Evolved Topology Generalized Multi-layer Perceptron (GMLP) for Anatomical Joint Constraint Modelling

Dr. Glenn L. Jenkins

School of Applied Computing
Swansea Metropolitan University
Swansea, Wales
email: glenn.l.jenkins@smu.ac.uk

Michael E. Dacey

School of Applied Computing
Swansea Metropolitan University
Swansea, Wales
email: michael.dacey@smu.ac.uk

Abstract— **The accurate simulation of anatomical joint models is becoming increasingly important for both medical diagnosis and realistic animation applications. Quaternion algebra has been increasingly applied to model rotations providing a compact representation while avoiding singularities. We propose the use of Artificial Neural Networks to accurately simulate joint constraints based on recorded data. This paper describes the application of Genetic Algorithm approaches to neural network training in order to model corrective piece-wise linear / discontinuous functions required to maintain valid joint configurations. The results show that Artificial Neural Networks are capable of modeling constraints on the rotation of and around a virtual limb.**

Keywords-GMLP; Unit quaternion; Constraint; Neural Network; Genetic Algorithm

I. INTRODUCTION

Joint systems are important constituents of anatomical models, they are used in simulation to retain anatomically correct movement and ensure limbs do not separate or intersect. Current techniques are limited by their underlying representation or their abstraction of the joint function. Demand is increasing for anatomically correct joints for applications in animation and medicine [1, 2]. However in current applications increasing accuracy incurs additional complexity and therefore computational cost [3-5].

Dynamics solutions can be used to produce realistic behavior based on input, contact and constraint forces [6]. Depending on the complexity of the simulation, the outcome of dynamics-based behavior can be difficult to predict. Inverse-Kinematics (IK) based approaches however allow the precise placement of end effectors as constraints [3]. IK solvers attempt to resolve constraints within a constraint system, a problem compounded by the existence of zero or more solutions [3].

Kinematics based solvers can be classified as analytical, often resorting to reduced coordinate formalisms, or numerical methods, using iterative approaches to solve a system of constraints. An important aspect of this is how the constraint of joints is represented. This work a

phenomenological [7] joints (whose behavior can be modeled without reference to the underlying joint anatomy).

Artificial Neural Networks (ANNs) are used to implicitly model the boundary between valid and invalid orientations by modeling a vector field in quaternion space. From the training data given, the network learns discontinuous corrective functions which model the behavior of the joint and ensure the joint configuration remains valid during movement. Using evolutionary techniques based on genetic algorithms, the topology of the network is configured dynamically to approximate the piece-wise linear properties inherent in discontinuous functions [8].

In previous research evolved GMLP neural networks have been applied to constraints on the orientation of the limb (or swing [9]) with regular (circular) [10, 11] and irregular [11, 12] bounded constraint regions. This paper focuses on the extension of these techniques to rotation around the limb (or twist [9]).

II. RELATED WORK

Primitive joint constraints have been parameterized using Euler angles [13-15]. However inter-dimensional dependencies are not represented [16] and singularities or "Gimbal Lock" are encountered [17]. Inter-dimensional dependencies between Euler angle components can be expressed using equations [18], that can provide mathematical descriptions of rotational constraint boundaries. Here geometric functions are fitted to a given dataset, examples include spherical [19] and conical polygons [1, 20].

Approaches such as special orthogonal matrices have been used to overcome the problem of singularities [2, 21]. More recent research has focused on the use of unit quaternions to model orientations and joint constraints. Unit quaternion algebra allows rotational models to be represented without the presence of Gimbal Lock [17].

Quaternions are an extension of complex numbers, composed of one real and three imaginary components where $q = <s, i, j, k>$. Multiplying complex numbers results in rotation in the complex plane, giving rise to the complex identity $i^2 = -1$. This is extended in a subset of quaternion space, where all quaternions are of unit length, to $i^2 = j^2 = k^2 = -1$. Unit quaternions occupy a three dimensional surface (a

hyper-sphere) in four dimensional space and can be used to represent rotations. This representation is redundant as the unit quaternions represent 4π rotations, hence quaternions at polar opposites (q and $-q$) represent the same rotation [22] i.e. they are antipodal.

Lee [23] decomposes a single unit quaternion into two unit quaternions each representing rotation in a single plane (effectively swing and twist for conic and axial constraints). Lee defines conic, axial and revolute constraints, more complex constraints can be modeled with a union of these basic types. Interrogation of these constraints (to ascertain the validity of a joint configuration,) is presented, but no method of calculating a correction to the nearest valid orientation is defined. Liu and Prakash [24] build on Lee's work. Using a sampled boundary they created a function to constrain the decomposed quaternion that can be used for both constraint validation and clamping to the boundary.

In the quaternion iso-surface approach of Herda, Urtasun, Fua and Hanson [25] limb rotations were recorded and represented in quaternion space. A set of four-dimensional unit quaternions describing the valid joint orientations are projected to a cloud of points in three-dimensions. This reduction in dimensionality overcomes the problem of ambiguity in quaternion space (q and $-q$ representing the same orientation). The initial approach [25] made use of spherical primitives to create an implicit surfaces but was limited by sparse data in difficult to sample (uncomfortable) areas. In the later approach [26] the point cloud is voxelized and the density of each voxel calculated, the voxels are subdivided until their density falls below some threshold. Each voxel is populated with a primitive and an iso-surface is then fitted to the primitives defining a boundary between valid and invalid orientations. An iterative approach can then be employed to resolve invalid joint configurations.

Johnson [27] reduced the dimensionality of the quaternion by projecting one half of the unit quaternion hyper-sphere onto a three-dimensional tangent space. A set of quaternions expressing valid joint and pose constraints are generated and constraints implemented based on a maximum deviation from their mean. Corrections are implemented by recursively moving an invalid point closer to the mean and the corrected point is then mapped back into unit quaternion space.

Evolved topology GMLP Neural Networks have been evolved and trained to provide a suitable unit quaternion based correction for a given orientation with zero correction for valid points [10, 11]. Unlike the approaches of the other authors [23-26, 28] this approach does not require pre-processing of the subject quaternion. Due to inaccuracy in the neural network some correction of valid orientations takes place. Research has shown that an SVM classifier can be used to separate valid orientations from those requiring

correction [29]. Recent research [30] has looked employ unsupervised learning more applicable to use with recorded actor or patient specific data (such as that gathered by Herda, Urtasun and Fua [26]).

Artificial neural networks are inspired by the structure of the human brain. Like biological neural networks they are composed of neurons which are linked together to form complex networks. However, they are significantly different in terms of complexity and the way nodes in the network communicate. There are many types of network architecture, from auto-associative memories such as the Hopfield network to unsupervised networks such as Kohonen's SOM [31].

Amongst the most popular types of architecture are feed-forward networks such as the Multi-layer Perception. These are trained to give certain outputs in response to given inputs by repeatedly adjusting the strengths of the interconnections between neurons within the network. Typically, neural networks use an optimization process to learn the best boundary to delineate regions within a multi-dimensional feature space. Recent developments have introduced the use of genetic algorithms to find the optimum network configuration and topology for a given network [32-36]. Huber, Mayer and Schwaiger [32] attempted to solve to this problem by means of Evolutionary Algorithms (EA) which search for a problem-adapted neural network topology. EA are used to evolve the topology while traditional a traditional supervised training is used on each of the networks in the evolved population.

In terms of modeling the rotation of a virtual limb current approaches using evolved GMLP, are capable of implicitly modeling constrained regions with regular boundaries [23, 28] and irregular boundaries [24-26] between valid and invalid limb orientations (relative to an attachment point). This work builds on these approaches to include constraints on the rotation around a virtual limb.

The remainder of this paper is structured as follows. Section 3 provides a description of our methodology with reference to the techniques employed. Section 4 reports the results of the experiments undertaken these are discussed in Section 5. Finally Section 6 draws conclusions from this work and highlights areas for future investigation.

III. METHODOLOGY

This paper describes the application of genetic algorithm approaches to neural network training in order to model piece-wise linear / discontinuous functions that approximate the behavior of anatomically correct joint constraints. The behavior of the joint constraint is modeled as a vector field in $\Re 4$ that maps invalid rotations to the nearest valid rotation, as illustrated in Fig 1.
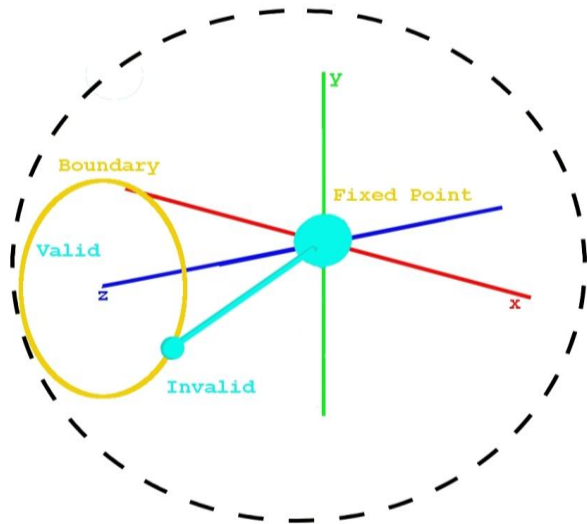
Figure 1. Model used for dataset generation. Valid region inside boundary invalid region outside.

values to each network using a fitness function. The fitness function is a combination of the network performance and network size.

Binary Tournament Selection is used to select the best networks of the population to breed, n individuals (typically two) are selected and the individual with the highest fitness is placed in the breeding pool. The selection is weighted, the higher an individual's fitness the more likely it is to be chosen [43]. An new generation of individuals is created through crossover and mutation of the fittest individuals selected from the last generation. This completes the evolutionary cycle that continues for a specified number of generations. It should be noted that the fittest individuals of the last generation will appear in the breeding pool more than once and inbreed generating identical offspring in the new generation [43]. This ensures that the best genetic patterns are passed on to the next generation

A GMLP neural network was topologically evolved and trained to model joint behavior, approximating a vector field in unit quaternion space. Each point in the feature space represents the orientation of a given joint model. Each point in the feature space represents the orientation of a given joint model. For each point, an appropriate correction must be modeled to map a given joint configuration to the nearest valid joint configuration. So for valid orientations, there is no correction, while for invalid orientations, a unit quaternion correction is stored to move the rotation to the nearest valid orientation. Discontinuities arise at the joint constraint boundary where the valid and invalid joint configurations meet, and opposite the constrained region. Here similar invalid orientations are corrected to diametrically opposite constrained region.

The NetJEN system used in our work is a Java based implementation of NetGEN [32] developed for research purposes at the University of Salzburg. NetJEN boasts several impressive features and provides an implementation of Huber *et al*'s [32] work in topology evolution. This hybrid system makes use of genetic algorithms and the resilient back-propagation learning algorithm. The following brief outline is based on published work [32-40].

The ANNs underlying structure is encoded such that GA techniques can be applied. A Direct Encoding method is used to convert between the phenotype (neural networks) and genotype (its genetic representation) minimizing the decoding effort. The system comprises of a Simple Genetic Algorithm (or SGA, an implementation of Smith *et al* [41]), the Genotype Phenotype Mapping (described above) and the Neural Network Manager (NNM). The NNM used is BOONE, developed by August Mayer at the University of Salzburg. The SGA generates blueprints for a random population of ANNs which are validated and passed to the NNM where they are constructed and trained using Resilient Back-propagation [42]. The SGA then assigns fitness

TABLE I EVOLUTION AND TRAINING SETTINGS

Parameter	Description	Value
Regularization function	Secondary fitness function.	Number of links
Hidden Nodes	Maximum no. of hidden nodes.	20
Number of Generations	No. of generations over which the ANN were evolved.	50
Population Size	Size of the populations evolved.	20
Fitness Function	Primary fitness function.	Inverse SSE
Regularization Weight	Controls the effect network size on the fitness function.	0.01
Evolve number of Links	Networks are pruned down from fully connected networks.	On
Evolve number of Hidden Nodes	Evolves the no. of hidden nodes.	On
Evolve number of epochs	Evolves the no. of training epochs	On
Learning Rate	Learning rate used when training the ANN.	0.1
Stopping Error	MSE at which the ANN are stopped.	0.001
Training Function	Training function used to train the weights of the ANN.	Resilient back-propagation
Max Epochs	Maximum number of training epochs	500

In each experiment the network was configured as follows. The input layer represents the current virtual limb orientation (as a unit quaternion), while the output layer represents the required correction to a valid orientation (also as a unit quaternion). The number of hidden nodes and connection topology are randomized and then evolved during the learning process using Generic Algorithms. The weights of the interconnections are randomized and updated using the back-propagation algorithm. The evolution and training parameters were set as shown in TABLE I.

Through experimentation, it was determined neural networks with sigmoid activation functions in the hidden layer and linear activation functions in the output layer produced good results. This distribution of activation functions was used throughout these experiments a similar distribution were employed for vector field approximation by Grzeszczuk *et al* [44], linear output layers have also been used with bi-polar sigmoid hidden layers [45, 46].

The number of generations, training epochs and hidden nodes were restricted to reduce training times. Each experiment was repeated five times to ensure the consistency of the results. The regularization weight was chosen based on publications by the systems authors [36], as was the learning rate [33], the stopping MSE for the networks was identified though experimentation. The size of the population, number of generations and the number of training patterns were suggested by a co-author of the NetJEN system Dr. Helmut Mayer in private correspondence. The evolution and training parameters were configured as shown in TABLE I.

Three datasets were prepared for each of the experiments; a training set, used to train each generation of ANN, a validation set, used to assess the fitness of the ANN for genetic selection and a test set which provided an unseen set of data on which to test the ANN.
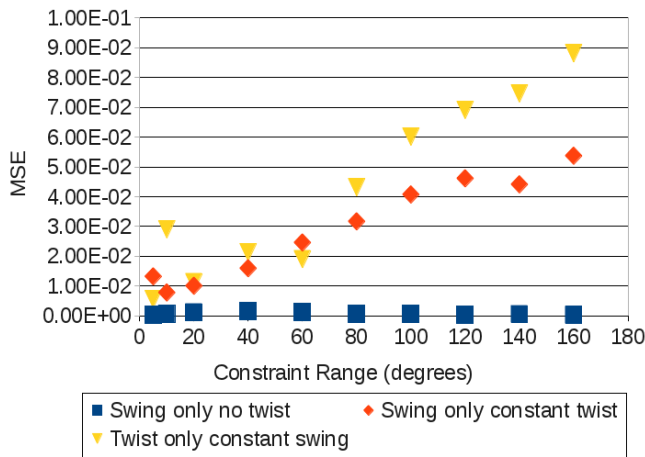
IV. RESULTS

Neural networks were successfully evolved and trained to model discontinuous vector fields in unit quaternion space representing both regular (spherical) constraints on the orientation of a virtual limb and constraints on the permitted rotation around the virtual limb. This is reflected both by the low Mean Squared Error (MSE) values (shown in Fig 2.) and the structure of the neural networks.

Two experiments were undertaken, in the first the rotation around the limb (twist) was maintained at 20° while the rotation of the limb (swing) was varied. In the second the swing was maintained at 20° and the (twist varied). Fig 2 shows the average MSE both experiments. Also shown for comparison are the results for constraint on the rotation of the limb with no rotation around the limb (twist) previously published by the author [11, 12].

The results show an increase in the swing constraint (with constant twist,) results in an increase MSE. An increase in the twist constraint (with constant swing) also results in an increase in an increase in MSE. There is a significant increase in error over the earlier experiment where twist was not considered. The evolution of the number of hidden nodes and limit on training epochs were limited to reduce training times (as indicated in TABLE I). The results show no clear relationship between the constraint range and the evolved network size or training epochs. However considering the average over all ranges (as shown in TABLE II,) shows an increase in neural network complexity (number of hidden nodes) and required training (number of training epochs) with the introduction of twist.

V. DISCUSSION

The results indicate that the introduction of a secondary constraint on the rotation around the limb has a detrimental effect on the performance of the neural network. This can be attributed to an increase in the complexity of the discontinuous vector field that the neural network is required to learn. This is supported by the increase the complexity of the evolved network. Previous work concluded that increases in modeling accuracy in the modeling of constraints on the limbs orientation could be achieved by increasing the limits on network size, extending training and other factors [11, 12]. The networks produced are close to the maximums dictated by the evolution parameters, indicating that improvements may be possible.

Figure 2. Graph showing the MSE for corrections to constraints on the rotation of the limb (swing) and the rotation around the limb (twist).

TABLE II PERFORMANCE, EVOLVED NETWORK SIZE AND REQUIRED TRAINING FOR CONSTRAINT MODELS.

	Averages Across Range		
	MSE	*Hidden Nodes*	*Epochs*
Swing no twist	0.00074	12.62	446.02
Swing with constant twist	0.02883	14.74	457.58
Twist with constant swing	0.04216	14.62	467.94

The increases in MSE with constraint size for constraints on both the rotation of and around the limb are attributed to changes in the distribution of training patterns in unit quaternion space. As patterns are evenly distributed between the valid and invalid region larger constraints have fewer patterns to represent the more complex invalid region. Pattern distribution has previously been shown to be significant in modeling joint constraint boundaries using neural networks [11].

VI. CONCLUSIONS AND FUTURE WORK

Previous research has shown evolved GMLP neural networks to be capable of modeling constraints on the orientation of a limb in unit quaternion space [11, 12]. The work presented here demonstrates that networks are capable of modeling constraints on both the rotation of the limb and around the limb. However due to increasing complexity the correction provided by the network is not as accurate.

The advantages of this approach is that constraints can be described and corrected with neural network models using quaternion representations directly, without resorting to reducing the dimensionality of (or otherwise pre-processing,) the quaternion as in Herda et al [25, 26], Lee [23] and Johnson [27]. Also unlike Lee [23] a method for correcting to a valid orientation is presented while avoiding the need for iterative methods as used by Herda *et al* [25, 26] and Johnson [27].

Future work will continue to address the neural network modeling of discontinuous vector fields in unit quaternion space for anatomical joint constraint. This will include the distribution of training patterns in quaternion space, both with respect to constraint range. Principally the performance of the neural networks on these more complex training sets will also be explored. This will include techniques used in previous work [10-12] and possibly the exploration of complex-valued neural networks. Which research indicates may provide advantages over real valued neural networks for quaternion based problems [47].

The results are encouraging and suggest that evolved GMLP neural networks are able to implicitly model constraints on the rotation of and around the limb with regular boundaries in unit quaternion space. They may prove as capable in modeling similar constraints with irregular boundaries and rotation around the limb while providing advantages over current approaches.

REFERENCES

[1] W. Manurel and D. Thalmann, "Human Shoulder Joint Modelling Including Scapulo-Thoratic Constraints and Joint Sinus Cones," Computers and Graphics, vol. 24, 2000, pp. 203-218.

[2] J. D. Feikes, J. J. O'Connor, and A. B. Zavatsky, "A Constraint-Based Approach to Modelling the Mobility of the Human Knee Joint," Journal of Biomechanics, vol. 36, 2003, pp. 125-129.

[3] A. Watt and M. Watt, "Forward Vs Inverse Kinematics in Computer Animation," in Advanced Animation and Rendering Techniques. New York: ACM Press, 1992, pp. 371-384.

[4] Y. Zhang and J. Wang, "A Dual Neural Network for Constrained Torque Optimization of Kinematically Redundant Manipulators," IEEE Transactions on System, Man and Cybernetics: Part B, vol. 32, 2002, pp. 654-662.

[5] A. D'Souza, V. S, and S. Stefan, "Learning Inverse Kinematics," Proc. International Conference on Intelligent Robots and System, Maui, Hawaii, USA, 29 Oct - 3 Nov 2001, pp. 298-303.

[6] P. M. Issacs and F. C. Micheal, "Controlling Dynamic Simulation with Kinematic Constraints, Behaviour Functions and Inverse Dynamics," ACM Transactions on Computer Graphics, vol. 21, 1987, pp. 215-223.

[7] A. E. Engin and S. T. Tumer, "Improvised Dynamic Model of Human Knee Joint and Its Response to Loading," Journal of Biomechanical Engineering, vol. 115, 1993, pp. 137-142.

[8] R. R. Selmic and L. L. Lewis, "Deadzone Compensation in Motion Control Systems Using Neural Networks," IEEE Transactions on Automatic Control, vol. 45, April 2000 2000, pp. 602-613.

[9] F. S. Grassia, "Practical Parameterization of Rotations Using the Exponential Map," Journal of Graphics Tools, vol. 3, 1998, pp.

[10] G. Jenkins and P. Angel, "Evolved Topology Genralized Multi-Layer Percerptron (Gmlp) for Joint Constraint Modelling," Proc. 9th International Conference on Computer Modelling and Simulation, United Kingdom Simulation Society, Oriel Collage Oxford, 4-6th April 2006.

[11] G. Jenkins and P. Angel, "Joint Constraint Modelling Using Evolved Topology Generalized Multi-Layer Perceptron (Gmlp)," International Journal of Simulation, Systems, Science and Technology, vol. 9, December 2008, pp. 15-27.

[12] G. Jenkins, "Evolved Neural Network Approximation of Discontinuous Vector Fields in Unit Quaternion Space (S³) for Anatomical Joint Constraint," University of Glamorgan, Treforest, PhD thesis, 2007.

[13] J. J. Faraway, X. Zhang, and D. B. Chaffin, "Rectifying Postures Reconstructed from Joint Angles to Meet Constraints," Journal of Biomechanics, vol. 32, 1999, pp. 733-736.

[14] J. Eng and D. A. Winter, "Kinematic Analysis of the Lower Limbs During Walking: What Information Can Be Gained from a 3d Model," Journal of Biomechanics, vol. 28, 1995, pp. 753-758.

[15] T. Furuta, T. Tawara, Y. Okumura, M. Shimizu, and K. Tomiyama, "Design and Construction of a Series of Compact Humanoid Robots & Development of Bipedal Walking Control Strategies," Robotics and Autonomous Systems, vol. 37, 2001, pp. 81-100.

[16] P. Baerlocher, "Inverse Kinematics Techniques for the Interactive Posture Control of Articulated Figures," Ecole Polytechnique Federal De Lausanne, Lausanne, PhD thesis, 2001.

[17] A. Watt and M. Watt, "Parameterisation of Rotation," in Advanced Animation and Rendering Techniques. New York: ACM Press, 1992, pp. 356-368.

[18] A. Maciel, L. P. Nedel, and C. M. Freitas, "Anatomy Based Joint Models for Virtual Human Skeletons," Proc. IEEE Computer Animation, Geneva, Switzerland., 19 - 21 June 2002, pp. 220-224.

[19] J. U. Korein, A Geometric Investigation of Reach. Massachusetts: MIT Press, 1984.

[20] A. E. Engin and S. T. Tumer, "Three Dimensional Kinematic Modelling of the Human Shoulder Complex Part 1: Physical Model & Determination of Joint Sinus Cone," Journal of Biomechanical Engineering, vol. 111, 1989, pp. 107-112.

[21] D. R. Wilson, J. D. Feikes, and J. J. O'Connor, "Ligaments and Articular Contact Guide Passive Knee Flextion," Journal of Biomechanics, vol. 31, 1998, pp. 1127-1136.

[22] A. Hanson, "Fundamentals of Rotation," in Visualising Quaternions, T. Cox, Ed. San Francisco, CA, USA: Morgan Kaufmann, 2006, pp. 43-56.

[23] J. Lee, "A Hierarchical Approach to Motion Analysis and Synthesis for Articulated Figures," Korean Advanced Institute of Science and Technology, Daejeon, PhD thesis, 2000.

[24] Q. Liu and E. Prakash, C, "The Parameterization of Joint Rotation with the Unit Quaternion," Proc. 7th Digital Image Computing: Techniques and Applications, Sydney, 10-12 December 2003, pp. 410-417.

[25] L. Herda, R. Urtasun, P. Fua, and A. Hanson, "Automatic Determination of Shoulder Joint Limits Using Quaternion Field Boundaries," International Journal of Robotics Research, vol. 22, June 2003, pp. 419-444.

[26] L. Herda, R. Urtasun, and P. Fua, " Hierarchical Implicit Surface Joint Limits for Human Body Tracking," Computer Vision Lab, Ecole Polytechnique Federal de Lausanne (EPFL), Lusanne CH-1015, 2004.

[27] M. P. Johnson, "Exploiting Quaternions to Support Expressive Interactive Character Motion," Massachusetts Institute of Technology, Massachusetts, PhD thesis, 1995.

[28] M. P. Johnson, "Exploiting Quaternions to Support Expressive Interactive Character Motion," Massachusetts Institute of Technology, Massachusetts, PhD thesis, 2003.

[29] G. Jenkins, "Support Vector Machines for Anatomical Joint Constraint," Proc. 11th International UKSim Conference on Computer Modelling and Simulation, IEEE Computer Society, Cambridge, UK, 25-27 March 2009, pp. 186-190.

[30] G. L. Jenkins and M. E. Dacey, "Sefl-Organizing Maps for Anatomical Joint Constraint Modelling," in 13th International Conference on Modelling and Simulation (UKSim2011). Cambridge, UK: IEEE, 2011, pp. 42-47.

[31] K. Mehrotra, C. K. Mohan, and S. Ranka, "Introduction," in Elements of Artificial Neural Networks. Massachusetts: MIT Press, 1997, pp. 1-40.

[32] R. Huber, H. A. Mayer, and R. Schwaiger, "Netgen - a Parallel System Generating Problem-Adapted Topologies of Artificial Neural Networks by Means of Genetic Algorithms," Proc. Beiträge zum 7. Fachgruppentreffen Maschinelles Lernen der GI-Frachgruppe 1.1.3, Dortmund 1995, pp. 91-98.

[33] H. A. Mayer, R. Huber, and R. Schwaiger, "Lean Artificial Neural Networks - Regularization Helps Evolution," Proc. 2nd Nordic Workshop on Genetic Algorithms and their Applications, Vaasa, Finland 1996, pp. p163-172.

[34] H. A. Mayer and R. Schwaiger, "Evolutionary and Coevlolutionary Approaches to Time Series Prediction Using Generalized Multi-Layer Perceptrons," Proc. Congress on Evolutionary Computation, Washington DC 1999, pp. 275-280.

[35] H. A. Mayer and R. Schwaiger, "Evolution of Cubic Spline Activation Functions for Artificial Neural Networks," Proc. 10th Portuguese Conference on Artificial Intelligence (EPIA 2001), 2001.

[36] H. A. Mayer and R. Schwaiger, "Differentiation of Neuron Types by Evolving Activation Function Templates for Artificial Neural Networks," Proc. World Congress on Computational Intelligence, International Joint Conference on Neural Networks, Honolulu, Hawaii, USA, 12-17 May 2002.

[37] H. A. Mayer, M. Strapetz, and R. Fuchs, "Simultaneous Evolution of Structure and Activation Function Types in Generalized Multi-Layer Perceptrons," Proc. WSES International Conference on Neural Networks and Applications, Puerto De La Cruz, Tenerife, Spain, Feburary 11-15th 2000.

[38] H. A. Mayer, "A Taxonomy of the Evolution of Artificial Neural Systems," Proc. Scientific Computing in Salzburg, Austrian Computer Society 2005, pp. 121-135.

[39] H. A. Mayer, "Symbiotic Co Evolution of Artificial Neural Networks and Training Data Sets," Proc. 5th International Conference on Problem Solving from Nature, Amsterdam, The Netherlands 1998, pp. 511-520.

[40] H. A. Mayer and P. Maier, "Evolution of Neural Go Players," Osterreichische Gesellschaft fur Artificial Intelligence, vol. 24, 2005, pp. 8-16.

[41] R. E. Smith, D. E. Goldberg, and J. A. Earickson, "Sga-C: A C-Language Implementation of a Simple Genetic Algorithm.," University of Alabama, Tuscaloosa TCGA 91002, May 1991.

[42] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The Rprop Algorithm," Proc. IEEE International Conference on Neural Networks, San Francisco, CA, March 28 - April 1 1993.

[43] D. E. Goldberg, "Genetic Algorithms Revisited: Mathematical Foundations," in Genetic Algorithms in Search, Optimization, and Machine Learning, D. Edward, Ed. Reading, Massachusetts: Addison-Wesley Publishing Company Inc., 1989.

[44] R. Grzeszczuk, Terzopoulos, D, Hinton, G, "Nuroanimator: Fast Neural Network Emulation and Control of Physics-Based Models," Proc. 25th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, New York, USA 1998, pp. 9-20.

[45] N. H. R. Goerke and R. Eckiller, "A Neural Network That Generates Attractive Vector Fields for Robot Control," Proc. Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen 1996, pp. 294-297.

[46] N. H. R. Goerke, F. Kintzler, A. Rabe, D. Roggisch, and R. Eckmiller, "Controlling the Khepera Robot by Neural Network Modules," Proc. First International Khepera Workshop, Paderborn: HNI-Verglasschriftenreihe 1999, pp. 119-128.

[47] A. Hirose, "Complex-Valued Neural Networks: Distinctive Features," in Complex-Valued Neural Networks. Berlin, Germany: Springer-Verlag, 2006, pp. 17-41.