EUROFUSION WPRM-CP(16) 15553

M Li et al.

Dynamic Model Identification Method of Manipulators for inside DEMO Engineering

Preprint of Paper to be submitted for publication in Proceedings of 29th Symposium on Fusion Technology (SOFT 2016)

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.

This document is intended for publication in the open literature. It is made available on the clear understanding that it may not be further circulated and extracts or references may not be published prior to publication of the original when applicable, or without the consent of the Publications Officer, EUROfusion Programme Management Unit, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK or e-mail Publications.Officer@euro-fusion.org

Enquiries about Copyright and reproduction should be addressed to the Publications Officer, EUROfusion Programme Management Unit, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK or e-mail Publications.Officer@euro-fusion.org

The contents of this preprint and all other EUROfusion Preprints, Reports and Conference Papers are available to view online free at http://www.euro-fusionscipub.org. This site has full search facilities and e-mail alert options. In the JET specific papers the diagrams contained within the PDFs on this site are hyperlinked

Dynamic Model Identification Method of Manipulators for inside DEMO Engineering

[1]Ming Li, [1]Huapeng Wu, [1]Heikki Handroos, [1]Yongbo Wang, [2]Antony Loving, [2]Crofts Oliver, [3]Matti Coleman, [2]Robert Skilton, [2]Guy Burroughes, [2]Jonathan Keep, and [2]RACE RM Control Team

[1]Laboratory of Intelligent Machines, Lappeenranta University of Technology, Finland
[2]Remote Applications in Challenging Environments (RACE), UKAEA, Culham Science Centre, United Kindom
[3]EUROfusion Consortium, Boltzmannstr.2, Garching 85748, Germany

In the inside engineering of DEMO, the robotic machines and manipulators are envisaged to be widely employed, which often have to deal with the demanding working conditions. The construction of the dynamic model of the robot and manipulator can be incorporated into the control system design to gain the adaptive control performance. A method of constructing the dynamic model with the unknown parts is proposed. The method can identify the unknown parts of the dynamic system by incorporating a BP neural network that will eventually substitute the unknown parts in the system dynamics after the well training. A modified Levenberg-Marquardt algorithm is developed for the training of BP neural network, which can back propagate the errors between entire actual system and the constructed dynamic model into the training process of the neural network. An example of constructing the dynamic model for a general Stewart structure mechanism is presented, in which the unknown parts, as well the entire dynamics are successfully identified. The proposed method can be extrapolated to the dynamics modeling of the blanket transporter in DEMO design, as well as the other general manipulators.

Keywords: robot, multi-layer neural network, system identification, dynamical model, Levenberg-Marquardt, demo

1. Introduction

In the inside engineering of DEMO, the robotic machines or manipulators are foreseeable to be widely employed, and the manipulators often have to deal with the demanding working conditions, and meanwhile respect the stringent positioning tolerance in the handling process. For example, the blanket transporter for handling the multi module segment (MMS) in DEMO concept has to handle the payload up to 80 tons while respecting tens of millimeter positioning accuracy. To meet these demanding requirements, the design of the adaptive positioning control system for the robot or manipulator has to be introduced. Herein, the dynamic modeling of manipulator plays a significant role to design the adaptive positioning control systems. Many researchers have been carrying out the research in utilizing the dynamical model of the robot in the adaptive control system design [1]. Additionally, the construction of the dynamic model of the manipulator can also benefit the performance evaluation and optimization work in the early design stage.

In practice, it is rather difficult to construct accurately the analytical dynamic model for the robots and manipulators. The reasons behind include, but not limited to, lacking the physical insight of some dynamic phenomenon, the inaccuracy or infeasibility of the direct measurements, the deviation of some dynamic properties after the robot and manipulator's assembly and deployment etc. For examples, it is difficult to construct the joint friction model individually in the parallel kinematic mechanism undergoing the deformation; it is also difficult to obtain the accurate inertial matrix as well as the geometrical mass center information for the structural complex manipulators. Quite often some components in the dynamic analysis process of the manipulator are deliberately omitted due to the identification complex, which results in the inaccuracy.

This paper proposes a method to construct the dynamic model of robots and manipulators that contain unknown components in the dynamics, which meanwhile can't be measured directly. The back propagation (BP) artificial neural network (ANN) is adopted to approximate and substitute the dynamics of the unknown components [2-4]. A modified Levenberg-Marquardt (LM) algorithm is developed, which can take advantage of the measurement data of the front-end input and output of the entire robotic system to train the BP neural network. In such a way, the direct measurement of the unknown components in the dynamics can be avoided since in practice the measurement of such unknown components for the ANN training is infeasible.

The proposed method in the paper is envisaged to be used to construct the dynamic model of a blanket transporter that will be applied in the DEMO remote maintenance (RM). Fig.1 shows the blanket transporter concept proposed in the work package WP RM by RACE in CCFE.

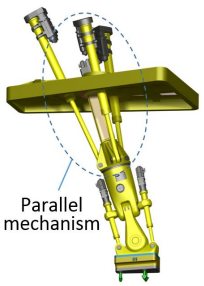


Fig.1 Concept of blanket transporter by RACE in CCFE

As an initial result, the parallel kinematic mechanism is considered as a highly potential choice being adopted in the blanket transporter design in the DEMO. Since the

author's email: Ming.Li@lut.fi

design of the blanket transporter is still a live process and the detailed concept is still undergoing evolution, for the general demonstration of applying the proposed method, a more general parallel kinematic mechanism – Stewart-Gough structure – is taken as the study object in the paper. To validate the proposed method, the inverse dynamics of the Stewart structure is constructed, in which the joint friction are taken as the unknowns and are identified and incorporated into the entire multi-body dynamics. The consideration behind is that the friction force in the joints of the potential blank transporter is envisaged to be significant due to the great payload and structural deformation, and thus can't be neglected in the dynamic modeling process. Moreover, the friction force in the joints of parallel structure presents the characters of highly non-linear, highly coupled and highly time-variant, which also render it infeasible to construct the model individually or measure it directly.

The rest structure of the paper is organized as follows: section 2 introduces the concept of the dynamic model identification of system with unknowns by using the BP ANN; section 3 describes the mathematics of the modified LM algorithm for the training of the BP ANN; section 4 demonstrates an example of the dynamic model identification of a parallel kinematic mechanism; section 5 brings forward the conclusions.

2. Concept of Dynamic Model Identification of Systems with Unknowns via ANN

For a general mechanical system, comprehensive knowledge of most parts of the dynamics is commonly available, while only a small part of dynamics present as unknowns which lead to the inaccuracy in the dynamics modeling process. Quite often the unknown parts are embedded in the dynamics and coupled with the systems fore and aft, which makes it difficult to measure the unknown components directly. In such scenario, the paper proposes to use the ANN to approximate the unknown components by using the front-end input output data from the entire dynamic system, and the scheme of the concept is shown in Fig.2.

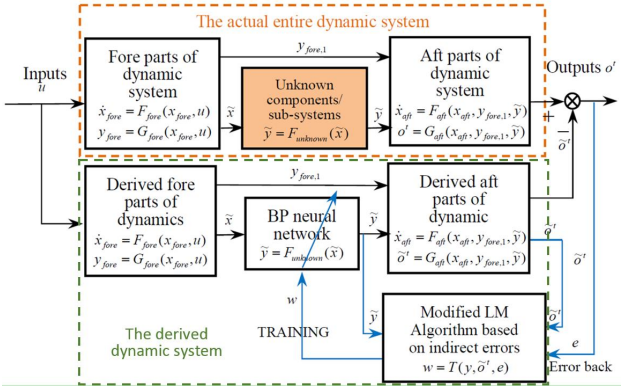


Fig.2 Neural network as an approximator for part of dynamic system

For generalization, the actual system with unknowns is expressed in the paper by the state space Eqs. from (1) to (4), which are derived from the fore and aft parts, as well as the unknowns respectively:

$$\dot{x}_{fore} = F_{fore}(x_{fore}, u), \tag{1}$$

$$y_{fore} = G_{fore}(x_{fore}, u), \tag{2}$$

where u is the input of system, x_{fore} is the state of the fore parts of system, and $y_{fore} = \left[y_{fore,1}^{\ T} \quad \tilde{x}^T \right]^T$ is the output of the fore parts, in which $\tilde{x}$ is the input to the unknown system and $y_{fore,1}$ is the variable coupling the fore and aft parts; F_{fore} and G_{fore} are the system matrix and output matrix of the fore parts respectively.

Similarly, the state space of the aft parts of entire dynamics is expressed as:

$$\dot{x}_{aft} = F_{aft}(x_{aft}, y_{fore,1}, \tilde{y}), \tag{3}$$

$$o^t = G_{aft}(x_{aft}, y_{fore,1}, \tilde{y}), \tag{4}$$

where x_{aft} is the state of the aft parts of system, $\tilde{y}$ is the output of the unknown system, and o^t is the output of the system; F_{aft} and G_{aft} are the system matrix and output matrix of the aft parts respectively.

The unknown components/sub-systems are expressed as:

$$\tilde{y} = F_{unknown}(\tilde{x}), \tag{5}$$

which is the identification object of the BP neural network.

A modified LM algorithm is developed in the paper to train the ANN, which can utilized the data that are from the entire dynamic system output, the derived dynamics of aft parts, along with the outputs of neural network. The training process is presented in Fig.2 by Eq. (6):

$$w = T(y, \tilde{o}^t, e). \tag{6}$$

By using the modified LM algorithm the direct training data for the unknown components are not needed, which can avoid the difficulty of measuring the unknowns. After the output error between the derived dynamics and the actual system converges to minimum, the unknown components are deemed as identified and the dynamics of entire system is deemed as constructed successfully.

3. Modified Levenberg-Marquardt Algorithm

This section first introduces briefly the original LM algorithm, then elaborates the modified LM training algorithm.

3.1 Levenberg-Marquardt Training Algorithm

The neural network training process by the original LM, which is evolved from Gauss-Newton algorithm, can be expressed by Eq. (7),

$$W_{k+1} = W_k - H_k^{-1} J_k^{\ T} E_k, \tag{7}$$

where the k and $k+1$ represent the current iteration and the next iteration; W_k and W_{k+1} represent the weight

vectors including all the individual weights of neural network; H_k is Hessian matrix representing a second order partial derivatives of the neural network output errors with respect to the neural network weights; J_k is Jacobian matrix representing the first order partial derivatives of the neural network output errors with respect to the neural network weights; E_k is output error vector between neural network and the target system;

To avoid the computation of the second order partial derivatives, the Hessian matrix is approximated in the LM algorithm by Eq. (8),

$$H_k = J_k^T J_k + \mu I \,, \tag{8}$$

where μ is a positive combination coefficient; and I is the identity matrix that is introduced to guarantee the invertibility of the Hessian matrix.

Substituting Eq. (8) into (7) gives weights updating rule for LM iteration training expressed as:

$$W_{k+1} = W_k - (J_k^T J_k + \mu I)^{-1} J_k^T E_k \,, \tag{9}$$

If the coefficient μ is very small, the Gauss-Newton algorithm dominates the Eq. (9) to give the advantage of speed; if the updating error and the combination coefficient μ increase, the steepest descent method dominates the Eq. (9) to guarantee a correct convergence training direction.

3.2 Modified LM Training Algorithm

In Eq. (9), the error vector is computed as the difference between the unknown system output and the neural network output. However, if the unknown system is a part of large system, and its outputs cannot be measured directly, as shown in Fig. 3, then the Jacobian matrix of Eq. (9) cannot be computed and the original LM algorithm is not applicable.

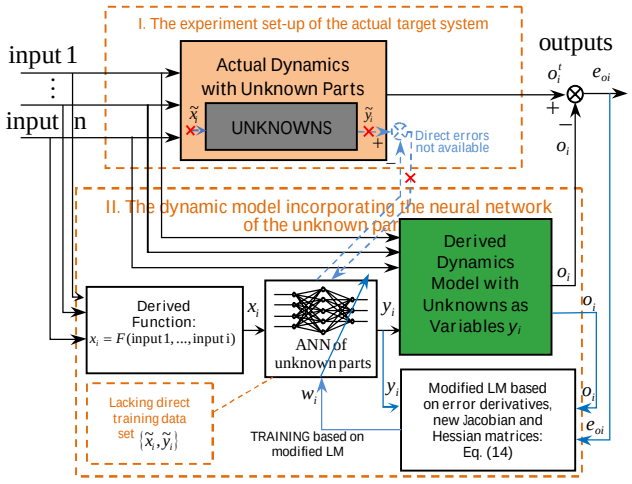


Fig.3 Neural network training by indirect error propagation

To train the neural network in such a scenario, a modified LM training algorithm is developed, where the new Jacobian and Hessian matrices are developed to represent the partial derivatives of the entire system output errors with respect to the neural network weights. The following notations are employed in this section:

o_i^t: the i*th* output of the actual entire dynamic system;

o_i: the i*th* output of the constructed dynamic model;

e_{oi}: the i*th* error between o_i^t and o_i, $e_{oi} = o_i^t - o_i$;

E_{o_k}: the error vector of all the errors e_{oi} in the k*th* iteration;

y_i: the i*th* output of the neural network;

w_i: the i*th* weight of the neural network;

M_k: the partial derivatives of the entire dynamics system output errors with respect to the neural network outputs in the k*th* iteration expressed as:

$$M_k = \begin{bmatrix} \frac{\partial(o_1^t - o_1)}{\partial y_1} & \frac{\partial(o_1^t - o_1)}{\partial y_2} & \cdots & \frac{\partial(o_1^t - o_1)}{\partial y_n} \\ \frac{\partial(o_2^t - o_2)}{\partial y_1} & \frac{\partial(o_2^t - o_2)}{\partial y_2} & \cdots & \frac{\partial(o_2^t - o_2)}{\partial y_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial(o_m^t - o_m)}{\partial y_1} & \frac{\partial(o_m^t - o_m)}{\partial y_2} & \cdots & \frac{\partial(o_m^t - o_m)}{\partial y_n} \end{bmatrix}_{m\times n} = -\begin{bmatrix} \frac{\partial o_1}{\partial y_1} & \frac{\partial o_1}{\partial y_2} & \cdots & \frac{\partial o_1}{\partial y_n} \\ \frac{\partial o_2}{\partial y_1} & \frac{\partial o_2}{\partial y_2} & \cdots & \frac{\partial o_2}{\partial y_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial o_m}{\partial y_1} & \frac{\partial o_m}{\partial y_2} & \cdots & \frac{\partial o_m}{\partial y_n} \end{bmatrix}_{m\times n} \tag{10}$$

J_{Y_k}: the Jacobian matrix between the neural network outputs and the weights expressed as:

$$J_{Y_k} = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_1}{\partial w_2} & \cdots & \frac{\partial y_1}{\partial w_i} \\ \frac{\partial y_2}{\partial w_1} & \frac{\partial y_2}{\partial w_2} & \cdots & \frac{\partial y_2}{\partial w_i} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_n}{\partial w_1} & \frac{\partial y_n}{\partial w_2} & \cdots & \frac{\partial y_n}{\partial w_i} \end{bmatrix}_{n\times i} ; \tag{11}$$

As a result, a modified LM algorithm for the neural network weights updating rule are expressed as:

$$W_{k+1} = W_k - (J_{o_k}^T J_{o_k} + \mu I)^{-1} J_{o_k}^T E_{o_k} \,, \tag{12}$$

where J_{o_k} is the new Jacobian matrix that is developed by applying the chain rule on Eqs. (10), (11) and is expressed by Eq. (13),

$$J_{o_k} = M_k J_{Y_k} \,. \tag{13}$$

It should be noted that Jacobian matrix J_{Y_k} in the modified LM algorithm is different from the one J_k in original algorithm: the J_{Y_k} is the derivatives of the neural network outputs with respect to the neural network weights, while the J_k is the derivatives of the neural network output errors with respect to the neural network weights.

To boost the training efficiency, the batch training of the modified LM algorithm can be implemented, and the updating rule of the neural network weights are realized by augmenting the Eq. (12) on both error vector E_{o_k} and Jacobian matrix J_{o_k} in the column direction, expressed as Eq. (14):

$$W_{k+1} = W_k - (J_{B_k}^T J_{B_k} + \mu I)^{-1} J_{B_k}^T E_{B_k} \,, \tag{14}$$

where E_{B_k} and J_{B_k} are the corresponding augmented error vector and the Jacobian matrix;

$$E_{B_k} = \left[{}^1E_{o_k}^T \quad {}^2E_{o_k}^T \quad \cdots \quad {}^pE_{o_k}^T \right]^T, \tag{15}$$

$$J_{B_k} = \left[{}^1J_{o_k}^T \quad {}^2J_{o_k}^T \quad \cdots \quad {}^pJ_{o_k}^T \right]^T, \tag{16}$$

where the superscript p is the index of data samples in a batch data.

4. Application of Dynamic Model Identification of Parallel Kinematic Mechanism

Fig.4 shows the schematic representation of the Stewart-Gough structure:

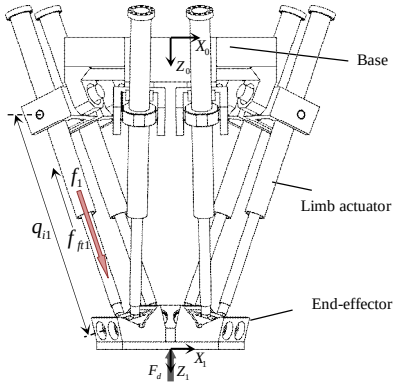


Fig.4 Schematic representation of Stewart structure

For the inverse dynamic modeling, all the obtained geometric parameters of this structure are assumed to be accurate. The friction models from the end-effector joints and limb actuator joints are deemed as the unknown systems. The friction models in the base joints are neglected in this study due to its minor effect.

4.1 Dynamic modeling of Stewart Structure with Friction Being Unknowns

By applying the Lagrangian formulation on the parallel structure, the inverse dynamics is obtained in Eqs. (17) and (18) with the friction force $F_{friction}$ being the unknown variables:

$$C_{q_d}{}^T \lambda = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{Q}_d}\right) - \frac{\partial L}{\partial Q_d} - F_d, \tag{17}$$

$$F_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{Q}_i}\right) - \frac{\partial L}{\partial Q_i} - C_{q_i}{}^T \lambda - F_{friction}, \tag{18}$$

where $\lambda = [\lambda_1 \ \lambda_2 \ \lambda_3 \ \lambda_4 \ \lambda_5 \ \lambda_6]^T$ represents the Lagrangian multipliers; L represents the kinematic and potential energy of the parallel structure; $Q_d = [q_{d1} \ q_{d2} \ q_{d3} \ q_{d4} \ q_{d5} \ q_{d6}]^T$ the 6 coordinates of end-effector reference frame $X_1Y_1Z_1$ in the global base frame $X_0Y_0Z_0$, which are the dependent coordinates in the Lagrangian equations; C_{qd} the kinematics constraint equations taking the partial derivatives with respect to the dependent coordinates; F_d the external force exerting on the end-effector; $Q_i = [q_{i1} \ q_{i2} \ q_{i3} \ q_{i4} \ q_{i5} \ q_{i6}]^T$ the length vector of the driving limbs in the Stewart structure, which are the independent coordinates in the Lagrangian equations; C_{qi} the kinematics constraint equations taking the partial derivatives with respect to the independent coordinates; $F_i = [f_1 \ \ldots \ f_6]^T$ the force vector that is needed from the limb actuators; $F_{friction} = [f_{ft1} \ \ldots \ f_{ft6}]^T$ the friction force vector in the limb actuators, which are an equivalent friction effect of the combined action of friction in the end-effector joints and limb actuator joints.

Due to the unknown equivalent friction models in the actuators, the computable inverse dynamics model can't be derived from Eqs. (17) and (18), which, in the sense of solving the linear equations, means the vector variable F_i that is the driving force needed from the actuator can't be computed thereby.

In order to obtain the computable inverse dynamic model of the Stewart structure, the BP neural network is used to approximate the equivalent friction models based on the input-output data set obtained by carrying out the control experiments on the Stewart structure. Fig.5 shows the experimental set-up for obtaining the training data of the neural network. As a convention, all the symbols denoting measured data or training data in the paper is capped by a tilde, hereafter.

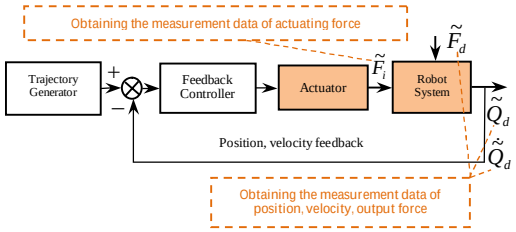


Fig.5 Experimental set-up for obtaining the training data set

In the experiment, the prescribed trajectories should cover domains of interest that the robot will be used for in practice. The accuracy of the trajectory tracking is not the focus herein, since the data set $\{\tilde{Q}_d, \dot{\tilde{Q}}_d, \tilde{F}_d\}$ of the position, velocity and payload of the end-effector and the data set $\{\tilde{F}_i\}$ of force from the actuators are the concerns. After the experimental data set are obtained, the BP neural network can be trained by utilizing the modified Levenberg-Marquardt algorithm. The training implementation is shown in Fig 6.

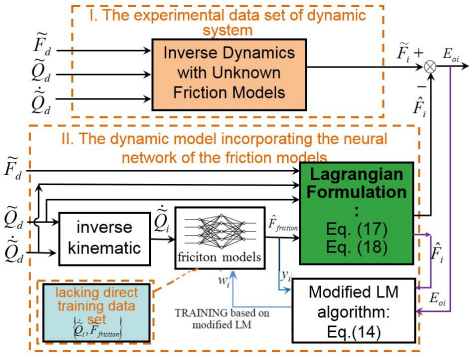


Fig.6 Neural network training by indirect error propagation

Although the direct training data set $\{\tilde{Q}_i, \tilde{F}_{friction}\}$ for the neural network of friction models are not available due to the immeasurability of friction in the joints, the errors E_{oi} between the $\tilde{F}_i$ from experiment (Fig.6.I) and the $\hat{F}_i$ from the constructed dynamic model (Fig.6.II) can still be utilized to train the neural network indirectly, because the errors herein are caused by the inaccuracy between the ANN model and the friction models. The modified

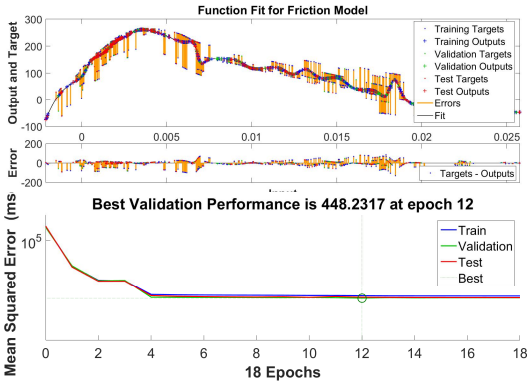
LM training algorithm is applied for the neural network training, which is represented by Eq. (14) in Fig.6.

4.2 Experimental Results

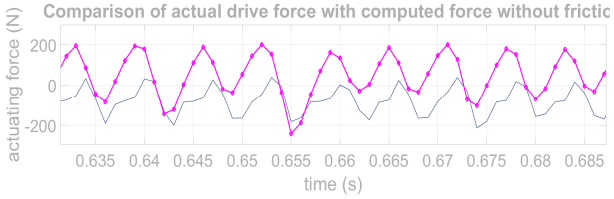
For demonstration, the Stewart structure dynamics modeling result is presented based on a specific end-effector trajectory where the end-effector moves along a trajectory in z direction with a speed of 10 mm/second. The motion of the limbs in this scenario is identical.

A multi-inputs-outputs neural network with a single hidden layer was employed for friction models of each limb, where the velocity vector of each limb comprises the 6 inputs of the neural network, while the friction in the limb is the output. 20 neurons are used in the hidden layer. For the training process, 2000 set of samples are used from actuator driving force and the corresponding end-effector positon and velocity.

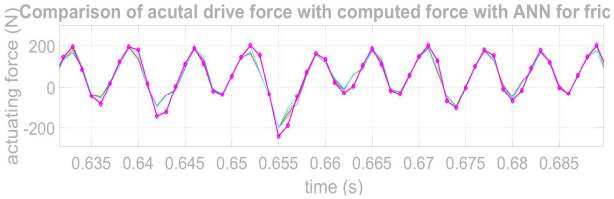
Results for the training process and the constructed inverse dynamics model in the given trajectory is shown in Fig.8.



a) ANN training of approximating friction in one of the actuators



b) Comparison of the actual drive force with the derived force of inverse dynamics without friction models



c) Comparison of the actual drive force with the derived force of inverse dynamics with ANN friction models

Fig.8 Inverse dynamic modeling results by incorporating the ANN friction models

Fig.8 (a) shows the convergence status of the friction model approximation, where the training converges fast, and reaches the best at epoch 12, and finished at epoch 18. The reason for the big mean squared error is the neural network is not fitting every training samples since the number of training samples are significant large. This is reasonable because the overfitting of neural network should be avoided.

Fig.8 (b) shows a piece of segment in the comparison of the actual drive force with the computed drive force from the derived inverse model without considering the friction, where the purple diamond line represents the actual drive force, and the rest lines represent the computed force of limbs from the inverse dynamics, and the mean error value of comparison in the entire trajectory is -1.330e+02.

Fig.8 (c) shows comparison of the actual drive force with the force from constructed inverse dynamic model with the ANN of friction, where the purple diamond line represents the actual drive force, and the rest represent the computed force, and the mean error value is 3.8723e-01. The results demonstrate that the construction of the inverse dynamic model for the given trajectory is successful by taking into account the unknown friction with ANN. It should be noted that, in Fig.8(c) there is still, to some extent, the mismatch between the actual drive force and derived drive force. The reason is the inertial matrix and mass information in the constructed dynamic model is slightly inaccurate from the practical machine. However, this is out of the discussion of this paper, and will be the future research work.

5. Conclusions

The dynamic model will notably benefit the control system design of manipulators in DEMO RM. The paper proposed a method using the BP ANN to approximate the unknowns in the dynamics with the comprehensive knowledge of the rest part of system. A modified LM algorithm is developed for the training of ANN. The method is applied successfully in the dynamics modelling of a parallel structure with unknown friction models. The method is general for the dynamic systems, and is envisaged to be extrapolated to the manipulators applied in DEMO.

Acknowledgements

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.

References

[1] M Li, H Wu, H Handroos etc. Vibration control for parallel manipulator based on the feedforward control strategy, ASME International Mechanical Engineering Congress and Exposition, San Diego, California, USA, Nov 15-21, 2013.

[2] K Narendra, K Parthasarathy, Neural networks and dynamics system, Part II: Identification, Tech. Rep. 8902, Center Syst. Sci., Dept. Elect. Eng., Yale Univ., New Haven, CT. Feb. 1989.

[3] H Yu, B Wilamowsik, Levenberg-Marquardt Training, the Industrial Electronics Handbook, Vol. 5 - Intelligent Systems, 2nd ed. (CRC Press, Boca Raton, 2011).

[4] S.B, H.S, Single layer neural networks for liner system identification using gradient decent technique, IEEE Tran.