RADIAL BASIS FUNCTION NETWORK (RBFN) APPROXIMATION OF FINITE ELEMENT MODELS FOR REAL-TIME SIMULATION

Madusudanan Sathia Narayanan
Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14260
E-mail: ms329@buffalo.edu

Puneet Singla
Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14260
E-mail: singla@buffalo.edu

Sudha Garimella, MD
Pediatric Nephrology
SUNY at Buffalo
Buffalo, NY 14260
E-mail: sgarimella@upa.chob.edu

Wayne Waz, MD
Pediatric Nephrology
SUNY at Buffalo
Buffalo, NY 14260
E-mail: wwaz@upa.chob.edu

Venkat Krovi
Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14260
E-mail: vkrovi@buffalo.edu

ABSTRACT

Nonlinearities inherent in soft-tissue interactions create roadblocks to realization of high-fidelity real-time haptics-based medical simulations. While finite element (FE) formulations offer greater accuracy over conventional spring-mass-network models, computational-complexity limits achievable simulation-update rates. Direct interaction with sensorized physical surrogates, in offline or online modes, allows a temporary sidestepping of computational issues but hinders parametric analysis and true exploitation of a simulation-based testing paradigm. Hence, in this paper, we develop Radial-Basis Neural-Network approximations, to FE-model data within a Modified Resource Allocating Network (MRAN) framework. Real-time simulation of the reduced order neural-network approximations at high temporal resolution provided the haptic-feedback. Validation studies are being conducted to evaluate the kinesthetic realism of these models with medical experts.

INTRODUCTION

Virtual reality (VR) simulation based training holds immense promise for offline development of skills in arenas ranging from flight simulation to medical procedural training. The promise lies in the ability to not only present controlled sets of stimuli but to simultaneously train both cognitive and sensorimotor skills of the user. One aspect of this work relies on development of haptic user interfaces (HUIs) capable of rendering the forces to the user, which is discussed elsewhere [1]. The other aspect focuses on development of computational haptic models capable of rendering the high fidelity haptic interactions and is discussed here. Over the years VR simulators for medical procedural training have gradually transitioned from a primarily unidirectional visual engagement process to a more bidirectional kinesthetic immersion [2]. Effective implementation of the force-feedback algorithms thus requires high fidelity models which are capable of being computed within the timing constraints prescribed by a deterministic real-time framework necessary for haptics.

Enhancing the haptic response (i.e.) to reflect the force-response behavior of real bio-physical systems (tissues, organs etc.) makes nonlinear models necessary. Current haptics models of soft tissues (used in COTS medical simulators) rely on relatively simple/ simplistic linear models (spring-mass-damper systems or linear finite element analysis systems) to compute the feedback at high update rates However, the complexity of computing the higher spatial and temporal fidelity response from such nonlinear models creates challenges. While methods such as finite element methods (FEM) offer potential means of computing such responses, they can be pursued only in an offline setting.

Hence, in this work, we seek to develop suitable approximation methods to effectively and parametrically capture the physics in reduced order models. The results obtained using FE systems were used as case studies for which radial basis function based approximation model was developed. A modified resource allocating network (MRAN) method was adopted to determine the neural network states and an extended Kalman filter (EKF) method was implemented for optimizing these parameters. The response of the approximation models tend to be faster and easier to simulate than the original nonlinear system. Moreover, methods based on radial basis functions can be trained offline to have an optimal network structure and deployed online in the predictive phase at very high update rates suitable for medical surgical applications. We examine the applicability by studying the approximation of FEM analysis of a linear elastic and plastic cantilever beam. Finally, a virtual-haptic environment was developed using MATLAB Simulink/ VRML to deploy the resulting force reflection models. Working through this simple problem provides a greater insight about the principal issues related to numerical computation of RBFN model as well as implementation of real time haptic interface.

BACKGROUND

Realistic real-time haptic applications typically require a feedback or update at greater than 30 Hz for visual sensations and minimum of 500 Hz for haptic sensations [3]. Currently VR-and haptic (VR-H) simulations are constrained by both limitations inherent to Haptic-User-Interfaces (HUIs) as well as haptic computational models. However, this is a coupled problem – choices of HUIs determine the sophistication of haptic computational models and the current technology only permits the simplest of models to be run in real-time [4] (i.e., lumped parameter spring/mass systems). While the use of linear FE models helps overcome some of the computational limitations, the accuracy and fidelity of haptic models are always in question. More realistic, complex 3D finite element based soft tissue models, are currently far outside of the realm of real time simulations.

Linear elasticity is used for modeling the deformable materials principally due to simplicity of ensuing computations. However, the physical behavior of soft tissue may be considered as linear elastic only for small deformations [5, 6] (typically less than 10% of the mesh size). Thus, linear elastic FE models are not valid for large displacements and are not invariant with respect to rotations [7]. Finally selection of mesh density as well as parameters for the linear elastic systems remains challenging exercise. Thus, currently many surgical haptic simulators depend upon the subject studies to tune these parameters to achieve “realistic performance”.

Some of the simpler deformation techniques use surface models, where the masses are concentrated in the mesh vertices connected by springs. Such a surface model contains less mass points than a volumetric model defining the same shape and is therefore more efficient [8]. A surface model, however, is inherently inaccurate and yields physically invalid deformations (e.g. self penetration). Thus, volumetric models are better suited for the simulation of deformable objects, especially when cutting is required. The most commonly used methods for developing soft tissue simulations are the Mass-Spring Method (MSM) and the FEM and method obtained by modification and combination of these two approaches [9]. In all these cases, the choice of the appropriate simulation method is influenced by factors such as computing efficiency, required accuracy and the types of manipulations that have to be performed. With the use of higher fidelity models, the computation time increases rapidly and as a result the real-time haptic performance for surgical simulation has to be compromised upon.

In this work, we focus on the issues surrounding numerical implementation of nonlinear models and an alternative novel approach to develop approximation models for the nonlinear FE systems using radial basis function network (RBFN). Such an approach can be made adaptive to different model parameters with nonlinear characteristics while remaining computationally tractable. In addition, using such methods can also prove to be valuable for implementing real-time simulation for other non-linear FE models outside the medical simulator domain.

RADIAL BASIS FUNCTION NETWORKS (RBFN)

Over the past few decades, Artificial Neural Networks (ANNs) have emerged as a powerful set of tools in pattern classification, time series analysis, signal processing, dynamical system modeling and control. The popularity of ANNs can be attributed to the fact that these network models are frequently able to learn behavior when traditional modeling is very difficult to generalize. Typically, a neural network consists of several computational nodes called perceptrons arranged in layers. The number of hidden nodes essentially determines the degrees of freedom of the non-parametric model. A small number of hidden units may not be enough to capture a given system’s complex input-output mapping and alternately a large number of hidden units may overfit the data and may not generalize the behavior.

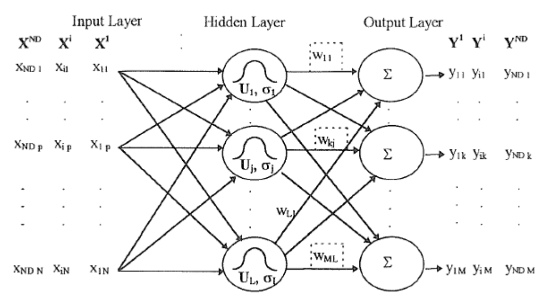


FIG. 1 MULTILAYER HIDDEN NODE NEURAL NETWORK WITH WEIGHTED OUTPUTS [10]

An RBF network is a two-layer feed-forward type network in which the input is transformed by the basis functions at the hidden layer. At the output layer, linear combinations of the hidden layer node responses are added to form the output. The name RBF comes from the fact that the basis functions in the hidden layer nodes are radially symmetric. In [11] authors report that the choice of the basis function is not crucial to the performance of the network. The most common choice however, is the Gaussian function which can be defined by a mean and a standard deviation. Figure 1 shows a schematic diagram of an RBF network with *n*, *l* and *m* respectively of input, hidden and output layer nodes for the general transformation of *nd* points of $\boldsymbol{X}(x_1, x_2, x_3, \ldots, x_{nd})$ in the input space to points $\boldsymbol{Y}(y_1, y_2, y_3, \ldots, y_{nd})$ in the output space.

In RBF networks, the connections between the input and the hidden layers are generally not weighted. The inputs therefore reach the hidden layer nodes unchanged. For an input $\boldsymbol{X_i}$, the j^{th} hidden node produces a response φ_j given by,

$$\varphi_j = \exp\left(-\left(\overrightarrow{X_i} - \mu_j\right) R^{-1} \left(\overrightarrow{X_i} - \mu_j\right)^T\right) \quad (1)$$

where, $\|\boldsymbol{X_i} - \boldsymbol{\mu}_j\|$ is the distance between the point representing the input, $\boldsymbol{X_i}$ and the centre of the j^{th} hidden node, $\boldsymbol{\mu}_j$ as measured by Euclidean norm. *R* is a state error noise covariance matrix. Generally, it is a positive definite matrix and for circular Gaussian functions, the off diagonal elements of *R* are zero and diagonal elements are $\sigma_k^{\boldsymbol{2}}$, for $k = 1 \ldots n$.

The output $\boldsymbol{y_{ik}}$ of the network at the output node is given by the weighted sum of RBFs,

$$y_{ik} = \sum^{h} \varphi_j w_j = W^T \phi\left(X, \mu_1, \mu_2, \cdots, \mu_h\right) \quad (2)$$

where, $W^T - \{w_1, w_2, \cdots, w_h\}^T$ is a vector of *h* linear weights or amplitudes, ϕ - vector of RBFs with centers at $\boldsymbol{\mu}_j$.

In the special case where the number of hidden layer nodes is equal to the number of data in the training set ($l = nd$) and the RBF centres coincide with the inputs ($\boldsymbol{U_j} = \boldsymbol{X_i}$, where $i = j = 1, 2\ldots, nd$), the hidden layer response according to (1) becomes unity for $j = i$. If the basis functions are truly localized, the response of the other hidden layer nodes will be near zero (i.e. $\boldsymbol{\sigma_j}$ for $j \neq i$ are such that $\boldsymbol{\varphi_j} \sim 0$ for $j \neq i$). It can also be seen that (2) gives the exact output when the output layer weight is equal to the output (the contribution to the weighted summation from j = i is $\boldsymbol{y_{ik}}$ and that from all $j \neq i$ is nearly zero). In the ideal case therefore, RBF network can be made to map points in *N*-dimensional input space exactly on to points in *M*-dimensional output space. This however, is not practical when *nd* is large in which case a few input points are chosen to represent the entire input data set.

The original RBF method requires that there be as many RBF centres as there are distinct data points in the input space. This however, is not possible in practice because that increases the complexity of the final RBF model tremendously. Moreover, the inputs usually occur in clusters making overlapping of receptive fields inevitable. Choosing all points as RBF centres will therefore lead to more number of redundant nodes with a huge network involving long training and computation times.

Modified Resource Allocating Network. The RBFN method that was implemented for our problem is Modified Resource Allocating Network (MRAN). MRAN adopts the basic idea of adaptively "growing" the number of radial basis functions where needed to null local errors, and also includes a "pruning strategy" to eliminate little-needed radial basis functions (those with weights smaller than some tolerance), with the overall goal of finding a minimal RBF network. RAN allocates new units as well as adjusts the network parameters to reflect the complexity of function being approximated. Though the optimal approximation is to add an impulse unit at the data point to match the output error, such a method actually lacks smoothness and is more error prone. Hence, in this work, the method explained in [12] using the Gaussian functions centered at the input data points to achieve the desired output was used. New nodes are not added at every point but are actually restricted by enforcing three main conditions as follows:

$$\|X_i - \mu_{nearest}\| > \varepsilon \quad (3)$$

$$e_i^{rms} = \sqrt{\sum_{j=i-(N_w-1)}^{i} \frac{\|e_j\|^2}{N_w}} > e_{r_{\min}} \quad (4)$$

$$\|e_i\| = \|y_i - f(X_i)\| > e_{\min} \quad (5)$$

Equation (3) ensures that a new node is added only if it is sufficiently far from all the existing nodes and equation (4) ensures that only if the approximation error using existing nodes exceeds the error specification. The final condition (equation(5)) compensates for the noise present in the observations data by determining the RMS error of last N_w observations and ensures that a new node is added only if the noise in data exceeds the specified threshold limit. For all the cases to be considered in this work only the standard Gaussian functions are used and the rotational parameters (off diagonal elements of the covariance matrix) are not learned along with the other network parameters, Θ.

$$\Theta = \{w_1, \mu_1^T, \sigma_1, \cdots, w_h, \mu_h^T, \sigma_h\} \quad (6)$$

In this case, if the input space dimension is *n*, output dimension of the network is *1* and the total number of nodes is *h*, then the size of the parameter vector considering the individual dimensions of w_j, $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma_j}$ for each node, will be:

$$DIM(\Theta) = h(n + n + 1) = h(2n + 1) \quad (7)$$

For each observation input to the network, feasibility of the constraints (3), (4) and (5) are determined and a new node will be added if all the conditions hold true. Irrespective of this result, all the network parameters will be updated using Extended Kalman Filter (EKF) as summarized below.

Extended Kalman Filter (EKF). This method is used for online adaptation of parameter of our nonlinear function approximation problem. So, the update relationships for each observation input are given below. First, it is necessary to determine the sensitivity (Jacobian) matrix of the RBFN for which a linearized model is used.

The measurement model for this problem becomes of the form:

$$\begin{aligned} &\hat{y} = h(x_k) + \nu_k \\ &with \\ &E(\nu_k) = 0 \\ &E(\nu_l \nu_k^T) = R_k \delta(l-k) \end{aligned} \tag{8}$$

The resulting EKF update equations are:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \tag{9}$$

$$\left.\begin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + K_k \left(\tilde{y} - h(x_k^-) \right) \\ P_k^+ &= (I - K_k H_k) P_k^- \end{aligned}\right\} \quad \text{where, } H_k = \left.\frac{\partial h(x_k)}{\partial x}\right|_{x=\hat{x}_k^-}$$

For determining the sensitivity matrix, the following relations were used:

$$\begin{aligned} \frac{\partial f}{\partial w_k} &= \varphi_k \\ \frac{\partial f}{\partial \mu_k} &= \left[w_k \varphi_k R_k^{-1} (x - \mu_k) \right]^T \\ \frac{\partial f}{\partial \sigma_{k_i}} &= \frac{w_k \varphi_k (x - \mu_k)^2}{\sigma_{k_i}^3}, i = 1 \cdots n \end{aligned} \tag{10}$$

Since, the inputs in all our problems were one-dimensional; rotation parameters of covariance matrix were not used in the formulation discussed in this work.

Formulation for Multi Output System. Instead of a scalar output, we now seek to predict the nodal displacements at more than one point based on application of load at one point, so it is necessary to formulate this problem for multi dimensional outputs based on above equations. Thus, this will result in additional RBFN parameters for learning and adapting to simulated data.

For this purpose, let us consider the dimension of the output vector to be m. Hence, this increases the number of weighting elements from 1 in the previous case to m for each input data. Thereby, total network parameters can be given as— n (for location of RBF center, $\boldsymbol{\mu}_j$), n (spread of the RBF covariance matrix, $\boldsymbol{\sigma}_j$) and m (for weighting element per output) per RBF node of the network. So, augmenting these additional parameters for a network of h nodes, the final network parameter vector is accumulated as:

Thus, the resulting dimension of the network becomes:

$$DIM(\Theta) = h(n+n+m) = h(2n+m) \tag{11}$$

$$\Theta = \left\{ w_{11}, w_{12}, \ldots w_{1m}, \mu_1^T, \sigma_1, \cdots, w_{h1}, w_{h2}, \ldots w_{hm}, \mu_h^T, \sigma_h \right\} \tag{12}$$

The network output in this case, can be derived as a summation of weighted Gaussian functions as follows:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \sum_{j=1}^{h} \left(\begin{bmatrix} w_{1j} \\ \vdots \\ w_{mj} \end{bmatrix} \varphi_j \right) = W^T \phi(X, \mu_1, \mu_2, \cdots, \mu_h) \tag{13}$$

where, $W^T = \begin{Bmatrix} w_{11} & w_{12} & \vdots & w_{1h} \\ w_{21} & w_{22} & \vdots & w_{2h} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1} & \vdots & \vdots & w_{mh} \end{Bmatrix}^T$ is a weighting matrix and ϕ is vector of RBFs with centers at $\boldsymbol{\mu}_j$.

Pruning. As continuous addition of nodes to the RBFN will increase the memory requirements and computational effort, it is desirable to remove the nodes whose contribution is not significant compared to the overall response of the entire RBFN that have been added to the network. For this purpose, contribution of each node is checked for last S_w observations and the nodes that contribute less than a threshold limit, δ during the last S_w observations are pruned from the network with parameters of the network and EKF appropriately updated after each pruning. The algorithmic implementation of pruning is given below:

a) Obtain the network parameters after EKF update ($k=1...h$ nodes)

b) For each $k=1....h$, determine the hidden node contribution of each of the nodes for the last S_w observations

c) Normalize this vector using maximum value of the hidden node contributions

d) Check if there are any RBFN nodes that contribute less than threshold limit, δ in the last S_w observations and store the corresponding nodal indices

e) Remove the corresponding hidden units from the network and update the EKF and network parameters appropriately

Modified MRAN (MMRAN). The modified MRAN can be implemented within the MRAN framework by incorporating the additional rotational parameters into the network formulation and additional entries in the sensitivity matrix. This method was studied basically to understand the underlying framework and validate if incorporating the rotation parameters of the covariance matrix enhances the performance of the existing framework. As a result, by including the rotational parameters of the Gaussian functions for learning into the RBFN state vector increases the overall dimension to

$$h\left(2n + m + n\left(\frac{n-1}{2}\right)\right) \Rightarrow h\left(m + 0.5n\left(\frac{n-1}{2}\right)\right) \tag{14}$$

where, n – dimension of the input space, m – dimension of the output space and h– number of network nodes. Otherwise, the algorithm to implement this is exactly similar to the earlier method. Thus, this results in additional network parameters that we can play with in order to obtain the more accurate approximation response. Essentially, such high dimensional network parameters are useful if the system under study has very high nonlinearities in a typical case of real tissues that are viscoelastic with time time-dependent and has to be considered from case to case. However, the results reported in this work are based only on the first method as the desired levels were performance was achieved with lesser parameters using MRAN method.

FINITE ELEMENT MODELING

In this work, the finite element simulation was carried out using COMSOL – a multiphysics simulation platform with a convenient MATLAB interface that allow us to execute custom scripts for repetitive analysis in an automated manner. In our problem, a parametric analysis was carried to conduct a parametric sweep of different values of loads. For the simplest case, only one dimensional load acting in a negative y direction was considered and for the most complex case, a two dimensional load vector acting in negative y direction was explored.

Two simple cantilever beams made of highly-elastic as well as plastic materials were developed using COMSOL (Fig. 2 & 3). Both these systems were subjected to different loads along y-direction (ranging from 0 – 10 N with 0.05 steps). The resulting displacements at all the nodes were recorded into a text file. The cantilever beams were basically assumed to be very thin and a constant cross-section (7.85e-5 m^2). The elements of the beam were Lagrangian quadratic type oriented along the length of the beam and comprised of 5 nodes. The length of entire beam is 0.1 m and length of each element is 0.02 m. Since, the beam was planar, nonlinear phenomena like warping or out-of-plane bending were considered negligible.

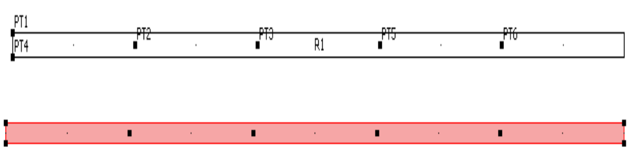


FIG. 2: NODE LABELING OF FE MODEL OF BEAMS

PRE-COMPUTATION ISSUES

For simple 1D examples that are not highly nonlinear, the performance of RBFNs were found to be poor. However, with increase in dimensionality and nonlinearity the prediction performance improved as shown for case 1 in the results section. Since, the FE model of the cantilever was assumed to be thin and long, the deformations were predominantly vertical and horizontal deflections were negligible. Hence, in all the FE approximation models, only the vertical deflection of each node is considered. This also helped in reducing the overall dimension of the RBFN problem in hand.

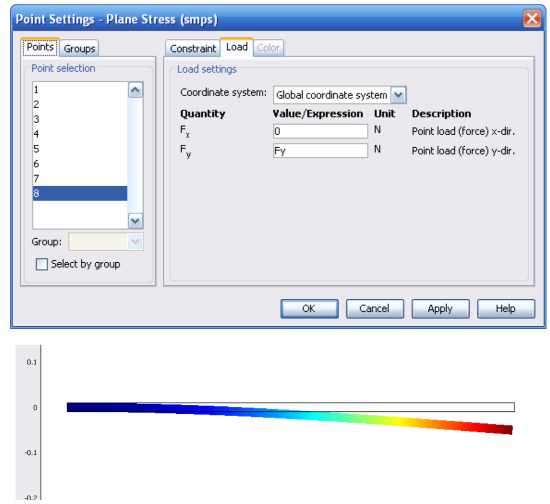


FIG. 3: COMSOL FEA MODELING OVERVIEW

Though, developing a neural network model for simulating the response of soft tissues was the primary goal of this work, due to time limitations, the FE model was simplified to a greater extent. Even for this simpler case, setting up of the overall problem and achieving the optimal set of tuning parameters to achieve better performance was challenging. Developing code for pruning proved to be a bigger challenge than any other part of the method. To avoid difficulty in the initial stages, hidden node contributions were checked at every instant (compared to last S_w observations) and appropriate nodes were instantaneously identified for pruning. Such a method resulted in loss of least vital nodes of the network and reduced the overall accuracy of the network output. In addition, with increase in input and output dimensions, updating the network and EKF parameters after each stage of pruning was one of the major issues faced and required careful manipulation of variables to accommodate for pruned nodes.

RESULTS

A number of reduced input-reduced output test cases were developed to improve the confidence levels in this framework (but are not reported here). Here, we present the 1D input- nD output (FEA approximation) for the beam in using material properties of both elastic (like copper metal) and plastic (like polymer fibers) materials.

1D Input- nD Output Problem (FEA Approximation)

The relevant material properties are tabulated in Table 1. The nodal point coordinate and element definition adopted for both the cases are shown in Table 2. In this work, the beam comprised of six nodes, one of them was fixed at one end. The vertical load ranging from 0 to 5 N was applied at the free end of the beam in the negative Y-direction.

TABLE I: MATERIAL PARAMETERS

Material	Properties	Values
Copper	Young's modulus (E)	110e9 [Pa]
	Poisson Ratio (υ)	0.35
	Density (ρ)	8700 [kg/m^3]
Nylon	Young's modulus (E)	2e9 [Pa]
	Poisson Ratio (υ)	0.43
	Density (ρ)	1150 [kg/m^3]

The displacements of 5 nodes of the beam were recorded at each force value and were used to learn the parameters of the neural network. By the symmetry condition, only forces along the negative Y direction were considered.

TABLE 2: NODE NUMBERING

Node Number	Node Coordinates
1	$[0,0]^T$
2	$[0.2, 0]^T$
3	$[0.4, 0]^T$
4	$[0.6, 0]^T$
5	$[0.8, 0]^T$
6	$[1, 0]^T$

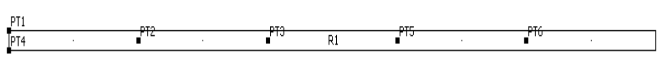
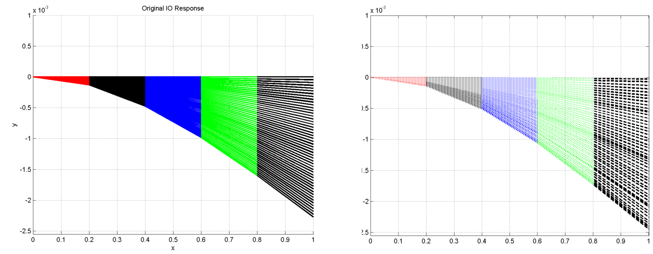


FIG. 4 NODAL NUMBERING OF FE MODEL OF BEAM

Case 1: Simple Elastic Model

The RBFN prediction of nodal displacements in case of copper beam is very accurate to the real solution (Fig. 5 (a) and (b)). In fact, as shown in and Fig. 6, the prediction error is nearly zero. However, the nodal displacements predicted by neural network show a tendency to drift from the test solution with higher forces especially along the boundaries as in Fig. 6. This becomes more visible and clear when results for case study 2 are discussed (Fig. 9).



(a) (b)

FIG. 5 RESPONSE OF FEA MODEL USING
(A) COMSOL FOR LOADS (FY = 0 TO -5 N IN 0.01 N STEPS)
(B) NEURAL NETWORK FOR LOADS (FY = 0 TO -5 N IN 0.01 N STEPS)

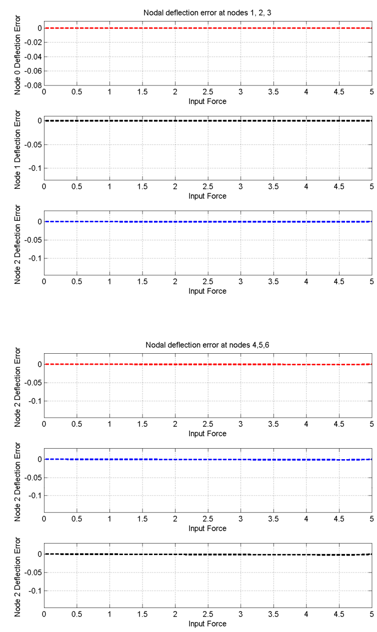


FIG. 6 NEURAL NETWORK RESPONSE ERROR AT 6 NODES VS INPUT FORCE (FY = 0 TO -5 N IN 0.01 N STEPS)

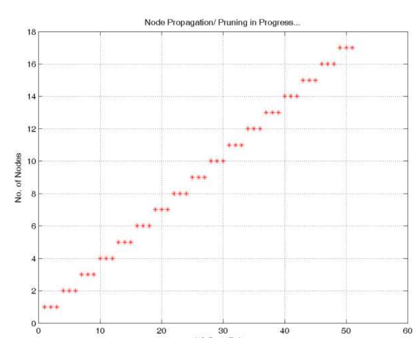
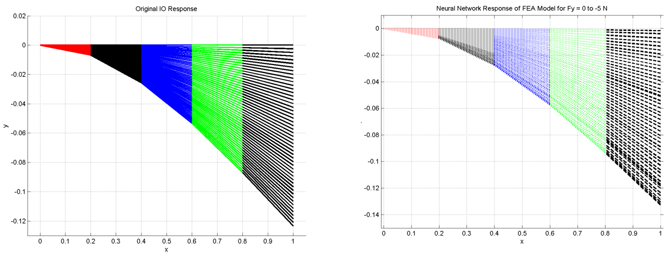


FIG. 7 GROWTH OF RBFN NODES

Case 2: Plastic Model

Since material library for soft tissues was not immediately available, a polymer fiber material (nylon) was chosen for implementing the RBFN based approximation. The mechanical properties of this fiber are depicted in the Table 1. However, efforts are underway to develop accurate nonlinear material model for soft tissues with custom material library to obtain simulation data for training the RBFN.



(a) (b)

FIG. 8 RESPONSE OF FEA MODEL USING
(A) COMSOL FOR LOADS (FY = 0 TO -5 N IN 0.01 N STEPS)
(B) NEURAL NETWORK FOR LOADS (FY = 0 TO -5 N IN 0.01 N STEPS)

The model prediction as shown in Fig. 8 (a) & (b) deteriorates as one proceeds to the outer node as well as for higher forces. The error in nodal displacements is shown in Fig. 10 for increasing values of force. This prediction error is partly because of the lesser degree of nonlinearity in these simpler models. Hence, the RBFN has to be carefully designed and tuned while testing the framework for more complex and realistic soft tissue models. The node propagation and pruning is shown in Fig. 10.

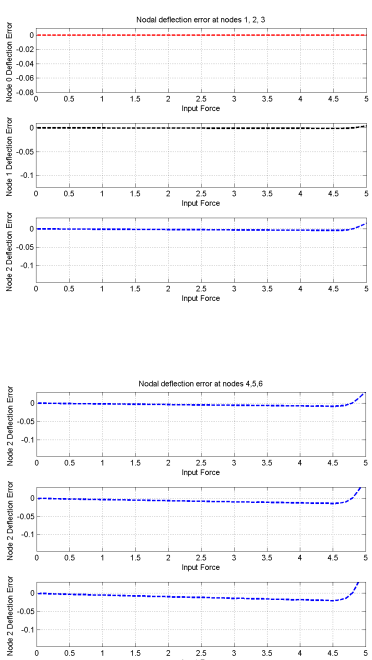


FIG. 9 NEURAL NETWORK RESPONSE ERROR AT 6 NODES VS INPUT FORCE (FY = 0 TO -5 N IN 0.01 N STEPS)

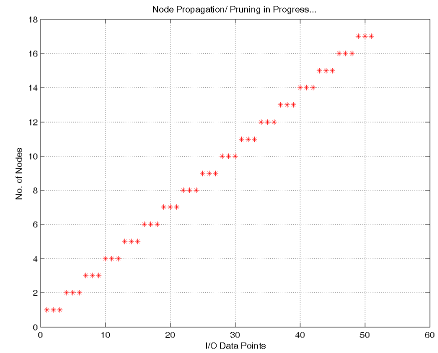


FIG. 10 GROWTH OF RBFN NODES

DISCUSSION

A Radial Basis Function Network (RBFN) based approximation method was studied in detail for applicability for creating real-time haptic computational response. The basic problem setup was developed for functional approximation purposes. An Extended Kalman Filter (EKF) was implemented for estimating and adapting the neural network parameters to improve the overall accuracy. Pruning of the network nodes was also implemented to reduce the size of the overall network by removing the least contributing nodes. We focused on implementing a RBFN approximation method for simulating the response of material interactions and deformations in real-time. Our next immediate task is to set-up the real-time interaction using appropriate visualization system and a haptic user-interface.

REFERENCES

[1] Lee, L.-F., Narayanan, M. S., Mendel, F. C., Karam, P., and Krovi, V. N., 2010, "Kinematics Analysis of In-Parallel 5 DOF Haptic Device," IEEE/ ASME International Conference on Advanced Intelligent MechatronicsMontreal, Quebec, Canada.
[2] Dorjgotov, E., Beneš, B., and Madhavan, K., "An Immersive Granular Material Visualization System with Haptic Feedback," Proc. EG UK Theory and Practice of Computer Graphics, 2007, D. D. Ik Soo Lim, ed.
[3] Chen, E., and Marcus, B., "Force Feedback for Surgical Simulation," Proc. IEEE, 1998, pp. 524-530.
[4] Delingette, H., "Toward realistic soft-tissue modeling in medical simulation," Proc. IEEE Special Issue on Surgery Simulation, 1998, pp. 512-523.
[5] Fung, Y. C., 1993, Biomechanics- Mechanical Properties of Living Tissues, Springer, Berlin.
[6] Maurel, W., Wu, Y., Thalmann, N. M., and Thalmann, D., 1998, Biomechanical Models for Soft Tissue Simulation, Springer, Berlin.

[7] Picinbono, G., Delingette, H., and Ayache, N., 2003, "Non-linear anisotropic elasticity for real-time surgery simulation."
[8] Chanthasopeephan, T., Desai, P. J., and Lau, C. W. A., 2005, "3D and 2D Finite Element Analysis in Soft Tissue Cutting for Haptic Display," 12th International Conference on Advanced Robotics, ICAR 2005 Seattle, WA, pp. 360-367.
[9] Kim, J., Choi, C., De, S., and Srinivasan, M. A., 2007, "Virtual surgery simulation for medical training using multi-resolution organ models," International Journal of Medical Robotics and Computer Assisted Surgery, 3, pp. 149-158.
[10] Jayawardena, A. W., Fernando, D. A. K., and Zhou, M. C., 1996, "Comparison of Multilayer Perceptron and Radial Basis Function networks as tools for flood forecasting," Destructive Water: Water-Caused Natural Disasters, their Abatement and ControlAnaheim, California, pp. 173-181.
[11] Chen, S., Cowan, C. F. N., and Grant, P. M., 1991, "Orthogonal least squares learning algorithm for radial basis function networks," IEEE Trans. Neural Networks, 2(2), pp. 302-303.
[12] Singla, P., and Junkins, J. L., 2007, Multi-Resolution Methods for Modeling and Control of Dynamical Systems, CRC Press.