Improved Kinematics Calibration of Industrial Robots by Neural Networks

Giovanni Legnani, Monica Tiboni

Department of Mechanical and Industrial Engineering, Brixia University, Italy
E-mail: {giovanni.legnani, monica.tiboni}@ing.unibs.it
URL: http://robotics.ing.unibs.it

Received October 15, 2009

Abstract. The paper presents a preliminary study on the feasibility of a Neural Networks based methodology for the calibration of Industrial Manipulators to improve their accuracy. A Neural Network is used to predict the pose inaccuracy due to general sources of error in the robot (e.g. geometrical inaccuracy, load deflection, stiffness and backlash of the mechanical members, etc. . .). The network is trained comparing the ideal model of the robot with measures of the actual poses reached by the robot. A back-propagation learning algorithm is applied. The Neural Network output can be used by the robot controller to compensate for the errors in the pose. The proposed calibration technique appears extremely simple. It does not need any information on the pose errors nature, but only the ideal robot kinematics and a set of experimental pose measures. Different schemes of calibration procedures are applied to a simulated SCARA robot and to a Stewart Platform and compared, in order to select the most suitable. Results of the simulations are presented and discussed.

Keywords. robot calibration, Neural Network, SCARA robot, Stewart Platform, compensation

1 Introduction

1.1 Kinematics calibration

A robot is a mechanical system in which constructive tolerances (geometrical inaccuracy), load deformations, stiffness and backlash of the mechanical members, etc. . . , cooperate to create inaccuracy in the gripper pose (position and orientation).

Industrial robots are generally quite repetitive while their accuracy is generally worse. Calibration is a methodology to improve the robot accuracy without mechanical means working only on its controller. Computer simulations and experimental verifications show that very often a proper calibration can improve the robot accuracy up to a value close to the robot repeatability [Mooring et al.(1991), Trevelyan et al.(1996)]. Calibration is possible whenever a procedure to predict the robot error can be established. Many research activities have been carried on this subject, nevertheless the search for a simple and effective calibration procedure for on-field applications is still open.

A procedure to improve the robot accuracy (which for not calibrated industrial robots is sometimes up to some millimeters) consists of two main parts:

1. the measurement of the gripper position and orientation error for a predefined set of gripper poses in the workspace;

2. the development of a mathematical technique to predict and to compensate for the measured errors.

These two problems can be considered quite independent. The authors attention in this work is focused on the second aspect: supposing given a data set of gripper pose measures, a new method to predict the gripper pose inaccuracy is proposed. This will make possible a compensation [Faglia et al. (1993), Legnani et al.(1996), Trevelyan et al.(1996)].

Classical methods are based on the well-known parametric approach and consist in two phases:

- the definition of a model of the robot considering some of the possible causes of inaccuracy (defining a priori the relating complexity);

- the identification of the unknown value of the parameters of the model.

Usually the models consider only geometrical inaccuracy. The complexity of the model and the high number of parameters involved often prevents considering other phenomena, such as the deflection of the elements, the backlash in the joints, etc.... This is the principal limit of the parametric approach.

1.2 The proposed methodology

To overlap the limits of parametric calibration we try to find a simple method, able to predict the pose errors in a way completely independent of the nature of their causes and without requiring any complex model [Tiboni et al. (2003), Fazenda et al. (2006)].

A Neural Network (NN) appeares a good instrument to achieve this goal. A robot can be considered a system that performs a transformation of the input (the joint angles) in a corresponding set of the gripper coordinates in the robot workspace (the output). The input/output transfer function in a real robot may be quite different from the theoretical one. Moreover, theory assures that a proper feed-forward Neural Network, with a back-propagation training technique is able to approximate any kind of mathematical transformation [Haykin (1999)]. The basic idea is to use a Neural Network to learn the input/output direct and inverse transfer function of the *real robot*.

This preliminary study has the aim to verify the effectiveness of this methodology comparing a number of different computational schemes involving a Neural Network and an ideal model of the robot kinematics. The methodology is applied to different serial and parallel manipulators. The network is trained and tested using simulated pose measures generated by an external program implementing the kinematics of a robot with geometrical errors, plus joint backlash and compliance. The results obtained by the different schemes are compared taking into account:

a. the accuracy in the prediction of the real pose;

b. the difficulty in the NN parameter tuning;

c. the capacity of generalization;

d. the immunity to the noise included in the measures.

Many different calibration schemes have been tested. Two of them, which will be more deeply analyzed in the paper, produced very good results. They are able to improve the accuracy up to a value comparable with the measuring error and the robot repeatability.

To have the possibility to examine different schemes with different variants, a suitable simulation procedure was adopted.

Different calibration scheme are initially compared on a simplified SCARA robot. As a second step, the more promising schemes are applied to a parallel manipulator and to a full model SCARA manipulator for their final validation.

2 The SCARA robot used for the preliminary tests

Being the aim of the first step of the research just a feasibility analysis, we consider a very simple SCARA robot. In spite of its simplicity, this robot is characterized by the same problems of more complex robots, as singularities, multiple solutions, etc.

The robot considered is formed by two links which move the gripper in the x-y plane by means of two rotoidal joints; the end-effector vertical translation is not considered. Let us denote with α and β the two joint variables, with l_1 and l_2 the link lengths, with x and y the end-effector cartesian coordinates (Fig. 1).

The nominal direct kinematic problem for this robot is solved by equations 1, the inverse one by eq. 2.

$$\begin{cases} x = l_1 \cos\alpha + l_2 \cos(\alpha+\beta) \\ y = l_1 \sin\alpha + l_2 \sin(\alpha+\beta) \end{cases} \tag{1}$$

$$\begin{cases} \alpha = \pm \arccos\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \\ \beta = atan2(y,x) - atan2(l_2 \sin\beta, l_1 + l_2\cos\beta) \end{cases} \tag{2}$$

By introducing geometrical inaccuracy the direct kinematic is solved by eq. 3, where the "prefix" δ denotes small unknown errors, while Δx,Δy represent the gripper error that can be measured for given values of α and β. The errors can be constant, random or depending on the external loads. The nature of the errors is better described in sect. 3.2.

$$\begin{cases} x_r = x + \Delta x = & (l_1+\delta l_1)\cos(\alpha+\delta\alpha) + \\ & +(l_2+\delta l_2)\cos(\alpha+\delta\alpha+\beta+\delta\beta) \\ y_r = y + \Delta y = & (l_1+\delta l_1)\sin(\alpha+\delta\alpha) + \\ & +(l_2+\delta l_2)\sin(\alpha+\delta\alpha+\beta+\delta\beta) \end{cases} \tag{3}$$

3 Calibration methodology

3.1 The Neural Network

In a problem of system identification (nonlinear input-output mapping) a right choice for a NN is a multilayer feed-forward network, trained in a supervised manner with the back-propagation learning technique [Haykin (1999)].

The chosen network contains just one layer of hidden neurons (Fig. 2) with sigmoidal activation functions. Linear activation functions are used for the output neurones. In accordance with the *"universal approximation theorem"*, this NN can approximate an arbitrary continuous function [Haykin (1999)]. Moreover, this choice is the optimum in the sense of easy implementation, learning time and generalization.

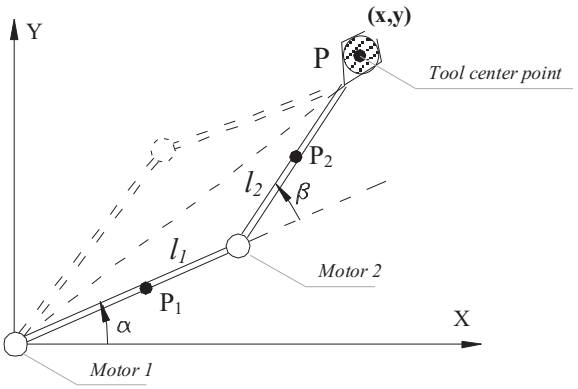


Figure 1: The SCARA robot joint variables.

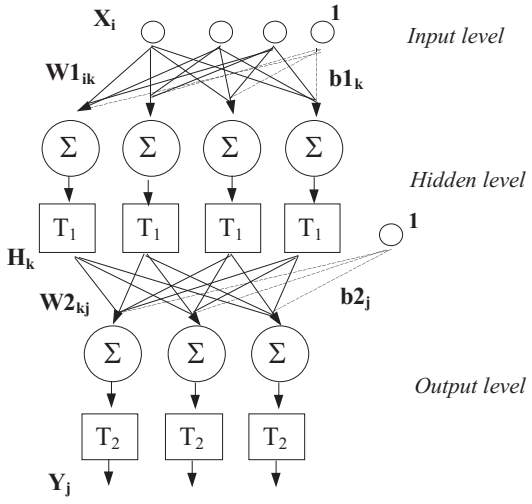


Figure 2: The feed-forward network used.

3.2 The tested calibration schemes

The authors attention has been focused first on the direct kinematic calibration and then on the inverse one.

For the direct kinematics, the idea is to create a neuro-kinematic (NK) model of the real robot (Fig. 3), merging a model of the ideal robot with a NN describing the manipulator errors. Different schemes have been analyzed (Fig. 5) to select the more effective one.

In Fig. 3 we denoted as $Q_{th} = [\alpha;\beta]^t$ the values of the joint variables measured by the joint transducers, while $S_{th} = [x;y]^t$ and $S_{re} = [x_r;y_r]^t$ are the theoretical and real gripper pose.

The NN is trained in order to reduce the quantity $E = \|S_{re} - S_{NK}\|$, which represents the error in the gripper pose prediction, where $\|\cdot\|$ is the Euclidean norm.

The schemes of NK model in Fig. 5 enable the prediction of the actual gripper pose S_{re}, knowing the joint rotations Q_{th} (direct kinematics calibration).

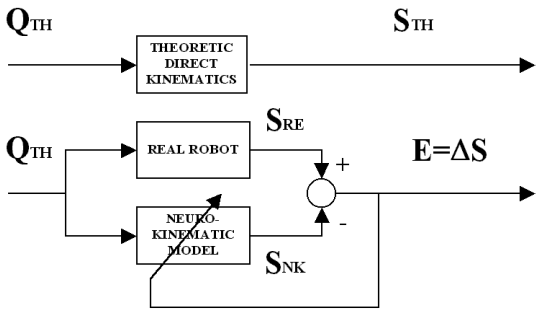


Figure 3: The direct kinematics calibration model.

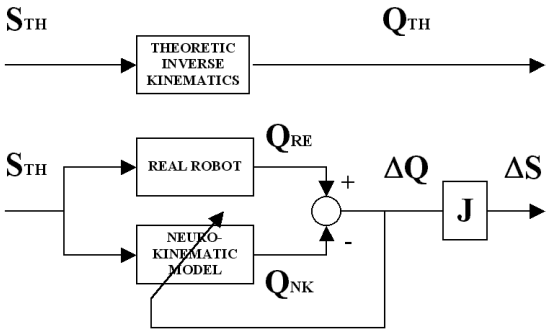


Figure 4: The inverse kinematics calibration model.

For the inverse kinematics calibration is required to predict the joint rotation Q_{re} that bring the gripper of an actual robot in a desired pose Sth. A NK inverse model has to be trained (Fig. 4); different possible schemes are shown in Fig. 6.

In order to make results comparable with those of the direct kinematics, instead of the joint error $E = ||Q_{re} - Q_{NK}||$, as performance index was used the equivalent pose error $E = ||\Delta S|| = ||J\Delta Q||$, where J is the jacobian matrix and $\Delta Q = Q_{re} - Q_{th}$

4 Comparison of the different Neuro-Kinematics schemes

4.1 Angular coordinates

The first simulations highlighted a problem related to large variations of the angles.
The robot behaviour depends on the sine and the cosine of rotation angles rather than on the angles themselves. And so a rotation of α or $\alpha \pm 2k\pi$ produces the same effect. However, since the NN activation functions are continuous and not periodical, if an angle is used as input to a NN, the output value would be different when the input is α or $\alpha \pm 2k\pi$. This fact makes the NN behaviour unreliable for large rotation of the joint angles and for the representation of the gripper attitude. After several tests, it was decided to replace each angular input of the NN with two inputs representing the sine and the cosine of the angles, that means that the joint vector $Q = [\alpha\ \beta]$ (trigonometric form) was defined as $Q = [sin\alpha\ cos\alpha\ sin\beta\ cos\beta]$ (angular form).

A second aspect of this problem was the identification of the best way to add joint rotations in the schemes like 2,3,4 and 6 of Fig. 5 and Fig. 6. This operation can be done in three different ways as described in Tab. 1. In the preliminary tests, all the considered NN performed better using the second type of angular addition.

Generalizing previous observations, every time a coordinates set X ($X = Q$ or $X = S$) contains some angular values, in order to avoid discontinuity and singularity, a transformation $T(\cdot)$ between the angular form X_a and trigonometric one X_t have to be defined:

$$X_t = T(X_a) \tag{4}$$

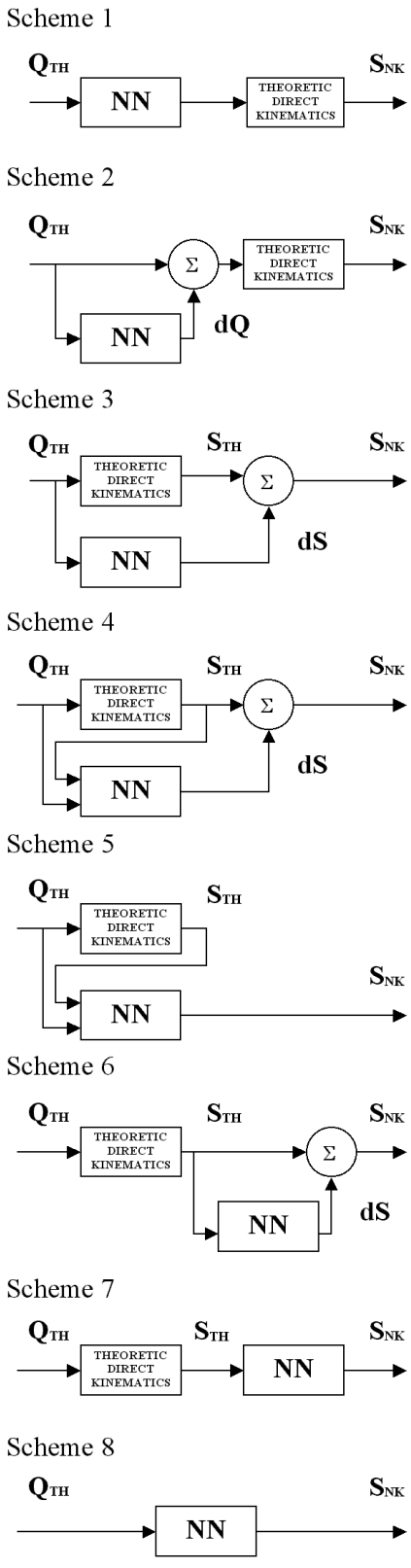


Figure 5: The direct Neuro-Kinematic models tested

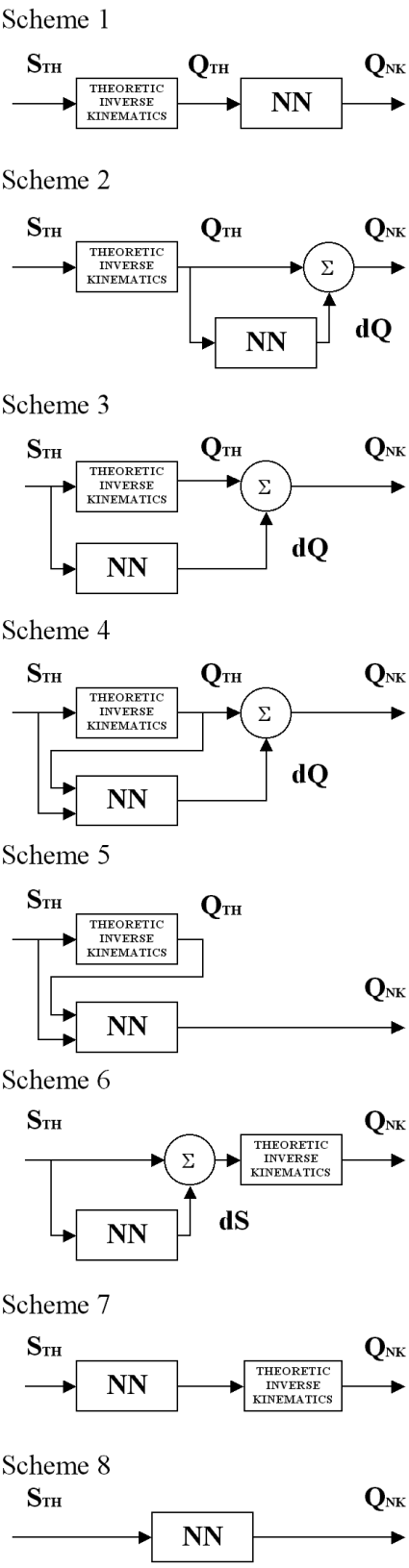


Figure 6: The inverse Neuro-Kinematic models tested

θ	$d\theta$	θ_c
θ	$d\theta$	$\theta + d\theta$
θ	$sind\theta$ $cosd\theta$	$\theta + atan2(sind\theta; cosd\theta)$
$sin\theta$ $cos\theta$	$\Delta sin\theta$ $\Delta cos\theta$	$atan2(sin\theta+\Delta sin\theta; cos\theta+\Delta cos\theta)$

Table 1: Different types of angular sum

X	dX	X_c
X_a	dX_a	$X + dX$
X_a	$T(dX_a)$	$X + T^{-1}(dX)$
$T(X_a)$	$dT(X_a)$	$T^{-1}(X + dX)$

Table 2: Different types of angular sum

The choice of the inverse transformation form $T^{-1}(\cdot)$ have to take in to account that, if X_t is redundant, some fundamental trigonometric relations may be not satisfied ($cos\alpha^2 + sin\alpha^2 \simeq 1$). The transformation chosen for the joint coordinates Q of the SCARA robot is:

$$\begin{aligned} X_a &= [\alpha\ \beta] \\ X_t = T(X_a) &= [sinX_{a1}\ cosX_{a1}\ sinX_{a2}\ cosX_{a2}] \\ &= [sin\alpha\ cos\alpha\ sin\beta\ cos\beta] \\ T^{-1}(X_t) &= [atan2(X_{t1}, X_{t2})\ atan2(X_{t3}, X_{t4})] \\ &= [atan2(sin\alpha, cos\alpha)\ atan2(sin\beta, cos\beta)] \end{aligned}$$

Gripper coordinates S do not contain angular values, so no transformation is necessary .

Generalizing cases of Tab. 1, there are three types of angular sum shown in Tab. 2.

4.2 Preliminary tests

The simulation tests were performed as follows.

A first computing program based on Eq. 3 and simulating the real robot was used to generate a set of robot poses S_{re} starting from a given set of joint coordinates. Some of the poses and the corresponding joint coordinates were used to train the NN of the NK models, while the others were used as validation set.

Proper tests suggested the correct values for the number of hidden neurons and for the training parameters: learning rate η (0.01) and momentum α (0.9). These values proved to be suitable for all the considered NK schemes. Scheme 8 using a plain NN without robot kinematics was discharged due to its poor performances.

The first tests were performed on the direct kinematics schemes considering the sources of the pose errors described in Tab. 3, excluding random components. Compliance is modeled by torsional springs representing elasticity in the kinematics transmissions; k_1 and k_2 are their stiffness constants. Constant forces are applied to the links at points P_1 and P_2 (Fig. 1) and produce extra joint angular deflections whose amplitude depends on the robot configuration. The tests were repeated three times, simulating the real robot with the numerical values reported in Tab. 4. The considered errors are much more severe than those usually present in actual robots.

The two last lines of Tab. 4 contain the mean and maximum position error before calibration.

Forty-six poses uniformly distributed in the working area were used to train the network. The back-propagation learning phase of the NN was performed in batch-mode [Haykin (1999)] and

dl_1	error in the length of the first link
dl_2	error in the length of the second link
$d\alpha$	error in the first joint rotation
$d\beta$	error in the second joint rotation
k_1	torsional stiffness of the first joint
k_2	torsional stiffness of the second joint
F_1	force applied on point P_1 (first link)
F_2	force applied on point P_2 (second link)

Table 3: Components of error considered

	case 1	*case* 2	*case*3
l_1 [mm]	330	330	330
l_2 [mm]	330	330	330
δl_1 [mm]	5	5	0.5
δl_2 [mm]	-6	-6	-1
$\delta\alpha$ [deg]	2	2	0.15
$\delta\beta$ [deg]	-1.3	-1.3	0.05
k_1 [Nm/rad]	rigid	1000	80000
k_2 [Nm/rad]	rigid	1000	70000
$F_{1x}; F_{1y}$ [N]	[0;0]	[100;0]	[200;-100]
$F_{2x}; F_{2y}$ [N]	[0;0]	[100;0]	[-300;200]
E_{ave} [mm]	6.67	8.93	0.76
E_{max} [mm]	10.2	17.0	1.26

Table 4: Values of the components of error considered

lasted for 1000 epochs.

During and after this learning process, the NK models were used to predict the pose of 1244 robot configurations forming the validation set.

Results are presented in Tab. 5 and in Fig. 7. Last column of Tab. 5 contains the optimal number of hidden neurons of the NN for each scheme (among those tested (10,15,20)).

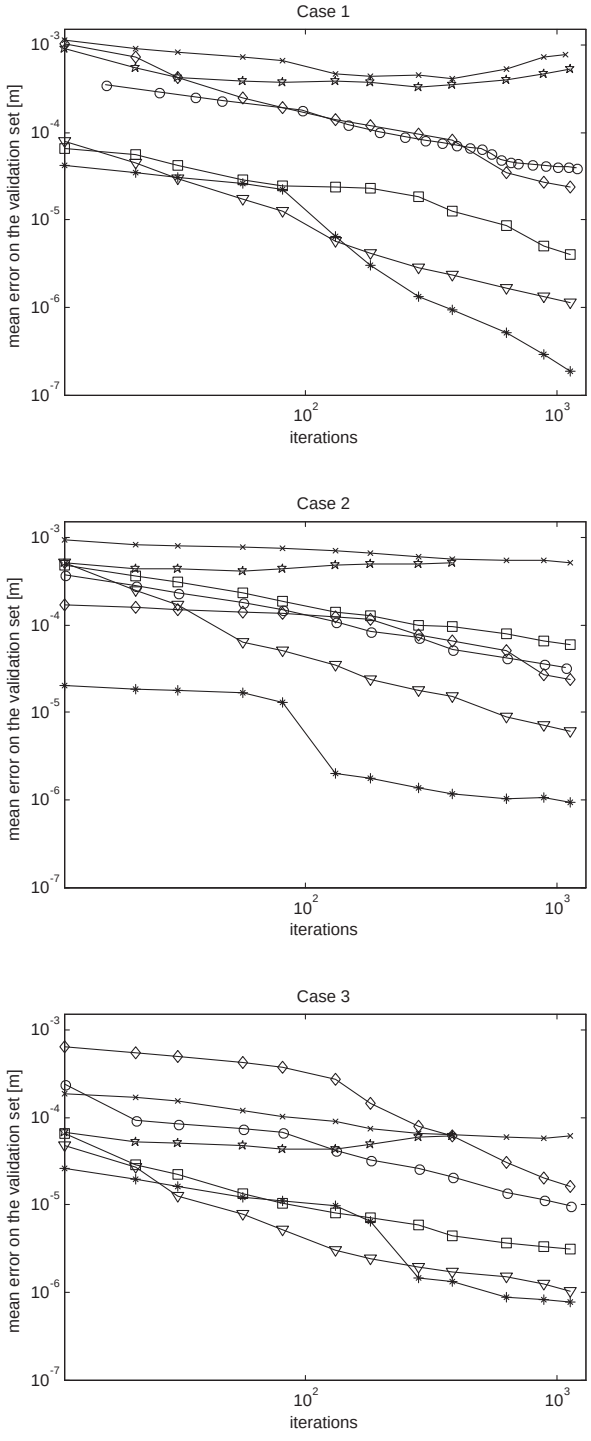
All the considered NK schemes (except for 6 and 7) performed well on the training set giving an error in the range $(10^{-7} \div 10^{-5})\,m$. Results on the validation set (Tab. 5) confirm the ability of some NK schemes to generalize the learning.

scheme	*case* 1	*case* 2	*case* 3	HN
1	0.03950	0.03230	0.00990	20
2	0.00019	0.00094	0.00077	20
3	0.00405	0.02880	0.00317	20
4	0.00114	0.00607	0.00105	20
5	0.02340	0.02350	0.00163	20
6	0.52100	0.50700	0.06140	15
7	0.77000	0.47300	0.06220	20

Table 5: Average error $[mm]$ on validation set after the training phase and optimal number of hidden neurons

4.3 Optimal NN selection

An analysis of the results shows that schemes 2 and 4 perform much better than the others in all the considered situations.



Scheme	1	2	3	4	5	6	7
Symbol	*o*	∗	□	▽	◊	⋆	×

Figure 7: Average error on the validation set, during the learning process, for the considered schemes in the 3 error cases

First at all, schemes 1 to 5 using joint coordinates Q as input for the NN performed better with respect to those based only on gripper coordinates S, probably because the same gripper position can be reached with different values of the joint coordinates resulting in different pose errors.

Moreover, the parallel schemes (2,3,4,5 and 6) are preferred than the series ones (1,7); probably because the NN has to learn just the difference between the nominal and the actual robot kinematics.

As already mentioned a scheme adopting only a NN without a nominal kinematic model was rejected after the preliminary tests because its performances were clearly worse.

After all the tests the NK models 2 and 4 confirmed to be a good tool to predict the kinematic behaviour of the actual robot. Few neurons were sufficient to reproduce the direct kinematics even in presence of load deflecting the joints. These two schemes were selected for further tests.

4.4 Tests with random errors

After having identified the more efficient NK schemes, further tests permitted to verify the NN behaviour in presence of noise or random errors like backlash.

All the poses constituting the training set were corrupted adding random errors with the maximum amplitude described in Tab. 6. E is the average pose error while the random component of the pose error E_r is computed as difference between the pose reached by the robot with constant and random errors and the pose reached by the robot with only the constant part of the errors.

Then the NN of the NK models 2 and 4 were trained using the corrupted set following the procedure denoted as "early stopping method" [Haykin (1999)].
The number of training poses was dropped to 36 and the number of the neurons was experimentally minimized in order to avoid the overfitting risk while keeping good convergence on the corrupted training set.

Finally the average and maximum pose error were evaluated with respect to the validation set. Results are reported in Tab. 7 for the direct kinematics and in Tab. 8 for the inverse one.

In this case for the NK schemes two performance indexes were evaluated: the "actual error" and the "apparent error". The actual error (E_{ac}) is the error evaluated comparing the gripper pose predicted by the NK model with the pose of the validation set not affected by random noise. The apparent error (E_{ap}) is the difference between the predicted poses and the poses of the validation set corrupted by the random noise. It is important to note that $E_{ac} < E_{ap}$. In other words, the NN reduces the effect of the random noise filtering it. With reference to the symbols of Tab. 6, 7 and 8, we evaluated the filtering index as:

$$FI = \frac{E_{ac}}{E_r} \tag{5}$$

Depending on the NK scheme, the filtering index was in the range $30\% \div 80\%$.

This result means that if the NN is trained using data having a certain amount of random error E_r, the apparent error in the validation set would be nearly equal to E_r, but the actual error would probably be much lower ($30\% \div 80\%$) because the NN produces an "average" effect on the data. This is quite desirable.

The same considerations can be made for the inverse NK schemes: using the same values of the errors of Tab. 6, the results obtained (Tab. 7, Tab. 8) show that schemes 2 and 4 are suitable for inverse kinematic calibration too.

parameters	*const*	*rand*
δl_1 [mm]	5	± 0.1
δl_2 [mm]	-6	± 0.12
$\delta\alpha$ [deg]	2	± 0.04
$\delta\beta$ [deg]	-6	± 0.026
δx [mm]	0	± 0.01
δy [mm]	0	± 0.01
pose error	*ave.*	*max.*
E [mm]	12.6	20.0
E_r [mm]	0.17	0.49

Table 6: Parameters for tests with random noise and corresponding pose error before calibration.

scheme	2		4	
HN	8		2	
error	*ave.*	*max.*	*ave.*	*max.*
E_{ap} [mm]	0.21	0.67	0.17	0.44
E_{ac} [mm]	0.14	0.43	0.05	0.17
FI	86%		32%	

Table 7: Results of the tests with random errors on validation set after direct kinematics calibration

scheme	2		4	
HN	7		5	
error	*ave.*	*max.*	*ave.*	*max.*
E_{ap} [mm]	0.18	0.54	0.19	0.66
E_{ac} [mm]	0.08	0.28	0.10	0.38
FI	49%		62%	

Table 8: Results of the tests with random errors on validation set after inverse kinematics calibration

4.5 Considerations

Schemes 2 and 4 proved to perform well even in presence of random errors and guarantee a final actual error close to the robot repeatability. Of course in experimental application only the apparent error can be estimated, but it is reassuring that the actual error would be a little better than it.

This good results encouraged the adoption of similar calibration techniques to more complicated robot models.

5 Application to a simulated Stewart platform

The main feature of the NN based calibration is that it does not require any kind of information about the causes of inaccuracy of the robot. This quality becomes fundamental when the robot has a complex structure.

The authors applied the proposed methodology to a Stewart platform robot having the configuration of Fig. 8.

The spherical hinges of the base (B_i) and the platform (P_i) are placed at the vertices of a regular hexagon and an equilateral triangle inscribed in a circle with a radius of 1 m and 0.5 m respectively. Their position with respect to the base and the platform frames are

$B_i = [x_{bi}\ y_{bi}\ z_{bi}]^t$
$P_i = [x_{pi}\ y_{pi}\ z_{pi}]^t$

Six translational joints (Q_i) link the platform to the base and control its position and orientation (6 dof).

Three frames are used: frame $\{b\}$ fixed to the base, frame $\{p\}$ fixed to the platform and frame $\{g\}$ fixed to the gripper (translated with respect to $\{p\}$ of $z_{pg} = 0.2$ m). The gripper pose is defined by the vector $S = [x_g\ y_g\ z_g\ \psi\ \theta\ \phi]$ (using Tait Brian orientation angles); the length of the six "legs" are used as joint coordinates $Q = [l_1\ l_2\ l_3\ l_4\ l_5\ l_6] = [Q_1\ Q_2\ Q_3\ Q_4\ Q_5\ Q_6]^t$.

The pose of the gripper with respect to the base is represented by the position matrix

$$
\begin{aligned}
&M_{bg} = Trans(X,x_g)\cdot Trans(Y,y_g)\cdot Trans(Z,z_g)\cdot \\
&Rot(X,\psi)\cdot Rot(Y,\vartheta)\cdot Rot(Z,\phi) = \\
&= \begin{bmatrix}
c_\theta c_\psi & -c_\theta s_\psi & s_\theta & x_g \\
s_\theta c_\phi + c_\psi s\phi & -s_\psi s_\theta s_\phi + c_\psi c_\phi & -s_\psi c_\theta & y_g \\
-c_\psi s_\theta c_\phi + s_\psi s_\phi & c_\psi s_\theta s_\phi + s_\psi c_\phi & c_\psi c_\theta & z_g \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{aligned}
$$

the position of the mobile platform with respect to the gripper is

$$
M_{gp} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -z_{pg} \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

The inverse kinematic problem is easily solved: the positions of platform hinges $[x'_{pi}y'_{pi}z'_{pi}]^t$ with respect to the base frame $\{b\}$ are computed using the transformation matrix M_{bp} which represents the position of the platform frame with respect to the base one

$M_{bp} = M_{bg}M_{gp}$

$$
\begin{bmatrix} x'_{pi} \\ y'_{pi} \\ z'_{pi} \\ 1 \end{bmatrix} = M_{bp} \begin{bmatrix} x_{pi} \\ y_{pi} \\ z_{pi} \\ 1 \end{bmatrix}
$$

the leg's length are computed as

$$
\begin{aligned}
l_i &= \sqrt{(x_{bi}-x'_{pi})^2+(y_{bi}-y'_{pi})^2+(z_{bi}-z'_{pi})^2} \\
i &= 1\ldots 6
\end{aligned} \tag{6}
$$

The direct kinematic problem is solved using iterative numerical methodologies (extended Newton-Raphson method).

Three different types of structural errors were added to this simulated robot:

- position error of the hinges of the base and the platform ($\delta x_i\ \delta y_i\ \delta z_i$);
- length error of the legs (δl_i);

- position (δx_g δy_g δz_g) and orientation ($\delta\psi$ $\delta\theta$ $\delta\phi$) error of the gripper with respect to the platform.

The actual value of the geometrical parameters of the robot L is obtained adding three components:

$$L = L_n + \delta L_c \pm \delta L_r \tag{7}$$

where L_n is the nominal (theoretic) value, dL_c is the constant part of the error and dL_r is the random component which varies in each pose. Random component represents backlash and the measurement tool uncertainty. For each geometrical parameter a value of the constant error L_c in the ranges specified in Tab. 9 was randomly chosen.

	δL_c	δL_r
δx_i δy_i δz_i [mm]	± 1	± 0.02
δl_i [mm]	± 3	± 0.02
δx_g δy_g δz_g [mm]	± 0.1	± 0.01
$\delta\psi$ $\delta\theta$ $\delta\phi$ [mrad]	± 0.25	± 0.02

Table 9: Ranges of the constant (δL_c) and the random (δL_r) geometrical error values.

Inside the work space of the robot, defined as

$$\begin{aligned} -0.2\,m &< x_g < 0.2\,m \\ -0.2\,m &< y_g < 0.2\,m \\ 1.0\,m &< z_g < 1.4\,m \\ -30\,deg &< \psi < 30\,deg \\ -30\,deg &< \theta < 30\,deg \\ -30\,deg &< \phi < 30\,deg \end{aligned}$$

50 poses (randomly distributed) for learning set (lea) and 100 for validation (val) were selected.

As a consequence of the (small) geometrical errors, the theoretical gripper pose differs from the actual pose. The homogeneous matrix M that describes the roto-translation between them has the following form

$$M \simeq \left[\begin{array}{ccc|c} 1 & -\Delta\phi & \Delta\theta & \Delta x \\ \Delta\phi & 1 & -\Delta\psi & \Delta y \\ -\Delta\theta & \Delta\psi & 1 & \Delta z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{8}$$

Calibration has the aim of minimizing the difference between predicted and actual pose, i.e. minimizing the extra diagonal elements of M. To measure the pose prediction quality two different error index were computed, the first for position, the second for orientation:

$$E_{xyz} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{9}$$

$$E_{ang} = \sqrt{\Delta\psi^2 + \Delta\theta^2 + \Delta\phi^2} \tag{10}$$

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave	*max*	*ave*	*max*	*ave*	*max*
δL_c	lea	*xyz*	5.98	14.3	5.98	20.7	5.98	20.6
		ang	9.60	17.1	9.34	20.5	9.34	20.5
		xyz_r	0.10	0.26	0.04	0.14	0	0
		ang_r	0.08	0.17	0.03	0.10	0	0
	val	*xyz*	5.99	28.8	6.47	50.0	6.48	50.1
		ang	9.62	26.0	9.93	59.9	9.96	60.1
		xyz_r	0.11	0.60	0.05	0.40	0	0
		ang_r	0.09	0.56	0.04	0.31	0	0
$\delta L_c/2$	lea	*xyz*	3.22	13.5	2.87	7.03	3.24	13.5
		ang	4.88	12.3	4.62	7.12	4.88	12.3
		xyz_r	0.10	0.38	0.05	0.18	0	0
		ang_r	0.08	0.27	0.04	0.13	0	0
	val	*xyz*	2.93	9.91	3.10	25.3	2.93	9.76
		ang	4.70	10.5	4.86	24.7	4.70	10.1
		xyz_r	0.10	0.46	0.05	0.24	0	0
		ang_r	0.08	0.38	0.04	0.23	0	0

Table 10: Pose errors for the Stewart platform before calibration (mm or mrand).

Average and maximum values on the validation set before calibration are shown in Tab. 10; to highlight the effects of constant and random errors, simulation tests were repeated 6 times with different combinations of the constant ($\delta L_c, \delta L_c/2$) and of the random ($\delta L_r, \delta L_r/2, 0$) errors. In Tab. 10 values labelled by *xyz* and *ang* are the average linear and angular error, while xyz_r and ang_r are the random components.

Contrary to serial robots, in parallel ones a joint configuration Q can bring the robot gripper in several different poses with different pose errors. This means that the robot configuration is completely defined only when the pose vector S is known. However, in actual uses of parallel robots, only one of the possible joint configuration is used because the change of the configuration implies the crossing of a singularity (where the structure is under-constrained). With this restriction even vector Q gives full information. For this reason, the calibration schemes that can be used are 2, 4 and 6.

Scheme 4 and 6 involve an angular sum on gripper coordinates S (orientation angles ψ,θ and ϕ). In order to avoid singularities of Tait Brian angles ($\theta = \pm\pi/2$) and discontinuity across $\pm\pi$ the transformation $T(\cdot)$ was built choosing eight elements of matrix M_{bp} and defining the angular S_a and the trigonometric S_t form of S:

$$\begin{aligned} S_a &= [x\ y\ z\ \psi\ \theta\ \phi]^t \\ S_t = T(S_a) &= [x\ y\ z\ c_\theta c_\psi\ \ -c_\theta s_\psi \\ &\quad\ s_\theta\ \ -s_\psi c_\theta\ \ c_\psi c_\theta]^t \\ &= [S_{t1}\ \ S_{t2}\ \ S_{t3} \therefore\ S_{t8}]^t \\ S_a = T^{-1}(S_t) &= [S_{t1}\ \ S_{t2}\ \ S_{t3}\ \ atan2(-S_{t7}, S_{t8}) \\ &\quad\ asin(S_{t6})\ \ atan2(-S_{t5}, S_{t4})]^t \end{aligned}$$

Pose error after calibration of the inverse kinematics are shown in Tab. 11. The average actual error on the validation set was strongly reduced but not enough to reach the repeatability of the robot E_r: for this reason the filter index FI is greater than 100%.

The proposed calibration methodology applied to a 6 dof robot had worse performance than the

2 dof SCARA. It is important to notice that NN's learning is, practically, an interpolation procedure so the density of the poses in the training set is a main factor for a good performance. If the robot has several dof, a great number of measured poses are necessary to cover the n-dimensional work-space of the robot (with n=dof).
Some comments are: the pose errors after calibration in the validation set is 2 or 3 times that of the calibration set having the same initial value of constant and random error. The two considered schemes have in average the same performances: the average accuracy error is reduced from some millimeters to about 0.1 mm. The presence of the random error significantly degrades the calibration performance.

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave.	*max.*	*ave.*	*max.*	*ave.*	*max.*
δL_c	lea	xyz ap	.074	.203	.054	.118	.026	.078
		ang ap	.160	.504	.111	.299	.061	.190
		xyz ac	.066	.182	.054	.110	.026	.078
		ang ac	.142	.501	.108	.338	.061	.190
	val	xyz ap	.133	1.03	.112	1.49	.087	.737
		ang ap	.327	3.40	.282	5.15	.225	2.47
		xyz ac	.123	.924	.111	1.55	.087	.737
		ang ac	.304	3.01	.280	5.32	.225	2.47
	FI	xyz	132%		217%		-	
		ang	386%		669%		-	
$\delta L_c/2$	lea	xyz ap	.040	.114	.019	.036	.010	.023
		ang ap	.101	.320	.041	.106	.025	.077
		xyz ac	.039	.098	.021	.049	.010	.023
		ang ac	.084	.280	.044	.121	.025	.077
	val	xyz ap	.096	.384	.045	.258	.032	.115
		ang ap	.198	1.19	.107	.755	.086	.465
		xyz ac	.083	.046	.040	.246	.032	.115
		ang ac	.173	1.50	.097	.689	.086	.465
	FI	xyz	217%		86%		-	
		ang	669%		258%		-	

Table 11: Pose errors after inverse kinematics calibration of the Stewart platform using scheme 2.

6 Full SCARA model

The set of structural errors used for the SCARA robot simulated in the previous case was not complete. This means that not all possible geometrical inaccuracy were considered for the simulated robot. In order to test the NN based calibration methodology with a full robot model, a 3 dof SCARA manipulator was used.

The kinematics model is shown in Fig. 9: reference frames are positioned on the robot using Denavit & Hartenberg conventions.

In a generic serial robot the number of geometrical errors is

$$N = 6 + 4R + 2P \tag{11}$$

were R is the number of rotational joint and P the number of translational one. However, depending on the measuring instrumentation, just some of them can be estimated. Assuming that G

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave.	*max.*	*ave.*	*max.*	*ave.*	*max.*
δL_c	lea	xyz ap	.039	.087	.025	.057	.021	.053
		ang ap	.080	.178	.056	.182	.051	.174
		xyz ac	.039	.092	.026	.054	.021	.053
		ang ac	.085	.286	.061	.155	.051	.174
	val	xyz ap	.115	.659	.068	.648	.081	.832
		ang ap	.287	2.01	.161	2.24	.183	2.41
		xyz ac	.103	.600	.064	.559	.081	.832
		ang ac	.264	1.62	.153	1.94	.183	2.41
	FI	xyz	110%		125%		-	
		ang	336%		366%		-	
$\delta L_c/2$	lea	xyz ap	.033	.108	.030	.076	.010	.033
		ang ap	.084	.382	.066	.417	.022	.110
		xyz ac	.032	.066	.030	.087	.010	.033
		ang ac	.068	.227	.067	.256	.022	.110
	val	xyz ap	.073	.259	.071	.213	.041	.303
		ang ap	.143	.750	.160	1.28	.095	1.05
		xyz ac	.061	.160	.061	.383	.041	.303
		ang ac	.113	.505	.149	1.18	.095	1.05
	FI	xyz	60%		131%		-	
		ang	137%		395%		-	

Table 12: Pose errors after inverse kinematics calibration of the Stewart platform using scheme 4.

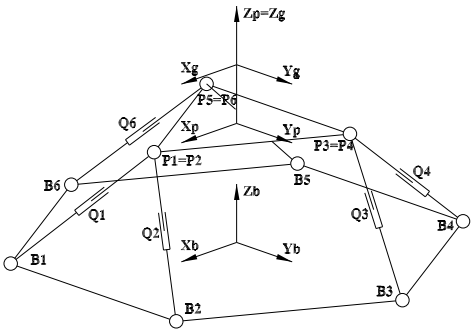


Figure 8: Simulated Stewart platform.

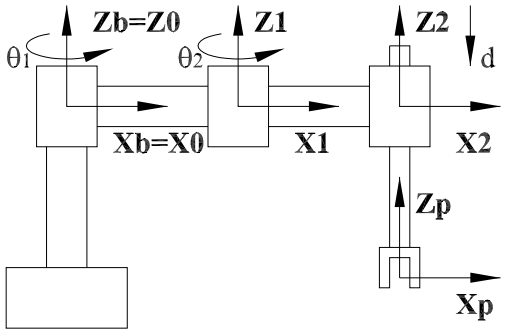


Figure 9: Simulated 3 DOF SCARA.

coordinates of the gripper pose can be measured the number of the identifiable parameters is

$$N = G + 4R + 2P \tag{12}$$

Eq. 11 and Eq. 12 are obtained generalizing the concepts described in [Mooring et al.(1991)]. Since we assumed that only the position $(x\ y\ z)$ of the gripper could be measured (and not its orientation), the number of geometrical error parameters which have to be defined to describe the robot geometry is $N = 3 + 4 \cdot 2 + 2 \cdot 1 = 13$.

A complete set of parameters describing the 3 dof SCARA geometry and compliance was selected [Omodei et al. (2000), Legnani et al.(2001)]. Nominal values and errors of the geometrical and compliance parameters are shown in Tab. 13 and Tab. 14 respectively.

A force F, whose components are function of the pose coordinates was applied to the gripper causing joint and gripper deflections (Tab. 15).

$$F_x = c_1 x + c_2 xy + c_3 z$$

L		L_n	δL_c	δL_r
X_0	trans(X_b) [mm]	0	2.5	$\pm$0.02
Y_0	trans(Y_b) [mm]	0	-2.2	$\pm$0.02
χ_0	rot(X_b) [mrad]	0	2.3	$\pm$0.02
ψ_0	rot(Y_b) [mrad]	0	-1.7	$\pm$0.02
θ_1	rot(Z_0) [mrad]	Q_1	-1.7	$\pm$0.02
a_1	trans(X_0) [mm]	330	0.4	$\pm$0.02
α_1	rot(X_0) [mrad]	0	0.1	$\pm$0.02
ψ_1	rot(Y_0) [mrad]	0	-0.3	$\pm$0.02
θ_2	rot(Z_1) [mrad]	Q_2	0.7	$\pm$0.02
a_2	trans(X_1) [mm]	330	-0.4	$\pm$0.02
α_2	rot(X_1) [mrad]	0	0.2	$\pm$0.02
ψ_2	rot(Y_1) [mrad]	0	1.7	$\pm$0.02
d	trans(Z_2) [mm]	Q_3	0.1	$\pm$0.02

Table 13: Geometrical parameters of the 3 dof SCARA.

L	L_n	δL_c	δL_r
K_{xy} [m/N]	*rigid*	10^{-5}	$\pm 10^{-6}$
K_z [m/N]	*rigid*	10^{-5}	$\pm 10^{-6}$
$K_{\theta 1}$ [rad/Nm]	*rigid*	10^{-5}	$\pm 10^{-6}$
$K_{\theta 2}$ [rad/Nm]	*rigid*	10^{-5}	$\pm 10^{-6}$
K_d [m/N]	*rigid*	10^{-4}	$\pm 10^{-6}$

Table 14: Compliance parameters of the 3 dof SCARA.

	L_n	δL_c	δL_r
c_1	0	3	$\pm$2
c_2	0	-6	$\pm$2
c_3	0	12	$\pm$2
c_4	0	-15	$\pm$2
c_5	0	30	$\pm$2
c_6	0	3	$\pm$2
c_7	0	6	$\pm$2
c_8	0	-18	$\pm$2

Table 15: Coefficients of Eq. 13: forces in [N] distances in [m].

$$\begin{aligned} F_y &= c_4 x_2 + c_5 xz \\ F_z &= c_6 + c_7 z + c_8 xy \end{aligned} \tag{13}$$

Inside the work space (which has a torus shape $R_{int} = 0.13$ m, $R_{ext} = 0.53$ m, 0 m $< z <$ 0.3 m) 72 uniformly distributed poses for the learning set (lea) and 430 for the validation set (val) were selected. The pose error is computed using Eq. 9. Average and maximum values of the pose error E and the random component E_r before calibration are shown in Tab. 16. To better analize the properties of the calibration technique in different situations, the algorithms have been tested with different combinations of constant (δL_c) and random (δL_r) errors values.

Apparent and actual pose error after direct kinematics calibration using scheme 3 and 4 are shown in Tab. 17 and Tab. 18 respectively. Gripper coordinates don't include angular values, so no transformation $T(\cdot)$ is needed.

Results reported in Tab. 16, Tab. 17 and Tab. 18 show that the calibration procedure proposed

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave	*max*	*ave*	*max*	*ave*	*max*
δL_c	lea	E	3.44	4.61	3.51	4.44	3.50	4.50
		E_r	0.28	0.65	0.15	0.29	0	0
	val	E	3.51	4.93	3.51	4.85	3.50	4.74
		E_r	0.30	0.65	0.14	0.32	0	0
$\delta L_c/2$	lea	E	1.80	2.60	1.77	2.36	1.76	2.28
		E_r	0.26	0.54	0.13	0.28	0	0
	val	E	1.76	2.85	1.76	2.35	1.74	2.42
		E_r	0.27	0.64	0.14	0.32	0	0

Table 16: Pose errors for the 3 dof SCARA before calibration.

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave	*max*	*ave*	*max*	*ave*	*max*
δL_c	lea	ap	0.27	0.52	0.21	0.45	0.07	0.16
		ac	0.18	0.39	0.16	0.38	0.07	0.16
	val	ap	0.38	1.01	0.23	0.60	0.12	0.48
		ac	0.23	0.70	0.18	0.44	0.12	0.48
		FI	76%		131%		-	
$\delta L_c/2$	lea	ap	0.29	0.65	0.12	0.25	0.04	0.10
		ac	0.21	0.60	0.10	0.20	0.04	0.10
	val	ap	0.37	0.96	0.17	0.39	0.05	0.18
		ac	0.24	0.80	0.10	0.34	0.05	0.18
		FI	89%		74%		-	

Table 17: Pose errors after direct kinematics calibration of 3 dof SCARA using scheme 3.

[mm]	set	E	δL_r		$\delta L_r/2$		$\delta L_r = 0$	
			ave	*max*	*ave*	*max*	*ave*	*max*
δL_c	lea	ap	0.29	0.57	0.19	0.50	0.03	0.08
		ac	0.19	0.34	0.15	0.41	0.03	0.08
	val	ap	0.36	0.96	0.21	0.53	0.05	0.30
		ac	0.21	0.54	0.15	0.41	0.05	0.30
		FI	69%		109%		-	
$\delta L_c/2$	lea	ap	0.25	0.58	0.12	0.25	0.006	0.02
		ac	0.14	0.31	0.09	0.18	0.006	0.02
	val	ap	0.30	0.63	0.17	0.43	0.01	0.10
		ac	0.14	0.37	0.10	0.30	0.01	0.10
		FI	51%		74%		-	

Table 18: Pose errors after direct kinematics calibration of 3 dof SCARA using scheme 4.

is very effective in the reduction of the positioning error. By using both scheme 3 and 4 the error is reduced to values close. For the considered manipulator the best results are achieved by scheme 4.

As observed for the 2 DOF SCARA robot, the NN partially filters the random errors.

7 Conclusions

The paper presents a preliminary study on an innovative Neural Network based calibration procedure. Different schemes of a Neuro-Kinematic model were tested in direct and inverse kinematic calibration both on serial and parallel simulated robots. Three of them obtained good performances: in all considered cases accuracy error was reduced nearly to robot repeatability. It was also shown that the NK model is able to filter part of the noise component present in the measures.

The NN based procedure seems to be suitable to the calibration of all robot types and extremely simple to use: only the nominal kinematic model of the robot is needed, the procedure is able to compensate for all type of ripetitive errors present in the robot (geometrical, load deflections, ...) without modelling them.

Similarly to other procedures, a proper (quite large) number of measured poses distributed in the workspace (or in a part of it) is needed. The results obtained in the simulated cases encouraged the authors to apply the proposed procedure to actual robots.

References

[Nowrouzi et al.(1988)] Nowrouzi A., Kavina Y.V., Kochekali H., Whitaker R.A., An overview of robot calibration techniques, Ind Robot, pp. 229-232, 15(4), 1988.

[Zierteg and Datseris (1989)] Zierteg J., Datseris P., Basic considerations for robot calibration, Int J Robot Aut 4(3), 1989, pp.158-166.

[Mooring et al.(1991)] Mooring B.W., Roth Z.S., Driels M.R., Fundamentals of manipulator calibration, New York, 1991.

[Kovac and Frank (1992)] Kovac I., Frank A., A novel industrial robot calibration device, Proc. of First RAA, Porotoz, Slovenia, pp. 162-167, 1992.

[Faglia et al. (1993)] Faglia R., Legnani G., Magnani P.L., Docchio F., Minoni U., Calibration of a SCARA robot by optical measurements: Methodology and experimental results, ISMCR'92 2^{nd} Int Symp on Measurement and Control in Robotics, Sept. 21-24, 1993.

[Legnani et al.(1996)] Legnani G., Mina C., Trevelyan J.P., Static calibration of industrial manipulators: Design of an optical instrumentation and application to SCARA robots, J Robot Syst, 13(7), 1996.

[Trevelyan et al.(1996)] Legnani G., Trevelyan J.P., Static calibration of industrial manipulators: a comparison between two methodologies, Robotics Toward 2000 Twenty-Seventh Int Symp on Industrial Robots, pp. 111-116, Oct. 6-8, 1996.

[Khoshzaban et al.(1996)] Khoshzaban M., Sassani F., Lawrence P.D., Kinematic calibration of industrial hydraulic manipulators Robotica v.14, pp. 541-551, 1996.

[Haykin (1999)] Haykin S., Neural networks: a comprehensive foundation, Prentice hall, 1999.

[Omodei et al. (2000)] Omodei A., Legnani G., Adamini R., Three Methodologies for the Calibration of Industrial Manipulators: Experimenthal Results on a SCARA Robot J. of Robotic System 17(6), pp. 291-307, Wiley, 2000.

[Legnani et al.(2001)] Legnani G., Adamini R., Jatta F., Calibration of a SCARA Robot by Force-Controlled Contour Tracking of an Object of Known Geometry, 32nd ISR, Korea, Apr. 19-21, 2001.

[Tiboni et al. (2003)] M. Tiboni, G. Legnani, P. L. Magnani and D. Tosi, A closed-loop neuroparametric methodology for the calibration of a 5 DOF measuring robot, Proceedings of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, July 2003, Kobe Japan, pp 1482- 1487.

[Fazenda et al. (2006)] Fazenda N., Lubrano E., Rossopoulos S., Clavel R. Calibration of the 6 DOF High-Precision Flexure Parallel Robot Sigma 6, Proceed. of the 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, April 25-26, 2006.