Teaching Grasping to a Humanoid Hand as a Generalization of Human Grasping Data

Michele Folgheraiter, Ilario Baragiola, and Giuseppina Gini

Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Piazza L. da Vinci 32, 20133 Milano, Italy
{folghera,gini}@elet.polimi.it

Abstract. Humanoid robotics requires new programming tools. Programming by demonstration is good for simple movements, but so far the adaptation for fine movements in grasping is too difficult for it. Grasping of natural objects with a natural hand is known as one of the most difficult problems in robotics. Mathematical models have been developed only for simple hands or for simple objects. In our research we try to use data directly obtained from a human teacher as in imitation learning. To get data from users we built a data glove, we collected data from different experiments, and generalized them through neural networks. Here we discuss the nature of the data collected and their analysis.

1 Introduction

Haptic sense is very important for human beings, especially during activities like manipulation. Sometimes it is possible to do a task without visual feedback using only tactile and force sensations. This is the reason why future Virtual Reality (VR) systems should be improved by devices capable to acquire somatic-sensory data (like articulation positions and velocities) and able to evoke touch and force feelings.

On the other side, the control of a robotic hand can be much easier if the complex data about positions and force are learned from a human teacher and not developed from geometric and dynamic equations. An expert performs the grasp, and phalanx positions and fingertip forces are acquired only when the object is firmly gripped. Then data are used by a Neural Network to learn how to generate position and force to grasp objects with some generalization. This is useful for example to control an artificial hand without computing the inverse kinematic and dynamic problem.

The challenge of our investigation is the possibility to teach grasping to a humanoid hand after learning from human grasping. To obtain data from the human teacher we designed a special glove, as we will illustrate in the following.

Grasping in humans is a complex activity and takes place in two steps: planning, which requires encephalon activity, and executing, which requires activity of the neuro-motor system. During execution the cerebellum plays an important role in comparing the reference from the encephalon and the sensorial data from the motor system.

Different postures are available for human grasping. They differ in the number of degrees of freedom and in the force exerted. The chosen posture depends on the properties of the object and on the task. Many authors have proposed interpretations about

J.A. López et al. (Eds.): KELSI 2004, LNAI 3303, pp. 139–150, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the generation of forces and positions. In [1, 2, 3] three basic directions (or primitives) are defined, as illustrated in Figure 1.

1. *Pad opposition* moves the fingers in a direction parallel to the palm. It identifies the direction x.
2. *Palm opposition* moves along a direction perpendicular to the palm, and identifies the z axis.
3. *Side opposition is along a direction along the palm and identifies the y axis.*

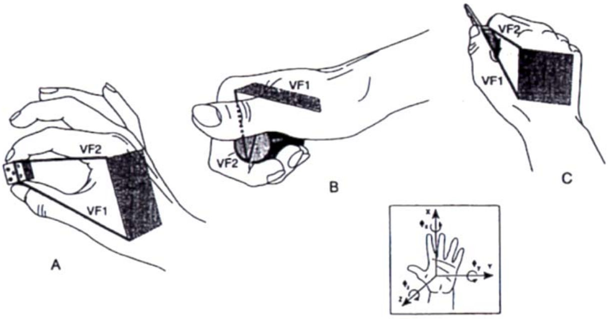

Fig. 1. A. Pad opposition; B. Palm opposition; C. Side opposition

In *Pad Opposition*, the hand can exerts small movements and applies small forces. It is typical of precision movements. In *Palm Opposition* the hand can use large forces. In *Side Opposition* the result is intermediate: medium precision and medium forces.

Cutkosky et al [4, 5] proposed a classification that integrates the relevance of the task considered with the precision and the power of grasping. Some classes have been modelled with the aim to automatically produce actuation commands for a given hand grasping a given object.

Considering the difficulties of mathematical models for the hand and the object, we see here how to learn from data acquired from a human teacher.

In Section 2 we illustrate the use of Neural Networks to learn grasping from different trials.

In Section 3 we describe our data glove used as a tool to get data, and in Section 4 we use data to infer grasping positions and force on new objects. We adopted in practice the pad opposition and the side opposition schemas to grasp simple objects and to analyse position and force data applied by a human to get a model for an artificial hand with similar kinematics.

In Section 5 we show the kinematics of the humanoid hand that will execute the motions. After we discuss the results and conclude.

2 Neural Networks for Grasping

The learning capability of neural networks has been applied to many fields in robotics as well as in many data analysis problem. Some NN architectures have been reported in literature about grasping.

Some authors proposed to learn the hand position for grasping. Kuperstein and Rubinstein [6] used a gripper with 5 degrees of freedom. Taha et al. [7] studied how to move a hand with 7 d.o.f. to grasp objects of different shapes. They applied those ideas to the control of prosthesis. They defined two postures of the hand: at time 0, when it is open, at time 1 when the object is grasped, for a given type of objects and sizes. The NN was able to compute the intermediate positions to obtain the grasp.

Moussa and Kamel studied how to learn generic grasping positions as a mapping between the reference frame of the object and the hand using the contact information from the tip of 5 fingers. A module makes the configuration of the hand position, another module defines the fingers positions. Fingers positions are determined by a network making the inverse kinematics. Taha et al. Have H. Huang et al. developed CANFM (Cascade Architecture of Neural Network with Feature Map) to classify 8 kinds of grasping using as input the EMG signals collected from 3 places on the patient arm. The final task is to classify them according to the 8 kinds of grasps. The first layer is a Kohonen network, which receives 3 input from sensors. The output from the SOM selects the coordinates x-y in a 2D topologic net, used to provide 6 input to a BPNN.

The last research to mention here is from Matsuoka [8], developed for the Cog robot with a hand of four fingers. Every finger has 4 dof, actuated by a motor through a cable (tendon), each with position sensor. Force sensor are distributed over the hand. The grasp activity of Cog imitates the human grasping reflex of babies. The hand controller has two parts: the first is a reflex control, the second is a NN. Data are collected from experiments on the real robot: time of closure and forcs, for different objects, are the input to the network (see Figure 2).

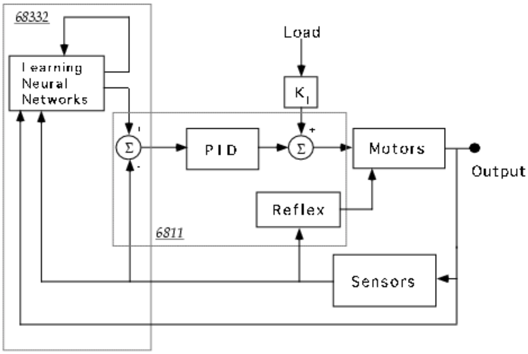

Fig. 2. The Cog controller

We chose neural networks for their easy implementation and integration in the Matlab/Simulink environment, and for their real-time behaviour after training.

3 Data Acquisition with Our Data Glove

The glove we designed is equipped with 16 sensors: 14 positions sensors and 2 force sensors. Position sensors measure the flexion of each phalanx for every finger, except the little finger. There are also two special sensors that measure the adduction for the thumb and the index. Indeed these two fingers have a first phalanx with an improved mobility, especially for the adduction and abduction movement.

Sensor signals are acquired using an electronic board, that is connected with an analog/digital card (Pcl 812) mounted on a Pc, and using the XpcTarget tool of Matlab. Data are sampled and converted into digital format with a frequency of 2KHz. We see in Figure 3 the glove.

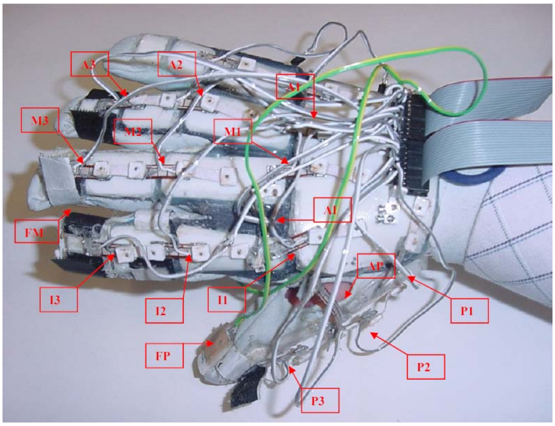

Fig. 3. The data glove

Our data glove acquires 16 signals:

- 12 positions for the phalanxes of first 4 fingers
- 2 abduction for thumb and index
- 2 forces for thumb and middle

The samples of grasps have been designed to learn different sizes and different materials.

The grasp objects are made of:

- Polystyrene (density=$0.029297\frac{g}{cm^3}$)
- Wood (density=$0.70898\frac{g}{cm^3}$)
- PVC (density=$1.4219\frac{g}{cm^3}$)

Objects of 7 different dimensions for each material are provided, as illustrated in Figure 4.

For each object size and material, ten grasps are monitored, and the positions of phalanxes, the abductions, the forces, are stored in an array of 16 columns. The rows are related to different times during grasping; with a sampling time of 0.0005 sec, 500 rows for each grasp are usual.

Three steps are needed before learning:

1. validation
2. deriving the force on the index
3. normalization

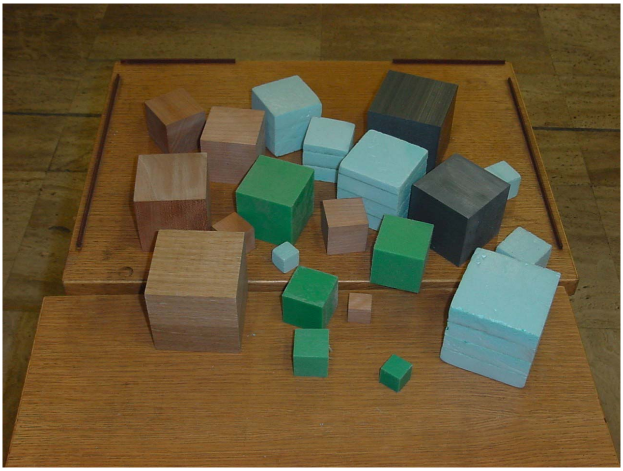

Fig. 4. The cubes used in the experiments

The grasp is different for different dimensions of the object. For the small cubes, of 2 or 3 cm size, the grasp involves only thumb and index, applying two forces of equal module and in the opposite directions.

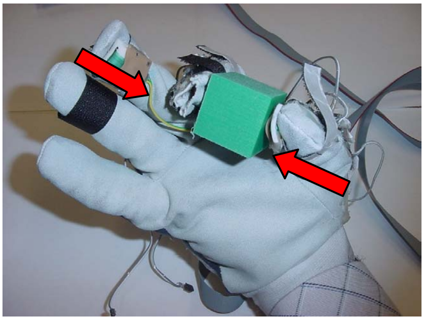

Fig. 5. Grasping a 3cm cube, with the directions of forces

For bigger cubes also the middle finger is used on the same surface of the index. The force of the thumb equals the sum of forces of index and medium.

The forces of the index are computed as

1. for small 2 or 3 cm cubes: $F_{index} = F_{thumb} = F_{middle} \approx 0$
2. cubes of 4 to 8 cm: $F_{index} = F_{thumb} - F_{middle}$

Input data to the network are two: cube dimension and density of the material, normalized in [0,1].

Densities are normalized using the maximum densities as before reported, while the dimensions are normalized to the maximum size of 10 cm

$$l_{norm} = \frac{l}{10} \tag{1}$$

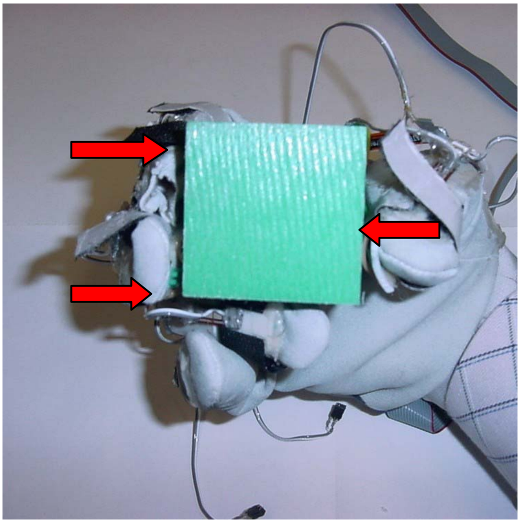

Fig. 6. Grasping a 6 cm cube

The output are position and forces for grasping, normalized in [-1,1] according to the formula:

$$X_{norm} = \left\{ \left[\left(\frac{X - X_{\min}}{X_{\max} - X_{\min}} \right) * 2 \right] - 1 \right\} \tag{2}$$

with $X_{\min}$ minimum value in the distribution, $X_{\max}$ maximum value in the distribution.

Examples of input and output are in Table 1. In all the tables numbers are with the comma notation for decimal part, as in out Matlab settings.

4 Training the Network

A feedforward network with 2 inputs, 17 outputs and 2 hidden layers with 20 neurons each, and a tangent sigmoid transfer function is the chosen architecture. The Matlab algorithm "traingdx", a fast backpropagation with heuristic techniques is used. The performance of the algorithm depends on the learning rate, which is adaptively adjusted.

After different learning, the best network in terms of number of neurons, training time, MSE is chosen. The MSE is illustrated in Figure 7. As we see, after 300 epochs, the error is stabilized.

We compare the real data obtained from the average on all the materials for the same cube size and compute the error and the variance. We see in Table 2 the results for position errors, while in Table 3 the results for force errors.

We applied early stopping to avoid overfitting of the network. This method requires that data are divided in 3 sets: the training set (used to compute the gradient

Table 1. Input and output samples

	Density	size
Input1	0,94793	0,2
Input2	0,94793	0,2
Input3	0,94793	0,2
Input4	0,94793	0,2
Input5	0,94793	0,2

	I1	I2	I3	M1	M2	M3	A1	A2
T1	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T2	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T3	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T4	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T5	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T6	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T7	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T8	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T9	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T10	0,54135	-0,42844	-0,10097	-0,094291	-0,50748	-0,065206	0,45685	-0,58024
T11	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T12	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T13	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T14	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T15	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T16	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T17	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841
T18	0,53884	-0,42494	-0,0953	-0,098169	-0,50475	-0,065206	0,44193	-0,5841

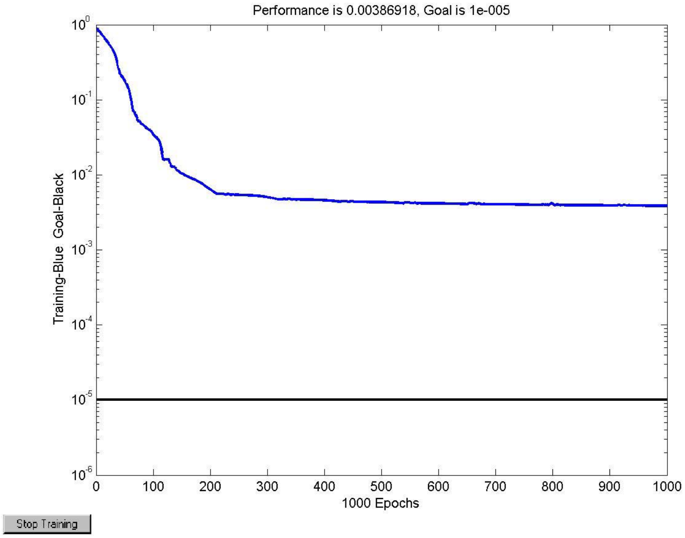

Fig. 7. MSE after training

and the weights), the validation set (used to monitor the error), and the test set. In Figure 8 we see in blue the MSE on the training set, in green on the validation set.

Table 2. Results of position error analysis

PVC	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	0,42869	-0,04124	0,29123	0,013478
Variance	2,4075	3,135	0,75822	0,32512

WOOD	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	0,25967	0,78452	0,84806	0,42862
Variance	2,0385	4,8306	4,7805	1,3845

POLYSTYRENE	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	0,033329	0,45582	0,2648	0,3304
Variance	0,48287	3,5469	1,8219	0,92951

Table 3. Force error analysis

PVC	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	-0,005613	-0,000561	-0,002300	-0,015903
Variance	0,000093	0,000017	0,000031	0,000704

WOOD	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	-0,014947	-0,003487	-0,000958	0,008940
Variance	0,000340	0,000765	0,000196	0,000128

POLYSTYRENE	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	-0,000378	-0,004563	-0,000323	-0,000960
Variance	0,000051	0,000014	0,000004	0,000013

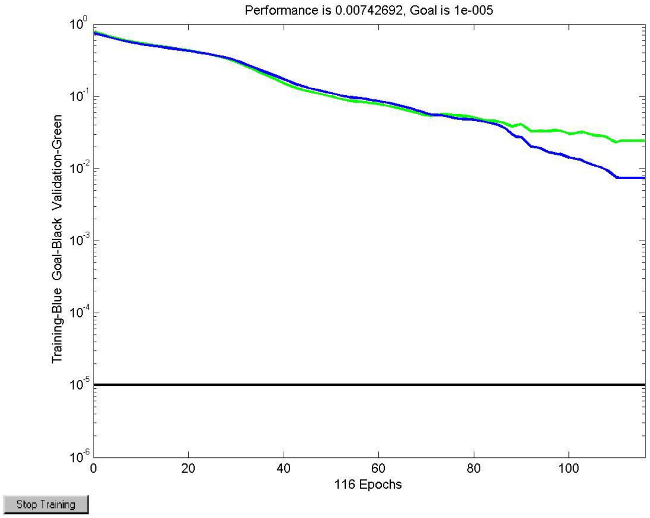

Fig. 8. MSE training set, validation set

The training stops after 117 epochs with MSE of 0.00654269 when the error on the validation set starts growing.

After training, we separately analyse the errors on the training set and on the test set. Results on the training set are reported in Table 4.

The test set contains instead cubes made of the three materials and of 3, 5, 7 cm. Repeating the analysis on the test set , we observe that the error (in degrees) for positions has a very low average (less than 2 degrees) but a high variance. The error for force is around 0.01 kilograms, and the variance is low.

Comparing results after controlling the overfitting, we observe a clear improvement on the test set in comparison with results obtained before checking overfitting.

Table 4. Errors on the prediction of the network on the training set

PVC	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	0,49529	-0,26398	0,58482	-0,92526
Variance	4,8442	8,6617	9,7215	7,8894

WOOD	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	0,54215	0,41388	-1,0726	-0,32047
Variance	11,556	8,6249	14,358	1,8077

POLYSTYRENE	*Cube 2 cm*	*Cube 4 cm*	*Cube 6 cm*	*Cube 8 cm*
Mean error	-0,34199	0,57926	0,15018	-0,49648
Variance	4,6829	5,0929	4,2389	9,318

Without overfitting checking, the position error ranges from ±0.5° to ±2.20°, but grows as far as ±8.05° for the test set only. Moreover the error is randomly distributed.

We see in Table 5 and 6 the data from Net1, the basic net, and Net2 trained with early stopping.

Table 5. Results from the net without and with early stopping on the training set

Standard deviation	*2 cm*	*4 cm*	*6 cm*	*8 cm*
PVC Net 1	1,5516	1,7706	0,87076	0,5702
PVC Net 2	1,1134	1,4791	2,2453	1,3917
Wood Net 1	1,4278	2,1979	2,1864	1,1766
Wood Net 2	2,1143	2,4236	1,4239	1,0625
Polystyrene Net 1	0,6949	1,8833	1,3498	0,96411
Polystyrene Net 2	1,7062	2,2723	1,7304	2,3702

Table 6. Results from the net without and with early stopping on the test set

Standard deviation	*3 cm*	*5 cm*	*7 cm*
PVC Net 1	5,0229	3,1659	3,2374
PVC Net 2	7,0026	5,4505	5,4312
Wood Net 1	6,4726	6,0083	4,4586
Wood Net 2	7,848	5,6188	7,0634
Polystyrene Net 1	8,0516	2,3718	4,8196
Polystyrene Net 2	7,4079	1,8793	4,5296

On the second network, the error on the training set is in the range ±1.06° to ±2.42°, and on the test set is ±1.9° to ±7.8°.

With the obtained values we can actuate a real anthropomorphic hand which has the kinematics described, The force value will be used by our control system which is able to apply given forces on the finger tips.

5 The Kinematics of the Humanoid Hand

We describe the humanoid hand we are using with our humanoid arm. More details are in [9, 10]. All the fingers but the thumb are equal, as illustrated in Figure 9. The hand mimics the human fingers in size and structure: it has a spherical joint with 2 degrees of freedom from metacarpus to the first phalanx, and cylindrical joints between the phalanxes. The activation is obtained through tendons actuated by McKibben pneumatic actuators.

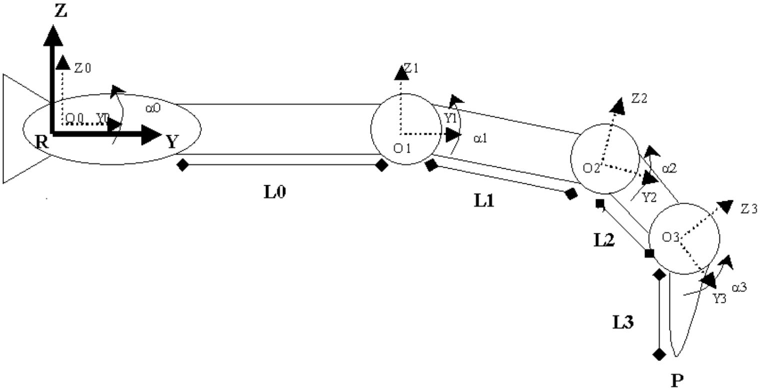

Fig. 9. Side view of the right index, with the joint reference systems

To compute the direct kinematics we use the Denavit Hartemberg notation to build the transformation matrices from the reference systems defined in the joints.

We assume that the coordinate system O_1, X_1, Y_1, Z_1 is moving with R,X,Y,Z**.** In this case the first matrix has $\alpha_0 = \theta_0 = 0$, and simplifies to:

$$T_{0\mapsto R} = \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \tag{3}$$

The second matrix is a simple translation as in equation 4:

$$T_{1\mapsto 0} = \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \tag{4}$$

The matrix from O_2, X_2, Y_2, Z_2 to O_1, X_1, Y_1, Z_1 contains the variables α_1 and $\theta 1$:

$$T_{2\mapsto 1} = \begin{array}{cccc} \cos\theta_1 & -\cos\alpha_1 \cdot \sin\theta_1 & \sin\alpha_1 \cdot \cos\theta_1 & -L2 \cdot \cos\alpha_1 \cdot \mathrm{sen}\theta_1 \\ \sin\theta_1 & \cos\alpha_1 \cdot \cos\theta_1 & -\sin\alpha_1 \cdot \cos\theta_1 & L2 \cdot \cos\alpha_1 \cdot \cos\theta_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \tag{5}$$

With post multiplication we obtain the matrix to transform a point given on the second phalanx system O_2, X_2, Y_2, Z_2 to the basic point R,X,Y,Z.

$$\begin{matrix} X_r \\ Y_r \\ X_r \\ 1 \end{matrix} = T_{0\mapsto R} \times T_{1\mapsto 0} \times T_{2\mapsto 1} \times \begin{matrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{matrix} \qquad (6)$$

We need to express a point P on the finger tip, so on the last phalanx, but the actuation of the last phalanx is not independent from the actuation of the first phalanx, as in the human hand. The analysis, as in Figure 10, gives the answer.

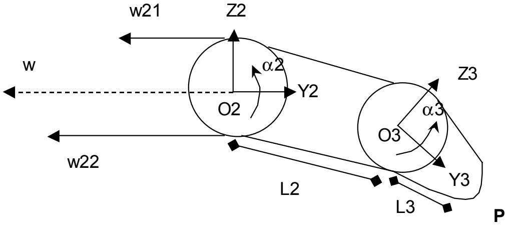

Fig. 10. Resolving the last phalanx

In fact we obtain the values to be used in the previous equation considering the contribution of the third joint.

$$\begin{aligned} &P(x_2) = 0 \\ &P(y_2) = L2 \cdot \cos\alpha_2 + L3 \cdot \cos(\alpha_2 + \alpha_3) \\ &P(z_3) = L2 \cdot \sin\alpha_2 + L3 \cdot \sin(\alpha_2 + \alpha_3) \end{aligned} \qquad (7)$$

Another important aspect is about the computation of the inverse kinematics of the hand to be able to find the values to actuate to reach a given position. In our case, we obtain directly angles values from the data glove, and we need only to transform the angles into actuators values, i.e. the length of the McKibben muscles to actuate to obtain the given joint angle. This transformation is easily computed from the actuator model.

6 Discussion and Conclusion

Future work will require to apply to the real robot hand the predicted values and check the resulting action. The way to improve the manipulation ability of our robot is still long. Other kinds of grasping will be studied, for instance considering the precision and pinch grasp. The idea is to generate different networks for the different grasping configurations of the hand, and to develop an arbitration network to get the good net according to the shape of the object.

Considering the data acquisition phase, data obtained from the same glove from different executors could be compared to understand the variability for different people making the same task and to find more standardized ranges of values. It is our opinion that some of the variance of the learned data is simply acceptable and different position/force patterns can reach a stable grasping on the object.

With respect to mathematical modelling our approach has the advantage to have a very low computational complexity and to be usable without a complete geometric description of the object to be grasped.

References

1. Iberall T.: The nature of human pretension: three dextrous hands in one, IEEE Trans RA **5**, 3 (1999)
2. Iberall T.: Human Prehension and dexterous robot hand, International Journal of Robotic Research (1989)
3. Iberall T., Bingham G., Arbib M.A.: Opposition space is a structuring concept for the analysis of skilled hand movements. In H. Heuer, H. and Fromm, C. (eds): Generation and modulation of action patterns, Springer-Verlag, Berlin(1986) 158-173
4. Cutkosky M.R., Wright P.K.: Friction, stability and the design of robotic fingers, International Journal Robotics Research **5**, 4 (1987) 20-37
5. Cutkosky M.R.: On grasp choice, grasp models and the design hands for manufacturing tasks, IEEE Trans. RA **5**, 3 (1989) 269-279
6. Kuperstein M., Rubinstein J.: Implementation of the adaptive controller for sensory-motor condition. In Pfeifer, R., Schreter, Z., Fogelman, F., Steels, L. (eds): Connectionism in perspective, Elsevier Science Publishers (1989) 49-61
7. Taha Z. et al: Modelling and simulation of the hand grasping using neural networks. Med Eng Phys. 1997 Sep;19(6):536-8
8. Matsuoka, Y: The Mechanisms in a Humanoid Robot Hand, Autonomous Robots **4**, 2 (2000) 199-209
9. Folgheraiter, M., Gini, G.: Blackfingers: an artificial hand that copies human hand in structure, size, and function, Proc. IEEE Humanoids (2000) Cambridge, Mass.
10. Folgheraiter M. Gini, G.: Human-like hierarchical reflex control for an artificial hand, Proc IEEE Humanoids (2001)Tokyo, Japan