Self-Organizing Maps 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 realistic animation and diagnostic medical applications. Recent models have exploited unit quaternions to eliminate singularities when modelling orientations between limbs at a joint. This has led to the development of quaternion based joint constraint validation and correction methods. In this paper a novel method for implicitly modelling unit quaternion joint constraints using Self Organizing Maps (SOMs) is proposed which attempts to address the limitations of current constraint validation and correction approaches. Initial results show that the resulting SOMs are capable of modelling regular spherical constraints on the orientation of the limb.**

Keywords-Self Organizing Map; Unit quaternion; Constraint; Neural Network

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, using iterative approaches to solve a system of constraints. An important aspect of this is how the constraint of joints is represented. This work builds on previous work in joint constraint modeling; specifically extending quaternion based phenomenological [7] joints (whose behavior can be modeled without reference to the underlying joint anatomy).

SOMs are used to implicitly model the boundary between valid and invalid orientations by modeling a group of valid rotations (expressed as unit quaternions). The SOM creates a set of prototype vectors representing the data set and undertakes a topology preserving projection of the prototypes from the n-dimensional input space onto a two dimensional grid [8]. When presented with an input orientation the network responds with the nearest prototype which can be used to ascertain the inputs validity and possibly provide a target for correction.

In this paper constraints on the rotation of the limb (or swing [9]) with regular (circular) bounded constrained regions are considered, while irregular boundaries and rotation around the limb (or twist [9]) are the subject of future work.

II. Related Work

Primitive joint constraints have been parameterized using Euler angles [10-12]. However inter-dimensional dependencies are not represented [13] and singularities or “Gimbal Lock” are encountered [14]. Inter-dimensional dependencies between Euler angle components can be expressed using equations [15], that can provide mathematical descriptions of rotational constraint boundaries. Here geometric functions are fitted to a given dataset, examples include spherical [16] and conical polygons [1, 17].

Approaches such as special orthogonal matrices have been used to overcome the problem of singularities [2, 18]. 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 [14].

Quaternions are an extension of complex numbers, composed of one real and three imaginary components where $q = \langle s, i, j, k \rangle$. 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 [19] i.e. they are antipodal.

Lee [20] 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 [21] 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 [22] 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 [22] 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 [23] 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 [24] also 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.

Generalised Multi-layer Perceptron Neural Networks have been evolved and trained to provide a suitable quaternion based correction for a given orientation with zero correction for valid points [25, 26]. Unlike the approaches of the other authors [20-23, 27] 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. It has also been demonstrated that an SVM classifier can be used to separate valid orientations from those requiring correction [28]. Both of these approaches are difficult to apply to recorded actor or patient specific data (such as that gathered by Herda, Urtasun and Fua [23]) as they rely on supervised learning.

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 [29].

The SOM is a popular neural network trained using unsupervised techniques [30]. The network is composed of two layers, an input and an output layer each containing nodes. Nodes in the output layer nodes are arranged in a topology (for example a grid), and each input node connected to every output node by a weight. Before training the weights are randomly assigned, then for each time step patterns (as vectors) are presented at the input nodes. The output nodes compete and the winning output node is that with the shortest Euclidean distance between its weight vector and the input vector [30]. The winning node and its topological neighbors are updated moving the weight vectors of the winning node and its neighbors towards the input, according to a learning rate which decreases along with the size of the neighborhood as training continues [30]. This reduces the size and effect of the training at each time step, with the network becoming less volatile [30]. Training ends when the network converges i.e. becomes stable [30] (there is no change in the winner for any pattern,) or some other stopping condition is reached [31-33].

In terms of modeling a virtual limb current approaches are capable of modeling regular boundaries [20, 27] and irregular boundaries [21-23] between valid and invalid limb orientations (relative to an attachment point). It is postulated that the SOM is capable of modeling both regular and irregular boundaries by identifying the prototype vector which is closest to the given orientation.

Current approaches model both the rotation of the limb and rotation around the limb with irregular rotation boundaries [20-23, 27]. This exploratory paper aims to study the capabilities of SOMs in modeling the rotation of the limb with a regular rotational boundary and no constraint on the rotation around the limb. This will provide a more practical alternative to the techniques developed in our earlier work [26, 34] which used supervised learning techniques and could not be trained from patient/actor recorded data alone. Future work will explore more complex constraints including irregular boundaries and rotation around the 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 SOMs to the correction of unit quaternions describing the orientation of an anatomical limb. In doing so the SOM models a set of valid orientations. SOMs were trained to identify the closest valid orientation for both valid and invalid input. Constraints of various sizes were investigated to ascertain their performance in the context of anatomical models. The input layer represents the current limb orientation, while the weights of the winning output node represent the nearest valid orientation. The number of output nodes and number of patterns where fixed were as indicated in Table 1.

The SOM training process (outlined by Mehrotra, Mohan and Ranka [30]) begins with the weights of the interconnections between input and output nodes being set to small random numbers. The output nodes are placed into a topology each having a position (in this case in a grid). Then for each time step the input set (comprising a number of patterns) is presented to the network. For each pattern the squared Euclidean distance (*D*) between the input pattern and the weight vector (the connections between the inputs and the output node) of each output node is calculated. The output node with the smallest value of *D* is the winner and is updated. Its weights are adjusted some portion of the distance towards the input vector according to the learning rate. Output nodes that are within the neighborhood (topological regions according to position), of the winning node also have their weights updated.

After a given time period (number of time steps) the learning rate and neighborhood were reduced according to a scaling factor, this process continued over successive periods until minimum values were reached for each. The initial values, minimum values and updating factors for each are provided in Table 1. along with the relevant time periods. These were identified by experimentation.

TABLE I. TRAINING CONFIGURATION

Parameter	***Description***	***Setting***
Input nodes	Number of input nodes	4
Output Nodes	Number of output nodes where constant.	100
Training Patterns	Number of training patterns where constant.	500
Learning Rate	Rate at which weights are updated. Updated for each period until the minimum is reached.	Initial: 1.0 Min.:0.02 Update: 0.5
Learning Rate Period	Periods over which the learning rate remains constant. Where *t* is the time step.	0 ≥ t < 10 10 ≥ t < 20 20 ≥ t < 30 30 ≥ t < 40 40 ≥ t
Neigborhood	Size of neighborhood updated at each period. Updated for each period until only the winning node updates.	Initial: 4.0 Min.:0.0 Update: 0.5
Neigborhood Period	Period over which the learning rate remains constant. Where *t* is the time step.	0 ≥ t < 10 10 ≥ t < 20 20 ≥ t < 30 30 ≥ t < 40 40 ≥ t
Maximum training time	Maximum number of time steps.	10,000

Each network was trained until it converged (entered a stable state in which the winning node for each pattern does not change) or a maximum number of training epochs (time steps) was reached. Each experiment was repeated ten times to ensure the consistency of the results. The SOM used in this work was based on that presented by Mehrotra, Mohan and Ranka [30] adapted to use a time step based termination criteria [31, 32].

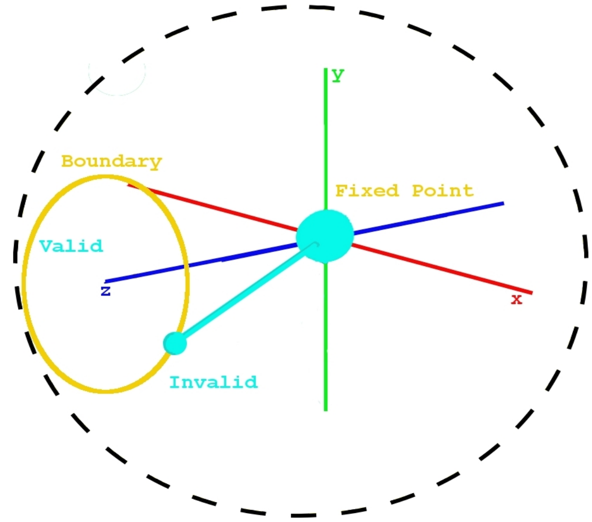

Figure 1. Model used for dataset generation. Valid region inside boundary invalid region outside.

Each SOM was trained using a dataset based on the simple model shown in Fig. 1. Each training set was created by the random generation of unit length vectors (virtual limbs), whose orientation relative to the x-axis was within a prescribed limit (in degrees). The orientation of such vectors (converted to a unit quaternion) provided the input set, representing a sample of the valid region in unit quaternion space. This essentially gives a cloud of valid orientations which the network attempts to generalize. When provided with a test orientation (valid or invalid) the network responds with the nearest valid orientation. This is similar to the approaches of Herda, Urtasun, Fua and Hanson [22] and Johnson [27].

The set of unit quaternion occupy the three dimensional surface of a hyper-sphere (S^3) in four dimensional space ($\Re^4$), this represents a double covering of the group of rotations (SO(3)) [35] (hence movement between two poles represents a 360° rotation). In the training set all valid patterns were placed one hemisphere of the unit quaternion hyper-sphere and all other patterns are considered invalid. This fits the context of anatomical constraints as a rotation though 360° should not be valid. It also has performance implications, as demonstrated in the authors earlier work [34], in that valid regions on both sides of the hyper-sphere (both *q* and its antipode –*q* being valid) disrupted learning.

Experiments were undertaken with output layers containing between 100 and 900 nodes, on datasets of between 500 and 5000 patterns. In experiments where the range was not varied a constant range of 90 degrees was used, defaults for the other parameters are given in Table 1. The training dataset contained only valid patterns, similar to those recorded from the movement of a human arm. A set of ‘ideal’ corrections (no correction for valid orientations and the nearest valid orientation for invalid ones,) was generated using the approach of Lee [20] and provided a measurement of the SOMs capabilities.

IV. RESULTS

The results show the effect of correcting the orientation to that suggested by the SOM (the quaternion represented by the weights of the winning node). The results indicate successful training of the neural network. An increase in the range (maximum angle between the limb and the x-axis,) of the constrained region results a decrease in performance as shown in Fig 2 (a). This increase appears independent of the number of output nodes, though increasing the output nodes does result in a reduction in error. This is confirmed in Fig 2 (b) which shows a decrease in error as the output layer size increases.

Experiments were also undertaken to investigate the effects of increasing the number of training patterns, this produced an increases in performance (as shown in Fig. 2(c)), which attenuates as the number of patterns increases. Some initial experiments have been undertaken into the distribution of training patterns. In the first the region was divided in two with a central and a boundary region. The results show that an increase in the density of patterns in the boundary region (shown as a pattern ratio in Fig. 2(d)), results in an increase in performance. This appears to attenuate as shown in Fig. 2 (d).

V. DISCUSSION

The results show that SOMs are capable of identifying the nearest quaternion representing the orientation of a virtual anatomical limb. They implicitly model a region occupied by valid orientations in unit quaternions space. The Mean Squared Error (MSE) compared to the test set (in which invalid orientations are corrected to the boundary) appears relatively low though higher than those for other neural network based approaches with comparable numbers of nodes [25, 34]. This is confirmed by applying a sample SOM constraint (using a 900 output node SOM, trained with 5000 patterns on a 90 degree constraint,) to a unit length virtual limb this results in a 3D error of 5.63 x 10^{-2}.

A key reason for this is the method in which the error is calculated, with the limb being corrected to the orientation provided by the weights of the winning node. This results in correction errors i.e. the correction of valid points and over correction of invalid points (to within the valid region rather than to the boundary).

However despite this the results provide an insight into the effects of problem, network and training attributes on performance. It is clear that the network is capable of learning constraints of varying sizes though larger constraints appear to demonstrate a higher error. This may be related to an increased correction of valid points as output nodes are more dispersed over the valid region and increased overcorrection of invalid points as fewer output nodes occupy spaces near the boundary.

Improvements resulting from the increase in output nodes can be ascribed to an increase in the density of output nodes over the valid region, reducing correction errors. The

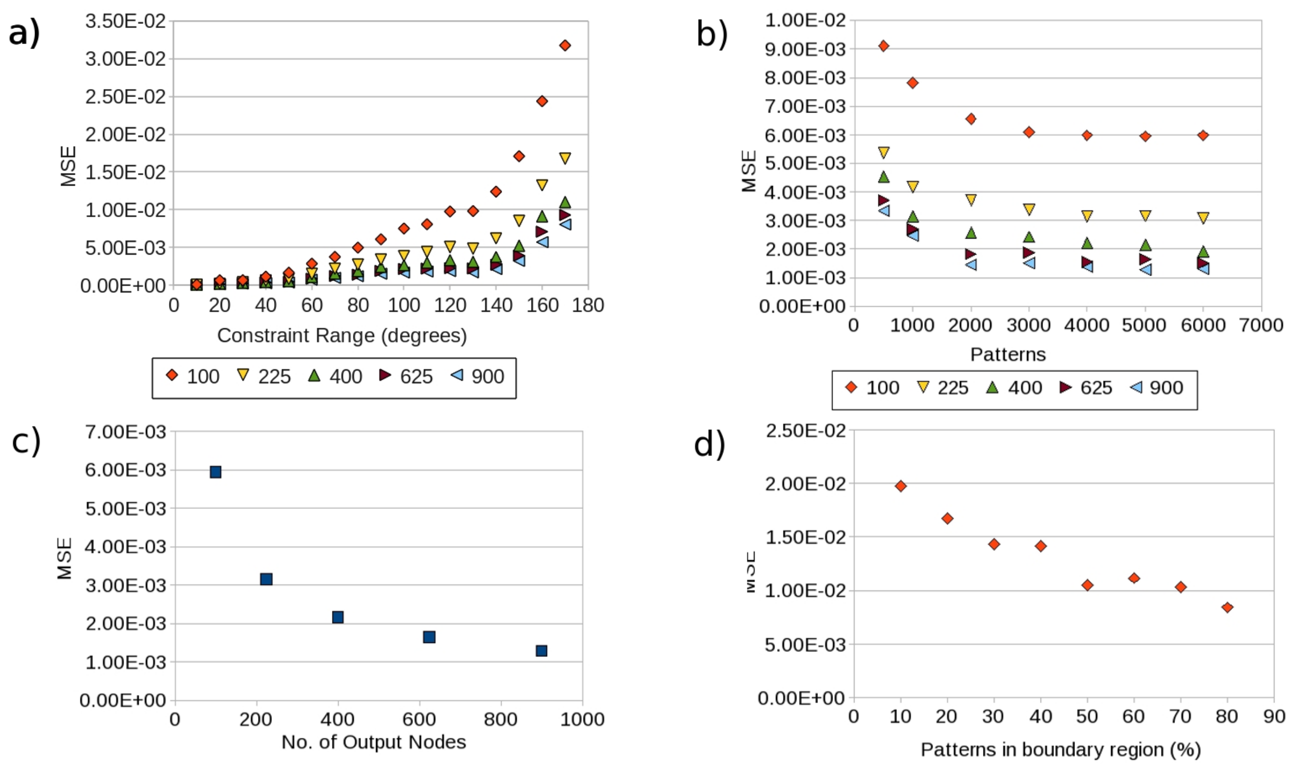

Figure 2. Performance of the SOM: (a) with range of constraint varied (number of output nodes as indicated by key), (b) with number of training patterns varied (number of ouput nodes as indicated by key), (c) with number of output nodes varied (d) with the changes in the distribution of patterns within the valid region.

reduction in error which accompanies an increase in the number of training patterns suggests that this improves the positioning of the output nodes over the valid region. This may simply improve their placement in dense regions but more importantly populate sparse regions.

Considering the change of pattern ratio between the central and boundary regions it is clear than increasing the density of the region at the boundary increases performance as shown in Fig 2. (d). This suggests that high error is directly related to sparse regions near the boundary and that a significant part of the correction error noted is caused by overcorrection of invalid patterns. This has implications for both small datasets and large constraints due to the low density of the valid region. It has been reported that poorly sampled regions are often those which are difficult or painful for an individual to reach or maintain [22, 23]. This may have implications for the future application of this technique.

The correction of valid patterns and the over correction of invalid patterns is a limitation of the current application of our approach (rather than the approach itself). As in other techniques [22], a threshold (based on the maximum Euclidean distance between the training patterns and the output of the trained network) could be employed to reduce the correction of valid patterns. Both Johnson [27] and Herda, Urtasun, Fua and Hanson [22] make use of iterative approaches to correct invalid patterns, moving them towards the valid region until they are valid.

Forcing valid orientations to one side of the unit quaternion hyper-sphere reduces the complexity of the modeled region and introduces a limitation. A valid rotation on the opposite side of the hyper-sphere would be reported as invalid. This is not a limitation in the context of anatomical constraints provided that the initial joint configuration is on the appropriate side of the hyper-sphere. The constraint correction (or clamping) system should ensure that the orientation remains within the boundary and thus prevent the limb from performing an anatomically impossible full rotation to the other valid configuration.

VI. Conclusions and Future Work

In conclusion SOMs are capable of implicitly modeling the boundary between valid and invalid orientations in quaternion space to a reasonable degree of accuracy, provided that the dataset is appropriately distributed in unit quaternion space. More importantly they can provide an indication of validity and focus for correction. In this they are similar to the approaches of Herda, Urtasun, Fua and Hanson [22], Herda, Urtasun, Fua and Hanson [23] and Johnson [27]. Unlike these approaches no decomposition or reformatting of the unit quaternion orientation is required.

More research is required into the effects of training data distribution especially data with sparse regions (such as that gained from motion capture [22, 23]) and the comparison of these results with other approaches [22, 23, 27]. Research is also required into the application of techniques to mitigate correction errors as used in other approaches [22, 23, 27]. The benefit of the SOMs topology preserving capabilities in modeling constraints in unit quaternion space over alternative techniques may also warrant investigation.

The results are encouraging and suggest that SOMs are able to implicitly model constraints on the rotation of 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, pp. 203-218, 2000.

[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, pp. 125-129, 2003.

[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, pp. 654-662, 2002.

[5] A. D'Souza, V. S, and S. Stefan, "Learning Inverse Kinematics," presented at International Conference on Intelligent Robots and System, Maui, Hawaii, USA, 2001.

[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, pp. 215-223, 1987.

[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, pp. 137-142, 1993.

[8] H. Ritter and K. Schulten, "On the stationary state of Kohonen's self organizing sensory mapping," Biological Cybernetics, vol. 54, pp. 99-106, 1986.

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

[10] J. J. Faraway, X. Zhang, and D. B. Chaffin, "Rectifying postures reconstructed from joint angles to meet constraints," Journal of Biomechanics, vol. 32, pp. 733-736, 1999.

[11] 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, pp. 753-758, 1995.

[12] 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, pp. 81-100, 2001.

[13] P. Baerlocher, "Inverse Kinematics Techniques for the Interactive Posture Control of Articulated Figures," in Department D'Informatique. Lausanne: Ecole Polytechnique Federal De Lausanne, 2001, pp. 156.

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

[15] A. Maciel, L. P. Nedel, and C. M. Freitas, "Anatomy Based Joint Models for Virtual Human Skeletons," presented at IEEE Computer Animation, Geneva, Switzerland., 2002.

[16] J. U. Korein, A geometric investigation of reach. Massachusetts: MIT Press, 1984.

[17] 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, pp. 107-112, 1989.

[18] D. R. Wilson, J. D. Feikes, and J. J. O'Connor, "Ligaments and articular contact guide passive knee flextion," Journal of Biomechanics, vol. 31, pp. 1127-1136, 1998.

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

[20] J. Lee, "A Hierarchical Approach to Motion Analysis and Synthesis for Articulated Figures," in Department of Computer Science. Daejeon: Korean Advanced Institute of Science and Technology, 2000, pp. 93.

[21] Q. Liu and E. Prakash, C, "The Parameterization of Joint Rotation with the Unit Quaternion," presented at 7th Digital Image Computing: Techniques and Applications, Sydney, 2003.

[22] 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, pp. 419-444, 2003.

[23] 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.

[24] M. P. Johnson, "Exploiting Quaternions to Support Expressive Interactive character Motion." Massachusetts: Massachusetts Institute of Technology, 1995.

[25] 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, pp. 15-27, 2008.

[26] G. Jenkins and P. Angel, "Evolved Topology Genralized Multi-Layer Percerptron (GMLP) for Joint Constraint Modelling," presented at 9th International Conference on Computer Modelling and Simulation, Oriel Collage Oxford, 2006.

[27] M. P. Johnson, "Exploiting Quaternions to Support Expressive Interactive character Motion," in MIT Media Lab. Massachusetts: Massachusetts Institute of Technology, 2003.

[28] G. Jenkins, "Support Vector Machines for Anatomical Joint Constraint," presented at 11th International UKSim Conference on Computer Modelling and Simulation, Cambridge, UK, 2009.

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

[30] K. Mehrotra, C. K. Mohan, and S. Ranka, "Unsupervised Learning," in Elements of Artificial Neural Networks. Massachusetts: MIT Press, 1997, pp. 155-216.

[31] T. J. Klassen and M. I. Heywood, "Towards the on-line recognition of Arabic characters," presented at International Joint Conference on Neural Networks, Honolulu, Hawaii, USA, 2002.

[32] M. Saito, I. Szakall, R. Toth, K. M. Kovacs, M. Oros, V. P. T. S. Prasad, M. Blumberg, and C. Vadasz, "Mouse striatal transcriptome analysis: effects of orgal self-administration of alcohol," Alcohol, vol. 32, pp. 223-241, 2004.

[33] M. Miyoshi, H. Kawai, and H. Masuyama, "Efficent SOM learning by data order adjustment," presented at International Joint Conference on Neural Networks, Honolulu, Hawaii, USA, 2002.

[34] G. Jenkins, "Evolved Neural Network Approximation of Discontinuous Vector Fields in Unit Quaternion Space (S^3) for Anatomical Joint Constraint," in Faculty of Advanced Technology. Treforest: University of Glamorgan, 2007, pp. 195.

[35] P. Bell and I. Potapov, "Reachability problems in quaternion matrix and rotation semigroups," Information and Computation, vol. 206, pp. 1352-1361, 2008.