally when the same signal with a distortion is processed. The procedure is very comparable with Slow Feature Analysis (Berkes and Wiskott 2005), but instead of of finding the temporally slowest features, the features that are most invariant to spatial transformations are found. Examples on synthetic data and handwriting recognition can be found in (Sakenas 2010).

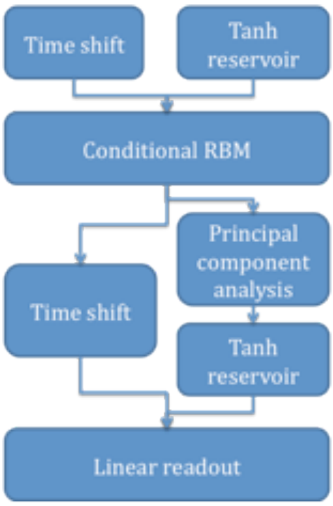
Temporal Reservoir Machine (UGent)

Guided by the advantageous and disadvantageous of previous approaches RBM-based approaches for time series modeling such as the CRBM and RTRBM, (Schrauwen and Busing, 2009) introduced a modification of these models called Temporal Reservoir Machine (TRM). The TRM is aimed at overcoming the limitations of a fixed model order (memory depth) of the CRBM and at the same time circumventing the problems associated with learning via BPTT which occur in the RTRBM, by using strategies from RC.

The basic idea of the TRM is the following. In addition to the visible and hidden random variables we introduce bias variables for each time step. These determine the biases of the visible and hidden variables of the RBM via a linear mapping. Further, the biases are given by a deterministic function of the previous visible variables. Inspired by the ESN, we consider here the setup where the bias is given by the activation of a recurrent neural network that receives the previous visible variables as input. In the spirit of the ESN approach, the reservoir weight matrices are randomly generated and they remain untrained. Only the connections from the reservoir to the visible and hidden variables are adapted using contrastive divergence. Learning, sampling and inference in this model is tractable and efficient.

Mimicking the hierarchical structure of a deep belief networks, TRMs can be stacked yielding a multi-layer model for sequential data analogous to the CRBM. Learning is performed layer-wise (“greedy”), i. e. at each layer a full TRM is trained using the inferred distribution over the binary hidden variables of the next lower layer as real-valued input data. The rational behind this hierarchical structure is that higher levels are thought to be advantageous for representing more abstract features and statistical relationships on longer time scales. Sampling from the multi-layer model is then performed by sampling from the undirected TRM at the top layer and a down-pass using directed connections.

In general, input is fed into an ESN via a randomly generated input matrix. This works well if the input dimension is reasonably low. If however the input dimension is high, especially if there is a lot of redundancy across input dimensions, such a random input mapping will drastically degrade the short-term memory of the ESN. Intuitively speaking, this is because redundant information is pushing the ESN dynamics in random directions, thereby destroying memory of past inputs. Therefore directly applying an ESN as the reservoir in higher layers of a TRM, where the visible layers often have a high dimension, will lead to a poor modeling performance because all memory of previous visibles is quickly lost. A straightforward solution for solving this problem is to apply standard dimensionality reduction techniques such as Principle Component Analysis to first significantly reduce the dimension of the visibles before feeding them into the ESN. The diagram below shows a single layer TRM representation as can be implemented in the ORGANIC Oger toolbox:



This architecture has with success been used to process speech information in the associated ORGANIC project (see (Schrauwen and Busing, 2009)). In currently unpublished work, Schrauwen et al. has shown that the TRM model can also effectively be used to model motion capture data, outperforming CRBMs. In this mode, the model can not only be used to classify motion capture data, but can also be used as a generative model to generate motion patterns sampled from the distribution of motion patterns seen during training. This would allow to seamlessly switch between e.g. different walking styles if an additional variable would be introduced to identify the style.

Neural Hierarchy of Timescales for Motor Control (UGent)

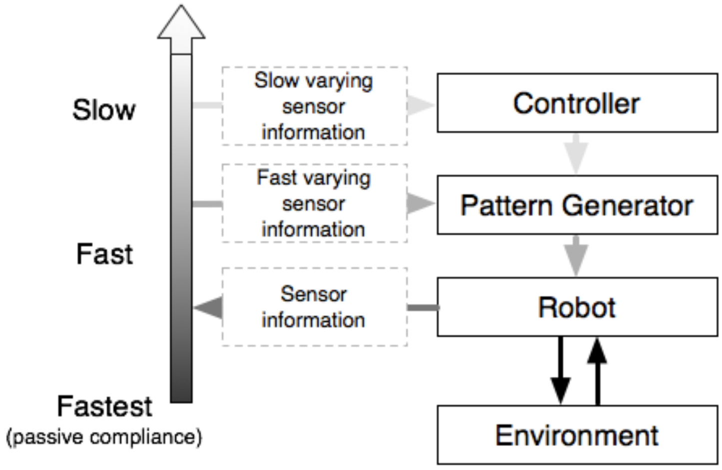
All previous hierarchical reservoir computing approaches are homogeneous in nature, where each layer basically is using the same learning paradigm and set up. The following architecture is heterogeneous in nature and is more specific in nature.

According to (Kiebel et al. 2008), many aspects of brain functions can be explained by a hierarchy of temporal scales at which representations of the environment evolve. The higher level encodes slower contextual changes in the environment or body while at the lower level faster variations due to sensory processing are encoded.

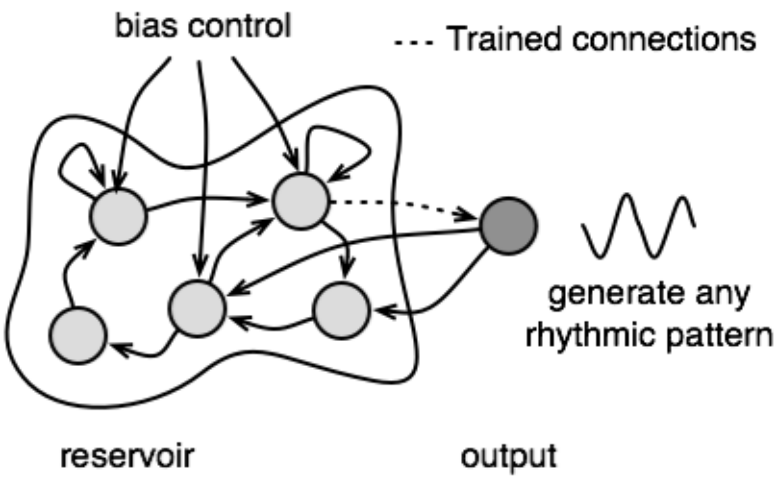
Other biological research suggests that specialized neural circuits, so called central pattern generators (CPGs), located in the spinal cord are responsible for generating rhythmic activations needed for body function including the contractions of a heart or lungs and control of muscles for walking (Stein et al. 1999). Implying that some aspects of brain functions are offloaded to regions outside of the brain. Researchers discovered (Cohen et al. 1980) after extracting and isolating the spinal cord from the body, that the spinal cord, when excited with electrical stimulations, will produce fictive locomotion. This indicates that sensory information is not needed to generate such rhythmic patterns. However, it plays a crucial role in shaping the generated pattern to keep the coordination between the CPGs and the body. These findings suggest that locomotion can be represented by both a modular and hierarchical structure. More specifically, a hierarchy of modules which interact with each other on different timescales, simplifies high level control and at the same time allows fast action against perturbations.

Sensory information propagates through this hierarchy and is handled in a multi-resolution manner. Slight irregularities in the terrain are compensated very fast by the morphology of the legs without the need of the brain to intervene. Larger irregularities are handled by a slower reflex motion of which the vertebra becomes finally conscious at the highest but slower contextual level.

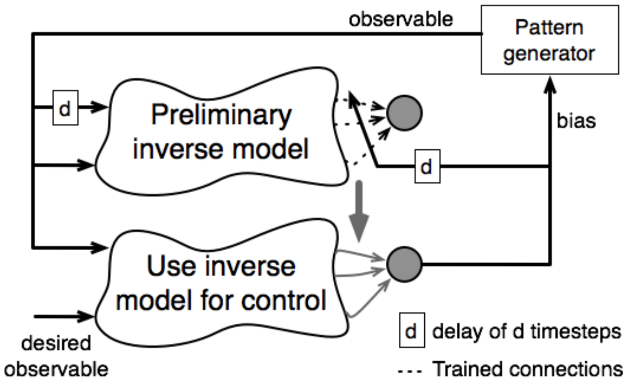
In this approach a multi-timescale control hierarchy is investigated and applied on a leg of the AMARSi Oncilla robot. Each of the layers in the proposed hierarchy uses a random dynamical system of which only the readout layer is trained (eg. Reservoir Computing systems (Verstraeten et al. 2007)). On the lowest level, the fastest timescale, the passive compliance of the leg interacts with the environment. The leg is driven by a pattern generator which generates learned rhythmic motor signals and gets feedback from the rotary encoders in the leg. On the highest level, we use a controller which reacts to slower contextual changes. This hierarchy separates the motor commands from the functional control which is done by the high level controller. This neural hierarchy of timescales can be illustrated as follows:



Corresponding to the biological central pattern generator we use a recurrent neural network based pattern generator of which only the output weights are trained (according to the Reservoir Computing philosophy). Such pattern generator is shown in the following illustration:



The dynamics of the network can be used to embed any imposed rhythmic pattern. Each neuron contains a bias term which can be controlled to achieve a desired modulation of the generated pattern. For instance, if the frequency or amplitude of the generated motion (not instantaneous but changes are slowly observed) needs to be controlled, this can be achieved by bias control. By determining which neurons are sensitive to such slow observables, a modulation vector can be calculated which is then controlled by a feedback controller. As shown in the following figure, this feedback controller includes two identical networks which are used to host an inverse model of the control problem.



In this hierarchy this control problem can be formulated as how much of bias needs to be added to the pattern generator neurons to achieve the desired slow observable. Both networks are identical and initialized randomly. As a result, initially the model does not correspond to a model of the control problem. However, by controlling the pattern generator randomly both its control value and the corresponding response can be used to improve the model. Each iteration new control pairs are observed which improves the model and will eventually allow the controller to track the desired properties of the generated pattern.

A detailed description and some preliminary experiments can be found in (Waegeman et al. 2012)

The mystery of top-down

Research within ORGANIC has led to an acute awareness of the importance, but also of the elusive nature, of top-down processing pathways in hierarchical architectures. In this section we report on preliminary insights that were obtained at JAC concerning the theme of top-down pathways.

The closer one looks at top-down processing, the more elusive this phenomenon becomes. An extensive literature search in the areas of machine learning, the cognitive neurosciences and robotics revealed that a large variety of functionalities have been associated with top-down processing:

- In the original, influential subsumption architecture for robot control proposed by (Brooks, 1989), higher levels (of control) *override the output* of lower levels.
- In some robot control architectures (Takahashi and Asada, 1999)(Jaeger and Christaller, 1998), higher levels of control *activate in a graded fashion* lower-level control modules.
- In classical-engineering hierarchical control architectures, or many AI-planning architectures, the top-down influence is *setting of target trajectories* or *subtasks* (e.g. (Albus, 1993)).
- In the Perceptual Control Theory (PCT) of (Powers, 1989), hierarchical control levels are paired with hierarchical perception levels; each level has an expectation about its perception and acts toward matching the perception with the expectation. What is passed down is *perceptual expectations* = targets for lower-level perceptual features.
- Within the conceptual framework of the bi-directional theory of sensory-motor integration, Miyamoto et al have proposed a homogeneous modular architecture (Miyamoto et al., 1996) where the top-down information consists in the cascaded construction of an *inverse kinematic model*. The connections between levels here are not passive information channels but active transformations.
- In the Bayesian approach to control by (Toussaint and Goerick, 2010) the layering does not refer to similar modules as in the other work listed here, but to a decomposition of a multiply-constrained motion task into a hierarchy of task and joint variables. Here, a "higher" level of variables passes down *messages* in the technical

sense of message passing algorithms for approximate inference in Bayesian networks.

- In some models of biological motor control (e.g. (Degallier et al. 2011)), higher levels pass down *control parameters* to lower-level motion pattern generators, which serve to *modulate* the dynamics of these generators, for instance adapting stepsize in gait production or target equilibrium positions in reaching motions.
- An illuminating mix is found in a survey study of (Prescott et al., 1999). The authors set forth to argue that biological behavior organization (as revealed by neuroanatomy and ethological field studies) can be mapped to Brooks' subsumption architecture. However, in two classical animal studies (behavioral repertoire of the herring gull, and rat defense system) surveyed in that paper, quite different functionalities are assigned to the various levels of behavioral organization. A closer inspection of what type of information is passed downwards reveals a heterogeneous collection. According to the case studies, what is variously passed down is described as *coordinating* lower levels; *substituting input*; *inhibiting*; and *gain control*. Thus, in fact, heterogeneity is found where homogeneity was sought – although this is not acknowledged by the authors.
- Today's mainstream approach for understanding top-down information passing, both in machine learning and theoretical neuroscience, is the Bayesian view of statistical information processing (Kiebel et al., 2008)(Tenenbaum et al. 2006). This view posits that higher levels pass down *statistical priors* (technically: hyperdistribution parameters, intuitively: context information). This is also the mathematical model underlying the (restricted and classical) Boltzmann machine models which are currently so popular in machine learning.
- In the human brain, top-down mechanisms associated with *goal-oriented attention* have been investigated (Posner and Petersen, 1990).
- Many researchers follow up on the general idea to understand cognition in terms of enabling the agent to make predictions in all situations of life (Hoffman, 1993)(Clark, 2012). Often but not always related to Bayesian models of information processing, this view perceives the nature of top-down information as providing *prediction error* information, allowing the lower layer to recalibrate or change modes.

The list illustrates the wide scope of functionalities that have been associated with top-down processing streams. Therefore, one fundamental challenge in architecture design is *to reach a better understanding of the functions served by top-down processing*. Several of the items listed above appear to be intuitively mutually related, and one could form groups of related functionalities, as a first step of bringing some order into that dazzling multitude:

- The functionalities of *prediction*, *statistical priors*, *perceptual expectations*, *prediction errors* appear to have a similar "flavor" of the higher level maintaining, and sharing with the lower layer, a more abstract representation of the information that is being processed in the lower one.
- The functionalities of *overriding output*, setting *control parameters*, setting of *target trajectories* or *subtasks*, and of *gain control* seem to share aspects of generic control hierarchies.

Other functionalities are not easily merged and stand alone: *inverse kinematic models* serve the purpose of coordinate transformations between layers – which one could generalize to *recoding of representations* beyond the typical robotics transformations; the *message passing* in Bayesian control appears to be a technical-algorithmical feature; *attention* may be a fundamental category of its own standing which has just not yet received due attention in machine learning and robotics.

Beyond this listing and tentative grouping we are at this time unable, like everybody else, to provide keys to a more integrated functional theory of top-down processing in hierarchies. However, we are able to provide another, albeit disturbing, insight. Regardless of the functionality that one might wish to realize by top-down interactions, there arise serious mathematical and algorithmical problems when one implements and runs multiscale learning architectures with bottom-up and top-down interactions between layers. These problems are generic consequences of establishing a complex dynamical system which operates on several timescales in different, bi-directionally interconnected layers. When such a system is trained, its parameters are gradually adapted, which necessarily leads to *bifurcations* if the ultimate system that one wishes to achieve is qualitatively different from the starting system. And of course, one wishes the ultimate system to be qualitatively different from the initial random guess from which one typically starts.

Bibliography

Albus, J. S. (1993): A Reference Model Architecture for Intelligent Systems Design. In Antsaklis, P.J., Passino, K.M. (Eds.), An Introduction to Intelligent and Autonomous Control. Kluwer Academic Publishers, 1993, Chapter 2, 27-56

Antonelo, E. A. and Schrauwen, B. (2012). Learning slow features with reservoir computing for biologically-inspired robot localization. Neural Networks, 25:178-190

Berkes, P. and Wiskott, L. (2005). Slow feature analysis yields a rich repertoire of complex cell properties. Journal of Vision 5(6):579-602

Brooks, R. (1989): The whole Iguana. In: M. Brady (ed.), Robotics Science, MIT Press, 432-456, 1989

Clark, A. (2012): Whatever Next? Predictive Brains, Situated Agents, and the Future of Cognitive Science. Behavioral and Brain Sciences, to appear.

Cohen A. and Wallen P. (1980): The neuronal correlate of locomotion in fish. Experi- mental brain research, 41(1):11–18

Degallier, S., Righetti, L., Gay, S., Ijspeert, A. (2011): Toward simple control for complex, autonomous robotic applications: Combining discrete and rhythmic motor primitives. Autonomous Robots, May 2011, 1-27

Frank, L. M., Brown, E. N., and Wilson, M. (2000). Trajectory encoding in the hippocampus and entorhinal cortex. Neuron, 27(1):169-178

Hoffmann, J. (1993): Vorhersage und Erkenntnis. Hogrefe Verlag

Jaeger, H., Christaller, T. (1998): Dual Dynamics: Designing Behavior Systems for Autonomous Robots. Artificial Life and Robotics 2, 108-112

Jaeger, H. (2007): Discovering multiscale dynamical features with hierarchical Echo State Networks. Technical report No. 10, Jacobs University Bremen

Kiebel, S.J. and Daunizeau, J. and Friston, K.J. (2008): A hierarchy of time-scales and the brain. PLoS computational biology 4(10)

LeCun, Y., Bottou, L., Bengio, J., Haffner, P. (1998). Gradient-Based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324

Lukosevicius, M. (2010): On self-organizing reservoirs and their hierarchies. Technical report No. 25, Jacobs University Bremen

Lukosevicius, M. (2012): Reservoir Computing and Self-Organized Neural Hierarchies. PhD thesis, Jacobs University Bremen, Bremen, Germany

Miyamoto, H., Schaal, S., Gandolfo, F., Gomi, H., Koike, Y., Osu, R., Nakano, E., Wada, Y., Kawato, M. (1996): A Kendama learning robot based on bi-directional theory. Neural Networks 9(8), 1996, 1281-1302

Moser, E. I., Kropff, E. and Moser, M.-B. (2008). Place cells, grid cells and the brain's spatial representation system. Annual Reviews of Neuroscience, 31:69-89

Posner, M. I., Petersen, S. E. (1990): The attention system of the human brain. Annu Rev Neurosci 13, 25—42

Powers, W. T. (1989): An Outline of Control Theory. In: Powers, W. T. (ed.), Living Control Systems. The Control Systems Group, Inc., 1989, 253-293

Prescott, T. J., Redgrave, P., Gurney, K. (1999): Layered control architectures in robots and vertebrates. Adaptive Behavior 7(1), 99-127, 1999

Sakenas, V. (2010). Distortion invariant feature extraction with echo state networks. Technical Report No. 24, Jacobs University Bremen

Schrauwen, B. and Buesing, L. A hierarchy of recurrent networks for speech recognition. NIPS workshop on deep learning (2009)

Stein P. (1999): Neurons, networks, and motor behavior. The MIT Press

Takahashi, Y., Asada, M. (1999): Behavior Acquisition by Multi-Layered Reinforcement Learning. Proceeding of the 1999 IEEE International Conference on Systems, Man, and

Cybernetics, 716-721
Tenenbaum, J. Griffiths, T. L., Kemp, C. (2006): Theory-based Bayesian models of inductive learning and reasoning. Trends in Cognitive Science 10(7), 309-318
Toussaint, M., Goerick, C. (2010): A Bayesian view on motor control and planning. In Sigaud, O., J. Peters (Eds.): From motor to interaction learning in robots. Studies in Computational Intelligence, Vol. 264, 227-252, Springer, 2010
Verstraeten D. , Schrauwen B., D'Haene M., and Stroobandt D. (2007): An experimental unification of reservoir computing methods. Neural Networks, 20:391–403
Waegeman T., Wyffels F., Schrauwen B. (2012): Towards a Neural Hierarchy of Time Scales for Motor Control. The International Conference on Simulation of Adaptive Behavior (submitted)
Wiskott, L. and Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. Neural Computation, 14(4):715-770