WPRM-CPR(17) 17428

M Li et al.

Comparison of Deformation Models of Flexible Manipulator Joints for use in DEMO

Preprint of Paper to be submitted for publication in Proceeding of
27th IEEE Symposium On Fusion Engineering (SOFE)

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Comparison of Deformation Models of Flexible Manipulator Joints for Use in DEMO

Ming Li, Huapeng Wu, Heikki Handroos, Robert Skilton, Jonathan Keep, Antony Loving

***Abstract*—The joints deformation contribute significantly to final end-effector displacement of a manipulator, especially when the manipulator is in the form a serial kinematics with long links. When the manipulators employed in the DEMO are under the heavy payload, the deformation of manipulator is inevitable and the magnitude is significant. In order to maneuver the large object through several via points with high positioning accuracy in the remote handling process of DEMO, the real-time computation of manipulator deformation has to be conducted, which is crucial to the DEMO adaptive position control system for the displacement compensation. Three computation-effective deformation modeling methods are proposed in the paper, which are parametric modeling method, non-parametric deterministic artificial neural network modeling method, and non-parametric Bayesian artificial neural network modeling method, respectively. A specific joint in a boom equipped in a telescopic articulated remote mast is taken as the study object in the paper. A nodal deformation in the joint are investigated by three modeling methods, respectively. The parametric deformation model are derived by using the structural mechanics, whose parameters are identified by using the Markov chain Monte Carlo method; the deformation model of deterministic artificial neural network is trained by using the Levenberg-Marquardt method; and the deformation model of Bayesian artificial neural network is trained by using the Markov chain Monte Carlo method. The results show that the parametric model from the structural mechanics is linear and is incompetent in the deformation modeling when the non-linearity presents; both the deterministic and Bayesian artificial neural networks are capable of model the nodal deformation of joint. The performance of both the deterministic network and Bayesian network cannot rival for one another in the application scenario of paper. The training of Bayesian network can provide the criterions for estimation of possible ranges of the modeling outputs from its probabilistic distribution curves, and the judgement of proper size of network.**

***Index Terms*—Artificial neural networks, Bayes methods, DEMO, deformation model, flexible Manipulators, Markov chain Monte Carlo method**

I. Introduction

Inside the DEMO machine, heavy duty manipulators are foreseen to be widely employed. A hybrid kinematic manipulator (HKM) is being designed at RACE (Remote Applications in Challenging Environments) of UKAEA (UK Atomic Energy Authority) to handle the large breeder blanket segments for DEMO [1]. The payload of this HKM is around 80 tonnes, and its trajectory requires stringent position accuracy as it passes key points, in order to manoeuvre the blanket into and out of position in the vacuum vessel. The TARM (Telescopic Articulated Remote Mast) at RACE is also under upgrading, and it is necessary to investigate its's deformation displacement due to its massive weight and the payload [2].

From the past experience of heavy duty robotic machines, it is noticed that deformation of the manipulator joints contribute significantly to the end-effector displacement [3-4]. In order to compensate such end-effector deformation displacement in the control system in real-time, it is necessary to develop computation-effective deformation model of the flexible joints, which can then be integrated and form a deformation model of the whole manipulator. In addition the deformation model of manipulator can be further utilized to optimize the end-effector trajectory by using the iterative algorithms.

In order to support the large payload, the joints of the manipulator are complex, making it unreasonable to employ the truss and beam simplifications from the structural mechanics. In addition to an abstract joint structure, the integrated mechanical transmission and drive system makes the joint deformation model even more complex. The finite element analysis (FEA) method can estimate the deformation of a complex structure with high accuracy given the payload, however, its computation consumption makes it prohibitive to apply to the control system in real-time and in the iterative algorithms.

The paper proposes three approaches to model the joints deformation, which calculate the deformation in a computation-effective way: a parametric stiffness model identified by the Markov Chain Monte Carlo (MCMC) method, a deterministic non-parametric artificial neural network (ANN) model and a non-deterministic ANN model trained by using the MCMC method. All the models are identified or trained off-line using a dataset obtained in advance from the finite element analysis (FEA) on an abstract target joint without transmission and drive system. In practice, the proposed methods can be applied to model the deformation of joints incorporating the transmission

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Ming Li, Huapeng Wu, and Heikki Handroos are with the Lappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland (e-mail: ming.li@ lut.fi).

Robert Skilton, Jonathan Keep, and Antony Loving are with the Remote Applications in Challenging Environments, UKAEA, Culham Science Centre, Abingdon, Oxfordshire, OX14 3DB.

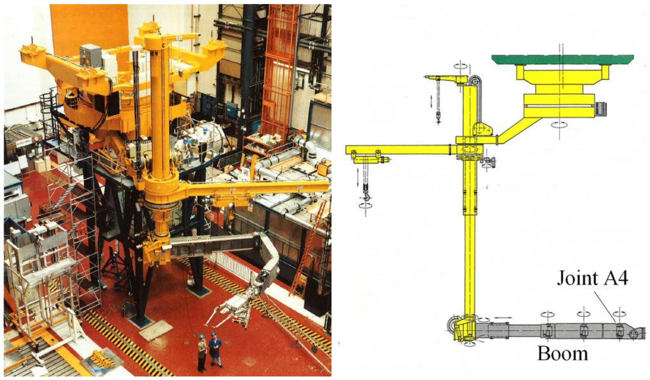


Fig. 1. TARM in-situ (left) and its schematic representation (right)

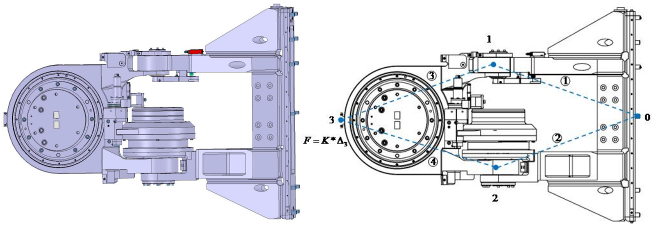


Fig. 2. Joint A4 in the boom of TARM and its deformation schematics

mechanisms and drive system, based on the real on-site measurement data, instead of using the FEA dataset on an abstract joint. The constructed models compute the deformation of joint in real-time, which is necessary in the control system.

The comparative results of applying proposed deformation models on a joint of heavy duty manipulator, namely TARM, are presented in the paper. The validation of parametric model, deterministic ANN model and non-deterministic ANN model are conducted, individually, by using different payload dataset and the deformation data obtained from FEA. The study can provide a good premise for constructing the real-time or computation-effective deformation model of entire manipulators that will be employed in the DEMO.

II. Deformation Analysis of Joint

Normally a joint of manipulator contains sub-joints structure, thus the matrix structural analysis (MSA) is employed herein to derive deformation kinematics of the joint, with the sub-joints structure is taken as the basic analysis element, which applies the same principle as FEA but in larger scale with much less nodes [5]. The advantage of applying the MSA in such scenario is its computation efficiency, whereas its weakness of less accuracy can be improved by deriving an accurate node stiffness matrix of each elementary structure. To present the application of MSA, the last second joint of a boom equipped in TARM is taken herein as an instance. Fig.1 shows the assembly of TARM, and Fig.2 shows the last second joint A4 of boom in TARM.

The joint A4 consists of two sub-joint structures, the upper and down joints in Fig.2, whereas applying the MSA, the corresponding nodes are represented by numbers 1, 2 and 3, respectively, and the elementary beams are represented from number 1 to 4 with a circle.

The analytic deformation model of this composed joint is expressed, as in

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} K_{11}^1 + K_{11}^3 & 0_{3\times3} & K_{13}^3 \\ 0_{3\times3} & K_{22}^2 + K_{22}^4 & K_{23}^4 \\ K_{31}^3 & K_{32}^4 & K_{33}^3 + K_{33}^4 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix}, \quad (1)$$

where F_1, F_2 and F_3 represent the external load acting on the nodes 1, 2 and 3; Δ_1, Δ_2 and Δ_3 represent the nodes' resultant deformation, respectively; the elementary item inside the big stiffness matrix (the first bracket part on the right side of (1)) is the individual nodal stiffness matrix and can be generalized in the form of K_{ij}^n, where the subscripts *i* and *j* denote the node numbers, and the superscript n denotes the beam, to which the nodes are attached. Physically the node stiffness matrix K_{ij}^n represents the needed equivalent force acting on node *i* of beam *n* while the node j undergoes a unit deformation.

In the joint deformation analysis, we concern more the deformation of the end-tip of joint, namely node 3 in this example. Since the external load only acts on the node 3 from its consecutive link, and the loads acting on the nodes 1 and 2 are assumed to be zero (neglecting the self-gravity effect), the stiffness model of node 3 can be derived from (1) and be expressed as

$$K = K_{22} - K_{21}K_{11}^{-1}K_{12}, \quad (2)$$

where *K* represents the stiffness matrix of node 3, and

$$K_{11} = \begin{bmatrix} K_{11}^1 + K_{11}^3 & 0_{3\times3} \\ 0_{3\times3} & K_{22}^2 + K_{22}^4 \end{bmatrix}_{6\times6}, \quad (3)$$

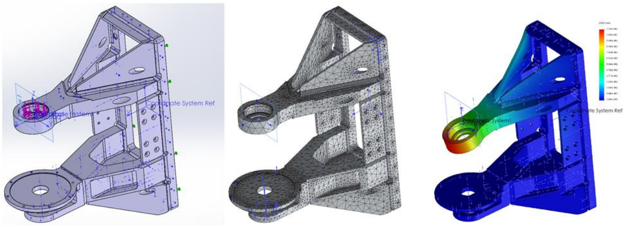
$$K_{12} = \begin{bmatrix} K_{13}^3 \\ K_{22}^4 \end{bmatrix}_{6\times3}, \quad (4)$$

$$K_{21} = \begin{bmatrix} K_{31}^3 & K_{32}^4 \end{bmatrix}_{3\times6}, \quad (5)$$

$$K_{22} = \begin{bmatrix} K_{33}^3 + K_{33}^4 \end{bmatrix}_{3\times3}. \quad (6)$$

In order to obtain the accurate stiffness model *K* of node 3, each elementary item K_{ij}^n inside the equations from (3) to (6) must be derived accurately. There are three methods feasible to construct K_{ij}^n: a) simplifying the node structure into the beam and truss mechanism and using the structural mechanics to compute a stiffness matrix; b) using the FEM to analyze the nodal structure and deriving a stiffness function from the FEM dataset; c) conducting the loading experiment and deriving a function from the measurement data. The first method is less accurate when the joint structure is complex; the second method is used in the paper, whereas the stiffness function is approximated by both parametric model and non-parametric ANN models; the third method will be used in future on-site.

In the paper, the stiffness K_{ij}^n are approximated, respectively, using a parametric stiffness model whose parameters are identified by using the MCMC method, a deterministic ANN



a. Applying random payload b. Meshing c. Analyzing deformation

Fig. 3. Obtaining data by using FEM in Solidworks COSMOS

model trained by Levenberg-Marquardt (LM) algorithm [6], and a non-deterministic Bayesian ANN model trained by using the MCMC [7-8].

III. Modelling Methods of Flexible Joint Stiffness

For demonstration, the stiffness modeling of upper joint (node 1 in Fig.2) is presented in the paper as an example. In order to obtain the training dataset used by the proposed three modelling methods, the CAD model of the joint are loaded into Solidworks (version 2017), and the FEA in its COSMOS package is employed to carry out the analysis. Fig.3 shows the procedures of generating the necessary data. A coordinate reference frame is set in the geometric center of the bearing face in node 1, with z axis pointing upwards along the bearing face geometric axis, x axis vertical to base plane of joint structure, and y axis determined by applying the right hand rule.

91 set of random loads are exerted on the bearing face of node 1, with magnitude ranging from 1000 to 5000 Newton, each of which consists of force along the z axis of the bearing face and normal to the bearing face, according to the practical loading scenario of this joint. All the force are distributed around the bearing face. The resultant deformation data of the bearing face geometrical centre is recorded consequently. The whole procedure is realized by a program written in VisualBasic which interfaces with the Solidworks COSMOS API [9], and automates the whole deformation data obtaining process. Part of the obtained data are presented in Table 1.

A. *Parametric Model by Markov Chain Monte Carlo Method*

According to the structural mechanics, if the nodal structure in the joint is simplified into beam, the parametric stiffness model of node 1 can be expressed as

$$K_{11}^1 = \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 \\ 0 & 0 & \frac{12EI_y}{L^3} \end{bmatrix}_{3\times 3}, \tag{7}$$

where *E* represents the Young's modulus of the beam; I_z and I_y are the quadratic moments; and *L* is the length of the beam.

Taking the diagonal items in (7) as the unknown lumped parameters $\theta = [\lambda_1 \quad \lambda_2 \quad \lambda_3]$ which need to be identified by MCMC, a linear model, with independent and Gaussian noise ε, can be presented in the form of

$$F_{11} = K_{11}^1(\theta)\Delta_{11} + \varepsilon, \tag{8}$$

where the F_{11} and Δ_{11} are the acting force and resultant deformation measurements, and vector θ is the unknown parameter being identified.

The corresponding Bayes formula is given, as in

$$p(\theta \mid F_{11}) = \frac{p(F_{11} \mid \theta)p(\theta)}{p(F_{11}) = \int p(F_{11} \mid \theta)p(\theta)d\theta}, \tag{9}$$

where $p(\theta)$ is the prior distribution of the unknown parameters; $p(F_{11} \mid \theta)$ is the likelihood function that gives the probability distribution of the observation F_{11} when given parameters value θ. The most likely values of the unknown parameters are those that give high values of the posterior distribution $p(\theta \mid F_{11})$.

TABLE I
Deformation Data Obtained Using FEM

Data No.		loads (unit: KN)			deformation (unit: mm)	
31	1.1852	-0.3176	-1.9860	-1.09	-3.3	-93.19
32	2.6665	0.7646	-2.3600	0.521	10.571	-110.37
33	1.8033	-0.5171	-0.7240	1.547	-6.31	-33.14
34	1.2145	0.3713	-1.8250	-0.72	5.347	-85.57
35	1.1074	-0.3386	-0.9170	0.274	-3.96	-42.51
36	1.7766	0.5772	-2.0830	-0.36	7.983	-97.37
37	1.0195	-0.3312	-1.2380	-0.21	-3.7	-57.67
38	1.0391	0.3578	-2.4670	-1.88	5.33	-116.21
39	1.5317	-0.5274	-2.3500	-1.11	-5.85	-109.82
40	2.2853	0.8318	-1.1410	1.489	10.931	-52.88
41	2.6311	-0.9577	-1.0880	2.084	-11.73	-49.51
42	2.7345	1.0497	-0.6070	2.882	13.452	-27.6
43	1.9437	-0.7461	-0.6170	1.702	-9.09	-27.73
44	1.4149	0.5716	-0.8640	0.66	7.461	-40.05
45	1.8507	-0.7477	-1.6020	0.113	-8.67	-73.7
46	2.5388	1.0776	-1.2480	1.555	13.876	-57.66
47	2.3418	-0.9940	-2.0490	-0.021	-11.54	-94.02
48	1.9093	0.8501	-2.4230	-0.82	11.424	-113.03
49	1.0889	-0.4848	-1.4510	-0.49	-5.54	-67.32
50	2.3446	1.0933	-1.8890	0.2057	14.1234	-87.28

[a] The payload consists of force along x, y, and z axis in the reference frame, and the units are kilo Newton.

[a] The deformation consists of deformation along x, y, and z axis in the reference frame, and the units are millimeter.

It is assumed that the residuals ε between the observed data F_{11} and the model outputs $\hat{F}_{11} = K_{11}^1(\theta)\Delta_{11}$ are normally and independently distributed with zero mean and constant variance σ_F, thus , the likelihood function is given by:

$$p(F_{11} \mid \theta) = \frac{1}{\sqrt{2\pi\sigma_F^2}} \prod_{i=1}^{N} \exp\left\{-\frac{[F_{11} - \hat{F}_{11}]^2}{2\sigma_F^2}\right\}. \tag{10}$$

When estimating the posterior of the parameters, it is common to ignore the evidence term $p(F_{11})$, and (9) can be further written in the form of proportionality

$$p(\theta \mid F_{11}) \propto p(F_{11} \mid \theta)p(\theta). \tag{11}$$

The adaptive Metropolis (AM) algorithm in MCMC is used to generate a sequences of random variables θ [10], whose empirical distribution can asymptotically approach to the posterior distribution $p(\theta \mid F_{11})$. The distinguished character of the AM is that the proposed prior distribution is updated based on the estimated posterior covariance matrix of the unknown parameters, whereas the posterior covariance matrix is computed at each simulation based on past simulations, thus, the proposed prior distribution is updated on the knowledge learnt so far from the posterior distribution. The proposal covariance matrix C_i can be computed, as in

$$C_i = \begin{cases} C_0 & , i \le i_0 \\ s_d Cov(\theta_0, ..., \theta_{i-1}) + s_d e I_d & , i > i_0 \end{cases}. \tag{12}$$

where the covariance C_i has a fixed value C_0 for the first i_0 simulations; s_d is a scaling parameter, and often computed by $s_d = (2.4)^2 / d$, where d is the dimensionality of parameter θ; *e* is a small parameter used to ensure the non-singularity of C_i; I_d is the d-dimensional identity matrix.

For $i > i_0$, the computation of covariance at simulation *i*+1 satisfies the recursive formula as in

$$C_{i+1} = \frac{i-1}{i}C_i + \frac{s_d}{i}(i\bar{\theta}_{i-1}\bar{\theta}_{i-1}^T - (i+1)\bar{\theta}_i\bar{\theta}_i^T + \bar{\theta}_i\bar{\theta}_i^T + eI_d), \tag{13}$$

where $\bar{\theta}_i = 1/(i+1)\sum_{x=0}^{i}\theta_x$.

The procedures of implementing the AM algorithm in MCMC are described as follows:

1) Compute C_i for current simulation, according (12), (13);

2) Generate a candidate vector θ^* for θ, where the candidate distribution $p(\theta^* | \theta_{i-1}) = N(\theta_{i-1}, C_i)$;

3) Compute the acceptance probability, α, of the proposed candidate:

$$\alpha = \min\left\{1, \frac{p(F_{11} | \theta^*)p(\theta^*)}{p(F_{11} | \theta_{i-1})p(\theta_{i-1})}\right\}, \tag{14}$$

where $p(F_{11}|\theta)$ is the likelihood function, $p(\theta)$ is the prior distribution of θ;

4) Generate $u \sim U[0,1]$, if $u < \alpha$, accept $\theta_i = \theta^*$, otherwise, set $\theta_i = \theta_{i-1}$;

5) Repeat step 1 for next simulation until the designated simulation iterations *K* has been reached.

The identified value of θ is computed as a mean by

$$\bar{\theta} = 1/(K-k)\sum_{i=k+1}^{K}\theta_i, \tag{15}$$

where *k* represents the initial simulations that need to be discarded to diminish the effects of initial distribution.

After obtained the stiffness matrix, the nodal deformation prediction can be computed by multiplying the inverse of stiffness matrix (which is called compliance matrix) with the vector of applied load.

B. Non-parametric Model by Deterministic ANN

When the underlying joint deformation model presents nonlinearity, the linear parametric model is no longer accurate, instead, a deterministic ANN model can be constructed based on the obtained FEA dataset or the on-site experiment data, which to some extent can reveal the underlying deformation physics when the scale of network is selected properly and the network is well trained. Fig.4 shows the general topology of a single hidden layer feedforward neural network.

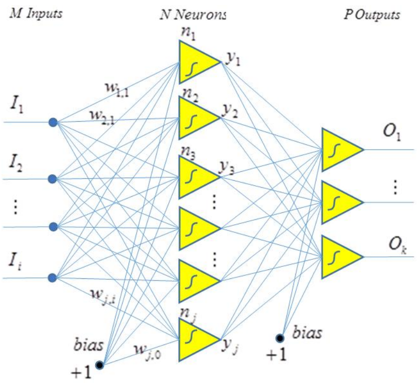


Fig. 4. Single hidden layer neural network architecture

M represents the number of inputs; *N* the number of neurons in hidden layer; *P* the number of outputs; I_i the *i*th input of network; O_k the *k*th output of the network; $w_{j,i}$ the weight of *i*th input of neuron *j*; $w_{j,0}$ the bias weight of neuron *j*; n_j the net value of all inputs of neuron *j*, and y_j the output neuron *j* which is computed as in

$$y_j = f_j(n_j), \tag{16}$$

where f_j is the activation function of neuron *j*, n_j is the sum of all inputs of neuron *j*, and expressed as

$$n_j = \sum_{i=1}^{M} w_{j,i}I_i + w_{j,0} . \tag{17}$$

The output O_k of neural network is computed by

$$O_k = f_k\left(\sum_{j=1}^{N} w_{k,j}y_j + w_{k,0}\right). \tag{18}$$

where f_k is the linear activation function of the output *K*.

In this paper, a feedforward neural network, with a hidden layer of twenty neurons, is employed to model the nodal deformation in the boom joint, of which the load vector F_{11} is taken as three inputs and the resultant deformation vector Δ_{11} is taken as three outputs. A sigmoid activation function is adopted in the hidden layer and a linear activation function is adopted in the output layer.

The Levenberg-Marquardt algorithm is utilized to train the network due to its high efficiency, which is represented by

$$W_{k+1} = W_k - (J_k^T J_k + \mu I)^{-1} J_k{}^T E_k . \tag{19}$$

where the subscripts k and $k+1$ represent the current and next iteration, respectively; W_k the weights vector of network; J_k the Jacobian matrix of network outputs error with respect to the weights; μ the combination coefficient that is used to adapt LM algorithm between the steepest decent method and Gauss-Newton algorithm; I the identity matrix which is introduced to guarantee the computation invertible in (19); E_k the error vector of network outputs.

In the training process, the initial value of μ is set to 0.001. When the current error increases as a result of weights update, μ increases by multiplying a factor of 10, otherwise, μ decrease by dividing a factor of 10. When μ reaches the minimum value 10e-5, it keeps the value constant until the error increases. The maximum training iteration in this example is set to 1000. In fact, the training process converges fast in this example, and has converged in around 60th iteration.

C. Non-parametric Bayesian ANN Model trained by MCMC

Compared with the single optimal weights vector of deterministic ANN, the ANN weights obtained from Bayesian inference typically performs better in the model prediction since the weights are selected widely by the overall information contained in the observed training data.

In the example of joint's nodal deformation modeling by Bayesian ANN, the same network structure is adopted with the previous deterministic ANN. The aim of ANN Bayesian training is to infer an acceptable approximation of the deformation model, which can be utilized to produce accurate predictions when presented with the new data. The nonlinear ANN deformation model, with independent and Gaussian noise, can be represented in the Bayes formula by

$$\Delta_{11} = g(F_{11} \mid w) + \varepsilon \,, \tag{20}$$

where $g(F_{11} \mid w)$ is the deformation function described by the ANN, and ε is the random noise term (measurement error) with zero mean and constant variance. The aim of the Bayesian training of ANN is to infer the weights posterior probability, providing the observed data $p(w \mid \Delta_{11}, F_{11})$.

The corresponding Bayesian model of ANN training is represented by

$$p(w \mid \Delta_{11}, F_{11}) = \frac{p(\Delta_{11} \mid w, F_{11})\, p(w)}{p(\Delta_{11} \mid F_{11}) = \int p(\Delta_{11} \mid w, F_{11})\, p(w) dw} \,, \tag{21}$$

where $p(w)$ is the prior weights distribution, $p(\Delta_{11} \mid w, F_{11})$ is the likelihood function, which depicts any information about w embodied in the data. In the actual implementation of the AM in MCMC in this paper, a Gaussian likelihood function is used, expressed by

$$L(w) = p(\Delta_{11} \mid w, F_{11}, \sigma^2) = \prod_{i=1}^{N=91} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{[\Delta_{11} - g(F_{11}, w)]^2}{2\sigma^2} \right\}, \tag{22}$$

where σ^2 is also estimated from the training data using Gibbs sampler (the simplest MCMC algorithm) [11].

In the multilayer ANN, the weights posterior distribution is typically complex and presents multimodal character, thus the assumption of a certain type of prior weight distribution is unreasonable. Herein, the AM algorithm is still adopted in MCMC for training ANN due to its adaptive proposal distribution.

The detailed procedures are similar to those used in the section A for parametric model identification, and presented as follows with slight difference:

1) Initialize the algorithm with an arbitrary set of weights w_0, and an arbitrary C_0 in the form of

$$C_0 = s_d \begin{bmatrix} \sigma_{w_1}^2 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & \sigma_{w_d}^2 \end{bmatrix}, \tag{23}$$

where $\sigma_{w_1}, \ldots, \sigma_{w_d}$ are standard deviations of the weights, initially set to some arbitrary positive values;

2) Compute C_i for current simulation;

3) Generate a candidate vector w^* for w, where the candidate distribution $p(w^* \mid w_{i-1}) = N(w_{i-1}, C_i)$;

4) Compute the acceptance probability, α, of the proposed candidate:

$$\alpha = \min\left\{ 1, \frac{p(\Delta_{11} \mid w^*, F_{11})\, p(w^*)}{p(\Delta_{11} \mid w_{i-1}, F_{11})\, p(w_{i-1})} \right\}; \tag{24}$$

5) Generate $u \sim U[0,1]$, if $u < \alpha$, accept $w_i = w^*$, otherwise, set $w_i = w_{i-1}$;

6) Repeat step 2 for next simulation until the designated simulation iterations *K* has been reached.

IV. Comparison of Results

The nodal deformation model identification and training results are presented in the Fig. 4, 5 and 6, respectively, from parametric model, deterministic ANN model and Bayesian ANN model.

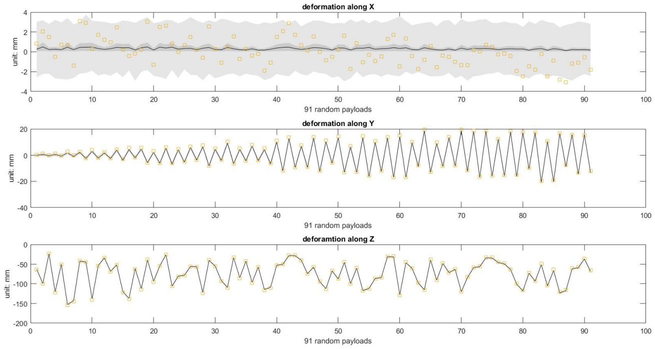


Fig. 4. Parametric deformation model identified by MCMC

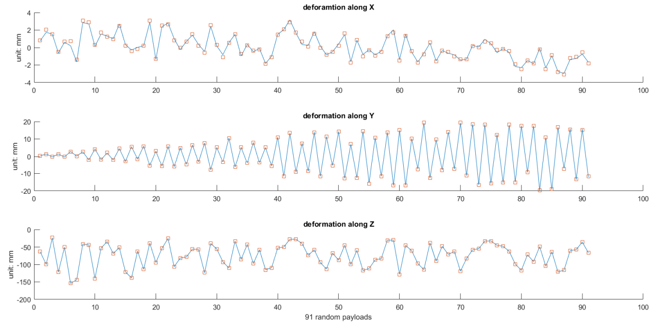


Fig. 5. Deterministic ANN deformation model trained by LM algorithm

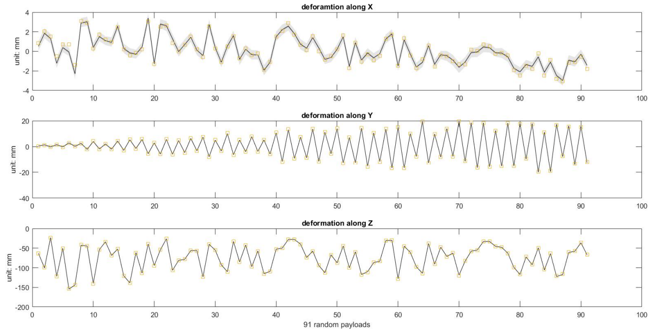


Fig. 6. Bayesian ANN deformation model trained by MCMC

The squares in figures represent the measured deformation data, with the unit in millimeter, after 91 random loads are applied, whereas the continuous lines represent the model predictions under the same payloads.

It is observed that in the linear parametric model, the prediction of deformation under the payload along x axis cannot approximate the actual deformation. The reason behind is that the actual deformation along the x axis of nodal 1 under the contraction (tension) in this joint is highly non-linear. However, the predictions of parametric model along y and z axis are promising, compared with the results along x axis. It reveals that the nodal 1 deformation along the y and z axis are approximately linear. The root mean squares (RMS) of deformation prediction in parametric model along x, y and z axis under random loads are 5.5863, 27.9727 and 2.6825, respectively.

Fig.5 and 6 both indicate that the deterministic ANN and Bayesian ANN under the training data can predict the

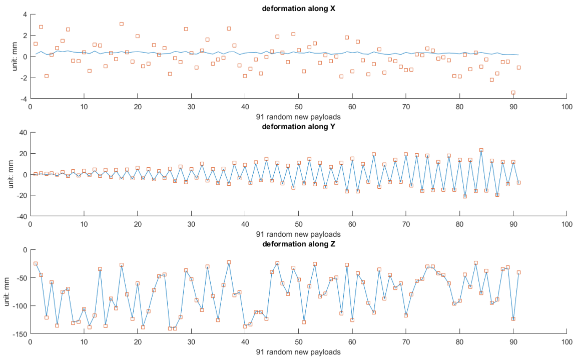


Fig. 7. Parametric deformation model prediction given new payload data

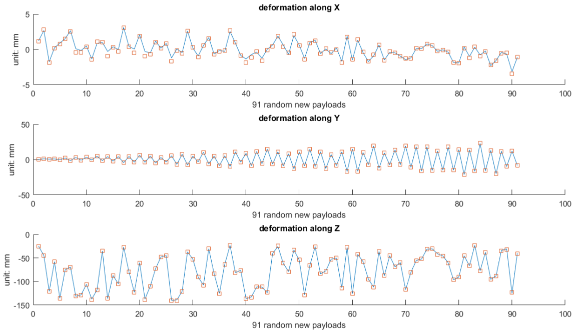


Fig. 8. Deterministic ANN deformation model prediction given new data

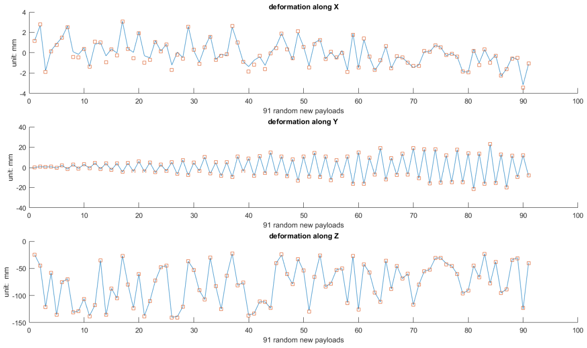


Fig. 9. Bayesian ANN deformation model prediction given new data

deformation accurately. The weights obtained in deterministic ANN is a single vector, whereas those obtained in Bayesian ANN present in the form of a probabilistic distribution. To compute the model prediction under certain payload in Bayesian ANN, the mean values of all the weights are utilized. The RMS of deformation prediction along x, y and z axis, in the deterministic ANN under the trained data, are 0.0498, 0.0020 and 0.0445, whereas those in Bayesian ANN model are 0.0165, 0.0088 and 0.0490, respectively.

Fig 7, 8 and 9 show the deformation predictions of parametric model, deterministic ANN model and Bayesian model, given the new payloads data (different from training data) in nodal 1.

Fig.7 indicates that the parametric model presents the similar performance in deformation prediction, given the new payload data, compared with the scenario where the training data is used. Herein we can conclude the actual deformation in nodal 1 along the x axis is highly non-linear, whereas those along the y and z axis are approximately linear, and the parametric model is incompetent to model such a deformation physics.

Given the new random payloads, Fig.8 and 9 both indicate that the underlying deformation physics in nodal 1 of the joint is identified successfully by both the deterministic ANN model and the Bayesian ANN model. The prediction performance between both models is close to one another in deformation modeling of nodal 1 of the joint. However, the Bayesian model can provide a probabilistic distribution of the prediction, which provides to some extend the perspective of deformation magnitude range. Meanwhile, the probabilistic distribution of the ANN weights can be obtained from the MCMC training process, which provides an efficient way to estimate if the proper size of the neural network is chosen. For convenience of comparison, all the RMSs of deformation prediction of nodal 1 in the joint, by three models under both the training data and the new data are presented in table 2.

TABLE 2
ROOT MEAN SQUARES OF MODELS UNDER TRAINING DATA AND NEW DATA

Under training data	deformation along X axis	deformation along Y axis	deformation along Z axis
Parametric Model RMS	5.5863	27.9727	2.6825
Deterministic ANN RMS	0.0498	0.0020	0.0445
Bayesian ANN RMS	0.0165	0.0088	0.0490
Under new Data	deformation along X axis	deformation along Y axis	deformation along Z axis
Parametric Model RMS	14.8158	30.6032	0.2154
Deterministic ANN RMS	0.4695	0.0420	1.8766
Bayesian ANN RMS	0.5102	0.0392	1.8745

V. CONCLUSIONS AND FUTURE RESEARCH

In order to compute the deformation of a complex joint, which consists of sub-joints, in a computation-efficient way, the overall joint deformation model can be derived in the kinematics form of sub-joints' deformation, by using the MSA method, whereas the sub-joints are taken as the elementary nodes by applying the same principle as in the FEA but in larger scale. Three modeling methods are proposed in the paper, to construct the accurate nodal deformation model in the joint, which are parametric model, deterministic ANN model and Bayesian ANN model, respectively. A specific sub-joint from a joint of a boom equipped in the TARM is taken as the study object of applying the proposed deformation modeling methods, and the comparative results are presented. We conclude that the parametric model of applying the structural mechanics is incompetent in modeling the nodal deformation when the non-linearity in deformation presents. Both the deterministic ANN and Bayesian ANN can model the underlying deformation physics successfully in the sub-joint structure, providing the amount of training data. In the specific example in the paper, the performance in both the deterministic ANN model and Bayesian ANN model cannot rival for one another, when the same size of network is used. The Bayesian ANN can provide the probabilistic distribution of the model which is useful in estimating the possible ranges of the model magnitude. In addition, the ANN weights probabilistic distribution can provide a perspective of estimating if the proper size of network is adopted.

The study of the paper is only limited to the deformation analysis of an individual abstract joint (without transmission mechanism and drive system), and its sub-joint deformation modeling, with the presence of only elastic deformation. When considering the realistic application of manipulator in DEMO, it may be not possible to conduct the measurement of an individual joint after the manipulator is assembled and deployed. Even the measurement data can be obtained in advance before the assembly of the manipulator or from the CAD model, it may deviate from the actual data after the manipulator is assembled and undergoes heavy duty operations. The backlash in the transmission mechanism of joint, the thermal and magnetic effect of the environment, and even the material property mutation due to the radiation, can all affect the final deformation physics of a joint, which are not investigated currently in the research of paper. The contribution of the paper, however, presents the feasibility and the potentials of applying the artificial neural network in modeling the nonlinearity of the deformation physics.

The future research will be focused on the deformation modeling of the entire manipulator, by integration of the individual joints, whose deformation physics will be identified by using the Bayesian ANN model, based on the measurement data from the end-effector of the manipulator.

References

[1] J. Keep, S. Wood, N. Gupta, M. Coleman, and A. Loving, “Remote handling of DEMO breeder blanket segments: Blanket transporter conceptual studies,” *Fusion engineering and design.*, Feb. 2017.

[2] A. C. Rolfe, “Remote Handling JET Experience,” JET Joint Undertaking, Abingdon, Oxfordshire, OX14 3EA, 1999.

[3] M. Li, H. Wu, and H. Handroos, “Static stiffness modeling of a novel hybrid redundant robot machine,” *Fusion engineering and design.*, vol. 86, no. 15-24, pp. 1838-1842, Feb. 2011.

[4] M. Li, H. Wu, and H. Handroos, “Stiffness-maximum trajectory planning of a hybrid kinematic-redundant robot machine,” in *IECON 2011*, Melbourne, VIC, Australia, 2011.

[5] M. Li, “Stiffness based trajectory planning and feedforward based vibration suppression control of parallel robot machines,” Doctoral dissertation, School of Energy Systems, Lappeenranta University of Technology, Finland: Acta Universitatis Lappeenrantaensis 611, 2014. [Online]. Available: http://urn.fi/URN:ISBN:978-952-265-703-9

[6] H. Yu, B. M. Wilamowsik, "Levenberg - Marquardt Training", in The Industrial Electronics Handbook, Vol. 5 − Intelligent Systems, 2nd ed. (CRC Press, Boca Raton, 2011).

[7] C. Andrieu, N. D. Freitas, A. Doucet, M. I. Jordan, “An introduction to MCMC for machine learning”, *Machine Learning*, Vol. 50, 5-43, 2003.

[8] G. B. Kingston, M.F. Lambert, and H. R. Maier, “Bayesian training of artificial neural networks used for water resources modeling”, *Water resources research*, Vol. 41, w12409, 2005.

[9] SOLIDWORKS API Help, Dassault Systèmes SOLIDWORKS Corp., http://help.solidworks.com/2017/English/api/sldworksapiprogguide/Welcome.htm

[10] H. Haario, E. Saksman, and J. Tamminen, “An adaptive Metropolis algorithm”, *Bernoulli 7(2)*, pp. 223-242, 2011.

[11] S. Geman; D. Geamn, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*. pp. 721–741. doi:10.1109/TPAMI.1984.4767596