A Self-Organizing State Trajectory Planner applied to an Anthropomorphic Robot Hand

Ruben C. Benante, Leonardo M. Pedro, Leandro C. Massaro, Valdinei L. Belini
Aluízio F. R. Araújo, Glauco A. P. Caurin, *Member, IEEE*

***Abstract*— A new incremental self-organizing map, called *State Trajectory Generator* (STRAGEN) is employed to plan state trajectory of a robot. STRAGEN can deal with different criteria to construct topological maps of the problem space, choosing neighbors that match these criteria and optimize different measures of the learned map. STRAGEN can also learn heterogeneous information, such as angles, torques and positions of a manipulator, preserving their characteristics. This algorithm was tested generating trajectories for a new robotic hand called Kanguera. The hand offers a suitable environment for experimental purposes due to its novel and more accurate transmission system. The implementation of adduction and abduction capacity for both the fingers and the thumb allows the execution of more complex movements. Simulations and experiments related to Kanguera hardware are also presented.**

I. INTRODUCTION

As extensively mentioned in the literature one of the main challenges in the recent robotic researches consists on the development of robot hands with dexterous capacity. If the robot can grasp and manipulate everyday objects in a human environment it is also able to interact with and modify its surroundings

The implementation of these capacities, at least as a rough approximation to the human dexterity, is not a trivial task. In fact, the capacity to manipulate unknown objects and even known objects requires a series of sophisticated abilities.

Initially, the system must be able to grasp and keep the object in a stable way, that is, to locate and reach the grasping points on the object surface, see [1] to [8] for details. The choice of these points involves matching form/force closure properties for the grasp synthesis. This synthesis may be extended to optimize the force required for the object grasping, meeting previously established quality criteria.

Additionally, for manipulation purposes the robot hand must be able to move the object grasped as in [8] and also, if necessary, perform finger repositioning (regrasping [9]). The combination of these important abilities in a sequential and suitable form (gaiting [10]) may lead to complex manipulations.

The simultaneous implementation of the different methods available for manipulation with multi-fingered robot hands is a complex problem; as a consequence, considerable computational effort is required, restricting its real-time implementation feasibility.

Taking these restrictions into account, in this paper a learning based approach is proposed, capable to generalize the fingers trajectories. The work is suggested as a contribution towards both the trajectory planning of anthropomorphic robot hands, but brings also contributions for the design of robot hand systems composed by independently articulated fingers.

A reliable mechanical design establishes a basis for the development of a successful manipulation planner. The Kanguera Hand uses an anthropomorphic model and its design is inspired by the human hand size, movability, and thumb opposability. The proposed design adopts biologically inspired contact joints. However, these joint type introduce nonlinear behavior to the transmission system connecting the actuators to the joints.

Several solutions have been presented by the scientific community to overcome these nonlinear behavior, most of them introduce springs between the pulley and the joint [11][12], and others simply connect the actuators directly to the joints [13].

The Kanguera Hand deals with the nonlinear behavior using a new transmission concept, where mechanical cams replace the traditional pulley mechanism. These cams provide a linear displacement for the cables between the motor and the joints, avoiding undesired dead zones and hysteresis.

This more reliable hand mechanism is now available for research projects in the robotic manipulation area. Three different projects are currently using it as an experimental environment: A project on Hardware-In-the-loop [14], Open Architecture Real Time Operating System [15], and different approaches on Trajectory Planning. In this paper only the aspects regarding the leaning based trajectory planner are considered

The State Trajectory Generator (STRAGEN) is a self-organized incremental neural network model that is able to deal with complex heterogeneous information, such as angles, torques and joint positions. Self-organized because it do not need a supervisor to teach how to create a topological map. The map is created using a given criterion that optimize its structure. It is incremental because the model can grow and shrink according to the amount of data and the quality of the representational map. STRAGEN keeps the heterogeneous information from being mixed with each other, by using n-dimensional pre-configured groups of similar data

This work was partially supported by FAPESP

Leonardo M. Pedro, Leandro C. Massaro, Valdinei L. Belini and Glauco A. P. Caurin are with the School of Engineering of São Carlos, University of São Paulo, SP, Brazil lmpedro@sc.usp.br, leandromassaro@gmail.com, belini@pnca.com.br and gcaurin@sc.usp.br

Ruben C. Benante and Aluízio F. R. Araújo are with the Center of Informatics, Federal University of Pernambuco, PE, Brazil rcb@cin.ufpe.br and aluizioa@cin.ufpe.br

represented in each node of the network. These information contain kinematics and dynamical data, they are stored in the nodes during the training phase and can be used at anytime, by pattern completion. Once the network had being trained with the problem space of the robot, it is sufficient and necessary to give the starting and target points, and STRAGEN will generate a trajectory from one to another using a diffusion energy algorithm. The trajectory can be generated to optimize different criteria, such as minimum distance, minimum torque variation or minimum joint angles variation.

This paper is organized as follows. In section II, the design of a new anthropomorphic hand is presented. Section III presents STRAGEN (*State Trajectory Generator*), the self-organized neural network previously mentioned. Section IV presents some STRAGEN simulation results. And finally some conclusions are presented in section V.

II. HAND DESCRIPTION

Dexterous manipulation imposes force/torque capacity and also fast motion capacity. Therefore it is an attractive challenge for the hand mechanical design. In this work a new concept of an anthropomorphic robot hand is presented. The prototype, called Kanguera (ancient indigenous word for bones outside the body) Fig.1(b), is the fourth generation of hands/grippers developed at the EESC - USP Mechatronics Lab [16][17][18] and it presents significant performance improvements when compared to the previous systems.

A. Actuators

Several robot hands have their actuators located outside its structure [11] to [19]. But there are also a number of solutions using actuators inside the hand [20] to [23]. Using actuators inside the structure increases the total weight, limiting acceleration and load capacity. On the other hand, if they are placed remotely, a transmission system is required. As in the previous versions, the Kanguera Hand uses DC servomotors located outside its structure.

B. Mechanical Design

The hand structure is made of biocompatible polyurethane based on ricinus oil. This material is lighter than the aluminum versions used in the first [16] and second [17] hand/gripper generation.

Each Kanguera finger and also the thumb have 2 rotational joints with 1 DOF and a new patented 2 DOF joint [24], Fig. 1(f), allowing adduction and abduction movements of the fingers and thumb opposability. The joints connecting the medial and distal phalanges of the 4 fingers, Fig. 1(e), are made dependent to the angular displacements of the previous joints (except for the thumb). As a consequence, the Kanguera Hand presents 16 DOF. A more complete description of the kinematical model of the hand can be found in [18]. All joints are biologically inspired surface contact joints, i.e. no axes are used to connect them to the linkages.

C. Transmission System

The contact joints introduce nonlinear kinematical behavior that must be considered when developing the transmission system. Similar conditions are also found in other robot hands, where the relation of motor angular displacements and joint angular displacements is not linear. To overcome this nonlinear behavior, elastic elements have been inserted between a conventional pulley system and the joints [11][12]. However, this approach adds undesired dynamical characteristics to the system.

Here an alternative solution is adopted using a new transmission system formed by cams replacing the conventional pulleys and specially designed to compensate the mentioned nonlinearities, Fig. 1(a) and Fig. 1(d). Another mechanical improvement was achieved introducing new routes for the transmission cables. The cables are conducted through the hand structure Fig. 1(c). This new transmission system allows independent movement of each joint and a more accurate positioning.

III. State Trajectory Generator (STRAGEN)

The algorithm will be described similarly to [25] to facilitate comparisons. The algorithm is composed of three phases: (a) the Training Phase: represents the topology of the solution space and adapts itself while reading samples from a database. (b) the Pruning Phase: eliminates unsuitable or unnecessary nodes and connections.(c) In the last phase, Trajectory Generator Phase: the algorithm tries to find the best trajectory between two points, according to a given criterion.

The database $\mathbf{B}_{L\times D}$, where L is the number of samples (lines) and D is the dimension of each sample, should be normalized before starting the training phase. Which is done in a pre-processing phase. Each sample $\mathbf{w}_j \in \mathbb{R}^D$ is normalized using its maximum and minimum values, assuring that $0 \leq \mathbf{w}_j \leq 1$.

The weight vector is defined as:

$$\mathbf{w} = [w_1 \dots w_D]^T$$

The weight vector $\mathbf{w}$ may contain heterogeneous information from various domains. To deal with this different information, the weight vector is divided into similar groups, to form possible neighborhood criteria:

$$\mathbf{w} = [\mathbf{V}_1 \mathbf{V}_2 \dots \mathbf{V}_m]^T \tag{1}$$

where the vector $\mathbf{V_i}$, $i = 1, \dots, m$ represents a set of variables of a domain that can be grouped together and operated as a whole. The dimension of each group is $\mathbf{V}_i \in \mathbb{R}^{D_i}$ and $\sum_{i=1}^{m} D_i = D$. In the case of a robot hand the groups are space positions, joint angles and joint torques.

We select $1 \leq \zeta \leq m$, to be the activity group used to calculate the activity of the network. We also select $1 \leq \eta \leq m$ to be the neighborhood criterion, used to evaluate the proximity of one stimulus and its representation in the topological map, among other proximity evaluations.

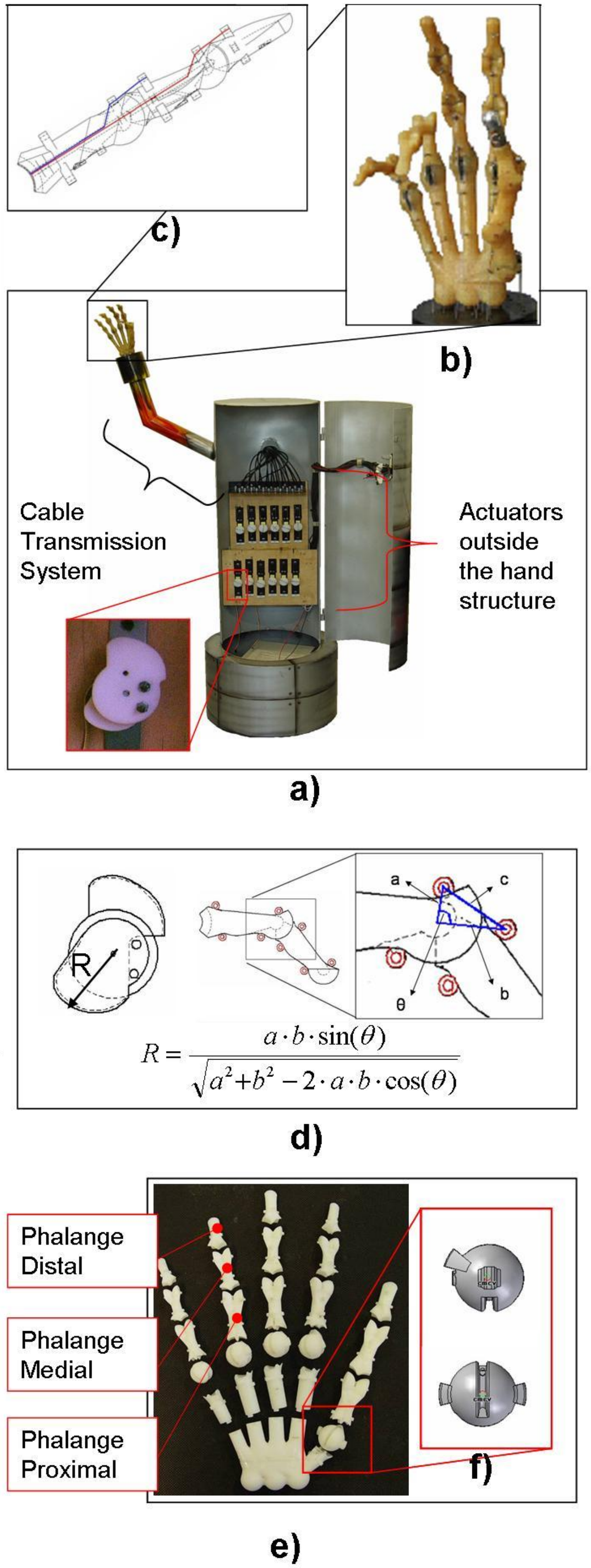

Fig. 1. Detailed Hand Hardware. a) The hand system emphasizing the cable transmission system and the actuators outside the hand structure. b) Detail of the hand. c) Cable guided inside the structure d) Formulation to design the cam system. e) Unmounted hand parts. f) The patented 2 DOF joint [24].

The activity threshold $\bar{\mathbf{a}}_k$ for the group $\mathbf{V}_\zeta$ is defined by a chosen percentage, $0 < P < 1$, of the maximum Euclidian distance in dimension D_{ζ_k}, for each subgroup inside $\mathbf{V}_\zeta$ that represents an independent information.

$$\bar{a}_k = \exp\left(P \cdot \sqrt{D_{\zeta_k}}\right), 1 \leq k \leq l$$

where l is the number of homogenous subgroups that compose $\mathbf{V}_\zeta$.

Training Phase: After the pre-processing, STRAGEN will start training all samples read from the database. The samples are not read totally random as other algorithms [26] [27], nor sequentially, but using a strategy called Motor Babbling ([28] used MB in a different way) that guides the data through possible trajectories, taking advantage of the *Structural Hill-climbing* [29], i.e., the SOM are know to be sensitive to the order which the data is presented. Therefore, presenting data in an meaningful order creates better connected maps. MB procedure is denoted $p(\boldsymbol{\xi})$ and is described as the following: for every iteration t, when a new sample is required for the Training phase, MB procedure $p(\boldsymbol{\xi})$ chooses Q random candidates not yet presented. For these Q points MB returns the nearest to the last presented input $\boldsymbol{\xi}$, according to some criterion η, unless it is the first sample. In that case, MB just returns any random sample.

Let $\mathcal{A}$ be a set of nodes and $\mathcal{C}$ be a set of connections between these nodes. Let the input distribution be $p(\boldsymbol{\xi})$, for inputs $\boldsymbol{\xi}$ as defined in MB procedure.

Let the learning rate be ϵ_b. Define the neighborhood criterion η and the activity criterion ζ. Let the final learning rate decay be $\alpha_f \approx 0$.

Before starting the algorithm, initialize the set $\mathcal{A}$ with 2 nodes n_1 and n_2 placed at $\mathbf{w}_{n_1}$ and $\mathbf{w}_{n_2}$, in $\mathbb{R}^D$, representing 2 random patterns from the database: $\mathcal{A} = \{n_1, n_2\}$ in which:

$$\mathbf{w}_{n_i} = [\mathbf{V}_1 \ldots \mathbf{V}_\zeta \ldots \mathbf{V}_\eta \ldots \mathbf{V}_m]^T$$

Initialize the connection set $\mathcal{C}$ with one connection between the first two inserted nodes $\mathcal{C} = \{c_{n_1,n_2}\}$.

The training algorithm is described bellow:

1) Generate a data sample from $p(\boldsymbol{\xi})$ as input to the network, according to the Motor Babbling procedure:

$$\boldsymbol{\xi} = [\mathbf{V}_1 \ldots \mathbf{V}_\zeta \ldots \mathbf{V}_\eta \ldots \mathbf{V}_m]^T$$

2) For each node i in the network, calculate the distance from the input $||\boldsymbol{\xi}_\eta - \mathbf{V}_{\eta,i}||$, and determine the best matching unit and the second best, $s_1, s_2 \in \mathcal{A}$, using the chosen neighborhood criterion η, such that:

$$||\mathbf{V}_{\eta,s_1} - \boldsymbol{\xi}_\eta|| \leq ||\mathbf{V}_{\eta,i} - \boldsymbol{\xi}_\eta||, \forall i \in \mathcal{A}$$

$$||\mathbf{V}_{\eta,s_2} - \boldsymbol{\xi}_\eta|| \leq ||\mathbf{V}_{\eta,i} - \boldsymbol{\xi}_\eta||, \forall i \in \mathcal{A} - \{s_1\}$$

3) Add one to the number of wins of s_1: $\sigma_{s_1} = \sigma_{s_1} + 1$.
4) Insert a new connection between s_1 and s_2 in $\mathcal{C}$, if there is not one yet: $\mathcal{C} = \mathcal{C} \cup \{c_{s_1,s_2}\}$
5) To calculate the activity of the stimulus $\boldsymbol{\xi}$ with respect to winner node s_1, it is necessary to use only part of the information in both vectors. The information used is the group number ζ, that defines the activity criteria. Then, $\mathbf{V}_\zeta$ is subdivided in l homogenous subgroups per joint, because the activity is calculated independently for each joint:

$$a_{k,s_1} = \exp\left(-||\boldsymbol{\xi}_{\zeta_k} - \mathbf{V}_{\zeta_k,s_1}||\right), 1 \leq k \leq l$$

6) If any activity is less than the established threshold for that group, ($\mathbf{a}_{1,s_1} < \bar{\mathbf{a}}_1$ OR $\ldots \mathbf{a}_{k,s_1} < \bar{\mathbf{a}}_k$), then a new node r must be added in the exact location of the input sample:
   a) Add the new node r to the $\mathcal{A}$ set: $\mathcal{A} = \mathcal{A} \cup \{r\}$.
   b) Create a new weight vector associated with node r, i.e., $\mathbf{w}_r = \boldsymbol{\xi}$.
   c) Remove the connection (s_1, s_2) from set $\mathcal{C}$.
   d) Calculate the distances:
   $\text{Dist} = \{\text{Dist}(r, s_1), \text{Dist}(r, s_2), \text{Dist}(s_1, s_2)\}$, where:
   $\text{Dist}(r, s_1) = ||\mathbf{V}_{\eta,r} - \mathbf{V}_{\eta,s_1}||$,
   $\text{Dist}(r, s_2) = ||\mathbf{V}_{\eta,r} - \mathbf{V}_{\eta,s_2}||$ and
   $\text{Dist}(s_1, s_2) = ||\mathbf{V}_{\eta,s_1} - \mathbf{V}_{\eta,s_2}||$.
   e) Select the 2 shortest distances Dist_1 and Dist_2 where:

$$\text{Dist}_1 = \text{arg min}(\text{Dist})$$
$$\text{Dist}_2 = \text{arg min}(\text{Dist} - \text{Dist}_1)$$

   f) Insert new connections between the nodes considered to determine Dist_1 and Dist_2.

$$C = C \cup \{c_{\text{Dist}_1,\text{Dist}_2}\}$$

7) If a new node was not inserted in step (6), update the positions of the winning node s_1:

$$\Delta\mathbf{w}_{s_1} = \rho \times (\boldsymbol{\xi} - \mathbf{w}_{s_1})$$

where:

$$\rho = \begin{cases} \epsilon_b \times \alpha_f^{(\sigma/\sigma_f)} & \sigma \leq \sigma_f \\ \epsilon_b \times \alpha_f & \sigma > \sigma_f \end{cases}$$

$0 < \epsilon_b < 1$ is the learning rate; $\alpha_f \approx 0$ is the final learning rate; σ is the counter of the number of times that a winner node has fired, and σ_f is the maximum number of times a node is supposed to fire.
8) Repeat from step (1) up to the maximum number of iteractions $t_{\max}$.

Pruning Phase: After the training phase, run the pruning phase for I iteractions to remove unsuitable nodes and links. The prunning phase will eliminate all unused links that do not disconnect the graph, all unused nodes and all isolated nodes.

1) Create a set $\mathcal{N} = \mathcal{A}$ of all nodes and a set $\mathcal{E} = \mathcal{C}$ for all links.
2) Repeat for I iteractions:
   - Genarate a data sample $p(\boldsymbol{\xi})$, according to MB procedure.
   - Exclude the winner node s_1 from the set $\mathcal{N} = \mathcal{N} - \{s_1\}$ and exclude the connection c_b between the best and the second best unit, from set $\mathcal{E} = \mathcal{E} - \{c_b\}$.
3) Repeat for all connections remaining in $\mathcal{E}$:
   - Check if a possible deletion of connection c_b between two nodes creates a disconnected graph.
   - If that is the case, do not delete the connection c_b. Otherwise, delete connection c_b permanently.
4) Eliminate all nodes remaining in set $\mathcal{N}$ (nodes that never won a competition).
5) Eliminate isolated nodes, i.e., disconnected nodes.

Trajectory Generation Phase: Given any initial point $\boldsymbol{\xi}_{\text{init}}$ and target point $\boldsymbol{\xi}_{\text{targ}}$ find the best matching node n_{init} for the initial point, and the best matching node n_{targ} for the target point.

Set an energy diffusion function [30] that defines a flux of energy diffusing through the links of the network such as $f(t, n)$, for all nodes n, aiming to find a chain of nodes $n_{S,S-1,\ldots,1,0}$, starting from the target point $n_{\text{targ}} = n_S$ to the initial point $n_{\text{init}} = n_0$, where S is the (unknow) size of the trajectory.

The procedure is the following:

1) Initialize the diffusion function $f(0, n) = 0, \forall n \neq n_{\text{targ}}$ and $f(0, n_{\text{targ}}) = 1$
2) Repeat for all $n \in \mathcal{A}$ until $f(t, n_{\text{init}}) \neq 0$

$$f(t+1, n) = \begin{cases} 1 & \forall t, \text{ if } n = n_{\text{targ}} \\ \mu \sum\limits_{j \in N_n} f(t, j) & \text{if } n \neq n_{\text{targ}} \end{cases}$$

where N_n is the set of all nodes that are neighbors of n and $|N_n|$ is its cardinality, and $\mu = K/|N_n|$, $K < 1$ such as $K = |N_n|/(|N_n| + 1)$.

This procedure diffuses an energy from target node to all other nodes until it reaches the initial node eventually (and necessarily [30] if there is a route). After the initial node received any amount of energy, the path can be followed from it to the target just choosing as the next point the neighbor with the maximum energy.

IV. STRAGEN Simulation and results

The STRAGEN was configured in the domain of robotics to employ the robot hand. The set of training samples $\mathbf{B}_{L\times D}$ contains $L = 1000$ points equally distributed in the interval of all possible points from a closed fist configuration

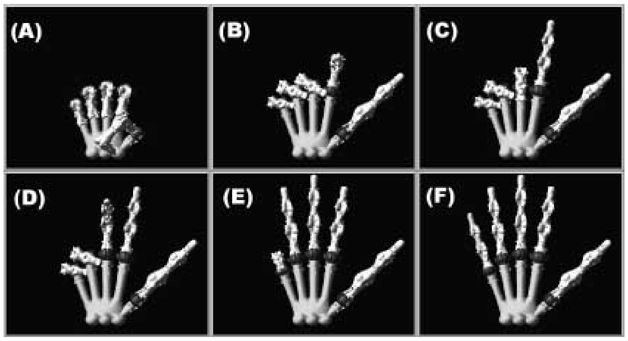

Fig. 2. Simulation 1: Trajectory from a closed fist to an open hand, generated by STRAGEN; 425 nodes; 360 trajectory points; $L = 1000$ samples.

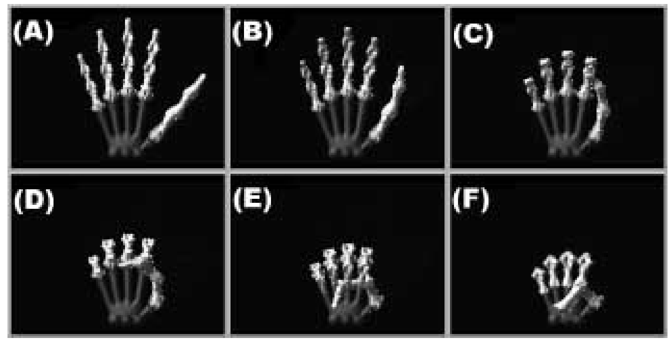

Fig. 3. Simulation 2: Trajectory from an open hand to a closed fist, generated by STRAGEN; 600 nodes; 597 trajectory points; $L = 2000$ samples.

to an open hand. Each weight vector is defined as the spatial position, the angle and the torque of each joint, with dimension $D = 55$, $\mathbf{w} \in \mathbb{R}^{55}$:

$$\begin{aligned} \mathbf{w}_{n_i} = [X_A Y_A Z_A X_B \dots Z_E \\ \theta_{A_1}\theta_{A_2}\theta_{A_3}\theta_{A_4}\theta_{B_1}\dots\theta_{E_4} \\ \tau_{A_1}\tau_{A_2}\tau_{A_3}\tau_{A_4}\tau_{B_1}\dots\tau_{E_4}]^T \end{aligned}$$

This weight vector has different types of information that are separated into $m = 3$ homogenous groups, subdivided by finger and joint groups:

$$\mathbf{V}_1 = \left[[X_A Y_A Z_A]_1^T \, [X_B Y_B Z_B]_2^T \dots [X_E Y_E Z_E]_5^T\right]^T$$
$$\mathbf{V}_2 = \left[[\theta_{A_1}\theta_{A_2}\theta_{A_3}\theta_{A_4}]_1^T \, [\theta_{B_1}\dots]_2^T \dots [\theta_{E_1}\dots]_5^T\right]$$
$$\mathbf{V}_3 = \left[[\tau_{A_1}\tau_{A_2}\tau_{A_3}\tau_{A_4}]_1^T \, [\tau_{B_1}\dots]_2^T \, [\tau_{E_1}\dots]_5^T\right]^T$$

where the dimensions are $D_1 = 15$ (and $D_{1,i} = 3, 1 \le i \le 5$), $D_2 = 20$ (and $D_{2,i} = 4, 1 \le i \le 5$), $D_3 = 20$ (and $D_{3,i} = 4, 1 \le i \le 5$), and $\sum_{j=1}^{m} D_j = D = 55$.

We used $\epsilon_b = 0.2$ as initial learning rate $\alpha_f = 0.1$ as the final learning rate, and $\sigma_f = 6$ as the maximum number of times a node will fire based on the number of samples in the database and the number of iterations $t_{\max}$, such as $\sigma_f = 2 \cdot t_{\max}/L$. The Motor Babbling candidates were defined as $Q = 10$.

The activity group vector used was $\zeta = 1$ and neighborhood criterion was also $\eta = 1$, the first group vector that defines the joint position coordinates. The percentage used to calculate the activity threshold $\bar{\mathbf{a}}$ was $P = 1\%$:

In this experiment the whole database was presented 3 times ($t_{\max} = 3000 = 3 \cdot L$) with 0.1% of noise in the samples. The number of iteractions for the prunning phase and the validation phase was also $I = 3000$.

After the training, it was proposed to STRAGEN a starting point, the close hand, and the target, an open hand, as show in Fig. 2.

STRAGEN generated a trajectory with 360 nodes in between, that was transferred to the robot hand to move according to it. The validation error, that measure the distance of the nodes and the real sample that it represents was $e = 0.046118$.

Another simulation with a new database of $L = 2000$ points and different finger positions was used to train the network. It was also decreased the percentage P of activity one node must present to prevent the network from creating a new node to $P = 0.3\%$. This made the network create more nodes and save more information that was learned from the database.

The same trajectory was tested in this second simulation, as shown in Fig. 3. The first simulation was trained with samples that presented the independent closing of all fingers. In the second simulation, the database contains samples of finger configurations closing in parallel. This allowed the network to choose a faster movement and close all the fingers simultaneously.

Because of the decrease in the percentage of activity, the generated trajectory of the second simulation had more points, precisely 597 positions. This reflected the more dense network created, with 600 nodes in the second simulation, and only 425 nodes in the first ones. This simple parameter P, measured in percentage, can be modified at will to choose the best precision for a given robot and problem space. The validation error for this simulation decreased to $e = 0.008248$.

After training, the network represented very well all joint angles, with error in each joint less than 0.0002%. The finger positions of each learned configuration had a maximum variation less than 5.93178%.

This trajectories were generated using the minimum cartesian distance between the previous and the next position of the end of each finger, represented by group $\eta = 1$, $\mathbf{V}_1 = \left[[X_A Y_A Z_A]_1^T \, [X_B Y_B Z_B]_2^T \dots [X_E Y_E Z_E]_5^T\right]^T$.

The diffusion energy function spread energy to all nodes, starting at the target node. After this process, to generate a trajectory is sufficient to go from the starting point to the next neighbor following the nodes with greater energy. The process of choosing the nodes for a trajectory is simple as that, because the main problem is the creation of a topological map that has neighbors set according to the chosen criterion. This map is created during the training phase, and after trained, the network allows instantaneous access to all information.

V. CONCLUSION

The Kanguera Hand presented above deals with the pointed out nonlinear behavior of the transmission using a new concept of transmission, where a cam is used instead of a traditional pulley.

These cams are able to provide a linear cable displacement between the motor and the joint finger by removing the undesired dead zone and hysteresis of the movement. As a result, the mechanical modifications introduced by the cams implies in a better positioning accuracy.

A new model of neural network was also presented and tested. This model can learn heterogeneous information and use such information as criteria to create new trajectories. The criterion tested was the minimum cartesian distance of the end effector, i.e., (x, y, z) positions of each finger. The model that implements this concept is called STRAGEN.

Two examples involving finding an optimum trajectory to a robot hand in a 3D space is given. The generated trajectory consisted in the minimum distance between two given points. The model is also capable of generating trajectories that optimize different criteria, such as minimum variation of joint angles or minimum variation of torques, despite it was not presented here.

It is shown that, for a given criterion, STRAGEN creates a suitable topology keeping the information of the nodes useful for the domain in question, avoiding to deform these information by the use of similar groups in the nodes. STRAGEN achieves it by comparing the Euclidian distances of the weight vector by groups of homogenous parts.

STRAGEN is concerned to choose, by the neighborhood criterion, the shortest way to connect a new node, when inserting it near the best and the second best matching node.

The Motor Babbling procedure allied with neighborhood criteria based on similar group of information improves STRAGEN's representation map.

REFERENCES

[1] E. L. Damian, D. Sidobre, and R. Alami, "A grasp planner based on inertial properties," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, 2005.

[2] K. Yamazaki, M. Tomono, T. Tsubouchi, and S. Yuta, "A grasp planning for picking up an unknown object for a mobile manipulator," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, 2006.

[3] T. Mouri, H. Kawasaki, K. Yoshikawa, J. Takai, and S. Ito, "Anthropomorphic robot hand: Gifu hand iii," *Proc. of Int. Conf. ICCAS*, pp. 1288–1293, 2002.

[4] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS*, 2003.

[5] A. Hourtash, "The cx-space: A unified paradigm for grasping using multifingered hands," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, 2006.

[6] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, 2003.

[7] R. Prado and R. Suárez, "Heuristic grasp planning with three frictional contacts on two or three faces of a polyhedron," *Proc. of the IEEE Intl. Assembly and Task Planning - ISATP*, pp. 112– 118, 2005.

[8] G. A. P. Caurin, A. R. L. Albuquerque, and A. L. A. Mirandola, "Manipulation strategy for an anthropomorphic robotic hand," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS*, 2004.

[9] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa, "Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, pp. 181–187, 2006.

[10] R. P. JR., A. H. Fagg, and R. A. Grupen, "Manipulation gaits: sequences of grasp control tasks," *Proc. of the IEEE Intl. Conf. on Robotics and Automation - ICRA*, vol. 1, pp. 801– 806, 2004.

[11] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[12] S. C. Jacobsen, J. E. Wood, D. F. Knutti, and K. B. Biggers, "The utah/mit dexterous hand: Work in progress," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 21–50, 1984.

[13] B. M. Jau, "Dexterous telemanipulation with four fingered handsystem," *Proceedings of IEEE Robootics and Automation*, pp. 338–343, 1995.

[14] A. R. L. Albuquerque, C. Policastro, L. M. Pedro, and G. A. P. Caurin, "The dynamic coupling identification between the artificial robotic hand and its acuators using genetic algorithms," *Proc. of the Int. Sym. on Dynamic Problems of Mechanics, XII DINAME*, 2007.

[15] R. V. Aroca, D. T. Matsuo, and G. A. de Paula Caurin, "Controling the scara robot with real time linux," 2007, submitted to AIM.

[16] C. M. O. Valente, A. F. R. Araújo, G. A. P. Caurin, and A. Schammass, "A neural gripper for arbitrary object grasping," *Journal of Connection Science*, vol. 11, no. 3, pp. 219–316, 1999.

[17] G. A. P. Caurin, P. M. Aguiar, and C. M. O. Valente, "Planning and control system for an anthropomorphic grippers," *IEEE/RSJ Int. Conf. on Intel. Robots and Systems - IROS*, vol. 1, pp. 340–349, 2002.

[18] A. R. L. Albuquerque, G. A. P. Caurin, and A. L. A. Mirandola, "Dynamic modelling of the brahma - brazilian anthropomorphic hand," *In.International Symposium on Multibody Systems and Mechatronics*, 2005.

[19] K. J. Kyriakopoulos, A. Zink, and H. E. Stephanou, "Kinematic analysis and position/force control of the anthrobot dextrous hand," *Trans. On System, Man, and Cybernetics-Part B: cybernetics*, vol. 27, no. 1, pp. 95–104, 1997.

[20] G. A. Bekey, R. Tomovic, and I. Zeljkovic, "Control architecture for the belgrade/usc hand," pp. 136–149, 1990.

[21] J. Butterfass, M. Grebenstein, H. Liu, , and G. Hirzinger, "Dlr-hand ii: Next generation of a dextrous robot hand," *Proc. of IEEE Int. Conf. on Robo. and Automa.*, pp. 109–114, 2001.

[22] A. Namiki, Y. Imai, M. Ishikawa, and M. Kanneko, "Development of a high-speed multifingered hand system and its application to catching," *Proc. of the 2003 IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, pp. 2666–2671, 2003.

[23] I. Yamano, K. Takemura, , and T. Maeno, "Development of a robot finger for five-fingered hand using ultrasonic motors," *Proc. of the 2003 IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, pp. 2648–2653, 2003.

[24] "Patent p.i. 01601422-4 articulação de dois graus de liberdade. 06/04/2006."

[25] B. Fritzke, "Growing grid - a self-organizing network with constant neighborhood range and adaptation strength," *Neural Processing Letters*, vol. 2, no. 5, pp. 9–13, 1995.

[26] ——, "Unsupervised ontogenic networks," *Handbook of neural computation*, vol. 1, pp. 1–16, 1997.

[27] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Networks*, vol. 15, no. 8-9, pp. 1041–1058, 2002.

[28] J. Fiala, "A network for learning kinematics with application to human reaching models," in *Proceedings of the 1994 WCCI conference*, Orlando, Florida, 1994.

[29] Angeline, Saunders, and Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks*, 1994.

[30] M. Zeller, R. Sharma, and K. Schulten, "Vision-based robot motion planning using a topology representing neural network," in *Applications of Neural Adaptive Control Technology*, J. Kalkkuhl, K. Hunt, R. Zbikowski, and A. Dzielinski, Eds. World Scientific Publishing, 1997.