ABCM Symposium Series in Mechatronics - Vol. 2 - pp.555-562

LEARNING BASED REGRASPING APPLIED TO AN ANTROPOMORPHIC ROBOT HAND

Glauco Augusto de Paula Caurin
EESC – USP, Av. Trabalhador Sãocarlense 400, 13566-590, São Carlos, SP, Brasil
gcaurin@sc.usp.br

Luiz Carlos Felicio
EESC – USP, Av. Trabalhador Sãocarlense 400 13566-590, São Carlos, SP, Brasil
feliciol@sc.usp.br

Abstract. *In this paper, a robot hand regrasping procedure is presented. Regrasping may be understood as the stage of the manipulation process when one of the fingers loses temporarily the contact with the object and moves to a better contact position. The motion to a different contact position improves grasping quality, optimizes the cooperation between the fingers and avoids finger configurations close to the mechanical limits of the finger joints.*
Regrasping was modeled as a combination of two different tasks: collision avoidance and trajectory planning. Motion planning which takes the existence of obstacles into account is a well established procedure in mobile robotics field. However, its application on robot hand manipulation processes, not restricted to the two dimensional case, has received almost no attention until now.
In the specific case of an anthropomorphic hand, each finger is considered as a single manipulator and during regrasping, a finger should reach the desired new contact position without colliding with the other fingers, nether touching an undesired object point.
Additionally, a Hopfield neural network was integrated to the system in order to reduce the computational time, required for the collision avoidance processing.
The approach was tested using a five fingers robot hand performing the manipulation of a single object.

Keywords*: Robot Hand, Learning, Collision Avoidance, Intelligent Manipulation*

1. Introduction

The knowledge acquired by the research and industrial community in the past two decades, in the different fields of robotics together with the increasing computational capacity, encourage new challenging research and development activities in the design of systems that gradually perform complex tasks with the robotic hands and fingers. Assembly and Disassembly of complex mechanical parts, eating with chopsticks, dressing and undressing clothes, may be understood as such complex manipulation tasks. Researchers have developed algorithms for dexterous manipulations of various kinds, for example, the synthesis of a suitable grasp for an object (Borst, Fisher and Hirzinger, 1999), how to measure and improve the quality of a grasp (Miller and Allen, 1999), and how to re-orient an object that is being grasped (Han and Trinkle, 1998), how to use visual information about the object to assist the grasping (Valente et.al. 1999), how to design a suitable hand for dexterous manipulation (Salisbury, 1982). Many studies have been also made on object manipulation by regrasping, the subject of this paper, such as Tournassoud et al. (1987) which proposed regrasping with one robot arm equipped with parallel-jaw end-effectors, and complementary studies on regrasping applied to robot hands may be found on Farooqi et.al. (1999) and Stoerer et.al. (1999)

There are different definitions for regrasping. Bicchi (2000) suggests that regrasping as a manipulation involving a sequence of grasps on the object, alternated with phases in which the object is left alone on a work table. End-effectors as simple as on–off grippers can be used to implement this effect. However, manipulation by regrasping has drawbacks, among which is the need for grasping and releasing the object several times during manipulation, and the consequent time consumption in the process.

Different from previous works on the field, in this work the definition above is extended to include situations where the object do not need to be left alone and also to include conditions where the object to be regrasped is moving. For instance, when a contact point reaches the limit position for the contacting finger, that finger must be detached from the object and reattached at a new contact point. This is only feasible using a hand with more than three fingers. Then, regrasping all fingers, arbitrary object manipulations can be performed.

To carry out the proposed regrasping approach, a collision avoidance algorithm was integrated to a trajectory planning as follows: first, an object surface point representation is transformed from the Cartesian space to the joint space for each finger. Second, the desired object motion are taken into account and the corresponding object surface point coordinates in the joint space can be analyzed. Finally, using the manifolds that are the results of the previous procedures, a new suitable contact positions can be elected.

The regrasping process is understood in this paper as a trajectory generation problem that allows the connection of the current finger contact position to a new contact point, assuming that both point coordinates are previously known. To avoid undesired and unplanned finger object touch, a collision avoidance task is added to the trajectory generation.

The paper is organized as follows: first a joint space representation of the system (including the object) is introduced as a powerful tool for the manipulation process analysis. The Cartesian representation is used in the text for clarity reasons. Then the collision avoidance strategy is integrated to the trajectory planning process. The collision avoidance strategy prepares a two dimensional surface where a Hopfield Network (section 2.2) can easily learn how to connect the desired initial and final positions. Finally, some simulation results for the trajectory calculation using a five fingers hand and a prismatic object are presented.

2. Modeling the Regrasping Phase

To allow a better understanding of the regrasping process, the set of points representing the object surface is initially presented in the Cartesian space (Figure 1) and then transformed in a corresponding manifold in the joint space for each single finger, as shown in Figure 2.

The approach brings advantages to the trajectory planning process because it provides:

- Visual explanation why the finger contact needs to be changed;
- Verification if the new contact position is appropriated; and,
- Inspection for the path generated by the trajectory planning algorithm.

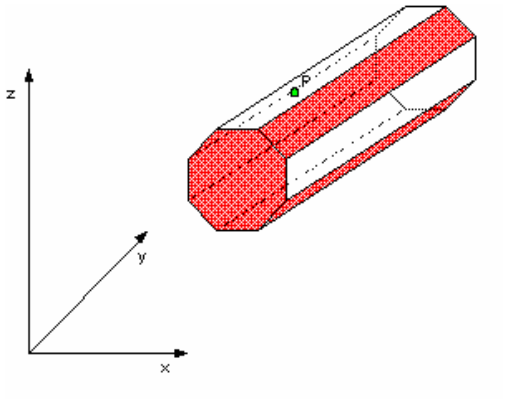

Figure 1. Prismatic object representation in the Cartesian space

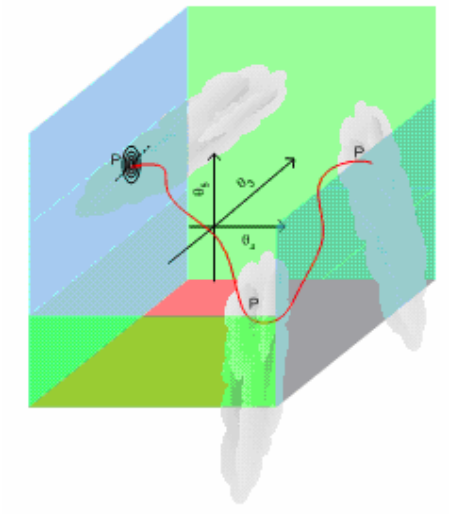

Figure 2. Step by step representation of the manifold displacement.

The angular coordinate limits are added for each one of the joint axes and they are represented as orthogonal (boundary) planes in joint space (Figure 2). The manifold moves in the joint space following the object motion in the Cartesian space. Assuming an arbitrary finger contacting the object surface at point P, the diagram allows a step by step study of the feasibility of keeping the contact while the object is moving. When the contact point P meets one of the

boundary planes it triggers the need for finger repositioning, consequently this allows the continuity of the object manipulation, performing the desired object motion in space.

The choice of the new contact point position Q on the object surface must satisfy simultaneously two conditions: (a) Q should lie inside the space limited by the boundary planes and additionally (b) the point should be strategically placed allowing the motion continuity for a longer time period; therefore, avoiding excessive new regrasping calls.

Applying the first condition, every point that lies on the manifold surface and do not exceed the limits established by the boundary planes may be considered as a candidate position for Q. The second condition is implemented as optimization criteria. Performing the requested motion of the manifold, Q is selected as the point that keeps inside the allowed space over a longer period of time.

2.1. Collision Avoidance

Knowing that the process has now a starting position and a final position, the definition of the path connecting them may be interpreted as a trajectory planner task. Since the object continues to move during the regrasping, collision avoidance ability is added to the trajectory planner as a form to ensure that there will be no intermediary contact with the object during the execution of the regrasping path.

To simplify the path generation, another coordinate transformation is performed. A Cartesian system with the origin on the object center of gravity is chosen. A limited object region containing points P and Q are also chosen. A mesh connecting the surface points of that region is generated and "virtually inflated" as shown in Figure 3. It can be noticed that the sections containing the initial and final position is not affected.

This procedure offers two direct advantages:

- It reduces the motion possibilities to a two-dimensional (mesh) repositioning problem,
- It ensure no unpredicted contact between a finger and the object

When the regrasping task is treated in a coordinate system attached to the object, it allows decouple the problem of trajectory generation for connecting the two points P and Q from the need of taking the object motion into account. Those two questions are treated separately; thus, the complexity is reduced.

Under this point of view, the procedure consists in finding a sequence of points (trajectory) over the inflated surface of the object and afterwards, transforming these point coordinates to the finger joints space (inverse kinematics). That first phase will be treated in section 2.2 where an approach based on the Competitive Hopfield Neural Network is introduced. The second phase, discussed in section 2.3, requires a synchronization of successive object motions with their simultaneous coordinate transformations.

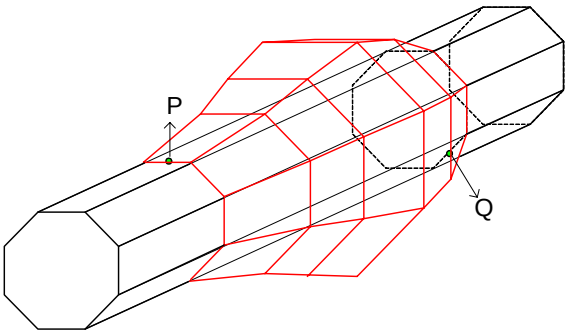

Figure 3. The object is "virtually" modified to avoid undesired contact during regrasping

2.2. Path Generation between Two Contact Points

Based on the mesh generated over the object, it is possible to construct a grid in the plane (Figure 4) which represents a very suitable form to solve trajectory planning problems using neural networks. The initial and final points P and Q are identified on the grid and the decision for the best path to connect them takes into account the distance to be traveled and the finger joint angle limitations.

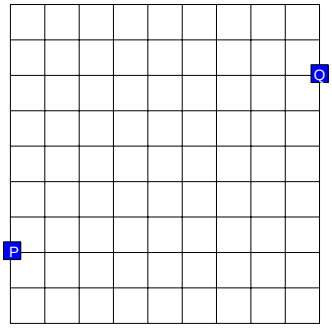

Figure 4. The 3 D trajectory generation is converted in a two-dimensional problem of connecting the initial point P to the final desired position Q

2.2.1. Modeling the polygonal approximation problem

The polygonal approximation is a classical problem. Describing a polygonal approximation with an energy function enables its solution using a Hopfield Network. When the network reaches a minimum energy state it will point to a polygon that minimizes the deviation between the original curve and an approximated route.

To formalize the polygonal approximation problem a closed curve $P = \{p_1 , p_2 , p_3 , \ldots ,p_n\}$ is considered as a set of n ordered points (Figure 5). In this definition, $p_X p_Y$ and $\overline{p_x p_y}$ represent respectively the original curve and the chord connecting point p_X to point p_Y. In Araújo and Tanaka (1995) the deviation between the curve and the polygon is defined as the sum of the perpendicular distances from the points belonging to $p_X p_Y$ up to the chord $\overline{p_x p_y}$:

$$h_{xy} = \sum_{p \in p_x p_x} d(p, \overline{p_x p_y}) \tag{1}$$

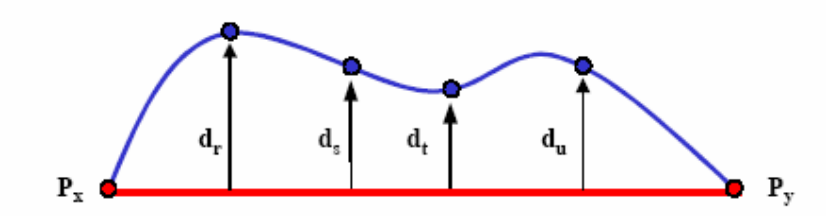

Figure 5. Point to chord deviation $h_{x,y}$. The chord (red) approximation error with respect to the curve $p_x p_y$ (blue) is defined as $d_r + d_s + d_t + d_u$.

The H matrix is defined as the connection matrix built upon these deviations. Observe that it is a non-symmetric matrix, i.e., $h_{x,y} \neq h_{y,x}$.

$$H = \{h_{xy}), x,y = 1,2, \ldots ,n \tag{2}$$

Based on this criterion, a solution for the polygonal approximation using a Hopfield Network was developed. For a given curve defined by n points and for a number of approximation points m , with m<n, the objective of the proposed method consists in finding a polygonal approximation $Q^{(m)}$ defined by m points of the curve P in such a way that the deviation between point and chord is minimized.

A classical Hopfield network may be implemented in a two-dimensional configuration (Figure 7) with *n* lines and *m* columns which comes up with *n* by *m* mutually connected neurons.

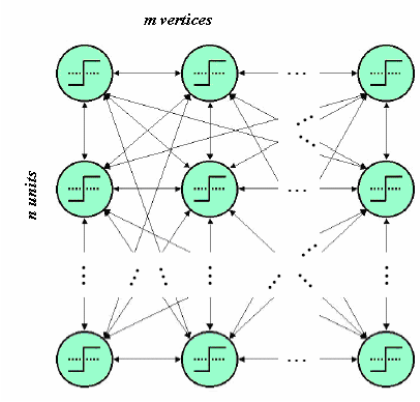

Figure 6. Representation of a Classical Hopfield Network. All units are interconnected.

Considering $V_{x,i}$ as the binary state of neuron (x,i) and $W_{x,i;y,j}$ as the weight value (connection) between the unit (x,i) and the unit (y,j), each neuron receives a signal $W_{x,i;y,j}\ V_{y,j}$ from each unit (y,j) besides a *bias* $I_{x,i}$ defined externally. An equation describing the network may be written as follows:

$$U_{x,i} = \sum_{y=1}^{n}\sum_{j=1}^{m} W_{x,i;y,j} V_{y,j} + I_{x,i} \tag{3}$$

The lines in the Hopfield matrix network represent the points of the curve, while the columns represent the vertexes positions on the approximation polygon. For example, an active neuron $V_{x,i}$ indicates that the x_{th} point on the curve was chosen as the i_{th} vertexes of the polygon

The stability of the two-dimensional Hopfield network is defined by the following Lyapunov function

$$E = -\sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{m}\sum_{j=1}^{m} W_{x,i\,;\,y,j} V_{x,i} V_{y,j} - 2\sum_{x=1}^{n}\sum_{i=1}^{m} I_{x,i} V_{x,i} \tag{4}$$

The network convergence is reached when the energy defined by the Lyapunov function is minimized. A possible objective function for the polygonal approximation is given by the following equation:

$$\begin{aligned} E = \quad & A \cdot \sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{m} \left(h_{y,x} V_{x,i} V_{y,i-1} + h_{x,y} V_{x,i} V_{y,i+1}\right) + B \cdot \sum_{x=1}^{n}\sum_{i=1}^{m}\sum_{\substack{j=1 \\ j\neq i}}^{m} V_{x,i} V_{x,j} + \\ & C \cdot \sum_{i=1}^{m}\left(1 - \sum_{x=1}^{n} V_{x,i}\right)^2 + D \cdot \left(m - \sum_{x=1}^{n}\sum_{i=1}^{m} V_{x,i}\right) \end{aligned} \tag{5}$$

Where the cyclic indices are given by: $V_{0,i} = V_{n,i}$, $V_{n+1,i} = V_{1,i}$, $V_{x,0} = V_{x,m}$ e $V_{x,m+1} = V_{x,1}$.

This function incorporates the boundary conditions for the polygonal approximation. The first term of the objective function defines the sum of the deviations between the arc and the chord defined by the adjacent points of the polygon. The second term determines that no point p_x on the curve appears twice on the polygon. The third term punishes the choice of point p_x and p_y in the same vertexes. The forth term certifies that a total of m vertexes where chosen for the approximation.

It should be noticed that the three former terms are penalty terms which try to keep the solution realistic while the first term minimizes the deviation between the curve the polygon. The quality of the solution is directly dependent of the weighting factors A, B, C and D. However, the definition of this terms is relatively time consuming.

2.2.2. Implementing the Competitive Hopfield Network

The Hopfield competitive network is proposed to give freedom to the definition problem of weighting factors, since the penalty terms will be defined in an implicitly way.

The *winner-take-all* rule (Bishop, C.M. 1995) is applied to perform the neuron updating. The neurons of the same layer compete among themselves to define which one will take the highest activation state. Thus, the winner unit will assume the value of 1 (one) and the others of the same layer will take the value of 0 (zero).

$$V_{x,i} = \begin{cases} 1 & if \quad U_{x,i} = \max\{U_{1,i}, U_{2,i}, \ldots, U_{n,i}\} \\ 0 & for\ the\ other\ cases \end{cases} \tag{6}$$

The *winner-take-all* rule avoids that the two points p_x and p_y would be defined for the same vertex. Moreover, it makes sure that a total of *m* points will be chosen for the vertexes of the polygon.

To keep the penalty imposed by the second term, a new definition for the connection matrix was taken, Equation 7. The change imposed to the arc-chord deviation assures that none of the points p_x will be in more than one polygon vertexes.

$$h_{x,y} = \begin{cases} \infty & if \quad x = y \\ \max\limits_{p \in \overline{p_x p_y}} \{d(p, \overline{p_x p_y}\} & if \quad x \neq y \end{cases} \tag{7}$$

The simplification imposed by equations 6 and 7 bounds the objective functions (Equation 5) to the first term of the minimization and eliminates the need of determining the weighting factors. The final equation is shown below and represents the energy function of the network.

$$E = \sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{m}(h_{y,x}V_{x,i}V_{y,i-1} + h_{x,y}V_{x,i}V_{y,i+1}) \tag{8}$$

Comparing the energy equation (Equation 8) with the Lyapunov function (Equation 4), one can obtain the weighting and bias values as:

$$W_{x,i\,;\,y,j} = -(\delta_{i-1,j}h_{y,x} + \delta_{i+1,j}h_{x,y})$$
$$I_{x,i} = 0 \tag{9}$$

where $\delta_{i,j}$ is the Kronecker delta function defined as: $\delta_{i,j} = 1$ if x = j and, $\delta_{i,j} = 0$ for the other cases. Applying Equations 9 and 10 to Equation 3, the total input to the units (x,i) is obtained as:

$$U_{x,i} = \sum_{y=1}^{n} -(h_{y,x}V_{y,i-1} + h_{x,y}V_{y,i+1}) \tag{10}$$

Analyzing the equation above it can be noticed that the network is not totally connected. The neurons corresponding to a specific layer receive their signs provided by the previous and the next layers; Figure 7. That new connection rule reduces significantly the network complexity.

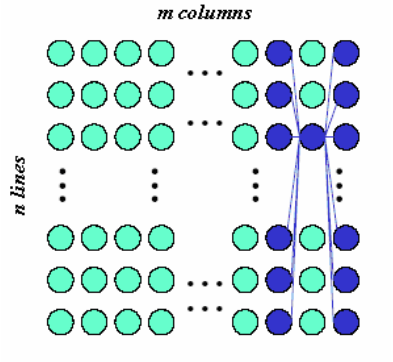

Figure 7. Competitive Hopfield Network architecture. The units of one layer are connected to every unit of the previous and the next layer.

The algorithmic of polygonal approximation used in the Hopfield Competitive Network can be summarized as follows:

1. The learning rule: construction of the connection matrix H, formed by Equation 1;
2. Definition of the network: adjustment of the initial state of the units as *m* points, the closest equidistant as possible, along the curve;
3. Propagation rule: calculation of the weighting sum for each unit (x,i) throughout the column *i*, using the Equation 6;
4. Activation rule: application of the *winner-take-all* rule (Equation 6) to obtain the new state for each column unit;
5. Repeat steps 3 and 4: until there are no changes in the network state;
6. Assembling the approximated polygon, whose vertexes q_1, q_2, ..., q_m are pointed out by the network active units.

2.2.3. Polygonal approximation process results

The Figure 8 shows a result obtained from the Hopfield Competitive Network applied to the polygonal approximating process. A trajectory is found and its outline is defined considering m equal to seven points, with five of them as intermediate points. After going through the polygonal approximating process, it follows the coordinate transformation, synchronized with step motions of the object.

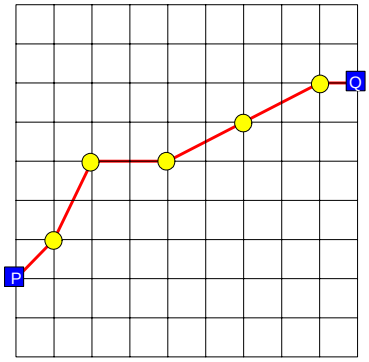

Figure 8. Results given by the Hopfield network output using seven points with five as intermediate ones.

2.3. Pos-Processing of the Results

In order to understand the results obtained form the Hopfield network, given in coordinates to be reached by a determinate finger, it is necessary some coordinate transformation.

Each active neuron of the Hopfield network corresponds to an "inflated" object coordinate. That coordinate is referred to a coordinate system attached to the object. Then, if the regrasping motion should follow the object motion, the process of going through the intermediate points should take place in a synchronized way with an equivalent number of intermediate points of the object trajectory, in the correspondent interval of time. The computation of the coordinate transformation is accomplished tanking into account the instantaneous object position with respect to the Cartesian coordinates.

The sequence of points generated by that process can supply the calculation of the inverse Kinematics and the corresponding fulfillment of the coordinates to be passed to the control of each individual joint.

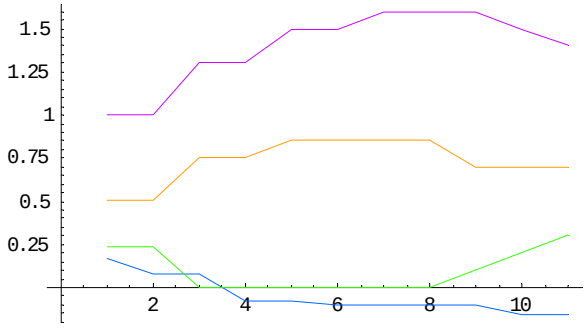

Figure 9. The curves of the joint angles for the forefinger of the BRAHMA artificial hand

3. Summary

In this paper it is presented details of a new approach for the regrasping process. The methodology considers the use of the concept of planning the trajectory combined with the concepts of deviating from obstacles. In order to carry out the regrasping strategy, it is applied a Hopfield Artificial Neural Network.

4. Acknowledgements

This work is supported by FAPESP #00/11321-2.

5. References

Araújo A.F. and Tanaka J.S., 1995. Variação de um modelo de rede neural competitiva de Hopfield para aproximação poligonal. Simpósio Brasileiro de Redes Neurais, 2., São Carlos, Anais. p.193-198.

Bicchi J., 2000, Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Toward Simplicity, IEEE Transactions on Robotics and Automation, vol. 16, no. 6.

Bishop, C.M. 1995 Neural networks for pattern recognition, Oxford University Press Inc., New York, EUA.

Butterfass, J., Grebenstein, M., Liu, H., and G. Hirzinger, 2001, "DLR-Hand II: Next Generation of a Dextrous Robot Hand". IEEE International Conference on Robotics and Automation - ICRA 2001. Seoul, Korea.

Borst C., Fischer M., and Hirzinger G., 1999, "A fast and robust grasp planner for arbitrary 3D objects," IEEE Int. Conf. Robotics and Automation.

Farooqi M. A., Tanaka T., Nagata K., Ikezawa Y., and Omata T., , 1999, "Sensor based control for the execution of regrasping primitives on a multifingered robot," . IEEE Int. Conf. Robotics and Automation.

Fukuda T., Mase K., and Hasegawa Y., 1999, "Robot hand manipulation by evolutionary programming," IEEE Int. Conf. Robotics and Automation.

Han L. and Trinkle J. C., 1998, "Dexterous manipulation by rolling and finger gaiting," IEEE Int. Conf. Robotics and Automation, pp. 730–735.

Miller A.T. and Allen P.K., 1999, "Examples of 3D grasp quality computations," IEEE Int. Conf. Robotics and Automation.

Nakamura Y., Nagai K., and Yoshikawa T., 1989, "Dynamics and stability in coordination of multiple robotic systems," Int. J. Robot. Res., vol. 8, no. 2, pp. 44–61.

Nakamura Y., Yamazaki T., and Mizushima N., 1999, "Synthesis, learning and abstraction of skills through parameterized smooth map rom sensors to behaviors," IEEE Int. Conf. Robotics and Automation.

Nilsson M., 1999 "Tactile sensing with minimal wiring complexity," IEEE Int. Conf. Robotics and Automation, pp. 293–298.

Okamura A. M., Smaby N., and Cutkosky M.R., 2000, "An overview of dexterous manipulation,". IEEE Int. Conf. Robotics and Automation, pp. 255–262.

Ota J. , Miayata N., Arai T., Yoshida E., Kurabayashi D. and Sasaki J., 1995, Transferring and Regrasping a Large Object by Cooperation of Multiple Mobile Robots, IROS International Conference on Intelligent Robots and Systems- Volume 3, 1995 Pittsburgh, Pennsylvania, USA.

Pati Y. C., Krishnaprasad P. S., and Peckerar M. C., 1992, "An analog neural network solution to the inverse problem of early taction," IEEE Trans.Robot. Automat., vol. 8, pp. 196–212.

Protzel P.W., et. al. , 1993, Performance and fault-tolerance of neural networks for optimization., IEEE Transaction on Neural Network, v.4, n.4, p.600-614.

Salisbury, J. K. ,1982, "Kinematic and force analysis of articulated hands". PhD Thesis, Stanford University, Stanford, CA.

Shegl T. and Buss M., 1998, "Hybrid closed-loop control of robotic hand regrasping," IEEE Int. Conf. Robotics and Automation, pp. 3026–3031.

Stoeter S. A., Vosst S., Papanikolopoulos N. P. and Mosemann H., 1999 "Planning of regrasp operations," IEEE Int. Conf. Robotics and Automation.

Tournassoud P. , Lozano-Perez T. and Mazer E., 1987 "Regrasping" IEEE Int. Conf. Robotics and Automation, pp. 1924–1928.

Valente, C.M.O.; Schammass, A.; Caurin, G.A.P.; Araújo; A.F. , 1999, A Neural Gripper for Arbitrary Object Grasping, Journal of Connection Science - Special issue of Adaptive Robots Vol.11 No. 3-4.

6. Responsibility notice

The authors are the only responsible for the printed material included in this paper.