ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 105-110

A RNN based Control Architecture for Generating Periodic Action Sequences

Thorsten Kolb[1], Winfried Ilg[2], Jörg Wille[1]

[1]Numerical and Applied Mathematics, Brandenburg University of Technology at Cottbus, 03044 Cottbus, Germany
[2]Department of Interactive Diagnosis and Service Systems, Research Center for Computer Science, 76131 Karlsruhe, Germany

Abstract. We introduce a type of fully connected Recurrent Neural Networks (RNN) with special mathematical features which allows us to determine its qualitative dynamical behaviour. Using these properties we describe a learning framework for the generation of sequences to be applied to nonlinear control problems. The potential of this approach is demonstrated by applying the learning framework to the adaptive leg control of the six-legged walking machine LAURON II.

1. Introduction

During the last years the application of RNNs gained constantly growing attention. The first attempt to formulate a learning algorithm for fully connected RNNs with deterministic discrete time dynamics (BPTT) dates back to one of the original publications of the backpropagation algorithm [12]. For practical applications convergence of this algorithm proved to be too slow and could not be guaranteed. So one favorite starting point to handle the classification or generation of time series was the application of several extensions to feedforward networks [5, 9]. Doing so questions about the computational power and convergence of RNNs were neglected in most cases. Only in recent years researchers became interested in this field, inspired for example by questions about generalizations of the Hopfield model [3, 6] or the description of chaotical behaviour [11, 7]. Application-oriented approaches were presented in terms of recombination of oscillators [1] and using biologically inspired time continuous models [10, 14]. On the other hand neuroethological research has shown that basic motion patterns are generated by small groups of neurons representing neural oscillators [13, 2]. The aim of this paper is to propose a time-discrete RNN model which is intended to be used for the generation of nonlinear time sequences based on a time invariant characterization in a biologically plausible manner.

2. A RNN Model for Generating Oscillating Behaviour

We use a fully connected deterministic RNN with d units and continuous outputs [8]. Let $\boldsymbol{x}_n \in \mathbb{R}^d$ denote the current state of the RNN at time step n, i.e. the vector of

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 105-110

outputs of each individual neuron at time step n. We assume a synchronous iteration mode. Therefore we can determine the RNN state in the next time step $n+1$ using the state transfer function $\boldsymbol{f}(\boldsymbol{x}) := \boldsymbol{f}_{act}(\mathcal{W}\boldsymbol{x})$ where $\mathcal{W} \in \mathbb{R}^{d\times d}$ is an arbitrary matrix and $\boldsymbol{f}_{act} : \mathbb{R}^d \to \mathbb{R}^d$ is a nonlinear normalization function defined as

$$\boldsymbol{f}_{act}(\boldsymbol{x}) = \begin{cases} \frac{\boldsymbol{x}\tanh|\boldsymbol{x}|}{|\boldsymbol{x}|}, & \boldsymbol{x} \neq \boldsymbol{0} \\ \boldsymbol{0}, & \boldsymbol{x} = \boldsymbol{0} \end{cases} \tag{1}$$

It is important to note that the proposed activation function $\boldsymbol{f}_{act}$ is *global* with respect to the RNN, i.e., it computes its new state using information about internal states of each individual neuron. This might cause problems in very large networks but the RNNs used here for sequence generation are mostly of sizes about ten units. Furthermore we can imitate the behaviour of an RNN of d units and global update by an RNN with $2d + 1$ units and local update functions in each neuron which is illustrated by figure 1.

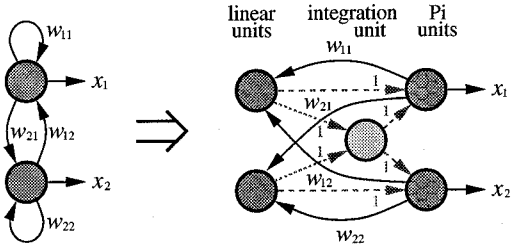

Figure 1: Mapping of RNN with global update to RNN with local update

Below we will describe some mathematical properties of the proposed RNN model which will prove to be helpful in the context of adaptive control systems.

Lemma 1 *Let $\boldsymbol{f}_{act}^{(k)}(\boldsymbol{x})$ denote component k of the result of applying $\boldsymbol{f}_{act}$ to $\boldsymbol{x}$. Let $\boldsymbol{v} \in \mathbb{R}^d$ with $|\boldsymbol{v}| = \|\boldsymbol{v}\|_2 = 1$. Then $\forall 1 \leq k \leq d$*

$$\left|\frac{\partial \boldsymbol{f}_{act}^{(k)}}{\partial \boldsymbol{v}}(\boldsymbol{x})\right| \leq 1 \quad \textit{and} \quad \max\left|\frac{\partial \boldsymbol{f}_{act}^{(k)}}{\partial \boldsymbol{v}}(\boldsymbol{x})\right| = \left|\frac{\partial \boldsymbol{f}_{act}^{(k)}}{\partial \boldsymbol{v}}(\boldsymbol{0})\right| = 1 \tag{2}$$

Lemma 2 *$\boldsymbol{f}_{act}(\delta\boldsymbol{x})$ is contractive for every $0 \leq \delta < 1$. In the limit case $\delta = 1$ $\boldsymbol{f}_{act}(\delta\boldsymbol{x})$ is not contractive, but it still holds that $\boldsymbol{x} = \boldsymbol{0}$ is a global asymptotic stable fixed point of $\boldsymbol{f}_{act}$.*

The first statement of lemma 2 follows directly from lemma 1. To see the correctness of the second statement we may assume an arbitrary point $\boldsymbol{x} \in \mathbb{R}^d$ which is not the origin. Let $g : \mathbb{R} \to \mathbb{R}$ be defined by $g(x) = \frac{\tanh(x)}{x}$ for $x \neq 0$ and $g(x) = 0$ for $x = 0$ then $\boldsymbol{f}_{act}(\boldsymbol{x}) = g(|\boldsymbol{x}|) \cdot \boldsymbol{x}$ and therefore $\boldsymbol{f}_{act}^n(\boldsymbol{x}) = g^n(|\boldsymbol{x}|) \cdot \boldsymbol{x}$. Furthermore $g'(x) < 1$ for $x \neq 0$ and $g'(0) = 1$, $g''(0) = 0$, $g'''(0) < 0$ which means that g is globally asymptotically stable with the attractive fixed point $x = 0$. Therefore

$$\lim_{n\to\infty} \boldsymbol{f}_{act}^n(\boldsymbol{x}) = \boldsymbol{x} \cdot \lim_{n\to\infty} g^n(|\boldsymbol{x}|) = 0 \cdot \boldsymbol{x} = \boldsymbol{0} \tag{3}$$

which concludes the proof of lemma 2.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 105-110

Theorem 1 (Eigenvector Theorem) *$\boldsymbol{f}_{act}(\mathcal{W}\boldsymbol{x})$ retains the eigenvectors of $\mathcal{W}$, i.e., every local linearization of $\boldsymbol{f}_{act}(\mathcal{W}\boldsymbol{x})$ has the same eigenvectors as $\mathcal{W}$.*

Theorem 2 (Eigenvalue Theorem) *Suppose $\lambda = re^{i\phi}$ is eigenvalue of $\mathcal{W}$. Then there exist a decreasing function $\beta : \mathbb{R}_0^+ \to (0,1]$ with $\overline{\lambda} = \beta(|\mathcal{W}\boldsymbol{p}|)re^{i\phi}$ is eigenvalue of the local linearization of $\boldsymbol{f}_{act}(\mathcal{W}\boldsymbol{x})$ in $\boldsymbol{p} \neq \boldsymbol{0}$. The local linearization in $\boldsymbol{p} = \boldsymbol{0}$ has the same eigenvalues as $\mathcal{W}$.*

Both theorems can be proven by tracing back the linearization at an arbitrary point $\boldsymbol{p} \in \mathbb{R}^d$ to the form $\boldsymbol{l}(\boldsymbol{x}) = \alpha(|\mathcal{W}\boldsymbol{p}|) + \beta(|\mathcal{W}\boldsymbol{p}|)\mathcal{W}\boldsymbol{x}$ with nonlinear differentiable functions $\alpha, \beta : \mathbb{R} \to \mathbb{R}$. Since α and β are constants to a given linearization the statements of the theorems result from basic linear algebra. The relation given by the Eigenvalue Theorem can be used to derive the following

Lemma 3 *Suppose the eigenvalues of $\mathcal{W}$ are ordered by $|\lambda_1| \geq \cdots \geq |\lambda_d| > 0$. Then for every local linearization with eigenvalues corresponding to theorem 2 holds $|\overline{\lambda}_1| \geq \cdots \geq |\overline{\lambda}_d| > 0$. Furthermore is $|\lambda_k| \geq |\overline{\lambda}_k|$ for $k = 1, ..., d$.*

The theoretical investigations given above result in the following dynamical properties which can be used to describe the qualitative dynamical behaviour of RNN.

Theorem 3 (Global Stability Theorem) *Suppose the eigenvalues of $\mathcal{W}$ are still ordered by their absolute values. Whenever $|\lambda_1| < 1$, i.e., the dynamical behaviour of the linear difference equation $\boldsymbol{x}_{n+1} = \mathcal{W}\boldsymbol{x}_n$ is globally attractive towards the origin the same is valid for the RNN. If $|\lambda_1| = 1$, i.e., the dynamical behaviour of the difference equation is stationary in at least one dimension the dynamics of the RNN are still globally attractive towards the origin.*

Theorem 4 (Oscillation Theorem) *If there exists a subset of eigenvalues of $\mathcal{W}$ with $|\lambda_k| > 1$, $1 \leq k \leq r$, then the RNN will exploit quasiperiodic behaviour on the space built of the eigenvectors corresponding to $\lambda_1, ..., \lambda_r$. The absolute value of λ_k determines the attraction rate of the corresponding attractor.*

The term quasiperiodic behaviour includes periodic (i.e. oscillatory) and fixed point behaviour as special cases. The following lemma is a direct consequence of the Global Stability Theorem and the Oscillation Theorem.

Lemma 4 *The proposed RNN model cannot exploit chaotical behaviour.*

3. A Learning Framework for Control Applications

Instead of learning a large neural network with recurrences which has to learn all the dynamical behaviour at once, we extract a normalized oscillating behaviour which describes the qualitative behaviour of the system. This behaviour is realized using the proposed RNN model and the adjustment to the task is accomplished by a simple feedforward network. Since we have a nonlinear oscillator with a nonlinear postprocessing by the feedforward network we can restrict ourselves to affine pre- and postprocessing

$$\boldsymbol{f}_{pre}(\boldsymbol{x}) = \mathcal{A}\boldsymbol{x} + \boldsymbol{a} \quad \text{and} \quad \boldsymbol{f}_{post}(\boldsymbol{x}) = \mathcal{B}\boldsymbol{x} + \boldsymbol{b} \quad \mathcal{A}, \mathcal{B} \in \mathbb{R}^{d \times d},\ \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d \qquad (4)$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 105-110

of the system state space. In system identification tasks we can choose the postprocessing as the inverse of the preprocessing. Suppose the output of the feedforward network is denoted by $f_{ff}(x)$, then the output o_n of the control system at timestep n depending on the initial system state s_0 and the time independent control parameter vector c is given by

$$o_n = f_{post} \circ f_{ff}(c, f^n \circ f_{pre}(s_0)) \tag{5}$$

The resulting control system is illustrated by figure 2. To construct an initial weight matrix $\mathcal{W}$ of the RNN one can combine several rotation and scaling matrices or determine an initial $\mathcal{W}$ directly from a set of given eigenvalues and eigenvectors. Given a representative set of optimal sequences it is also possible to estimate the characteristic system using finite differences or HOUSEHOLDER transform. According to the second statement of the Oscillation Theorem it is possible to revise $\mathcal{W}$ to match a given attracting behaviour by simply multiplying $\mathcal{W}$ with a factor $\alpha > 1$ if the global attraction behaviour is to be adapted or to enlargen the eigenvalues of special eigenvectors of $\mathcal{W}$ to achieve a specific attracting behaviour.

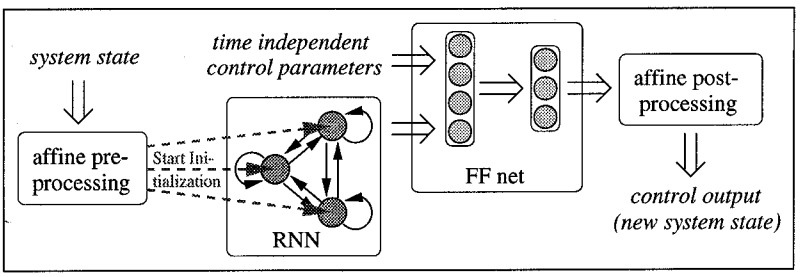

Figure 2: Integration of the RNN in an adaptive control system.

Now we can train the feedforward net with one of the usual backpropagation algorithms. We need to establish an external analysis module which monitors the differences between the system state predicted by the control system and the real system state according to sensory information. If the differences become too large the analysis module has to decide which component of the control system should be recalibrated. This can be achieved by estimating the characteristics of the dynamical system in the same way the initial weight matrix of the RNN is obtained. In case the analysis module detects changed system characteristics it derives a new weight matrix for the RNN in parallel to the running system and updates the weight matrix at an arbitrary time step. If the characteristics have not changed the occured differences have to be minimized by further training of the feedforward network. The result of this approach is a fast adaptation of the control system in case of varying environmental conditions.

The process of developing a control system is thus a combination of constructive elements using context information and knowledge about dynamical characteristics, and adaptive elements using the advantages of feedforward networks, both in a context of neural networks.

The applicability of our approach is tested by the generation of joint angel sequences for adaptive leg control as illustrated by figure 3 of the six legged walking machine LAURON II ([4]), which was built at FZI Karlsruhe, Germany. The aim of

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 105-110

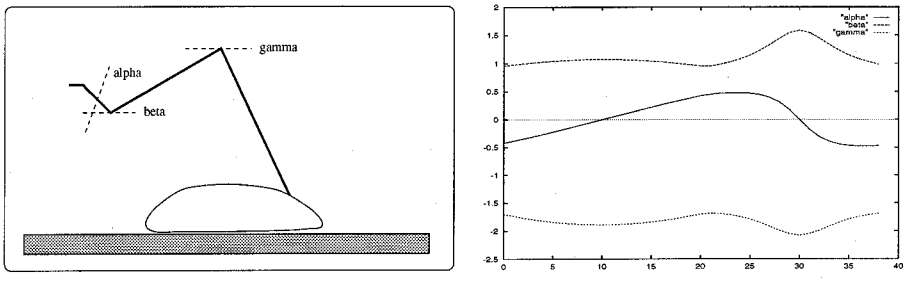

Figure 3: The left figure shows a schematic drawing of a single leg. In the right figure the sequences of the joint angles for one complete leg trajectory are shown (in radians).

the research project LAURON is to realize an autonomous legged robot which can adapt its behaviour online while exploring unstructured environments. This leads to the following requirements for the representation of leg trajectories.

- Memory and computational efficiency to match the limited computing resources and hard realtime requirements of an autonomous system.
- Robust and flexible trajectories. Main problems are the large variety of possible initial situations and disturbances during motion.
- Integration in hybrid learning architectures. To control the walking behaviour of LAURON II we have to integrate the single leg controller into a control system for leg-coordination and path planning.
- Integration of available domain knowledge about the dynamical behaviour. Especially for an online learning process on the walking machine it is extremely important to guarantee a basic behaviour even during the learning process.

Using our method we were able to construct a family of oscillators that fulfill the given specifications in every respect. We obtained some standard trajectories for walking on even terrain by *programming by demonstration*. From these trajectories we extracted the description of the oscillator and the training examples for the feed-forward network. It showed up that only three neurons are necessary for the RNN to describe the qualitative dynamical behaviour given by the standard trajectories.

In figure 4 the results of some experiments to generate leg trajectories are shown. A two dimensional grid of starting positions for the leg deviating from the optimal starting point is choosen and the resulting trajectories are recorded. We see that the attracting behaviour of the recurrent network representation compensates for deviating positions of the leg by gradually driving it back to the standard trajectory.

4. Conclusions

We developed a new RNN model which has several mathematical features which proved to be useful in sequence generation tasks. Based on these results we described a learning framework which separates the control system into an oscillating RNN and a task-oriented adaptive component which supports fine-tuning, faster learning and

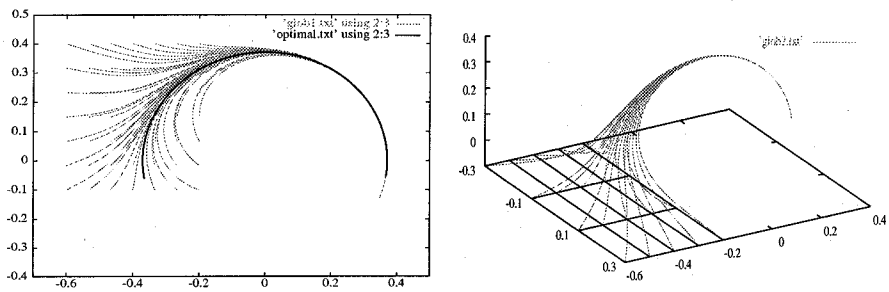

Figure 4: Plot of two different return stroke trajectory fields created with our approach.

execution. Future work will include the extention of our approach to leg coordination with several gaits including the transitions between them.

References

[1] P. Baldi and K. Hornik. Universal approximation and learning of trajectories using oscillators. *NIPS 8*, p.727–732, MIT Press, 1996.

[2] F. Delcomyn. The walking of cockroaches – deceptive simplicity. In *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, p.21–43. Academic Press, 1993.

[3] E. Goles and S. Martínez. *Neural and Automata Networks – Dynamical Behavior and Applications*. Kluwer, 1990.

[4] W. Ilg and K. Berns. A learning architecture based on Reinforcement Learning for adaptive control of the walking machine LAURON. *Robotics and Autonomous Systems*, 15:321–334, 1995.

[5] M. I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. of the 8th Ann.Conf. of the Cognitive Science Society*, p.531–546, Erlbaum, 1986.

[6] P. Koiran. Dynamics of discrete time, continuous state hopfield networks. *Neural Computation*, 6:459–468, 1994.

[7] T. Kolb and K. Berns. Concerning the formation of chaotic behaviour in recurrent neural networks. *ESANN'94*, D facto, Brussels, p.1–6, 1994.

[8] T. Kolb. *Adaptive Verfahren zur Folgengenerierung*. Doctoral Dissertation, Brandenburg University of Technology at Cottbus, 1998.

[9] C.-C. Ku and K. Y. Lee. Diagonal recurrent neural networks for dynamic systems control. *IEEE Trans. NN*, 6(1):144–156, 1995.

[10] K. Matsuoka. Sustained oscillations gernerated by mutually inhibiting neurons with adaptation. *Biol.Cybern.*, 52:367–376, 1985.

[11] F. Pasemann. Discrete dynamics of two neuron networks. *Open Systems & Information Dynamics*, 2(1):49–66, 1993.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *PDP I*, p.318–362. MIT Press, 1986.

[13] G. M. Shepherd. *Neurobiology*. Oxford University Press, 1988.

[14] G. Taga. A model of the neuro-musculo-skeletal system for human locomotion. *Biol.Cybern.*, 73:113–121, 1995.