Efficient and practical determination of grasping configurations for anthropomorphic hands ⋆

Josep-Arnau Claret * Raúl Suárez *

* *Institute of Industrial and Control Engineering (IOC)*
Technical University of Catalonia (UPC), Barcelona, Spain
(e-mail: josep.arnau.claret@estudiant.upc.edu, raul.suarez@upc.edu).

Abstract: The paper presents a methodology to rapidly solve the inverse kinematics of anthropomorphic hands, which is particularized for a mechanical hand considering 27 degrees of freedom. Given the contact points and normal directions on an object surface, the proposed algorithm finds the joint values and the wrist position and orientation that make the fingertips satisfy the contact constraints. The approach combines an iterative algorithm with an off-line analysis that allows significant reductions of the execution time. The approach has been implemented and the paper includes application examples. The effectiveness and fast execution of the algorithm is demonstrated with statistical results.

Keywords: robotics, robot kinematics, inverse kinematic problem, robotic manipulators

1. INTRODUCTION

Robotics is a technology applied in different scenarios: medical assistance, industry, space exploration, among several others. A robot can also have different objectives in a certain environment like for instance inspection, location, transporting or manipulation. These wide range of applications involve a very large number of robotic actions that need to physically interact with the environment and, in particular, need to grasp and manipulate different objects. Robot hands, as their versatility is very high, are one of the most adaptable tools for grasping objects. The advances in the developments of robot hands are significant (Bicchi, 2000), but they have some associated problems that need better solutions than the current existing ones. One of these problems is the grasp planning, where the first decision is the selection of the desired type of grasping (Cutkosky, 1989): power grasp, closing the hand around the object without knowing the final contact points between the hand and the object; or precision grasp, where the contact points are known on the object and take place only on the hand fingertips. Many works were focused on finding appropriate contact points on the object (e.g. for 2D objects: (Nguyen, 1988) (Park and Starr, 1990) (Liu, 1998) (Cornellá and Suárez, 2009), and for 3D objects: (Ponce et al., 1997) (Borst et al., 1999) (Li et al., 1989) (Pollard, 2004) (Roa and Suárez, 2009)), but there are not general formulations to solve precision grasp including the kinematics constraints of a given hand. Solving the hand inverse kinematic is an interesting problem, that is, the search of an appropriate set of joint values of a robot hand that satisfies the constraints imposed by some contact points (Rosell et al., 2005) (Rosales et al., 2011) (Suárez and Claret, 2009). The main difficulty of this problem is to quickly find a valid hand configuration in the very high dimensional space defined by the hand joints. The approach proposed in this work uses the hand Jacobian to iteratively find hand configurations closer to the desired constraints imposed by the contact points on the objects, together with a statistical study to select initial hand configurations that speeds up the iterative procedure.

2. PROBLEM DESCRIPTION

The objective of this work is to find a reachable hand configuration that satisfies the constraints imposed by the desired contact points on the object using the fingertips, i.e. the fingertips must be properly located and oriented, the finger joints must be within the corresponding range, and there must be no collision among the hand elements (palm and fingers). Checking for collisions between the hand and the environment is outside the scope of this work.

The mechanical hand used in this work is the Schunk Anthropomorphic Hand (SAH) shown in Fig. 1. This hand is anthropomorphic and has four fingers (thumb, index, medium and ring fingers). Each finger has four joints (Fig. 2a): two independent (joints 1 and 2, as abduction and flexion, respectively) and two coupled (joints 3 and 4, both flexion), which makes three independent degrees of freedom (DoF). The thumb has an extra joint in the base (joint 0). Then, the total number of DoF of the hand is 19, 13 from the fingers plus 6 from the hand wrist movements.

The contact points on the hand must be on the proper region of each fingertip. Each fingertip is considered spherical, and the accepted contact regions are shown in Fig. 2b. Two parameters are needed to identify a contact point on each fingertip, meaning the existence of 2 additional virtual DoF per finger. Then, the total number of DoF of the hand system is 27, 19 from the finger joints and the wrist plus 8 from the fingertips.

⋆ This work was partially supported by the Spanish Government through the projects DPI2010-15446 and DPI2008-02448

Fig. 1. Mechanical hand SAH assembled on an industrial robot arm.

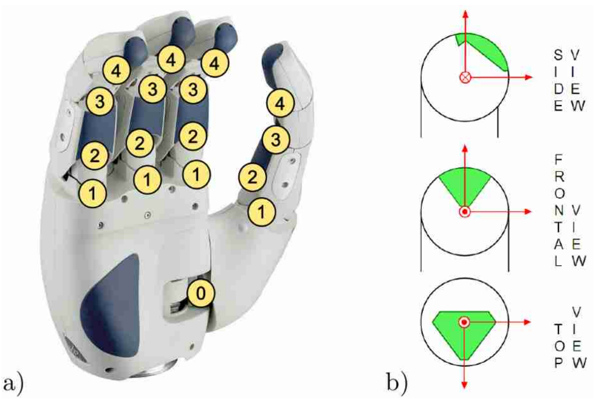


Fig. 2. a) Hand and finger joints; b) Contact region on the fingertip surface.

A contact between a fingertip and a point on the object boundary imposes 5 constraints: a fingertip point must coincide with the point on the object (3 parameters) and the normal to the fingertip must coincide with the surface normal at the object contact point (2 additional parameters).

The assignment between each finger and its contact point on the object is assumed to be known; if this is not the case, all the possible combinations should be checked until a solution is found or no solution at all can be determined.

3. PROPOSED SOLUTION

The proposed approach has two parts, one is based on a traditional iterative algorithm that, starting from a given initial hand configuration, uses the hand Jacobian to determine virtual movements of the hand that try to satisfy the contact constraints on each fingertip. The other part is an off-line study to determine a sequence of hand configurations that works well as initial configurations in the iterative algorithm. These two parts are described in detail in the following two subsections.

3.1 Iterative Algorithm

The iterative algorithm has to be executed each time it is necessary to determine a hand configuration satisfying some contact constraints on the object surface. The algorithm has two loops, the first (outer) loop is devoted to change the initial configuration of the hand if no solution is found with the current selected one, and the second (inner) loop is devoted to the search of hand virtual movements that iteratively change the hand configuration from the initial to a final one satisfying the contact constraints. The iterative algorithm is formally described as follows. Let:

- i_{max} be the maximum number of initial configurations.
- k_{max} be the maximum number of jacobian iterations.
- $\mathbf{C}_k$ be the hand configuration in the k-th jacobian iteration.
- $\mathbf{P}_k$ be the contact constraints on the fingertips in iteration k.
- $\mathbf{P}^*$ be the contact constraints on the object, i.e. the desired final contact constraints for the fingertips.
- the subindices I, M, R and T indicate the fingers index, middle, ring and thumb, respectively.
- $\mathbf{p}_i$, $i \in \{I, M, R, T\}$, be the position of the contact point on finger i.
- $\mathbf{r}_i$, $i \in \{I, M, R, T\}$, be the position of the center of the fingertip i.

Algorithm 1 Iterative Algorithm

1: **for** $i = 1$ **to** i_{max} **do**
2: $\quad(\mathbf{C}_1, \mathbf{P}_1) \leftarrow$ Obtain_Initial_Conf($i, \mathbf{P}^*$)
3: $\quad k \leftarrow 1$
4: $\quad flag\_collision \leftarrow$ FALSE
5: $\quad flag\_progress \leftarrow$ TRUE
6: $\quad$**while** $k \leq k_{max}$ **and** $flag\_collision =$ FALSE **and** $flag\_progress =$ TRUE **do**
7: $\qquad$**if** $\mathbf{P}_k \simeq \mathbf{P}^*$ **then**
8: $\qquad\quad$**if** Check_Collisions($\mathbf{C}_k$) = FALSE **then**
9: $\qquad\qquad$**return** ($\mathbf{C}_k$)
10: $\qquad\quad$**else**
11: $\qquad\qquad flag\_collision \leftarrow$ TRUE
12: $\qquad\quad$**end if**
13: $\qquad$**else if** $k \geq n$ **and** $\mathbf{P}_{k-n+1} \simeq \ldots \simeq \mathbf{P}_k$ **then**
14: $\qquad\quad flag\_progress \leftarrow$ FALSE
15: $\qquad$**end if**
16: $\qquad(\mathbf{C}_{k+1}, \mathbf{P}_{k+1}) \leftarrow$ Comp_Nxt_C($\mathbf{C}_k, \mathbf{P}_k, \mathbf{P}^*$)
17: $\qquad k \leftarrow k + 1$
18: $\quad$**end while**
19: **end for**
20: **return** ("No solution")

The main functions and steps of Algorithm 1 are the following.

Function "Obtain_Initial_Conf($i, \mathbf{P}^*$)" (Step 2)

This function returns an initial hand configuration based on the number $i - 1$ of initial configurations already used and the given desired contact constraint $\mathbf{P}^*$.

The reference system used to describe $\mathbf{p}_i$ and $\mathbf{r}_i$ has the origin at the thumb contact point, $\mathbf{p}_T$, and the z-axis is the vector of $\frac{\mathbf{r}_T - \mathbf{p}_T}{|\mathbf{r}_T - \mathbf{p}_T|}$; the x- and y-axis are chosen randomly to complete an orthonormal basis.

The set $(\mathbf{p}_i, \mathbf{r}_i)$ fully defines the contact constraints of finger i, i.e. the contact point position on the fingertip and the direction normal to the fingertip at the contact point. Note that knowing $\mathbf{p}_i$ only two parameters are needed to determine $\mathbf{r}_i$, since $\|\mathbf{p}_i - \mathbf{r}_i\|$ is a constant distance (the radius of the spherical fingertip).

Now, the set of contact constraints on the fingertips can be generically expressed, for a hand configuration $\mathbf{C}_k$, as:

$$\mathbf{P}_k = (\mathbf{p}_I, \mathbf{r}_I, \mathbf{p}_M, \mathbf{r}_M, \mathbf{p}_R, \mathbf{r}_R), \tag{1}$$

where for simplicity the subindex k is not included in each component of $\mathbf{P}_k$; analogously, using the supraindex '*' to indicate desired values, the desired contact constraints on the fingertips are:

$$\mathbf{P}^* = (\mathbf{p}_I^*, \mathbf{r}_I^*, \mathbf{p}_M^*, \mathbf{r}_M^*, \mathbf{p}_R^*, \mathbf{r}_R^*). \tag{2}$$

Note that the thumb contact constraints are not included in $\mathbf{P}_k$ (i.e. $\mathbf{p}_T$ and $\mathbf{r}_T$) nor in $\mathbf{P}^*$ (i.e. $\mathbf{p}_T^*$ and $\mathbf{r}_T^*$); this is because the initial hand configuration is chosen such that the thumb contact always satisfies its corresponding contact constraints, as it is explained immediately below.

The 27 DoF fixing an initial hand configuration $\mathbf{C}_1$ are computed as follows:

(1) The hand is initially positioned such that the constraints imposed by the thumb contact are satisfied. This is done by imposing the conditions:

$$\mathbf{p}_T = \mathbf{p}_T^*, \qquad \mathbf{r}_T = \mathbf{r}_T^*. \tag{3}$$

This is equivalent to five independent constraints, so there are still 27-5=22 DoF to be fixed in order to determine completely the hand configuration.

(2) The joint values of the hand (both the mechanical and the virtual DoF detailed in Section 2) are fixed following a predetermine sequence of hand poses $\mathbf{Q}$ given by the 25-dimensional vector

$$\begin{aligned}\mathbf{Q} = (&\phi_{T0}, \ldots, \phi_{T6}, \phi_{I1}, \ldots, \phi_{I6},\\ &\phi_{M1}, \ldots, \phi_{M6}, \phi_{R1}, \ldots, \phi_{R6}),\end{aligned} \tag{4}$$

where for each component the first subindex identifies the finger and the second subindex identifies the finger joint. In the latest case, values 0 to 4 identify the finger real joints (Fig. 2a), and values 5 and 6 identify the two virtual joints defining the contact point on each fingertip (Fig. 2b); note that subindex 0 exists only for the thumb, according to the hand structure described in Section 2.

The generation of a proper sequence of poses $\mathbf{Q}_i$, $i = 1, ..., i_{max}$ required to generate the i_{max} hand configurations for Algorithm 1 is one of the key points of this work and is detailed below in Section 3.2.

Each hand pose fixes the 25 joint values, but since the 3rd and 4th joint of each finger are coupled (Fig. 2a) there are 4 joints (one per finger) that are not independent and therefore only 25-4=21 DoF are actually fixed. Then, there is 22-21=1 DoF left to define completely the hand configuration.

(3) The remaining degree of freedom corresponds to the rotation, ψ, of the hand around the direction normal to thumb contact point (i.e. normal to $\mathbf{p}_T - \mathbf{r}_T$), and it is fixed locating the hand such that the index and ring fingers are well oriented with respect to their expected final positions, which is done as follows. Let Π_T be the plane orthogonal to $\mathbf{p}_T - \mathbf{r}_T$ containing the thumb contact point $\mathbf{p}_T$, and let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the projections on Π_T of the vectors $\mathbf{p}_R^* - \mathbf{p}_I^*$ and $\mathbf{p}_R - \mathbf{p}_I$ on Π_T respectively. Now, ψ is selected such it minimizes the angle between $\mathbf{v}_1$ and $\mathbf{v}_2$.

In this way all the 27 DoF of the hand are fixed.

Then, the initial hand configuration can be written as

$$\mathbf{C}_1 = (\mathbf{p}_T, \mathbf{r}_T, \mathbf{C'}_1) \tag{5}$$

where

$$\begin{aligned}\mathbf{C'}_1 = (&\psi, \phi_{T0}, \ldots, \phi_{T6}, \phi_{I1}, \ldots, \phi_{I6},\\ &\phi_{M1}, \ldots, \phi_{M6}, \phi_{R1}, \ldots, \phi_{R6}).\end{aligned} \tag{6}$$

$\mathbf{C}_1$ is a vector with 32 elements but only 27 of them are independent, representing the 27 DoF of the hand in the workspace, and $\mathbf{C'}_1$ has 26 elements with only 22 of them being independent. Note that $\mathbf{P}_1$ can be computed directly by solving the direct kinematics of the hand at $\mathbf{C}_1$.

Function "Check_Collisions($\mathbf{C}_k$)" (Step 8)

This function checks if there are collisions between the elements of the hand (fingers and palm) for the hand configuration $\mathbf{C}_k$, returning TRUE if so or FALSE otherwise.

Function "Comp_Nxt_C($\mathbf{C}_k, \mathbf{P}_k, \mathbf{P}^*$)" (Step 16)

This function computes a new hand configuration from the current one, $\mathbf{C}_k$, and the desired contact constraints $\mathbf{P}^*$. Once $\mathbf{p}_T$ and $\mathbf{r}_T$ are given (note that they are constant $\forall \mathbf{C}_k$), $\mathbf{C'}_k$ has all the information needed to fully know the hand configuration, and since it has a smaller dimension it will be used to compute the next hand configuration in the iterative algorithm. Let:

- $\Delta\mathbf{P'} = \alpha(\mathbf{P}^{*'} - \mathbf{P}_k^{'})$, where $\mathbf{P}^{*'} = (\mathbf{P}^*, 0, 0, 0, 0)$, $\mathbf{P}_k^{'} = (\mathbf{P}_k, 0, 0, 0, 0)$ (the reason for adding these zeros will become evident below), and α is a constant value empirically determined to obtain a good convergence of the algorithm. Note that $\mathbf{P}^{*'}$ and $\mathbf{P}_k^{'}$ are extended vectors of dimension 22, and so is $\Delta\mathbf{P}$.
- $\Delta\mathbf{C'} = \mathbf{C'}_k - \mathbf{C'}_{k-1}$. $\mathbf{C'}_k$ and $\mathbf{C'}_{k-1}$ are vectors of dimension 26, and so is $\Delta\mathbf{C'}$.
- J be the hand Augmented Jacobian (Siciliano and Khatib, 2008) that is obtained by adding to the standard hand Jacobian four additional rows that include the coupling constraints between joints 3 and 4, i.e. $\phi_{i3} = \phi_{i4}$ with $i = \{T, I, M, R\}$. Each of these rows is of the type $(0, ..., 1, -1, ..., 0)$, i.e. a row of zeros with the exception of the positions corresponding to the joints ϕ_{i3} and ϕ_{i4} in $\mathbf{C'}_k$ for each of the four fingers. Note that J is a matrix with dimension 22×26.

The effect of the rows added in J and the zeros added in $\Delta\mathbf{P'}$ makes that in the relation $\Delta\mathbf{P'} = J\Delta\mathbf{C'}$ the elements 19 to 22 become $0 = \phi_{i3} - \phi_{i4}$ for each i, i.e. $\phi_{i3} = \phi_{i4}$.

Now, $\Delta\mathbf{C'}$ can be approximated as,

$$\Delta\mathbf{C'} = J^+ \Delta\mathbf{P'}, \tag{7}$$

where J^+ is a pseudoinverse of J.

Then, the next configuration in the iterative procedure is simply computed as,

$$\mathbf{C'}_{k+1} = \mathbf{C'_k} + \Delta\mathbf{C'}, \tag{8}$$

and $\mathbf{C}_{k+1}$ is obtained from $\mathbf{C'}_{k+1}$, $\mathbf{p}_T$ and $\mathbf{r}_T$; finally $\mathbf{P}_{k+1}$ is computed from $\mathbf{C}_{k+1}$ using the hand direct kinematics.

Ending conditions (Steps 1, 6, 7 and 13)

One of the following ending conditions must be satisfied to finish the iterative search algorithm:

(1) i_{max} initial configurations $\mathbf{C}_1$ have been tested without finding a solution, i.e. $i = i_{max}$ (Step 1).
(2) The hand configuration satisfies the desired contact point constraints on the object surface (Step 7), i.e. $\mathbf{P}_k \cong \mathbf{P}^*$, with a hand configuration without self-collisions (Step 8). This is checked verifying that in the iteration k the following conditions are satisfied for $i \in \{I, M, R\}$,

$$\|\mathbf{p}_i^* - \mathbf{p}_i\| < d_{min}, \qquad \|\mathbf{r}_i^* - \mathbf{r}_i\| < d_{min}, \tag{9}$$

where d_{min} is a predefined constant parameter;

and one of the following ending conditions must be satisfied to exit the inner loop of the iterative search algorithm:

(1) A number k_{max} of iterations have been computed without finding a solution (Step 6).
(2) The hand configuration does not progress enough during a predefined number n of consecutive iterations. This is checked verifying that the following condition is satisfied during n consecutive iterations of k for $i \in \{I, M, R\}$ (Step 13):

$$\sum_1^4 (\|\mathbf{p}_{i_k} - \mathbf{p}_{i_{k-1}}\| + \|\mathbf{r}_{i_k} - \mathbf{r}_{i_{k-1}}\|) < s_{min}, \tag{10}$$

where s_{min} is a predefined constant parameter.

3.2 Determination of the Initial Configurations Sequence

As mentioned in Subsection 3.1, a key point of the approach is the determination of a proper sequence of initial hand configuration in Step 2 of Algorithm 1 (function "Obtain_Initial_Conf($i, \mathbf{P}^*$)"). A sequence of initial configurations $\mathbf{C}$ is equivalent to a sequence of initial hand poses $\mathbf{Q}$, which is determined as follows.

A large enough set $\mathcal{S}$ of hand poses samples are randomly generated and a Principal Component Analysis (PCA) (Jolliffe, 2002) is used to find the direction in the hand working space with larger dispersion of samples. This is done by computing the eigenvalue decomposition of the covariance matrix of the samples (after a mean centering each data attribute) and selecting the eigenvector corresponding to the largest eigenvalue. Repeating this procedure, a new base of the hand workspace space is obtained, with the vectors in this base ordered according to a decreasing dispersion along each direction. Taking the first n vectors of this base, it is possible to define a subspace that approximates the hand workspace with a more tractable lower dimension n. This procedure is very often used to reduce the dimension of multidimensional data sets, and was already used to reduce the hand workspace in works dealing with grasp searching (Santello et al., 1998) (Tsoli and Jenkins, 2007) (Ciocarlie and Allen, 2009) (where the set of sampled is composed of grasping poses, and the directions of the base are called *eigengrasps*), with the synthesis of human-like motions in graphic applications (Safonova et al., 2004), and with motion planning of a hand-arm system (Rosell et al., 2007) (where the set of samples is obtained by mapping poses of the operator hand during unconstrained movements, and the directions of the base are called *principal motion directions*).

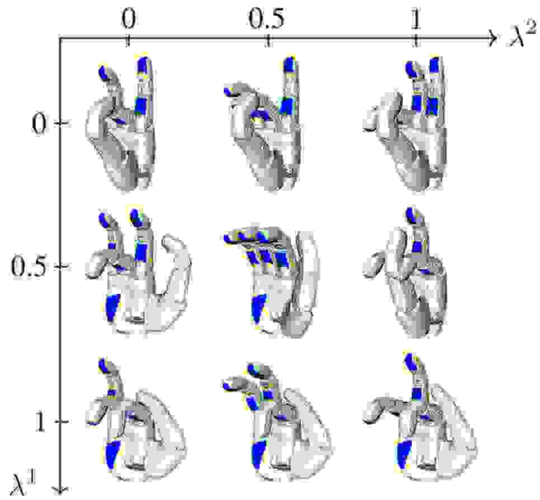


Fig. 3. Initial hand poses.

In this work the first two vectors of the base of the hand workspace obtained with the PCA described above were selected to generate the initial hand configurations in Algorithm 1. Each of these two vectors indicates a direction in the hand workspace that corresponds to a coordinated motion of all the joints in a single DoF. Then, with only two parameters, λ_i^1 and λ_i^2, it is possible to vary these two DoF and determine the initial hand pose $\mathbf{Q}_i$ in a 2-dimensional space, i.e. the dimension of the subspace where $\mathbf{Q}$ is determined is reduced from 21 to 2, trying to cover as much as possible of the hand workspace. Let:

- $\mathbf{m}$ be the mean of the set of samples $\mathbf{Q}_i \in \mathcal{S}$.
- $\mathbf{c}_j$, $j = 1, 2$, be the unitary vectors along the selected directions of the hand workspace.
- σ_j, $j = 1, 2$, be the standard deviation of the set of samples $\mathcal{S}$ along $\mathbf{c}_j$.
- λ_i^1, λ_i^2 be two real values between 0 and 1.

Then, the i-th initial hand configuration $\mathbf{Q}_i$ in the sequence is computed as the following function of the values λ_i^1 and λ_i^2 as:

$$\mathbf{Q}_i = \mathbf{m} + 3\,\sigma_1\,(2\lambda_i^1 - 1)\,\mathbf{c}_1 + 3\,\sigma_2\,(2\lambda_i^2 - 1)\,\mathbf{c}_2. \tag{11}$$

Note that for $\lambda_i^1 = \lambda_i^2 = 0.5$ results $\mathbf{Q}_i = \mathbf{m}$, and that $\lambda_i^1, \lambda_i^2 \in \{0, 1\}$ produce extreme poses at $\pm 3\sigma_i$ in the considered 2-dimensional subspace determined by $\mathbf{c}_1$ and $\mathbf{c}_2$ (this covers 99% of the dispersion of $\mathcal{S}$ on the 2-dimensional subspace). Some initial hand poses are shown in Fig. 3 for different values of λ_i^1 and λ_i^2.

Thus, determining a sequence of initial poses is equivalent to determine a sequence SEQ of $\lambda_i = (\lambda_i^1, \lambda_i^2)$, which can be done off-line for a particular hand using Monte Carlo simulations to look for a sequence that allows a good performance of Algorithm 1. A sequence SEQ of i_{max} poses is determined as follows.

(1) Discretize the domains of λ_i^1 and λ_i^2 into a finite and uniformly distributed set of N values $\lambda_{i_j}^1$ and $\lambda_{i_k}^2$, respectively (this generates N^2 potential duplas $\lambda_{i_{jk}} = (\lambda_{i_j}^1, \lambda_{i_k}^2)$, with $j, k = 1, ..., N$).
(2) $SEQ = \emptyset$

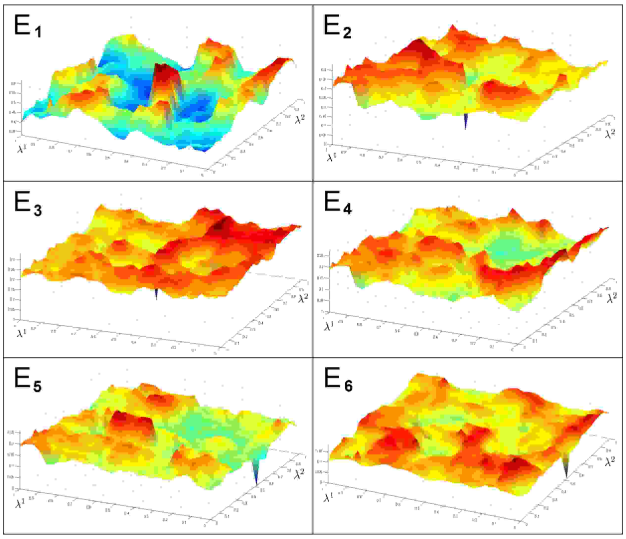


Fig. 4. $E_{i_{jk}}, i = \{1, ..., 6\}$ for the SAH hand (the dark red color corresponds to the highest values).

(3) For $i = 1$ to i_{max} do:
   (a) Generate a large enough set $\mathcal{P}$ of M random contact constraints $\mathbf{P}_l^*$ that cannot be solved with any $\lambda_{h_{jk}} \in SEQ$, i.e. the already selected duplas $\lambda_{h_{jk}}$ $\forall h < i$ (this means that no solution was found for $\forall \mathbf{P}_l^* \in \mathcal{P}$ using the values of $\lambda_{h_{jk}}$ already included in *SEQ*).
   (b) For each $\lambda_{i_{jk}}, j, k \in \{1, ..., N\}$ do:
      (i) Obtain the initial hand configuration $\mathbf{C}_{i_{jk}}$ using $\lambda_{i_{jk}}$ in Eq. (11) and the function Obtain_Initial_Conf$(i, \mathbf{P}^*)$ (Section 3.1).
      (ii) Use Algorithm 1 with the only initial configuration $\mathbf{C}_{i_{jk}}$ to look for a hand configuration satisfying each constraint $\mathbf{P}_l^* \in \mathcal{P}$ and save the rate of success $E_{1_{jk}}$, defined as the percentage of the M constraints $\mathbf{P}_l^*$ that were solved using $\mathbf{C}_{i_{jk}}$.
   (c) Select as the i-th element of the sequence the value $\lambda_{i_{jk}}$ with associated highest rate of success, i.e. add $\lambda_{i_{\alpha\beta}}$ to *SEQ*, with $\lambda_{i_{\alpha\beta}}$ such that $E_{i_{\alpha\beta}} \geq E_{i_{jk}}$ $\forall j, k$.
(4) Return(*SEQ*)

Fig. 4 shows the values of $E_{1_{jk}}$ to $E_{6_{jk}}$ obtained in the selection of λ_1 to λ_6 for the SAH hand with $M = 5000$ and $N = 30$.

4. EXAMPLES AND PERFORMANCE

4.1 Examples

The approach has been implemented in C++. The parameter values used in Algorithm 1 were $i_{max} = 12$, $k_{max} = 100$, $d_{min} = 1.5$ mm and $s_{min} = 1$ mm. A sequence of 12 initial hand poses has been computed using parallel computing on fourteen regular PC CPUs using the MPI library (MPI, 2010). The obtained values of λ_1 to λ_{12} are shown in Table 1, and Fig. 5 shows the six first initial configurations obtained with λ_1 to λ_6.

A first application example is shown in Fig. 6. It is the grasp of a fork that needed 1 initial configuration and 6 jacobian iterations to solve that particular inverse kinematic problem.

i	λ_i^1	λ_i^2	i	λ_i^1	λ_i^2	i	λ_i^1	λ_i^2
1	0.517	0.483	5	0.690	0.448	9	0.690	0
2	0.793	0.552	6	0.138	0.241	10	0.690	0.448
3	0.310	0.690	7	0.034	0.827	11	0.724	0.276
4	0	0.517	8	0.655	0.241	12	0.276	0.827

Table 1. Values of $\lambda_i = (\lambda_i^1, \lambda_i^2), i = \{1, ..., 12\}$.

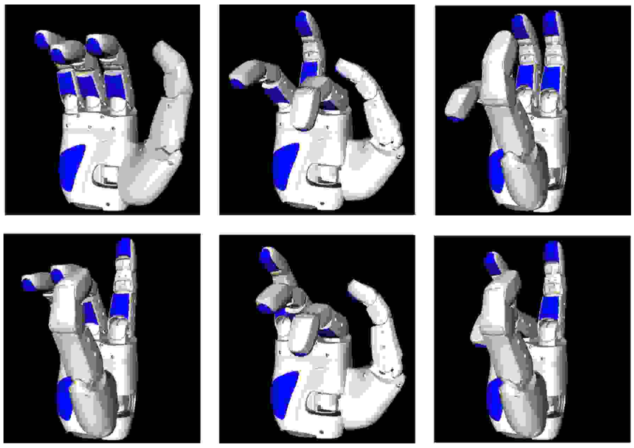


Fig. 5. First six initial hand configurations (ordered from left to right and from top to bottom).

Note that the evolution of the hand configuration is very significant in the first iterations, while in the last iterations the hand configurations are rather similar (Fig. 6b). Note also that in the second of the intermediate hand configurations there are collisions between the ring and middle fingers, this is not a problem at all since the only configuration that must not have collisions is the final one.

Another three examples are illustrated in Fig. 7, showing two views of the object with the contact constraints and two views of the final grasp configuration satisfying the constraints. The solution for the cup in Fig. 7a needed 1 initial configuration and 13 jacobian iterations, for the statuette of buddha in Fig. 7b needed 3 initial configurations and 25 jacobian iterations, and for the cup of coffee in Fig. 7c, 7 initial configurations and 38 jacobian iterations.

A final example with a can is shown in Fig. 8, with the contact constraints in Fig. 8a (side and top views) and the final configuration in Fig. 8b, where it can be seen the resulting grasp of the SAH on a can in the IOC Robotics Lab with the joint values of the mechanical hand set at the solution configuration. The solution for the can needed 1 initial configuration and 22 jacobian iterations.

4.2 Performance

The success percentage and the average execution time per initial configuration generated has ben calculated (in a AMD Athlon 64 X2 Dual Core Processor 5400+ at 2800 MHz). The results can be seen in Fig. 9a. See that with one initial configuration around the 62% of all inverse kinematics problems are solved; with two, around the 80%; until a 97%, using twelve (of course, a higher percentage could be solved by generating more than twelve initial

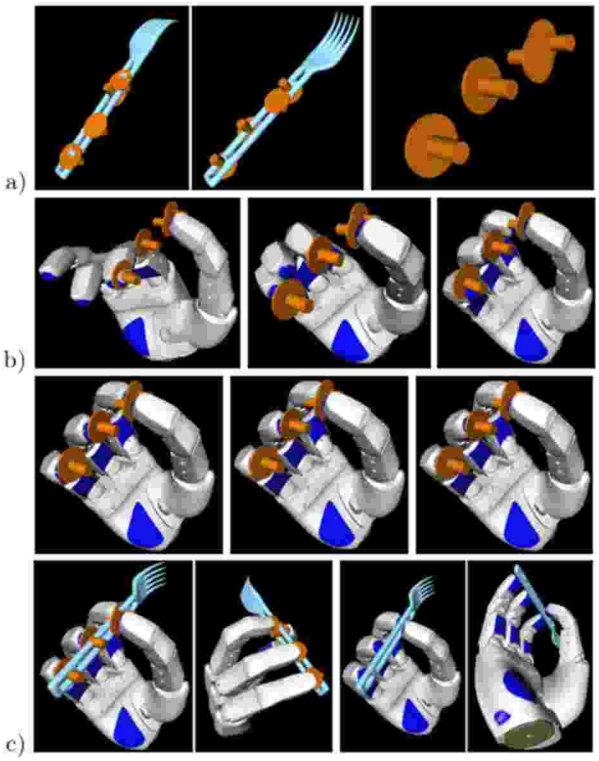


Fig. 6. a) The desired contact points with and without the fork (the position and orientation constraints are given by the center of each disk and the attached cylinder respectively); b) The initial hand configuration (first image) and the the hand configurations obtained in the next five jacobian iteration; c) Views of the final grasp configuration after 6 jacobian iterations (the first two images remark the contact points).

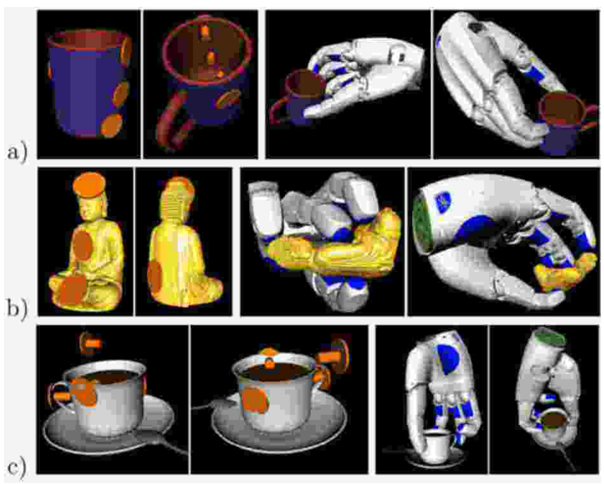


Fig. 7. Three examples with two views of the contact constraints on the object and two views of the solution.

configurations, with the previous off-line computation of the corresponding λ_i). The average execution time for one initial configuration is 5 ms, 7.5 ms for two, and 15 ms if twelve. With both measures it is easy to see that the average execution time to solve a 62% of all cases is 5 ms, and so on, as shown in Fig. 9b.

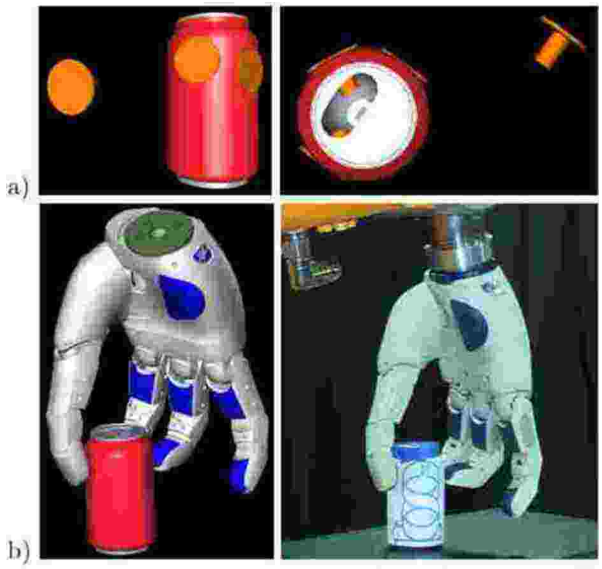


Fig. 8. a) Two views of the contact constraints on a can, and, b) obtained solution in simulation, and execution of the solution with the real hand SAH.

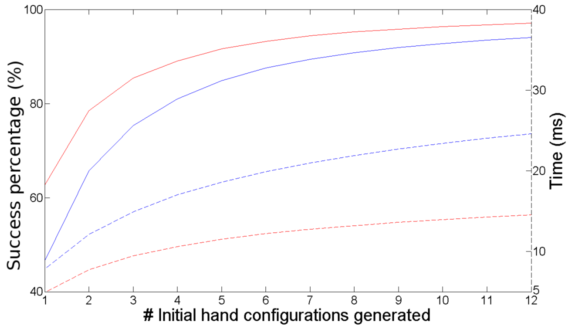
In order to have a certain mesure of the efficiency of the proposed approach with the off-line work (Section 3.2), the results shown have been compared with the ones obtained by generating random initial configurations; this means that at any time that Algorithm 1 reaches Step 2, the initial configuration has been computed by assigning random values to the λ_i. The *random* results correspond to the blue lines in Fig. 9a and 9b.

Two main conclusions can be drawn from Fig. 9a. First, the success rates of the random initial configurations are always inferior than the ones obtained with the off-line work; in the worst case the difference is a 16%, while in the best, around a 3%. Second, the time of the random initial configurations is always between a 50% to 67% slower. In Fig. 9b the comparative between the two methods can be more easily done; to obtain a 62% of exit percentage the off-line initial configurations need 5 ms, while the randoms spend 12 ms; to obtain a 94%, off-line configurations take 12 ms, and the random, 24 ms. That is, the off-line algorithm makes the proposed solution from a 50% to a 58% faster; therefore, it doubles the computation speed.

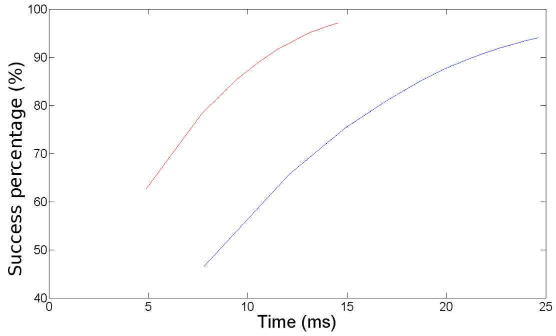
5. CONCLUSIONS AND FUTURE WORK

This paper presents a procedure to obtain a grasp configuration of the Schunk Antropomorphic Hand when a set of contact constraints is given (for instance, by a grasp planner). Statistically, the approach solves around 97% of the cases in a very reasonable time, and the percentage can be improved if increasing the execution time is acceptable. The approach combines a classical use of the Jacobian in an iterative algorithm with an off-line procedure that doubles the execution speed of the algorithm. The approach was implemented and the results are satisfactory.

Future work includes considering the potential collisions of the hand with the objects in the work environment and the inclusion of the implemented procedure in a general hand motion planner. Another interesting proposals to speed up the approach are the generation of a proper initial hand



a)



b)

Fig. 9. Statistical performance of the approach. Red lines correspond to results obtained with initial hand configurations obtained with the proposed off-line work (Section 3.2), and blue lines to results obtained with random initial hand configurations: a) Success percentage (continuous lines) and execution time (dashed lines) vs. the number of initial configurations generated; b) Success percentage vs. execution time.

configuration as a function of the constraints imposed by the contact points on the object, thus it would change depending on the particular problem to be solved, and the use of learning methods to improve the initial hand configurations based on the results of real applications.

ACKNOWLEDGMENTS

The authors would like to thank Leo Palomo for his support on the software, and Jan Rosell, Carlos Rosales and Jose Fortín for their thoughts and help in the experiments.

REFERENCES

A. Bicchi, (2000). Hands for dexterous manipulation and powerful grasping: a difficult road towards simplicity. *IEEE Trans. on Robotics and Automation*, 16 (6), pp. 652-662.

C. Borst, M. Fischer and G. Hirzinger, (1999). A fast and robust grasp planner for arbitrary 3d objects. *In Robotics and Automation. Proceedings, IEEE International Conference on*, pp. 1890-1896.

Ciocarlie MT, Allen PK, (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28 (7), pp. 851-867.

Cornella, J. i R. Suárez. Efficient determination of 4-point form-closure optimal constraints of polygonal objects, (2009). *IEEE Transactions on Automation Science and Engineering*, (ISSN: 1545-5955), Volume 6, Issue 1, Jan. 2009, pp. 121-130.

M.R. Cutkosky, (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans. on Robotics and Automation*, 5 (3), pp. 269-279.

I.T. Jolliffe, (2002). Principal Component Analysis. *Springer Series in Statistics*, Springer.

Z. Li, P. Hsu and S. Sastry, (1989). Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8 (4), pp. 33-50.

Y.H. Liu, (1998). Computing n-finger force-closure grasps on polygonal objects. *In Robotics and Automation. Proceedings, IEEE International Conference on*, pp. 2734-2739.

MPI Library. *www.mcs.anl.gov/research/projects/mpi/*. June 2010.

V.D. Nguyen, (1988). Constructing force-closure grasps. *The International Journal of Robotics Research*, 7 (3), pp. 3-16.

Y.C. Park and G.P. Starr, (1990). Grasp synthesis of polygonal objects. *In Robotics and Automation. Proceedings. IEEE International Conference on*, pp. 1574-1580.

N.S. Pollard, (2004). Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research*, 23 (6), pp. 595-613.

J. Ponce, S. Sullivan, A. Sudsang, J.D. Boissonnat and J.P. Merlet, (1997). On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16 (1), pp. 11-35.

M. A. Roa i R. Suárez, (2009). Finding locally optimum force-closure grasps. *Robotics and Computer-Integrated Manufacturing*, (ISSN: 0736-5845), Volume 25, Issue 3, June 2009, pp. 536-544.

C. Rosales, Ll. Ros, J. M. Porta i R. Suárez, (2011). Synthesizing grasp configurations with specified contact regions. *International Journal of Robotics Research*, 30 (4), April.

J. Rosell, X. Sierra, L. Palomo and R. Suárez, (2005). Finding grasping configurations of a dexterous hand and an industrial robot. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1178-1183.

J. Rosell, R. Suárez, C. Rosales, J.A. Garcia and A. Perez, (2009). Motion planning for high DOF anthropomorphic hands. *IEEE International Conference on Robotics and Automation*, pp. 4025-4030.

Safonova A., Hodgins J.K. and Pollard N.S., (2004). Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions Graph*, 23 (3), pp. 514-521.

Santello M. and Flanders M., (1998). Soechting JF postural hand synergies for tool use. *Journal of Neuroscience*, 18 (23), pp. 10,105-10,115.

B. Siciliano and O. Khatib, (2008). *Springer Handbook of Robotics*. Springer, pp. 257-258.

R. Suárez and J.A. Claret, (2009). Searching a valid hand configuration to perform a given grasp. *40th International Symposium on Robotics*, pp. 151-156.

Tsoli A. and Jenkins O.C., (2007). 2D subspaces for user-driven robot grasping. *In: Proc. of the RSS 2007 Workshop on Robot Manipulation: Sensing and Adapting to the Real World*.