Calibration of the 6 DOF High-Precision Flexure Parallel Robot “Sigma 6”

Fazenda, N.; Lubrano, E.; Rossopoulos, S.; Clavel R.

Abstract

In this paper, we present a new method for calibrating a 6 degree-of-freedom (DOF) high-precision flexure parallel robot.

The innovative contributions of this work are either in the technological challenge of calibrating a 6 DOF robot with sub-micrometer accuracy either in the way of processing the measurement data in order to correct the robot pose errors.

The first part of this work describes the procedure adopted for collecting a set of 6 D (3 translations + 3 rotations) measurement data from the robot. In this procedure, the robot was programmed, using closed-loops with external measurement devices, in order to execute either “pure translational” or “pure rotational” motions. All measurements were carried out on a thermally-stabilized environment.

The second part describes the method used to process the acquired data in order to correct the pose errors. We show in which optimal way neural networks (NN) have been used to perform such task. In particular, we show that the use of NN avoids to the robot user the complex task of formulating an analytical geometric model that takes into account many geometric or non-geometric sources of inaccuracy.

1 Introduction

1.1 Robot calibration

Calibration is the art of accuracy improvement. As long as robots and machine-tools were used in applications requiring accurate manipulations, the problem of calibration has been of major concern.

In fact, no matter how is the design of a given robot, its end-effector is always affected by pose errors. These errors can either have a *deterministic* (systematic) or a *non-deterministic* (random) nature. Systematic errors are for example errors in the robot geometry created by manufacturing tolerances, whereas random errors can be attributed to the measurement system.

In a proper calibration procedure, the systematic part of the error has to be eliminated or, in other words, after a perfect calibration the robot pose accuracy is reduced to its non-deterministic component.

In any calibration procedure, there are several steps:

1. formulation of a judicious calibration model or approach;
2. acquisition of reliable experimental data from the robot to be calibrated;
3. data processing (linking 1. and 2.) and implementation of the new model in the robot controller.

The methods found so far in the literature for the calibration of a given robot (serial or parallel) can be classified in the following 2 categories:

■ model-based approaches: in this case, a corrected geometric model (or a correction function) has to be computed, establishing an analytical relationship between motor (or articular) coordinates and end-effector (or operational) coordinates.

This model has a given number of parameters directly or indirectly related to the geometry of the robot. In the calibration procedure, these parameters are determined as a result of a numerical optimization of an error function properly defined.

■ modeless approaches: in this case, the relationship between motor and end-effector coordinates is a pure numerical correspondence.

This correspondence can be obtained either by using spline interpolation (or approximation) or by means of neural networks.

1.2 Presentation of the robot

The robot considered throughout this paper is a parallel mechanism called “Sigma 6” with 6 degrees-of-freedom (3 translations + 3 rotations).

Each of the 6 kinematic chains of the robot is made of a prismatic and a space transmission. The prismatic joint consists on a simple elastic stage made of 4

flexure hinges (Fig. 1a). The universal joint consists of 2 pairs of flexure hinges at 90º placed in a serial way (Fig. 1b).

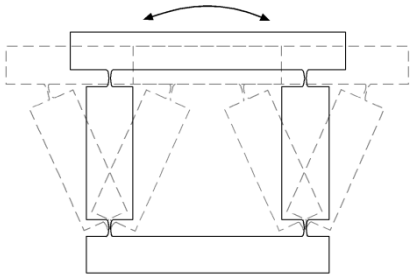

Figure 1a: *4-hinge prismatic joint*

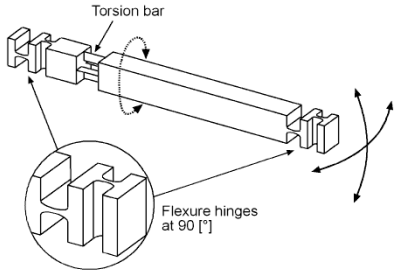

Figure 1b: *Universal joint*

Thanks to its parallel kinematics and the particular design of joints (flexure hinges), the robot's motions are free of dry friction or mechanical backlash; the repeatability is therefore of a few nanometers. Thanks also to nanometer resolution linear encoders controlling the displacements of the prismatic joints, the robot is able to perform highly-repeatable 6D motions with a resolution of 10 [nm] and 0.25 [µrad] over its working volume 4 [mm] x 4 [mm] x 4 [mm] x 3 [º] x 3 [º] x 3 [º].

In Figures 2a and 2b, we provide schematic representations of the Sigma 6 robot, showing a simple kinematic structure and a more detailed representation. The reader might notice a resemblance with the classical Stewart platform.

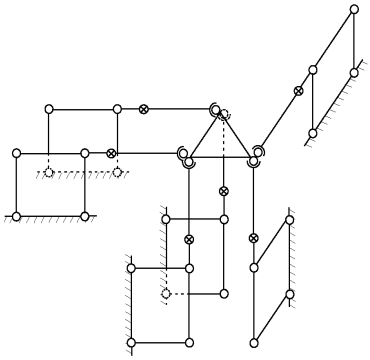

Figure 2a: *Kinematic structure of the Sigma 6 robot.*

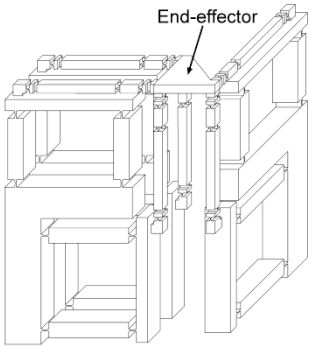

Figure 2b: *Detailed schematic view of the Sigma 6 robot*

Assuming that all flexure hinges act like perfect pivots and that the real lengths of all mechanical parts are equal to their corresponding nominal value, the inverse geometric model of the robot can be easily obtained by means of classical transformations between link frames. However, this model only provides a first approximation of the robot kinematics.

In order to make full use of the high-precision positioning capabilities of this robot, a calibration is needed. Our goal is of course to obtain a pose accuracy approaching the motor resolutions.

We believe that is extremely tedious and difficult to obtain an analytical kinematic model that, by taking into account all the main sources of inaccuracy, is able to predict (after parameter identification) the robot pose errors within the resolution of the motors.

In this paper, we will demonstrate, after having observed the smooth behaviour of the robot pose errors over its working volume, that an appropriate use of neural networks will enable us to approximate (within the desired accuracy) the correspondence between the articular and operational worlds and therefore to calibrate the robot without any physical knowledge on the possible causes of inaccuracy.

This approach has already been tested successfully on 2 different 3 DOF Delta high-precision flexure parallel robots. Tests on other parallel robots are planned.

2 Experimental issues

Acquiring reliable data for the calibration of a high-precision 6 DOF parallel robot is not an easy task.

In fact at the sub-micrometer range, only non-contact measurement techniques can provide robust data. Moreover, data provided by high-precision measurement devices (in particular interferometers) is strongly influenced by environmental perturbations, in particular, temperature variations.

Especial care has therefore been given to the measurement environment.

2.1 Measurement environment

When a temperature variation occurs in a measurement environment there are expansions not only in the robot mechanical parts but also in the measuring loop. Previous Finite-Element simulations have shown that reducing the variations in the level of 0.01-0.02 [°C] leads to a very small cumulative (robot + measurement system) error (in order of 10 - 20 [nm]).

Working in a thermally-controlled environment is therefore a crucial point for us.

The robot and all the measurement devices have been mounted inside a thermal insulated box, in which the temperatures are monitored and controlled. The temperatures are acquired using differential resistor temperature captors. The sensors are placed in order to monitor all the parts involved in the measurement loop (robot, measuring devices, robot frame, measuring devices frame, ambient air). Those readings are then converted in digital values and processed by a computer.

The “active part” of the thermal stabilization are 2 Peltier cells controlled by this computer and used to act on the air inside the thermal box in a closed-loop way. This control is performed using a classical P-I-D regulator.

In order to stabilize the robot to a certain consign temperature at least 6-8 hours are typically needed. The thermal stabilization has to be active also during the measurement phase, because the robot motors and the measuring devices will produce heat.

Figure 3 shows, in a schematic view, the overall thermal stabilization system.

In Figure 4, we present a graph of the evolution of the different temperatures during a typical thermal stabilization run achieved under “real” calibration conditions (robot moving continuously and measuring instruments inside the thermal box).

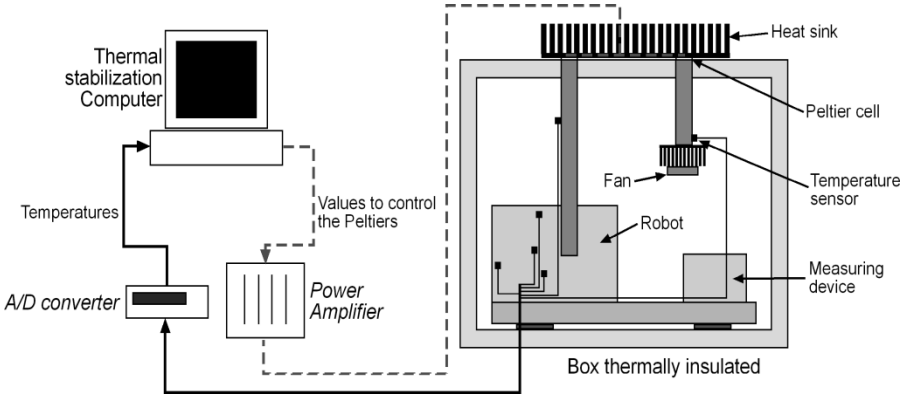

Figure 3: *Principle of the thermal stabilization (overview of the complete system).*

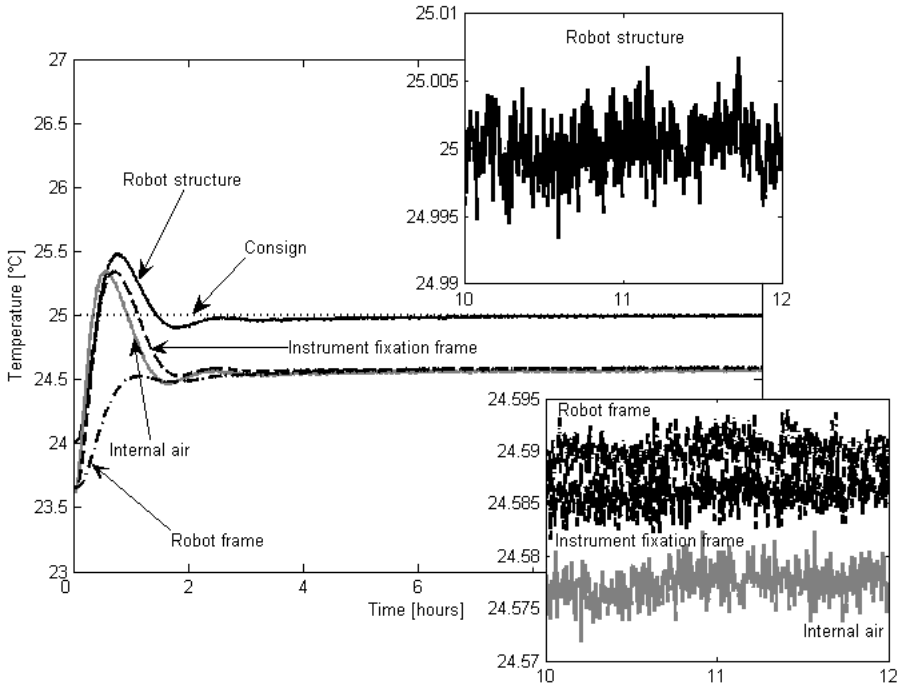

Figure 4: *Evolution of the different temperatures involved in the measuring loop. The only temperature regulated to 25 °C was the one of the robot structure. As a consequence of this regulation, other temperatures were set to stable and repeatable values (not necessarily 25 °C).*

2.2 Measurement devices

For measuring distances, we have used a SIOS SP 2000 **interferometer** (resolution: 1 [nm], accuracy: ~ 5 [nm], range: ± 2 [m]) and 3 Keyence LC 2420/2430 **laser displacement sensors** (resolution: 10/20 [nm], accuracy: ~ 50 [nm], range: ± 0.2/0.5 [mm]) mounted orthogonally in the 3 directions of the Euclidean space.

For measuring angles, we have used 2 NewPort LDS 1000 electronic **autocollimators** (resolution: 0.02 [arcsec], accuracy: 0.16 [arcsec]). Since these devices have a limited range (± 400 [arcsec]), a calibrated polygon prism (facets of ± 1.5 [º] and ± 3 [º]) has been used for measuring "large" angular motions.

The interferometer and the 2 autocollimators readings are taken on a mirror cube mounted on the effector whereas the 3 laser sensors detect the displacement of a sphere mounted also on the end-effector. When the sphere is being measured, the cube is removed from the end-effector and replaced by a support simulating its weight on the same centre of mass.

2.3 Measurement protocol

The major difficulty encountered in collecting the 6D high-precision data from the robot using state-of-the-art measurement devices lies in the fact that is not possible to measure at the same time (without loosing accuracy) translations combined rotations. When measuring the displacements of a reflecting mirror with an interferometer, it is in fact well known that when an angular variation with a certain magnitude (± 120 [arcsec] in our case) occurs, the beam is lost. Even if this tilt angle is within the tolerated range, an angular variation will still have an influence on the distance measured by the beam between 2 points.

For this reason, we have proposed a calibration procedure with different phases in order to able to acquire data having translations combined with rotations. In these different phases, only 3 degrees-of-freedom are calibrated per time.

■ The first part of the procedure was called "**translations without rotations"** (TWR) and consists in calibrating only the translational degrees-of-freedom of the robot, imposing null rotations on the robot end-effector. At the end of this phase, the robot is able to move its end-effector always with "pure translation" motions all over its working volume.

This part can be decomposed on 2 sub-phases: first, with the help of the 2 autocollimators measuring the orientation of the mirror cube, the angular variations are suppressed for a certain number of points in the robot working volume; second, with the help of the interferometer, the real X, Y and Z translations of these "pure translational" points are determined (Figure 5a)

The threshold under which rotations are considered to be suppressed has been set 0.20 [arcsec].

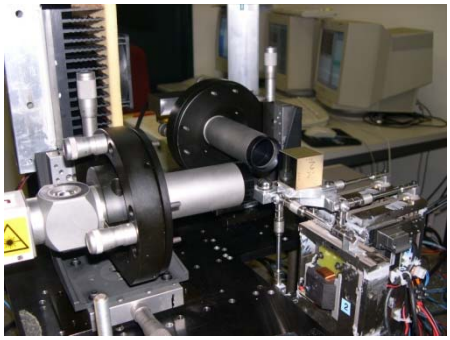

Figure 5a: *"Translations without rotations": performing the closed-loop with the 2 autocollimators*

Figure 5b: *"Rotations without translations" programmed with the help of 3 Keyence laser displacement sensors detecting the 3D position of the centre of the sphere.*

■ The second part consists in performing **rotations without translations** (RWT) with the robot. At this point, a closed-loop is done with the help of the 3 Keyence laser displacement sensors. These sensors are used in our case as zero detectors to read and correct the 3D position of a sphere mounted on the end-effector (Fig. 5b).

The threshold under which translations are considered to be suppressed has been set to 40 [nm] (twice the resolution of the worst sensor).

In order to merge these 2 phases, the absolute frame in respect to which to robot will be calibrated has been defined as having its origin at the centre of the sphere (when the robot is in the reference position). The directions are defined by those of the mirror cube.

In order to merge the previous 2 measurement configurations, the following calibration protocol has been adopted:

PHASE 1 – Calibration of the robot orientation: The 2 autocollimators are mounted and a correspondence is established between the 6 coordinates of the Inverse Geometric Model and the 3 angles measured. At the end of this phase, the user will be able to move the robot all over its working volume with correct orientation. "Small" angles (up to ± 400 [arcsec]) are determined by direct reading of the autocollimators on the surfaces of the mirror cube, whereas "big" angles (up to ± 3 [º]) are measured with the help of calibrated polygon prisms.

PHASE 2 – Calibration of the pure translations of the robot (TWR).

PHASE 3 – The laser displacement sensors are mounted in a certain number of positions. For each position of the laser sensors, the sphere mounted on the robot end-effector is therefore moved to the zeroes of the 3 laser sensors:

● the first point (from the reference position) is a "pure-translation" point, calculated according to the results of phase 2;

● the following points, determined by the RWT closed-loop, are points having non-trivial angular contributions and are determined according to the results of phase 1.

Notice that all the points obtained (except the first one) in phase 3 are combinations of translations and rotations (that could not be measured in a direct way): for each location of the laser sensors, the translation is the one of the "pure-translation" point whereas the rotations are already calibrated from phase 1.

3 Pose correction using Neural Networks

3.1 Basic considerations

According to Haykin [1], a Neural Network (NN) can be defined as "*a machine that is designed to model the way in which the brain performs a particular task or function of interest*".

A neuron is the basic processing unit used to build neural networks.

From the mathematical point of view a neuron is described by the following equations (Fig. 6)

$$u_k = \sum_{j=1}^{m} w_{k,j} x_j \text{ and } y_k = \varphi(u_k + b_k) \quad (1)$$

A neural network is a group of neurons connected in an appropriate way (Fig. 7).

Because of their ability to learn the correspondence between input-output data, neural networks have been widely used in the past 20 years in function approximation applications

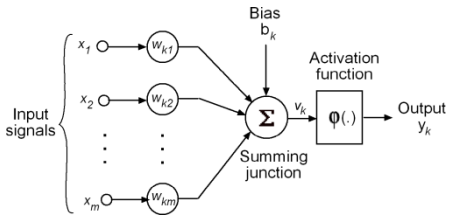

Figure 6: *Mathematical scheme of a neuron*

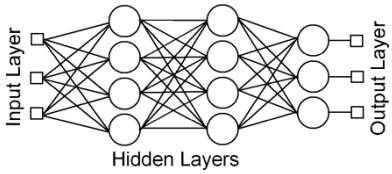

Figure 7: *Representation of a [4 4 3] feed-forward neural network*

In order to give to the network a good prediction capability, it has to go through a *training phase* in which a *learning algorithm* is used.

During this process the free parameters of the network (the weights $\boldsymbol{w_{k,l}}$ and the biases $\boldsymbol{b_k}$ of Eq. 1) are adapted through optimization schemes by comparing the network output to the target that has to be predicted.

In this paper, we only consider feed-forward networks and we will adopt the same convention of *MatLab*® for representing the structure of a given neural net:

[*number of neurons in the 1st hidden layer*; *number of neurons in the 2nd hidden layer*; … … ; *number of neurons of the output layer*].

According to this convention, the network of Figure 7 is therefore a [4 4 3] neural network.

3.2 Applying Neural Networks for robot calibration

3.2.1 State-Of-The-Art

Although the era of Neural Networks (NN) has started more than 60 years ago with the pioneering work of McCulloch and Pitts [2], their use in robot calibration has less than 20 years.

Josin [3] has been the first author to use neural networks in the context of robot calibration. The network considered received as input both the end-effector coordinates and the motor coordinates generated by the Inverse Geometric Model (IGM) and returned, as output, the corrections to be made in the motor coordinates.

A similar procedure has been used by Takanashi [4] for the calibration of a PUMA 560 robot.

In Kozakiewicz's paper [5] the NN used for the calibration of a 3 DOF SCARA robot was for the first time in parallel with the IGM.

In Renders work [6], the author showed through simulation the calibration of only 3 out of 6 DOF of a PUMA robot.

Watanabe [7] proposed a methodology to speed up the process of training of a neural network. The method consisted in splitting the network into 2 different sub-networks, with two different training phases.

The innovative contribution of Lewis [8] and Zhong [9] lies in the use of a so-called "Pi-sigma" network, a particular network that uses a linear summing unit in the hidden layer and a product unit in the output layer.

Xu [10] used 2 networks for calibrating a 2 DOF robotic joint. The second network in this case was trained by learning the residual errors of the first one. The author showed that 90% of the total error was corrected by the first network whereas the second network corrected only 5%.

Zhong [11, 12] continued the work [8, 9], adding the comparison with another kind of NN, the recurrent continuous-valued Hopfield NN, in an unsupervised learning way.

Wang [13] calibrated a 2 DOF planar robot using a Kohonen self-organized neural network.

Dreiseitl [14] presented a method to calibrate a robot composed by only revolute joints. The method consisted in mapping the kinematic equations of the robot in a NN representation.

Jung [15] proposed to compensate uncertainties in the robot kinematic Jacobian by NN in an 'on-line' way, at the trajectory planner level.

Tiboni [16, 17] has been the first author to consider different connections between the NN and the IGM. The main conclusion was that a pure neural approach (without IGM) could not provide satisfactory results (in comparison to a model-based calibration with parameter identification) and that the best connection to use was the parallel one.

3.2.2 Optimal Use of Neural Networks for robot calibration

3.2.2.1 Connections with the Inverse Geometric Model

The correction of the robot pose errors can be achieved in several ways considering (or not) the information already provided by the Inverse Geometric Model:

- **Serial 1:** NN in serial with the IGM, before the IGM: the NN learns the correspondence between the measured end-effector coordinates and the end-effector coordinates given by the IGM – Fig. 8;
- **Serial 2:** NN in serial with the IGM, before the IGM (error): in this case, the NN has the same input as in previously but the output to fit is the difference between the measured end-effector coordinates and the end-effector coordinates given by the IGM (the error) – Fig. 9;
- **Serial 3:** NN in serial with the IGM, after the IGM: the NN receives as output the motor coordinates and as input the motor coordinates generated when the measured end-effector coordinates pass through the IGM – Fig. 10;
- **Serial 4:** NN in serial with the IGM, after the IGM (error): this case is identical to the case "Serial 2", except that it occurs in motor coordinates – Fig. 11;
- **Parallel:** NN in parallel with the IGM: the input will be the measured end-effector coordinates and the output will be the difference between the motor coordinates corresponding to the measured end-effector coordinates and the motor coordinates generated when the measured end-effector coordinates pass through the IGM – Fig. 12;
- **Without IGM:** NN without IGM: the NN learns direct the correspondence between the measured end-effector coordinates and the corresponding motor coordinates – Fig. 13.

Figure 8: *Case "Serial 1"*

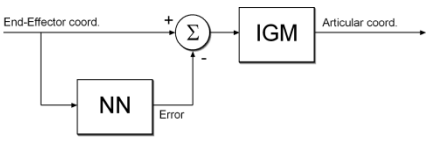

Figure 9: *Case "Serial 2"*

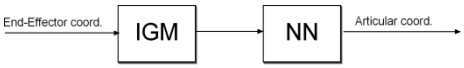

Figure 10: *Case "Serial 3"*

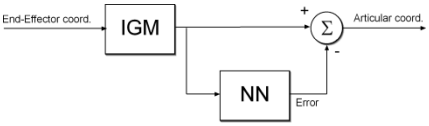

Figure 11: *Case "Serial 4"*

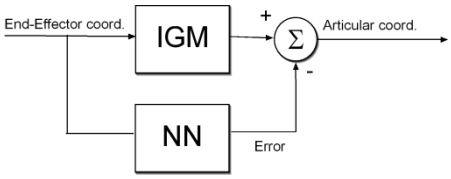

Figure 12: *Case "Parallel"*

Figure 13: *Case "Without IGM"*

3.2.2.2 Finding the optimal Network structure using a decision-tree

The main difficulty in using Neural Networks in robot calibration lies in the fact that there is no standard algorithm nor procedure to find the optimal neural network to do this task.

In general, the most common procedure used by previous researchers consists in varying the architecture of the network and do several trainings until the results are good. The kinds of adjustment that can be done are the following:

- Adding/subtracting neurons to the network.
- Adding/subtracting hidden layers to the network.
- Changing training algorithm and/or the parameters of it.

As a limit to this procedure there is the computational cost of the network, and as a consequence of that, the training time. This because complex networks need too much time to be trained. On the other side, networks that don't have enough neurons will not learn very well functions that are highly non-linear (which is often the case for parallel robots).

In order to find the optimal structure (number of hidden layers, number of neurons on each hidden layer) of the Network to be used, a search algorithm based on a *decision-tree* has been developed and act as follows:

The research always starts from a very simple neural network. This network, called *Father Network,* is a network having only 1 hidden layer with the same number of neurons than the output layer. The number of layers in the output layer corresponds to the number of degrees of freedom of the robot or the number of joint coordinates.

Starting from this network, the algorithm does the following steps:

1. Generating the Children Networks from the Father Network: there are 2 different rules for this creation:

• *rule 1*: add a neuron in each hidden layer (one layer per time);

• *rule 2*: duplicate a given layer and put this new layer before or after the layer from which it has been copied.

2. Training of all the Children Networks: all the children generated in the previous iteration are trained for a certain number of epochs.

3. Choice of the new Father Network: all the children that have been generated and that have not yet become fathers will be compared. The one that gives best results will become the father of the next iteration.

Therefore, the tree is expanded until a certain level, for which:

Maximum number of neurons in any layer = *Number of iterations* + *Number of neurons in the output layer* (2)

Maximum number of iterations + 2 = *Maximum number of layers* (3)

Example.

In the lines below we present an example of how the previous algorithm works.

The *score* of the network corresponds to the *absolute value of the prediction error* with data not seen during training.

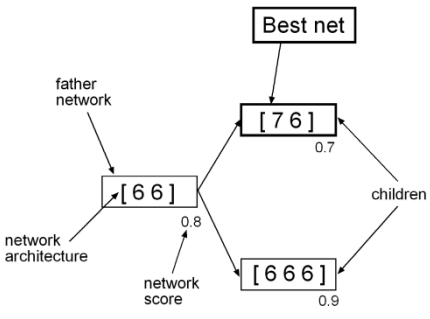

Figure 14: *Iteration 1*

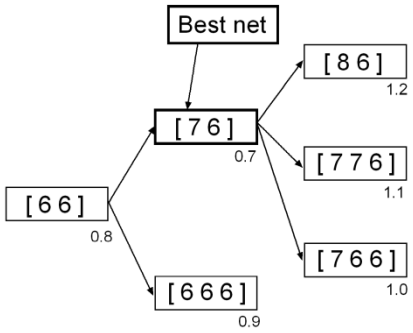

Figure 15: *Iteration 2*

■ Iteration 1: Starting from the father network [6 6], the 2 children networks [7 6] and [6 6 6] have been generated (Fig. 14).

These networks have been trained and the network with the best score has been chosen to become the father network ([7 6] in Fig. 14) for the next iteration.

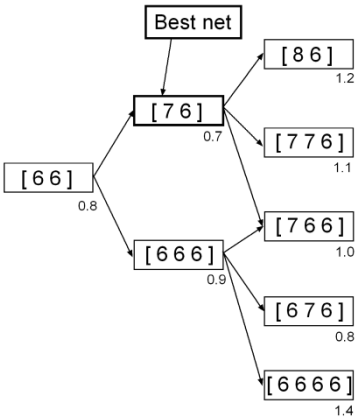

Figure 16: *Iteration 3*

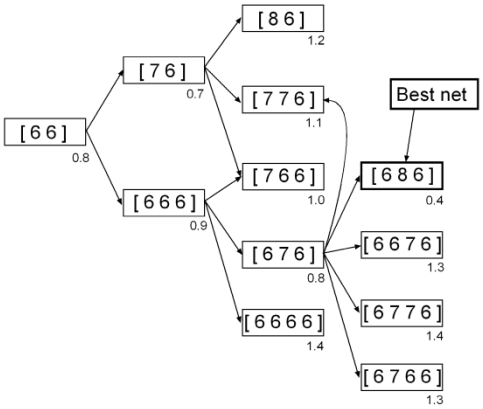

Figure 17: *Iteration 4*

■ Iteration 2: In this iteration, the network [7 6] generates 3 children: the networks [8 6], [7 7 6] and [7 6 6] (Fig. 15).

Like before, these 3 networks have been trained and the score of the children networks have been compared. However, the network having the best score is still the [7 6], but this network has already been the father previously.

Therefore, the next father network for the next iteration will be the one with the best score that has not yet been father previously: in our case, it is the network [6 6 6].

■ Iteration 3: In this iteration, the network [6 6 6] generates the children [7 6 6], [6 7 6] and [6 6 6 6]. Notice that the network has already been generated in the second iteration, so this time it will not be trained (Fig. 16).

The 5 children networks have been compared and the one having the best score has been chosen as the next father. The network [7 6] is still the best one, but it can not become father twice. Thus, the next father has been chosen to be [6 7 6] since it is the children network having the best score.

■ Iteration 4: In this last iteration, the network [6 7 6] has generated the children [7 6 6], [6 8 6], [6 6 7 7], [6 7 7 6] and [6 7 6 6] (Fig. 17).

All the children networks have been compared after training and seeing that it is the last iteration, the network [6 8 6] has been adopted as our final calibration network.

Remarks:

1. Notice that this algorithm is not of "best-path" type: if the algorithm stops now (because the maximum number of iterations has been reached), the network chosen for the calibration will still be the [7 6], even if a lot of networks have been checked afterwards;

2. We can see that in our case, according to the prediction of equations 2 and 3, the maximum number of neurons in any given layer is 10 and the maximum number of layers of any generated networks is 5.

3.2.2.3 Overfitting and training algorithm

The networks have been trained using a back-propagation scheme on which a Levenberg-Marquardt algorithm is used to update the network's weights and biases.

For the training phase, the data has been divided in 3 sets (each set being representative of the whole data diversity): an "input" set composed by 80 % of the data, a "validation" set composed by the 16 % of the data and the last 4 % called the "test" set.

The error in the validation set will normally decrease during the initial part of the training; however, when the network begins to ***overfit*** the data, the error on the validation set starts to increase despite the fact the error on the input set continues to decrease. This well-known phenomenon decreases the network's generalization capability.

The input set is used to actually train the network while the validation set is only used to monitor continuously the error during the training and therefore to evaluate the generalization capability of the network.

The test set is also used to monitor the error during the training and for displaying the results after training.

Two different ways have been used to prevent overfitting issues: first, the training algorithm used ("trainbr") implements a so-called "**regularization**" on which a trade-off factor is used to make some adjustments in the error function during the training;

second, an **early-stopping** method was also used: the training is stopped whenever error in the validation set starts to increase.

3.2.2.4 Activation functions

After a large number of tests, we found that networks having sigmoidal functions in the hidden units and linear functions in the output neurons have the best prediction capability for calibration.

3.3 Calibration results

3.3.1 Predicting angular motions

Using the decision-tree search algorithm, we have found that the network [6 8 7 3] was the best candidate for the prediction of angles in the phase 1 of our calibration procedure. When our calibration procedure was limited to the measurement of "small angles" directly on the mirror cube prediction errors were within ± 0.33 [arcsec] (within a confidence level of 90 %). When "big angles" were introduced in the data to be trained by the network, the same prediction error increased to ± 2.70 [arcsec], since the measurement area in the facets of the prism was reduced by a factor 10 in respect to the on of the mirror cube.

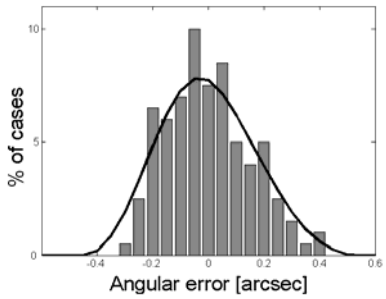

Figure 18: *Histogram of the prediction errors of the phase 1 – case of "small angles"*

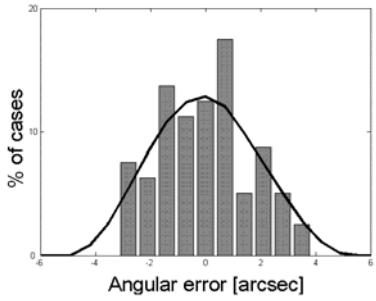

Figure 19: *Histogram of the prediction errors of the phase 1 – case of "big angles"*

3.3.2 Predicting translational motions

In the phase 2 of our calibration procedure, we have also applied the decision-tree search algorithm to the different 5 connection cases with the IGM (Figures 8 to 12) as well as to the case without IGM (Figure 13). We have been able to find "equivalent" networks in terms of prediction capability (around 50 [nm], with a confidence level of 90 %). Table 1 lists the structures of these networks.

The last point of the calibration phase consists in verifying, with the help of the 3 laser sensors measuring the sphere, if the robot is able to turn around any centre of rotation. Preliminary results have shown errors up to 300 [nm] – this result is currently being improved.

Table 1: *Structure of optimal "equivalent" networks found fort he prediction of "pure translational" motions of the Sigma 6 robot.*

Connection	*Best Net Architecture*
Serial 1	*[7 8 10 10 9 8 6]*
Serial 2	*[7 9 9 12 9 6]*
Serial 3	*[12 17 6]*
Serial 4	*[9 9 11 9 8 9 6]*
Parallel	*[7 7 6 7 10 7 6]*
No-IGM	*[6 9 10 10 10 6]*

4 Conclusion and future work

The contributions of this paper in respect to published literature are the following:

■ from the experimental point of view, we have proposed a procedure to calibrate a 6 DOF high-precision parallel within sub-micrometer accuracy. In the near future, this calibration will be confirmed using the robot as a nano-indenter;

■ from the theoretical point of view, we have shown in which "optimal" way neural networks can be used to perform calibration of parallel robots. The major advantage of neural networks lies in the fact that they act in a "black-box" approach" correcting

the robot pose errors even without any physical knowledge on the sources of inaccuracy.

In the future, we are planning to use NN for correcting also the influence of external perturbations in data acquisition so that sub-micrometer accuracy can be reached in an industrial environment (without thermal stabilization).

Literature

[1] S. Haykin, Neural Networks - A comprehensive foundation, 1999, Prentice-Hall.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp 115-133.

[3] G. Josin, D. Charney and D. White, "Robot Control using Neural Networks", Proceedings of the IEEE International Conference on Neural Networks, 1988, Vol. 2, pp 625-631.

[4] N. Takanashi, "6 D.O.F. manipulators absolute positioning accuracy improvement using a neural-network", Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS), 1990, pp 635-640.

[5] C. Kozakiewicz, T. Ogiso and N. Miyake, "Calibration analysis of a Direct Drive Robot", Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, 1990, pp 213-220.

[6] J.-M. Renders, J. Millán del R. and M. Becquet, "Non-geometrical parameters identification for robot kinematic calibration by use of neural network techniques", Proceedings of the European Robotics and Intelligent Systems Conference, Corfu – Greece, 1991.

[7] T. Watanabe, T. Hidekatsu, K. Kurokawa, A. Kawano, S. Kubo and K. Kenichi, "Calibration of position and orientation of robot manipulators using a neural network", Proceedings of the 1992 Japan-USA Symposium on Flexible Automation – Part 1, San Francisco, California, USA, pp 219-225, 1992.

[8] J. M. Lewis, X. L. Zhong and H. Rea, "A neural network approach to the robot inverse calibration problem", Proceedings of the IEE International Conference on Intelligent Systems Engineering, September 1994, pp 342-347.

[9] X. L. Zhong and J. M. Lewis, "Kinematic identification and compensation of robot manipulators using neural optimization networks", Proceedings of the 3rd International Conference on Automation, Robotics and Computer Vision, Nanyang Technology University Singapore, 1994, Vol. 3, 1472-1476.

[10] W. L. Xu, K. H. Wurst, T. Watanabe and S. Q. Yang, "Calibrating a Modular Robotic Joint Using Neural Network Approach", Proceedings of the IEEE International Conference on Neural Networks, Vol. 5, July 1994, 2720-2725.

[11] X. L. Zhong, J. M. Lewis and H. Rea, "Neuro-accuracy Compensator for Industrial Robots", Proceedings of the IEEE International Conference on Neural Networks, Vol. 5, July 1994, pp 2797-2802.

[12] X. Zhong, J. Lewis and F. L. N-Nagy, "Inverse Robot Calibration Using Artificial Neural Networks", Engineering Applications of Artificial Intelligence, Vol. 9, No. 1, pp 83-93, 1996

[13] D. Wang and A. Zilouchian, "Solutions of Kinematics of Robot Manipulators Using a Kohonen Self-Organizing Neural Network", Proceedings of the 12th IEEE International Symposium on Intelligent Control, July 1997, Istanbul – Turkey, pp 251-255.

[14] S. Dreiseitl, W. Jacak. T. Kubik and R. Muszynski, "Neural processing based robot kinematics modelling and calibration for pose control", Systems Science, Vol. 23, No. 3, 1997, pp 61-68.

[15] S. Jung and B. Ravani, "On-Line Kinematic Jacobian Uncertainty Compensation for Robot Manipulators Using Neural Network", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 1998, pp 3483-3488.

[16] M. Tiboni, A. Omodei, G. Legnani, D. Tosi and P. L. Magnani, "Calibration of a 5 DOF Measuring Robot by Neural Network", XXX Convegno Nazionale AIAS – Alghero (SS), September 2001, pp 1467-1476.

[17] M. Tiboni, G. Legnani, P. L. Magnani and D. Tosi, "A closed-loop neuro-parametric methodology for the calibration of a 5 DOF measuring robot", Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, July 2003, Kobe – Japan, pp 1482-1487.

[18] H. Demuth and M. Beale, "Neural Network Toolbox – User's guide", Version 4, 2004, The MathWorks.