HAL
open science

On Neural Networks' Ability to Approximate Geometrical Variation Propagation in Assembly

Loïc Andolfatto, François Thiébaut, Marc Douilly, Claire Lartigue

To cite this version:

Loïc Andolfatto, François Thiébaut, Marc Douilly, Claire Lartigue. On Neural Networks' Ability to Approximate Geometrical Variation Propagation in Assembly. 12th CIRP Conference on Computer Aided Tolerancing, Apr 2012, Huddersfield, United Kingdom. pp.224 - 232, 10.1016/j.procir.2013.08.035 . hal-01094434

HAL Id: hal-01094434

https://hal.science/hal-01094434v1

Submitted on 12 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On neural networks' ability to approximate geometrical variation propagation in assembly

Loïc Andolfatto[ab]*, François Thiébaut[bc], Marc Douilly[a], Claire Lartigue[bc]

[a]*EADS Innovation Works, 7 Route de Bray, 80300 Méaulte, France*
[b]*LURPA, ENS de Cachan, 61 Avenue du Président Wilson, 94235 Cachan Cedex, France*
[c]*IUT de Cachan, 9 Avenue de la Division Leclerc, 94235 Cachan Cedex, France*

Abstract

Tolerance analysis is an important step to validate assembly process planning scenario. Simulations are generally performed to evaluate the expected geometrical variations of the assembled product. When the simulation models take into account part compliance, assembly sequence and contact interaction, the resulting behaviour of the assembly are generally non-linear and simulations – mainly performed using finite element analysis – require high computing efforts. This paper investigates the ability to approximate the non-linear propagation of geometrical variations in assembly with artificial neural networks. The aim is to drastically reduce the computing efforts required for the simulation and therefore allow its use for the geometrical tolerances allocation optimisation. The influence of the neural network design parameters on the approximation quality is presented in a case study. The quality of the neural network approximation is also evaluated and discussed.

Keywords: geometrical variation propagation; assembly; contact influence; neural network

1. Introduction

Due to several sources of geometrical variations, the actual geometrical state of an assembled product differs from its nominal state. Some key characteristics (KC) of the product – related to the fulfilment of functional requirements – are identified and bounded to define the acceptable differences between the actual state and the nominal state of a product. A product with all its KC within the specified boundaries is declared conform.

Tolerance analysis consists in predicting the expected conformity rate – the percentage of conform assembled products – of a production. It requires the use of an appropriate method to simulate the KC deviations according to the geometrical variations of the components and to the dispersions of the assembly processes. When the assembly sequence, the compliance of the components and the influence of contact at interfaces are taken into account, the geometrical variation propagation during the assembly follows a non-linear relation, especially when the key characteristics are gaps at interfaces between components. This relation is generally obtained thanks to a finite element analysis (FEA) [1, 2, 3, 4, 5].

If one knows the propagation relation for the assembly of one product, a stochastic analysis can be used to evaluate the probability density function (PDF) of each key characteristic deviation and evaluate the value of the expected conformity rate. The stochastic analysis may require numerous evaluations of the propagation relation of the assembly. The resulting intensive use of non-linear FEA is highly computationally expensive. Thus, the computing cost required for a tolerance allocation method based on iterative tolerance analyses seems to be a prohibitive one.

Several papers present solutions to reduce the computational cost of geometrical variation propagation in non-linear cases. Cai *et al.* proposed a second order Taylor series development in [2]. The second order terms extend the method of influence coefficient (MIC) proposed by Liu and Hu [1] to give an approximation of the non-linear relation. It can drastically reduce the

* Corresponding author. Tel.: +33 (0)3 22 64 32 15; fax: +33 (0)1 46 97 34 04
. *E-mail address*: loic.andolfatto@eads.net ; @lurpa.ens-cachan.fr

number of FEA required but it only provides a local approximation in the neighbourhood of the evaluated point. Dahlström and Lindkvist [3] proposed a contact search algorithm coupled with an adapted MIC to model the influence of contact. The method proved to be approximately five times faster than non-linear FEA with little loss of accuracy. Xie *et al.* presented the direct evaluation of the KC deviations probability density functions thanks to a statistical analysis based on Taylor series development obtained with a small amount of FEA results and a problem dimension reduction [4]. This method leads to a sizeable gain in computing time for the evaluation of statistical moments of the KC distribution. Ungemach and Mantwill investigated the use of a retroactive contact consideration to correct MIC results [5]. This method also led to lower computing efforts than FEA to evaluate a close approximation of the KC deviations considering the contact influence.

This paper deals with an original method to approximate the non-linear geometrical variation propagation in assembly with an artificial neural network. This leads to a sizeable gain in computing time once the neural network is trained and allows performing multiple tolerance analyses with various input conditions (*e.g.* optimisation of tolerance allocation) with low computing efforts.

The following section presents several approximation methods among which the neural networks seem to be the most suitable. It also focuses on the topology and architecture of artificial neural networks (NN). The third section gets into the details of the experiment plan performed to identify relevant neural network design parameter sets for geometrical variation propagation approximation and to evaluate the associated approximation quality. The fourth and fifth sections present a case study and the experimental results obtained. Section six is the conclusion.

2. Approximation of non-linear geometrical variation propagation relations

2.1. Metamodels for non-linear relations

Considering a problem entity (*e.g.* geometrical variation propagation in assembly), a causal simulation model can be established (*e.g.* the parametric FEA-based geometrical variation propagation relation in the assembly). Kleijnen defines a metamodel as an approximation of the multi-input/multi-output relation given by the simulation model [6].

Considering $\boldsymbol{x}$ as the input vector and $\boldsymbol{y}$ as the output vector resulting from the model of the problem entity, the simulation model can be mathematically expressed by a function $f(\boldsymbol{x}) = \boldsymbol{y}$. Evaluating the probability density function of $\boldsymbol{y}$ (*e.g.* the KC deviations gathered in a vector) requires mapping the input parameters space. The computing effort required to evaluate the simulation model f on numerous $\boldsymbol{x}$ samples can be an obstacle to a suitable method. A computationally efficient solution consists in computing a learning set of simulation results $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i\in\{1\ldots N\}}$ to build a metamodel f^* that minimises ε as defined in eq. (1):

$$\varepsilon = \sum_{i=1}^{N} |f^*(\boldsymbol{x}_i) - \boldsymbol{y}_i| \qquad (1)$$

Such a metamodel should provide a good prediction $\boldsymbol{y}^* = f^*(\boldsymbol{x})$ for untested inputs $\boldsymbol{x}$ although built from a small size N of the learning set. Generally, those two characteristics are antagonistic.

A common approach consists in fitting the simulation results with a linear model. This is basically the case of the method of influence coefficients [1]. This method is not suitable when the problem entity is non-linear. Second order polynomials methods are also commonly used but they generally only provide suitable approximations in a local area in the input space. Splines can be used to build a global model according to multiple local fitting. Unfortunately, the most common multivariate spline methods are based on interpolating splines, which are not applicable for metamodelling purposes [7]. This class of methods is therefore limited to univariate or bivariate input space.

Inverse distance methods, including local linear approximation methods, define another class of simple metamodels, but the computing cost to evaluate f^* increases with the size of the learning set which is related to the number of components of $\boldsymbol{x}$. It becomes an issue when the number of input variables of the problem entity involves a large learning set (*i.e.* a large N). The same problem exists with spatial correlation (kriging) metamodels. Kriging can briefly be described as a spatial-dependant linear combination of the observations $\{\boldsymbol{y}_i\}_{i\in\{1\ldots N\}}$ of the learning set.

Artificial neural networks can deal with multi-input/multi-output approximation problems [8]. It can be seen as a general formulation of several classes of metamodels aforementioned [7].

Finally, kernel regression methods can provide better approximation quality than neural networks with a lower algorithmic cost [9] but it requires a strong mathematical background to be handled properly.

The artificial neural networks seem to be suitable for metamodelling the geometrical variation propagation in assembly, as they can provide good approximations of large multi-input/multi-output simulation models while requiring a reasonable mathematical background.

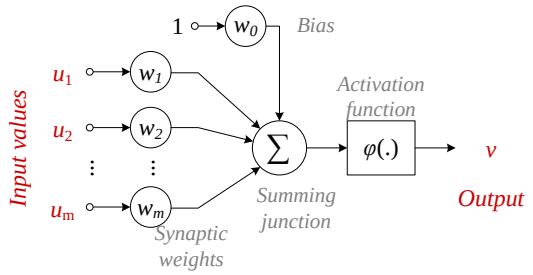

Fig. 1. Non-linear model of a neuron (adapted from [8]).

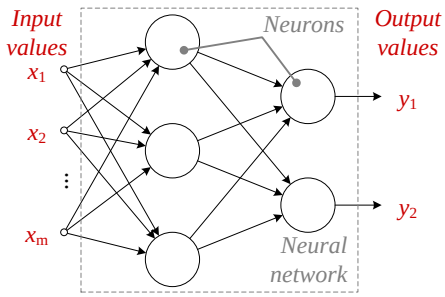

Fig. 2. Example of a neural network with 5 neurons, m inputs and 2 outputs.

2.2. Description of an artificial neural network

The artificial neuron is the fundamental information-processing unit of a neural network. From the mathematical point of view, it is a function of m variables $(u_1 \dots u_m)$ returning a scalar v. Fig. 1 is a block diagram representing the non-linear model of a neuron. Its transfer function is given in eq. (2), using the conventions $u_0 = 1$ and $w_0 = b$.

$$v = \varphi\left(b + \sum_{j=1}^{m} w_j \cdot u_j\right) = \varphi\left(\sum_{j=0}^{m} w_j \cdot u_j\right) \qquad (2)$$

The w_j are called synaptic weights and b is the neuron bias. The function φ is called activation function. Sigmoid functions are usually employed as activation functions for approximation purposes [10].

An artificial neural network is a function obtained by composition of artificial neurons, as depicted in Fig. 2. The graphical representation is more convenient than an equivalent mathematical expression of the neurons' transfer function composition. The topology of the network is characterised by the number of neurons and the existing connections between them. A particular topology called multilayer perceptron is known to be efficient for metamodelling purpose [8]. Fig. 2 shows the example of a perceptron with a two neurons output layer and one layer of three neurons, called *hidden layer*. The approximation capabilities of the network depend on the number of hidden layers and also on the number of neurons in each layer. The higher the number of neurons and the number of hidden layers, the more complex the pattern that can be approximated. A high number of neurons also leads to a complex training.

When the topology of the network and the activation functions are chosen, the metamodel function f^* only depends on the synaptic weights and bias of the neurons. The final step of the metamodelling – the *training* of the neural network – aims at estimating the values of those parameters (weights and biases) to minimise an objective function evaluated on a training data set $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i\in\{1\dots N\}}$. In the case of approximation neural networks, the objective function commonly used is the mean squared error defined in eq. (3) (optimised with respect to the w_j parameters).

$$mse(\{w_j\}) = \frac{1}{N}\sum_{i=1}^{N}\left(f^*(\boldsymbol{x}_i, \{w_j\}) - \boldsymbol{y}_i\right)^2 \qquad (3)$$

The training is commonly performed with an algorithm called *back propagation*.

2.3. NN-based geometrical variation propagation in assembly

Fig. 3 schematically displays how the key characteristics are simulated according to the FEA-based geometrical variation propagation relation (blue frame). It also explains how they can be approximated with a trained neural network (orange frame). The following section focuses on the neural network design (pictured in the green frame).

3. Neural networks design and assessment

3.1. Experiment plan

Fig. 4 summarises the main parameters that influence the approximation quality of the neural network. In our case, some design parameters (in light grey in Fig. 4) can be chosen according to the literature on neural networks. The choice of sigmoid activation functions, multilayer perceptron topology or back propagation training algorithm with mean squared error as objective function are not discussed in this paper as they appear to be relevant for metamodelling purpose [8]. Formal relations known between components of the input vector $\boldsymbol{x}$ and output vector $\boldsymbol{y}$ are also previously assigned to avoid letting the neural network learns it, what reduces the number of neurons required, and the number of parameters estimated during the training.

Other parameters are more problem-dependant. An experiment plan is performed to identify relevant sets of neural network design parameters for the approximation of geometrical variation propagation in assembly and to evaluate the associated approximation quality. The following subsections are describing the design parameters and the neural network approximation quality indicators chosen in this study.

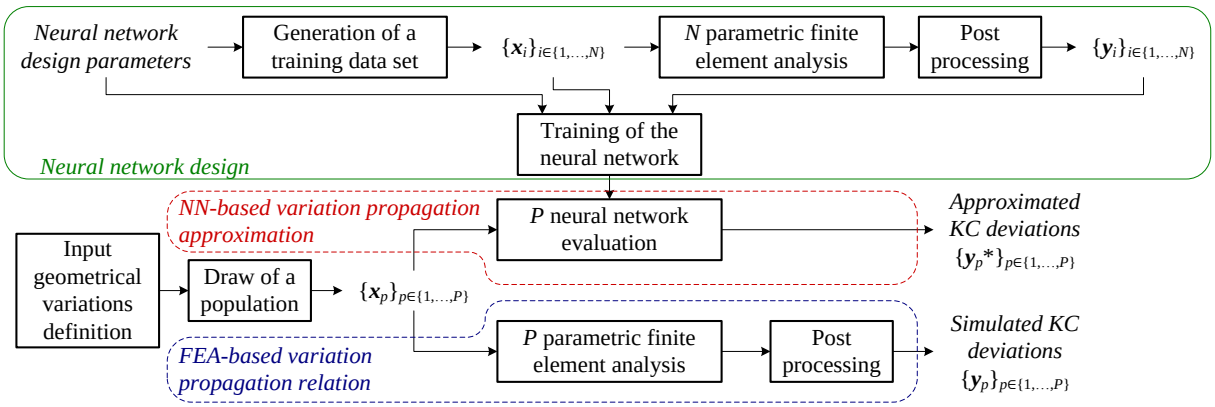

Fig. 3. Principle of the geometrical variation propagation in assembly approximated with a neural network compared to direct finite element analysis variation propagation simulation.

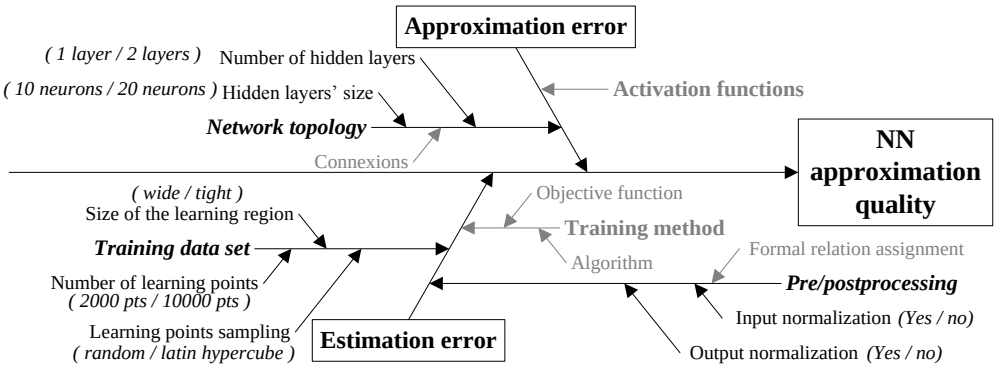

Fig. 4. Cause-and-effects diagram inventorying characteristics of a neural network (design parameters investigated in dark).

3.2. Design parameters selection

As mentioned previously, the number of hidden layers and the number of neurons in each hidden layer (called hidden layers' *size*) can have a significant impact on the approximation quality. For those two parameters, two levels are chosen: one and two layers and 10 and 20 neurons per layer.

The inputs $\{\boldsymbol{x}_i\}_{i\in\{1\dots N\}}$ and outputs $\{\boldsymbol{y}_i\}_{i\in\{1\dots N\}}$ can be normalised to reduce scale effects between components and to reduce estimation errors due to related training issues. That gives two other design parameters with two levels each.

The training data set $\{\boldsymbol{x}_i\}_{i\in\{1\dots N\}}$ must be designed to map efficiently the input space with a minimum number N of finite element analyses to perform and nevertheless providing a good generalisation, *i.e.* an estimation of the w_j parameters that allows a good prediction for $\boldsymbol{x} \notin \{\boldsymbol{x}_i\}_{i\in\{1\dots N\}}$. The impact of the number N of training points is evaluated: two levels $N = 2000$and $N = 10000$ are chosen.

Two different sampling methods are commonly used to generate the training data set $\{\boldsymbol{x}_i\}_{i\in\{1\dots N\}}$: random uniform distribution for each variable and Latin Hypercube Sampling of the learning region. Theoretical pros and cons are not discussed in this paper. The two methods are evaluated and compared, leading to another design parameter with two levels.

The size of the learning region is defined by the volume of the convex hull formed by the training data set in the hyperspace of dimension m, where m stands for the number of components of the input vector $\boldsymbol{x}$. The learning region must obviously include the entire region where the geometrical variation propagation relation needs to be approximated. A neural network trained on a wide region provides the ability to approximate the geometrical variation propagation relation for a large variety of inputs (*i.e.* provides a wide search space for tolerance allocation optimisation) but can lead to higher approximation errors. To evaluate the impact of the size of the learning region on the approximation quality, the training of the neural network is performed either on a wide or on a tight learning region (Fig. 5). This is the last design parameter.

The seven design parameters (summarised in dark in Fig. 4) with two levels lead to 128 (2^7) experiments to identify relevant sets of parameters for the geometrical variation propagation approximation.

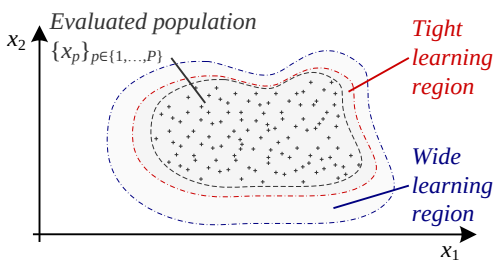

Fig. 5.Example of a wide and a tight learning region compared to the approximation region of the neural network in the case of a two dimensions input space.

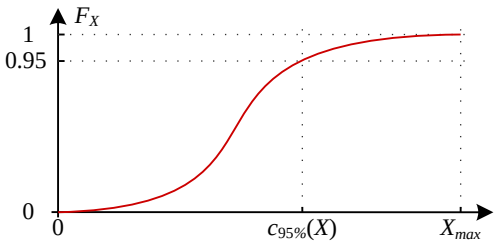

Fig. 6. Example of a cumulative density functions F_X and illustration of the 95% coverage interval $c_{95\%}(X)$.

3.3. Approximation quality indicators

The paper deals with the evaluation of the expected conformity rate associated with assembly processes and with geometrical tolerances of components. The quality of the metamodel may thus be expressed with indicators representing the approximation of this quantity at the population level. Meanwhile, it is still interesting to evaluate the generalisation quality at the individual level (*i.e.* the ability to predict the value of the output for an untested input) to assess the confidence in the neural network approximation.

Considering a population of P input vectors $\{\boldsymbol{x}_p\}_{p\in\{1\dots P\}}$ representing the geometrical variations of parts and assembly processes for the assembly of P products, the reference simulation model responses $\{\boldsymbol{y}_p\}_{p\in\{1\dots P\}}$ are computed thanks to P finite element analyses. That gives the reference used to evaluate the metamodel response $\{\boldsymbol{y}_p^*\}_{p\in\{1\dots P\}} = \{f^*(\boldsymbol{x}_p)\}_{p\in\{1\dots P\}}$.

The indicators presented in eq. (4) are described for a scalar valued population $\{y_p\}_{p\in\{1\dots P\}}$. As the output $\boldsymbol{y}$ is assumed to be a vector, the indicators are calculated for each of the components of $\boldsymbol{y}$ separately and the worst case along the dimension of $\boldsymbol{y}$ is chosen to characterise the global quality of the approximation.

$$\begin{cases} \delta c_{95\%} &= \left|c_{95\%}\left(\{y_p\}\right) - c_{95\%}\left(\{y_p^*\}\right)\right| \\ \overline{\delta y} &= \frac{1}{P}\sum_{i=1}^{P}|y_i - y_i^*| \end{cases} \qquad (4)$$

where $c_{95\%}(X)$ denotes the 95% coverage interval of the set X as defined in Fig. 6. The associated indicator $\delta c_{95\%}$ expresses the error on the predicted boundary of y^* for a 95% target conformity rate.

The generalisation quality at the individual level is evaluated through $\overline{\delta y}$ which is the mean of the absolute generalisation error among the population.

4. Case study

4.1. Problem entity

The case presented in this paper is the pre-assembly of a mechanical structure depicted in Fig. 7. The structure is composed of two feet, a frame and a square. The feet and the frame are positioned with a tooling before assembly (see Fig. 7 (b)). The assembly sequence is given in Fig. 7 (a).

The key characteristics of the pre-assembled structure are the maximum gaps between each couple of surfaces nominally in contact (see Fig. 7 (b)). The three corresponding key characteristic deviations are the problem entity outputs.

The arrows in Fig. 7 (b) expresses a contact between two surfaces of two components or a kinematic link between two components [11]. The problem entity inputs model the components and assembly processes geometrical variations. They are decomposed in two categories.

The dispersions of the situation of each component are due to geometrical variations of the kinematic links. Each relation on degrees of freedom imposed by a kinematic link is prone to geometrical variations. To reduce the size of the problem, the positioning of the feet by the tool is supposed to be performed without introducing geometrical variations. The five other kinematic links considered are leading to nine relations on degree of freedom prone to variations. Those variations are expressed with nine components of the vector $\boldsymbol{x}$.

The geometrical variations of each component are described by modes. The concept of mode-based description and tolerancing of geometrical variations has been explained by Huang and Ceglarek in [12] and by Samper and Formosa in [13]. The magnitude of each of the five unitary modes depicted in Fig. 7 (c) is also an input of the problem entity, giving five more components of the vector $\boldsymbol{x}$.

Although they can be considered as an input of the problem entity due to their impact on the resulting gaps, the clamping forces introduced by the staples (steps *iv-v-vi*) are not considered to be prone to variations in this study.

The outputs of the problem entity are the maximum gaps at interfaces between surfaces. As the simulation takes the contact interaction into account, the gaps are necessarily positive and the multi-input/multi-output geometrical variation propagation relation is non-linear.

4.2. Simulation model and neural network approximation

The training data sets and the reference responses are generated with parametric non-linear finite element analyses conducted with Cast3M. Each kinematic link in the assembly is converted into boundary conditions the parameters of which are the associated geometrical deviations. The meshes of the components are also deviated according to the magnitude of their modes. To take the assembly sequence into account, the boundary conditions of each step of the sequence are updated according to the solution of the finite element problem of the previous step.

The reference data set $\{\boldsymbol{x}_p\}_{p\in\{1\ldots P\}}$ has 10000 points ($P = 10000$). Each variable $\boldsymbol{x}$ has a normal distribution with mean values and standard deviations detailed in Table 1.

This table also summarises the boundaries of the tight and wide learning regions for each input variable.

The experiments are conducted with the *Neural Networks* toolbox of MATLAB®. The training of a neural network requires initial values before optimising the weights. The introduction of randomness in those initial values is recommended to avoid introducing prior information [8]. The training of each network is then performed 30 times with random initial weights and the worst value of each of the indicator defined in eq. (4) among those 30 trainings is considered.

Table 1. Characteristics of the reference data set ($[\mu - 3\sigma, \mu + 3\sigma]$ intervals) and boundaries of the learning regions (in mm for the kinematic links (indexes corresponds to Fig. 7 (a)) variations and modes magnitudes.

Kinematic link	*dof*	Ref.	Tight	Wide
Tool-Foot	*Tx, Ty, Tz*	[0 0]	[0 0]	[0 0]
Tool-Frame *(1)*	*Tx*	[0, 0.2]	±0.3	±0.5
	Ty	[-0.7, 0.7]	±0.8	±1
	Tz	[-0.7, 0.7]	±0.8	±1
Tool-Frame *(2)*	*Tx*	[0, 0.2]	±0.3	±0.5
	Tz	[-0.7, 0.7]	±0.8	±1
Tool-Frame *(3-4)*	*Tx*	[0, 0.2]	±0.3	±0.5
Foot 2-Square *(5)*	*Tx*	[-0.3, 0.8]	±0.8	±1
	Ty	[-0.5, 0.7]	±0.8	±1

Part	Mode	Ref	Tight	Wide
Frame	1	[0, 0.75]	±0.75	±1
Foot 1	1	[0, 0.2]	[0, 0.4]	[0, 1]
Foot 2	1	[0, 0.2]	[0, 0.4]	[0, 1]
Square	1	[-0.3, 0.3]	±0.4	±1
	2	[-0.5, 0.8]	±0.8	±1

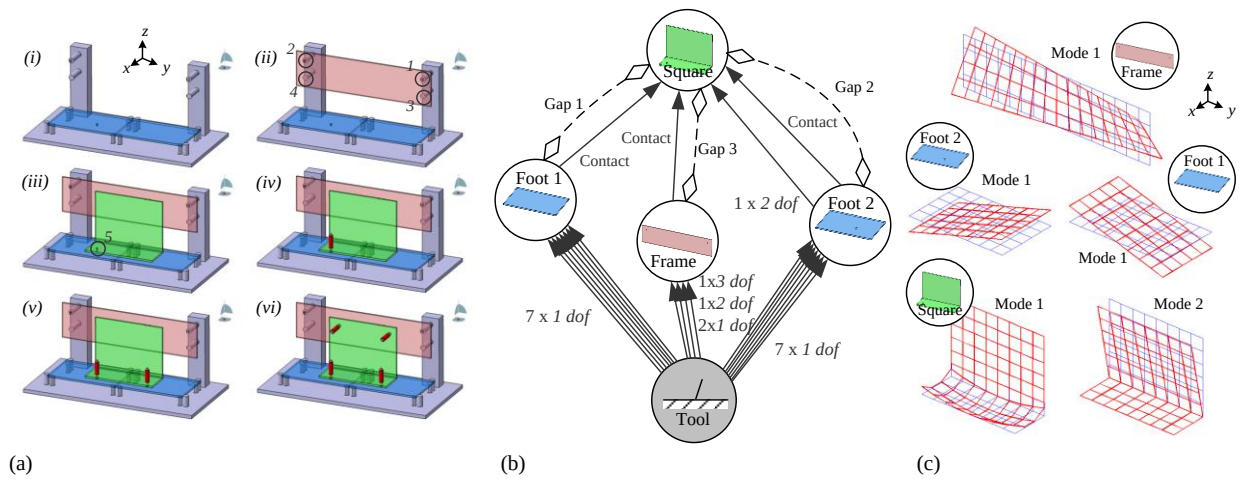

Fig. 7. (a) Assembly sequence of the use case: (i) Feet positioned by tool; (ii) Frame positioning positioned by tool; (iii) Square positioning positioned on foot 2 (hole to hole) and frame;(iv) Clamping of square on foot 2; (v) Counter-drilling and clamping of square on foot 1;(vi) Counter-drilling and clamping of square on frame. (b) GAIA® graph [11] of the positioning link of the components. Uncertainties are associated with each relation on degree of freedom. (c) Geometrical variation modes of the components.

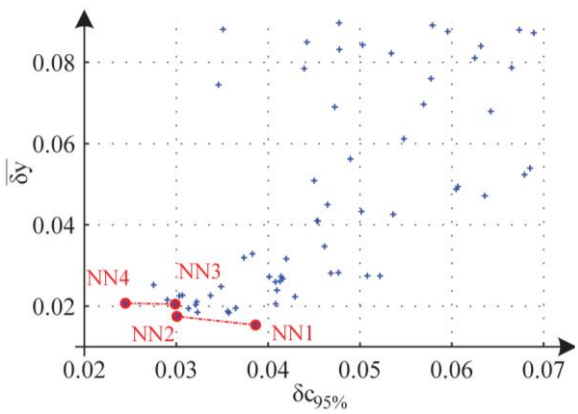

Fig. 8. Indicators for the 128 tested networks of the experiment plan (blue crosses) and Pareto front (red line) in the $(\overline{\delta y}, \delta c_{95\%})$ plane for the case study.

Table 2. Design parameters of the Pareto-optimal networks (LHS and RU stand for *Latin Hypercube Sampling* and *Random Uniform* respectively).

Design parameters	NN1	NN2	NN3	NN4
Nb. of hidden layer(s)	1	1	1	1
Hidden layer size	10	10	10	20
Input normalization	Yes	No	No	Yes
Output normalization	No	Yes	No	Yes
Nb. Of learning points	10000	10000	10000	10000
Sampling	LHS	RU	RU	RU
Training region size	Tight	Tight	Tight	Tight

Table 3. Approximation performed with the Pareto-optimal networks in the worst case among the 30 trainings compared to the reference simulation (values in *mm*).

	Gap 1		Gap 2		Gap 3	
	$c_{95\%}$	$\overline{\delta y}$	$c_{95\%}$	$\overline{\delta y}$	$c_{95\%}$	$\overline{\delta y}$
Ref.	0.262	/	0.276	/	0.231	/
NN1	0.263	0.015	0.273	0.013	0.193	0.006
NN2	0.263	0.017	0.277	0.014	0.201	0.006
NN3	0.264	0.016	0.274	0.021	0.202	0.008
NN4	0.262	0.016	0.265	0.021	0.207	0.002

5. Results

5.1. Relevant sets of design parameters

The search of optimal sets of design parameters is a multi-objective optimality problem due to the two indicators considered. The approach chosen for solving this problem consists in searching the Pareto front in the $(\overline{\delta y}, \delta c_{95\%})$ plane, as shown in Fig. 8. In this case, there are four optimal networks (see Fig. 8) the design parameters of which are given in Table 2.

The four networks are trained on a tight learning region mapped with 10000 points. As far as the training data are not prone to noise, this result can be generalised: the higher the density of learning points, the better the approximation. They all have a single hidden layer, what may be due to the smoothness of the geometrical variation propagation relation. In a general case, we can assume that this relation is smooth enough to be approximated with a single layer perceptron.

The input and output normalisation do not seem to affect the approximation quality in the tested case. It is probably because of the natural homogeneousness of the data in this case. Yet, the training of networks with normalised data is generally faster and the NN4 – which appears to be a good trade-off among the optimal networks – is built with normalised data. The Latin Hypercube Sampling generally allows a faster training than the random uniform sampling. But in matter of approximation quality, no strong trend can be observed among the networks close to the Pareto front.

The hidden layer size is always a hard parameter to select [8]. The more neurons, the larger the estimation error but the lower the expected approximation error. NN4 has 20 neurons in its hidden layer but the worst results (not visible in Fig. 8) are also obtained with 20 neurons in the hidden layer. Thus, further experiments need to be performed to find the optimum hidden layer size relatively to the approximation quality in the general case.

5.2. Associated approximation quality

Table 3 presents $c_{95\%}$ and $\overline{\delta y}$ of each of the three key characteristics for the reference data set $\{\boldsymbol{y}_p\}_{p\in\{1\ldots P\}}$ and for the $\{\boldsymbol{y}_p^*\}_{p\in\{1\ldots P\}}$ obtained with the four Pareto optimal networks.

The $c_{95\%}$ of the three gaps obtained with NN1, NN2, NN3 and NN4 need to be compared to the reference values.

The highest $\delta c_{95\%}$ of 0.039 mm (neural network 1, gap 3) is around twenty per cent of the reference $c_{95\%}$. This error is the worst case among 30 neural network trainings. For most of the tested case, the $\delta c_{95\%}$ error is quite lower than ten percent, especially for gap 1 and 2. It is therefore interesting to notice that the size of the coverage interval is widely under-evaluated for gap 3, what can denote that the neural networks tend to be bad at approximating the tails of cumulative density functions. This can be an issue to evaluate the conformity rate.

The average generalisation error is around five per cent of the reference $c_{95\%}$.

The global approximation performance evaluated in this case study can be seen as satisfactory for many tolerancing applications. Therefore, this good approximation ability may be seen cautiously. The reference data set is nearly centred in the learning region, with normal distribution of the variables. Those particularities may favour a good approximation. Nevertheless, it can be difficult to evaluate a confidence indicator with a reasonable computing time.

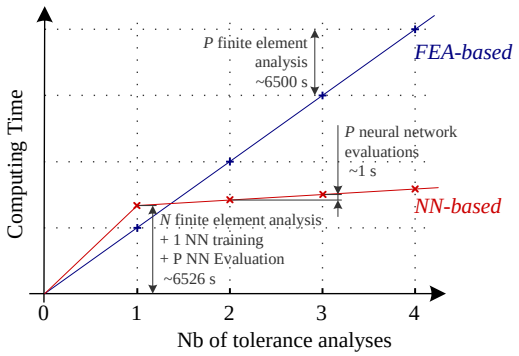

Fig. 9. Estimation of the computing time required *vs.* number of tolerance analyses.

5.3. Potential gain in computing time

The evaluation of the FEA-based variation propagation relation in the presented case takes approximately 0.65 s for a single assembly simulation. Then, the simulation of a population of 10000 assembled products takes approximately 6500 s and the total computing time required for multiple tolerance analyses is linear with respect to the number of analyses (see Fig. 9).

In our case, the same computing time is also required to generate a training data set because of its 10000 elements. The training of the neural network takes approximately 25 s, what leads to a total 6525 s to obtain an operational neural network. After that, the evaluation of the approximated NN-based variation propagation relation takes less than 1.10^{-4} s, leading to 1 s to approximate a population of 10000 assembled products.

Those figures are summarised in Fig. 9. They are average values observed along the experiments. They can only serve the relative comparison between FEA-based simulation and neural network approximation.

The Fig. 9 shows the potential gain in computing time when multiple analyses are performed, like in an optimum tolerance allocation research. The metamodel is valid as long as the evaluated region is included in the learning region. The neural network approximation can also be a relevant solution for off-line preparation before performing a quick evaluation of different scenario.

6. Conclusion

The computing time may become a hurdle in the simulation of geometrical variation propagation in assembly for tolerance analysis, particularly when the influence of part compliance, assembly sequence and contact interaction are taken into account (what generally leads to non-linear geometrical variation propagation relations).

This paper presents an original method to approximate the non-linear geometrical variation propagation relation with neural networks. This approximation aims to reduce drastically the computing time required for multiple tolerance analysis.

The impact of several design parameters of the neural network on the approximation quality is investigated through a case study. Several appropriate sets of design parameters are identified. Further work is still required to find the optimum size of the neural network hidden layer.

The quality of the approximation of those relevant networks is also evaluated on the case study. The average generalisation error – at the individual level – and the error in the prediction at the population level – like the 95% coverage interval evaluated in the paper – are satisfactory in the tested case. The errors are generally below ten per cent for the 95% coverage interval. Yet, the method proposed lacks a way to assess the confidence in the approximation.

Finally, the potential gain in computing time is pointed out. The approximation of the geometrical variation propagation in assembly with neural networks proved to be particularly efficient when multiple tolerance analyses need to be performed, leading to a very low computing time required while the network is trained and as long as the evaluated population is included in the learning region.

Acknowledgements

This research work has been carried out in partnership between EADS Innovation Works and the LURPA – ENS de Cachan in the frame of the GRC-Flexible Assembly of the INNO'CAMPUS collaboration.

References

[1] S. Liu, S. J. Hu, A parametric study of joint performance in sheet metal assembly, *International Journal of Machine Tools and Manufacture* 1997;**37** (6):873–884.

[2] W. W. Cai, C.-C. Hsieh, Y. Long, S. P. Marin, K. P. Oh, Digital panel assembly methodologies and applications for compliant sheet components, *Journal of Manufacturing Science and Engineering* 2006;**128** (1):270–279.

[3] S. Dahlstrom, L. Lindkvist, Variation simulation of sheet metal assemblies using the method of influence coefficients with contact modeling, *Journal of Manufacturing Science and Engineering* 2007;**129** (3):615–622.

[4] K. Xie, L. Wells, J. A. Camelio, B. D. Youn, Variation propagation analysis on compliant assemblies considering contact interaction, Journal of Manufacturing Science and Engineering 2007; **129** (5):934–942.

[5] G. Ungemach, F. Mantwill, Efficient consideration of contact in compliant assembly variation analysis, *Journal of Manufacturing Science and Engineering* 2009; **131** (1):011005.

[6] J. P. Kleijnen, Kriging metamodeling in simulation: A review, *European Journal of Operational Research* 2009;**192** (3):707–716.

[7] R. R. Barton, M. Meckesheimer, Chapter 18 metamodel-based simulation optimization, in: S. G. Henderson, B. L. Nelson (Eds.),

Simulation, *Handbooks in Operations Research and Management Science* 2006;**13**:535–574.

[8] S. Haykin, *Neural networks and learning machines*, Pearson International Edition, Prentice Hall, Upper Saddle River, 2010.

[9] E. Vazquez, Modélisation comportementale de systèmes non-linéaires multivariables par méthodes à noyaux et applications, Ph.D. thesis, Université Paris XI Orsay 2005.

[10] A. Barron, Approximation and estimation bounds for artificial neural networks, *Machine Learning* 1994;**14** (1):115–133.

[11] H. Falgarone, N. Chevassus, An innovative design method and tool for structural and functional analysis, in: *Proceedings of CIRP Design Seminar*, Cairo, Egypt, 2004.

[12] W. Huang, D. Ceglarek, Mode-based decomposition of part form error by discrete-cosine-transform with implementation to assembly and stamping system with compliant parts, *CIRP Annals - Manufacturing Technology* 2002;**51** (1):21–26.

[13] S. Samper, F. Formosa, Form defects tolerancing by natural modes analysis, *Journal of Computing and Information Science in Engineering* 2007;**7** (1):44–51.