2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

A Learning-free Method for Anthropomorphic Grasping

David Flavigné[†], Véronique Perdereau[†]

Abstract— **This work deals with grasping using an anthropomorphic hand. The main idea is to easily compute a grasp for a robotic hand in the context of a given task. This paper describes a method that does not require learning. Starting from works in the neuroscience field on human hand postural synergies, we introduce a two-level algorithm that uses a mathematical model of relationships between muscles and degrees-of-freedom of the hand and a set of five parameters to define synergies between muscles according to some grasp properties taken from an existing taxonomy of grasps. The two-level architecture presented in this paper aims to provide the flexibility needed for working with a real robotic hand. This algorithm is validated both in simulation using Gazebo and on the Shadow Robot Hand.**

I. INTRODUCTION

Manipulation skill provides robots with more autonomy because it enables them to interact with their environment. Currently, several kinds of manipulators exist to fulfill heterogeneous tasks. From industrial robotics to teleoperation, the number of these tools keeps increasing, also increasing their specifity. Indeed, they are very specialized devices built to perform one specific task, often requiring additionally the environment to be more or less controlled. The need for more generic manipulators that can handle a large set of tasks, and that can adapt to human environment, has led researchers to build new manipulation organs, such as simple grippers and dextrous robotic hands.

Recent attempts to build a dextrous manipulation organ have focused on human-inspired strategies. Human-like hands with numerous degrees-of-freedom (dofs) are used in laboratories to test various grasping algorithms. The advantage of the human hand is its high dexterity and its ability to rapidly execute heterogeneous tasks on a wide variety of objects. The most challenging point resides in the contrast between the known complexity of a task and the apparent simplicity for a human to perform the task. This suggest that humans have an efficient way to deal with high-dimensional configuration spaces that defeats common motion planning algorithms.

An obvious strategy to mimic the behaviour of a human hand is to learn from human observation. Common solutions for useful learning from human often require specific platforms with expensive and/or complex sensors. A significant sample of subjects must be prepared and asked to execute repetitive actions. Another strategy consists in modeling a mechanical hand close to the human hand. The accurate modeling of a human hand requires a precise knowledge of hand biomechanics.

In this paper, we choose to present a simple mathematical model of the human hand that can be tuned and extended using learning. This novel method is based on postural and muscle synergies, and is divided into two levels. The first level describes relationships between muscle activations and joint movements. The second defines synergies between muscles using five main grasp properties from an existing grasp taxonomy. To test our methodology, we performed grasping simulations using the Gazebo simulator software and the Shadow Ethercat Hand equipped with five ATI Nano17 6D force-torque sensors at fingertips.

As part of the HANDLE project, this method belongs to a chain of several other modules. Hence, it focuses on how to execute a grasp without handling related parts such as grasp stability, object detection or learning.

In the next section, we present related work on manipulation and synergies of the hand. Then, the two different parts of the method are explained, followed by implementation details. In the fourth section, computer simulations and experimental results on a real anthropomorphic hand are presented. Then a discussion is presented. Finally, we conclude by synthesizing core ideas of the paper and presenting main directions for future work.

II. RELATED WORK

In general, grasp synthesis methods have to satisfy three main sets of constraints: the hand geometry and kinematic structure, the object shape, and the task. Analytical approaches are focusing on finding force-closure grasps [1], [2] or on task compatibility using wrench spaces [3]. Empirical approaches based on human observation rely on data gathered using vision devices, data gloves or other tactile devices to record human movements [4], [5]. Other empirical approaches achieve better grasp stability by using observation of the object [6]–[8]. In [9], the authors address the problem of uncertainty by applying contact sensor feedback to modify the posture of the hand while it is moving to grasp an object.

Studies in the field of neuroscience have focused on postural synergies occuring during grasping. For instance, the authors of [10] observed that several subjects exhibited similar patterns when grasping everyday objects. They suggested that most of the grasp postures could be described using a reduced number of parameters. In [11], authors use this notion of synergy to define subspaces for hand postures that express coordination patterns between multiple dofs. These subspaces are used to optimize the grasp during an interactive

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n^{o}231640 (http://www.handle-project.eu)

[†]UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France

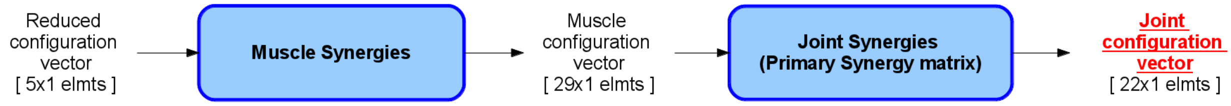


Fig. 1. Two levels of synergies are used to command an anthropomorphic hand using only a reduced set of parameters. 'elmts' stands for elements

approach to the object. The authors of [12] reviewed different methods that use the notion of synergy to generate grasps and to model robotic hands. Further, they present approaches using synergies to control the distribution of force on the surface of the object. They propose a design for an artificial hand using the principles explained in the paper. In [13], the authors present an extensive analysis of the forces involved in grasping. They include in their model the friction, the joints compliance and the postural synergies of the hand. They show that few postural synergies are sufficient to obtain force-closure grasps. Further, they show that the addition of high-order synergies (above 3) have almost no effect on the quality of the grasp.

Numerous realistic physical models for the human hand have been proposed. The authors of [14] model complex physiological actions, such as musculo-tendon and neuro-muscular control, which occur during hand grasping. Their models are based on biomechanics, ergonomics, and existing robotic hands.

However, the main motivation of our work is to provide our robotic hand with a simple synergy-enabled grasping method that is flexible and easy to use. The main control parameters should be easy to understand. The method should provide different solutions for tuning, modifying and extending grasping possibilities. The method presented on the next section is build on these specifications.

III. METHOD

The main idea of the method described here is the generation of human-like movements using a simple model that does not require learning the synergies. Specifically, we want a method that does not require learning, but can be easily configured to be improved using learning. Methods using learning require the construction of large databases, which is time consuming and/or an expensive process. Even if data is shared between labs and research teams, numerous adaptations may be required to fit the data to the specific requirements of a particular application.

In defining the model, we assume that some of the synergies in the hand are implied by the biomechanics of the hand, and in particular by the biomechanics of its muscles. The remaining synergies are mainly derived from synergies between muscles. The literature in the fields of physiology and anatomy provides detailed information about the biomechanics of the hand. However, understanding muscle synergies is still an active research topic. Therefore, we divide our method into two main parts. In the first part, we use the biomechanics of the hand to define a simple mathematical model. In the second part, we define our own model of synergies between muscles using parameters that can be easily understood and intuitively modified by a human, i.e. human-understandable parameters.

Our method provides a modular structure, where each part can be tuned or replaced by a custom method. The two main parts of our method, displayed in figure 1, are represented in the model by synergy matrices containing constraint-defined coefficients. The muscular synergy vector, which is defined on the basis of the grasp taxonomy [15], computes a 29-element vector from the main control vector (which is a 5-element vector). The primary synergy matrix transforms this 29-element vector containing the activation level of muscles to a 22-element vector representing the dofs of the hand.

The grasp types (as defined in [15]) to be executed are given by a higher level method that searches the best grasp sequences for a given task using Markov Decision Processes [16]. Our synergy method has to execute a grasp given this grasp type and has to inform about its success. The output is a sequence of joint references that will be executed on the hand by an external controller such as [17]. Quality metrics are the same than those used in [18].

In the following sections, the nomenclature used in the paper is given, then we describe each part of the method: first level synergies (section III-B), and second level synergies (section III-C).

A. Nomenclature

Fig. 2 shows the nomenclature used in this paper for dofs and muscles. In joint names, the first letter stands for the finger (**T**humb, **I**ndex, **M**iddle, **R**ing and **L**ittle). The other letters are for the joint type :

- **M**eta**C**arpo**P**halangeal
- **Rot**ational
- **D**istal **I**nter**P**halangeal
- **C**arpo**M**eta**C**arpal
- **P**roximal **I**nter**P**halangeal
- **Abd**uction/adduction

For example, *IMCP* stands for the **I**ndex **M**eta**C**arpo**P**halangeal dof.

B. Synergies between joints

This section describes the equations defining the relationships between the muscles and the dofs of the hand. We construct the first level of synergy (primary synergy matrix) using information obtained from biomechanical literature [19]. The qualitative descriptions of the muscle hierarchy of the hand are expressed by equations that combine the muscle activations and produce values for the dofs of the hand. We weight these combinations by a set of coefficients that we obtain using constraint programming. It allows us to define a 22×29 matrix that will be used to compute a configuration of the hand from muscle configuration. The rows of the matrix represents dofs of the hand, while the columns represent

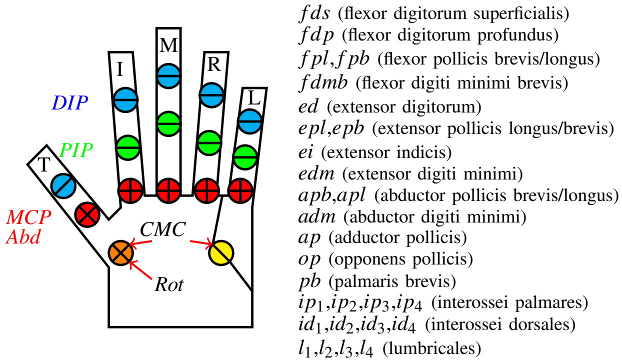


Fig. 2. Terms used for naming joints (left) and muscles (right). Circles with a cross represent joints with 2 dofs.

muscles. Thus, each cell contains a coefficient determining the action of one muscle on one dof. All non-zero coefficients of the matrix ($A1$ to $C22$) are defined according to the constraints presented in paragraph III-B.1.

Equations in Table I show the different actions of muscles on each dof. For example, the position of the *IMCP* joint is defined by the actuation of seven muscles: two for extension (*ed* and *ei*) and five for flexion ($ip2$, $id1$, $l1$, fds and fdp).

TABLE I

EQUATIONS MODELING SYNERGIES BETWEEN FINGER JOINTS.

TCMC = -A1.epb -B1.apb +C1.op
TMCP = -A2.epb -B2.apl +C2.fpl +D2.ap +E2.fpb +F2.ip1
TPIP = -A3.epl +B3.fpl -C3.ip1
TAbd = -A4.epb -B4.apl -C4.apb +D4.fpb +E4.ap +F4.op +G4.ip1
TRot = -A5.epb -B5.apl +C5.op
IMCP = -A6.ed -B6.ei +C6.ip2 +D6.id1 +E6.l1 +F6.fds +G6.fdp
IPIP = -A7.ed -B7.ei -C7.ip2 -D7.id1 -E7.l1 +F7.fds +G7.fdp
IDIP = -A8.ed -B8.ei -C8.ip2 -D8.id1 -E8.l1 +F8.fdp
IAbd = -A9.id1 +B9.ip2
MMCP = -A10.ed +B10.id2 +C10.id3 +D10.l2 +E10.fds +F10.fdp
MPIP = -A11.ed -B11.id2 -C11.id3 -D11.l2 +E11.fds +F11.fdp
MDIP = -A12.ed -B12.id2 -C12.id3 -D12.l2 +E12.fdp
MAbd = -A13.id2 +B13.id3
RMCP = -A14.ed +B14.ip3 +C14.id4 +D14.l3 +E14.fds +F14.fdp
RPIP = -A15.ed -B15.ip3 -C15.id4 -D15.l3 +E15.fds +F15.fdp
RDIP = -A16.ed -B16.ip3 -C16.id4 -D16.l3 +E16.fdp
RAbd = -A17.id4 +B17.ip3
LCMC = -A18.adm +B18.pb +C18.odm
LMCP =-A19.ed-B19.edm+C19.ip4+D19.l4+E19.fdmb+F19.fds+G19.fdp
LPIP = -A20.ed -B20.ip4 -C20.l4 -D20.edm +E20.fds +F20.fdp
LDIP = -A21.ed -B21.ip4 -C21.l4 -D21.edm +E21.fdp
LAbd = -A22.adm +B22.ip4 +C22.odm

1) Defining the set of coefficients: In physiology literature, muscle actions are often described qualitatively [19]. Some papers are gathering quantitative data including muscle strength [20], [21]. Data about hand extrinsic and intrisic muscles vary from one experiment to another, mainly because muscle strength can be very different from one subject to another [22]–[25].

Rather than mixing different sources on muscle strength, we preferred a more "logical" approach to describe muscle-joint interactions by defining constraints on muscle actions from the qualitative indications in literature. These constraints are represented by coefficients in front of each muscle actuation value.

To define these coefficients, a learning approach would require time and a specific equipment. Instead, we choose to define some constraints that coefficients must respect in order to mimic the human hand behaviour. The goal is to provide a set of coefficients that can be used to control our robotic hand, without using learning techniques.

Most of the constraints are based on intuitive assumptions. This gives us hypotheses to start on and to build an initial set of coefficients. This set, which turns out to be good enough for experimentations on a real robot (see section IV) does not fully reflect the human hand biomechanics, but is meant to be tuned for better results.

The constraints are defined using different pieces of information. For brevity reasons, only few examples of the equations derived from these constraints are presented.

Some muscles have equivalent effects on different dofs. This means the concerned coefficients are equal. This assumption allows to determine which dofs are influenced in the same manner by a specific muscle. For example, *ed*, (resp. fdp,fds) acts in the same way on the index, middle, ring and little fingers: the four *ed* (resp. fdp,fds) coefficients for *MCP*, *PIP* and *DIP* joints are equivalent. Relations 1 and 2 are derived from these constraints.

$$A6 = A10 = A14 = A19 \qquad (1)$$

$$A7 = A11 = A15 = A20 \qquad (2)$$

For each dof, the different muscles involved in its computation have more or less influence. This assumption allows to weight the action of the different muscles on one dof. For example, *epb* and *apb* act to extend the $TCMC$ dof, but *epb* is the main extender for this joint. Thus, coefficients for *epb* are superior to coefficients for apb. On the other side, *op* is the only flexor for $TCMC$. Its coefficient has to ensure that it will at least cover half the size of the range of the dof defined for the hand. The same reasoning applies for all other dofs. In relations 3 and 4 below, we present some examples of constraints:

$$A1 > B1 \qquad (3)$$

$$C1 > halfrange[TCMC] \qquad (4)$$

One muscle can act on different dofs. For each of these dofs, the action of the muscle can be different. This assumption allows to order the different coefficients of a muscle for the different dofs they act on. For example, fds primary function is flexion of *PIP* joints but it also flexes *MCP* joints. Thus, coefficients corresponding to fds muscle are superior in *PIP* joint equations than in *MCP* joint.

The last constraints ensure the coverage of the range of each dof. This reflects the fact that one muscle can antagonize several other muscles and maintain the current joint configuration, while other muscles contract. When the hand is in rest position, the action of extensors (resp. flexors) alone should allow to reach extension (resp. flexion) limits of the concerned dofs.

The matrix obtained from these equations provides synergy primitives. They allow different movement primitives for each finger (such as simple flexion, extension, abduction, flexion with extended phalanxes, etc.). This provides flexibility by allowing higher-level methods to use synergy primitives instead of independent finger joints. Moreover, the set of coefficients can be tuned or completely modified using learning techniques. The usage of other data such as physiological cross-sectional area of muscles is also possible. Additionally, we can adjust them to fit the needs of different robotic anthropomorphic hands. The matrix does not depend on the hand current configuration and is static during manipulation.

This model gives more parameters to control than in the beginning. The high-dimension problem of manipulation is not yet solved. This is the purpose of the second level of synergies presented in the next section.

C. Synergies between muscles

The model described above represents synergies between finger movements using relationships between muscles and joints. To reduce the problem dimensionality, we introduce synergies between muscles.

This second level of synergies represents synergies between muscles activation given a limited number of parameters that can describe the desired grasps. Four parameters are taken from the grasps properties described in the taxonomy. One parameter is added to represent the closure of the hand. These parameters were chosen to be human understandable, and ease the description of a grasp. All the considered parameters are real values in the $[0,1]$ range. They can change continuously during the planning of a grasp. These five parameters are the following ones:

- **Power/Precision** parameter: Depending on if we execute a power grasp or a precision grasp, different muscles are activated. It affects synergies during flexion and extension. It acts mainly on extrinsic muscles.
- **Number of Finger** parameter: Determines which fingers will oppose the thumb.
- **Thumb Opposition** parameter: Determines if the thumb is adducted or abducted. It acts on opposition muscles.
- **Hand closure** parameter: Controls the global configuration of the hand which ranges from fully opened (0) to fully closed (1). It acts on flexion/extension muscles.
- **Fingers Adduction** parameter: Controls the adduction of fingers. It mainly acts on muscles responsible for the adduction of fingers (*interossei* muscles).

The five parameters presented above are "human-understandable", i.e. a human can read, understand, and intuitively set the values of these parameters. For example, humans may have an understanding about how the hand closure parameter affects the model, and intuitively set its value to fulfill their requirements.

Each parameter is used to fill a 29-element vector. For a vector, each element represents activation value for one muscle. Let M_{pow}, M_{nf}, M_{thOpp}, M_{clos} and M_{add} be the vectors obtained from the parameters. These vectors are obtained as shown in 5:

$$M_p = \begin{bmatrix} m_{p_1} \\ \vdots \\ m_{p_{29}} \end{bmatrix} \text{ with } \begin{cases} m_{p_i} = f_i(P) & \text{if the } i^{th} \text{ muscle is used} \\ m_{p_i} = 0 & \text{otherwise} \end{cases} \tag{5}$$

where P is the value of the parameter and $f_i(P)$ depends on the parameter type and the muscle. For example, $f_{fds}(P) = 1 - P$ for M_{pow}.

These five vectors are used to form a single vector representing the final muscle activations. This final vector is obtained using the formula 6:

$$M_{muscle} = [m_{muscle_1} \cdots m_{muscle_{29}}]^T \tag{6}$$

where

$$m_{muscle_i} = (m_{add_i} + m_{thOpp_i} + m_{clos_i}) \times m_{nf_i} \times m_{pow_i} \tag{7}$$

i represents the i^{th} element of the corresponding vector.

In Equation 7, the m_{nf_i} vector and the m_{pow_i} values represent the level of involvement of each muscle rather than explicit activation. That is why they globally influence other parameters. As for adduction, closure and thumb opposition vectors, their values are added since each of these movements require additional strength on the concerned muscles. The final values are then bounded to the $[0,1]$ range. These five parameters define the synergy space.

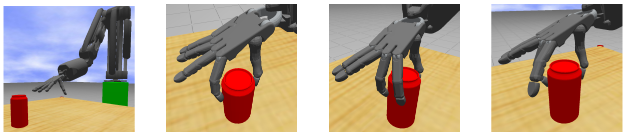
Using this muscle synergies level, we now have a complete model that computes a twenty-two dofs configuration from a reduced set of five human understandable parameters. These parameters are meant to be modified along a defined trajectory to execute a given grasp and reach the target object.

This algorithm is validated on a simulator including a realistic model of the Shadow hand and on the real platform. These tests are described in the next section.

IV. SIMULATIONS AND EXPERIMENTS

Both simulations and experiments are made using ROS operating system and ROS nodes developed specifically for our experimental platform (described in section IV-B). ROS nodes are containers for processes that perform computation. The nodes we used include a software for dealing with contact sensors on fingertips for both the simulator and the real platform, a controller node for the hand and a specific node for the simulator. Some of these nodes are entirely developed for the project.

The core algorithm is integrated as a library in a ROS node implementing an action server. The grasp execution is done by sending a sequence of configurations of the hand computed "on-the-fly" and sent to the hand controllers for immediate execution. The node can receive a request containing a "Grasp ID" refering to the taxonomy. In which case, it retrieves the corresponding parameters in an internal database. The request can also contain raw parameters, if the database does not contain the desired grasp. The last part of the request is a "contact scheme" and is optional. It



(a) (b) (c) (d)

Fig. 3. The simulation scene is reproducing the real platform with accurate meshes of the arm and hand, and a can representing the object to grasp. Three examples of tested grasps: palmar pinch (b), tripod grasp (c) and tripod grasp when the contact scheme specifies that the index should not be in contact (d).

(a) (b) (c)

Fig. 4. Different grasps tested during experiments on real platform. A palmar pinch grasp (a), a modified tripod grasp (b) like in Fig 3(d), and a large diameter grasp (c)

provides information about which finger is expected to be in contact at the end of the grasp. It can be specified if it is different from the information contained in the database. The algorithm will take this information into account to return feedback about the success of the grasp. This feedback is mandatory in order for the higher level nodes to validate the grasp. For flexibility purposes, this optional contact scheme can override the number of fingers parameter.

A. *Simulation*

To test our algorithm, we used the Gazebo simulator. The simulation setup is shown in Fig. 3(a). Three nodes must be launched for the algorithm to be able to interact with the simulated hand: the sensors node, the controllers node, and our own node that wraps the algorithm.

In Fig. 3, the final configurations of the hand for three computed grasps are shown. The palmar pinch grasp in Fig. 3(b) involves only two fingers, the thumb and the index. It is a precision grasp, as it requires little strength and only fingertips to be in contact. The index is a little bit abducted, to avoid collision of other fingers with the grasped object. The thumb is opposed to the fingers. The closure parameter is always set fully close, so that starting from an opened configuration, the fingers are continuously flexing to reach the target configuration. This gives us our five parameters for setting the target. The concerned fingers are stopping when detecting a contact with the can during the grasp. This allows the configuration of the hand to be mold by the object shape.

Fig. 3(c) shows a tripod grasp, which is defined as a grasp with three fingers in contact, precision type, with thumb in opposition and fingers abducted. In Fig. 3(d), we send the same Grasp ID, but a contact scheme is given to specify which fingers we expect to be in contact. In this case, we put only middle finger and thumb, so that the index is not used for the grasp. As this command was sent directly after the normal tripod grasp command (the one shown in Fig. 3(c)), the index is releasing contact while other fingers in contact are told to maintain contact.

B. *Experiments*

In this section, we describe experimentations on a real anthropomorphic robotic hand, which is the Shadow Ethercat Hand. Fingertips are equipped with *ATI nano17* 6-dof force-torque sensors. Not counting the wrist, the hand has eighteen dofs as the distal and proximal joints are coupled for the fingers (both are added for each finger before sending configuration to the hand controllers). The objects used are a standard coke can and a sugar dispenser. They are standing on a flexible surface.

The ROS nodes used include low-level Ethercat drivers, position controllers, force sensors drivers and contact point computation for *nano17* sensors, and a library for accessing to joint commands and positions. The last node is the one containing our synergy grasping method. The tests are presented in the attached video. Due to technical constraints, the motor velocities were deliberately slowed down.

For each test, the arm and wrist positions were modified manually to a suitable configuration in order to test the grasping method. All grasp commands are sent via command line using ROS tools.

In the first scene, the hand goes from an opened pregrasp configuration to a grasp with two fingers. The dofs of the concerned finger are moving at the same time to execute the given grasp. We can remark that the fingers stop when contact is detected. Contact points are not strictly opposed on the surface of the can and the fingers are not reaching it at the same time. This is due to the initial pose of the hand relative to the object that is not ideal for having a stable grasp. As the stability of the can is not taken into account, the hand has to be placed carefully.

In the following grasp, the pressure exerted by the new contact slightly moves the object as it is not compensated by a better opposition of the thumb. In the release phase, the can is moving towards the thumb. This is due to the thumb releasing its pression on the can before the index.

In the second scene, the second command includes a modifier for the contact scheme. It tells the algorithm which finger we expect to be in contact. The resulting grasp is a two finger grasp using middle and thumb, letting the index free for another task such as finger gaiting.

In the third scene, the *Large Diameter* power grasp involves contacts with phalanxes and palm. The hand has to be placed very close. The can stands on a higher support because the bounds of wrist and arm joints did not permit a lower position to be reached. The initial configuration of the hand specifies abducted fingers, while the final one is adducted. This results in fingers getting closer to each other linearly during the execution of the grasp. The fourth scene shows the grasp *Fixed Hook* with a sugar dispenser.

The remaining part of the video shows the variety of grasps that can be obtained using our method, in simulation and on the real platform.

V. DISCUSSION

The described method allows to generate a large set of grasps using a reduced number of parameters. These parameters are easily understandable by a human and can be adapted to each situation. The generated grasps are molded by the object shape using contact sensors.

Several works [10]–[13] in the literature agree that most of the variability of human grasps is dominated by only two or three synergies. However, these synergies are computed using algorithms such as PCA decomposition on data learned from human experiments. The synergies obtained are not meant to be intuitively understood or tuned by a human operator. They depend on the subjects and may vary according to the task executed. These methods did not fit our requirements since our main motivation was to provide a simple and flexible method to grasp objects, without having to learn it from observation.

The video shows that the grasp quality and stability depend on the placement of the arm and wrist. They have to be correctly placed for each considered grasp. This placement may be different depending on the grasp and on the environment constraints.

Although a 22-dof configuration is computed by the algorithm, some grasps of the taxonomy cannot be executed due to physical constraints on the hand. Indeed, the hand has only 18 dof: the two last phalanxes of the fingers are coupled. This limits the types of grasp that can be actually executed. These constraints were adressed by remapping the output of the method. In a more general way, as long as the manipulator is an anthropomorphic model with reasonably simple variations from the Shadow hand kinematic structure, remapping seems to be sufficient. If the kinematic structure varies a lot from the Shadow hand model (while staying anthropomorphic), the method may simply be adapted by modifications of the set of coefficients and/or of muscle synergies. The hand movement may sometimes seem clumsy. That is due to the linear interpolation in the synergy space. This results in all parameters moving at the same time to complete the grasp. This is illustrated in the video by the two last scenes: finger adduction is not completed until the end of the grasp, while a human may prefer to pre-shape his hand before grasping the object. A prioritisation of parameters may solve this problem.

VI. CONCLUSION

In this paper, we presented a two-level method to easily generate human-like grasps that adapts to the object shape. This method relies on biomechanics-based synergies between fingers and defined synergies between muscles to generate a configuration of the hand from a human-understandable set of parameters. The method does not require learning from human data but can be tuned using such data for defining coefficients of the synergy matrix. This method was tested on the Shadow Hand using different types of grasps. Different possible improvements are exposed in the last section. Some of them are addressed by other modules of the *HANDLE* project, while others are ongoing work. We plan to adapt an existing force controller to maintain contact of all concerned fingers during the whole grasp and to manage transitions between two grasp types. Future works also include the implementation of in-hand manipulation synergies based on this method, as well as a better interpolation method for the parameter set.

REFERENCES

[1] D. Ding, *et al.*, "An efficient algorithm for computing a 3d form-closure grasp," in *IEEE/RSJ IROS*, 2000.
[2] X. Zhu and H. Ding, "Planning force-closure grasps on 3-d objects," in *IEEE ICRA*, 2004.
[3] M. Prats, *et al.*, "Task-oriented grasping using hand preshapes and task frames," in *IEEE ICRA*, 2007.
[4] F. Kyota, *et al.*, "Detection and evaluation of grasping positions for autonomous agents," in *Int. Conf. on Cyberworlds*, 2005.
[5] J. Romero, *et al.*, "Human-to-robot mapping of grasps," in *IEEE/RSJ IROS*, 2008.
[6] Y. Li, *et al.*, "Data-driven grasp synthesis using shape matching and task-based pruning," *IEEE TVCG*, 2007.
[7] A. Saxena, *et al.*, "Robotic grasping of novel objects using vision," *IJRR*, 2008.
[8] M. Stark, *et al.*, "Functional object class detection based on learned affordance cues," *ICVS*, 2008.
[9] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *IEEE/RSJ IROS*, 2009.
[10] M. Santello, *et al.*, "Postural hand synergies for tool use," *J. Neurosci.*, 1998.
[11] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *IJRR*, 2009.
[12] A. Bicchi, *et al.*, "Modeling natural and artificial hands with synergies," *Phil. Trans. of the Royal Society B*, vol. 366, pp. 3153 – 3161, 2011.
[13] M. Gabiccini, *et al.*, "On the role of hand synergies in the optimal choice of grasping forces," *Auton. Robots*, vol. 31, no. 2-3, pp. 235–252, Oct. 2011.
[14] J. L. Sancho-Bru, *et al.*, "Towards a Realistic and Self-Contained Biomechanical Model of the Hand," *Theoretical biomechanics*, pp. 212–240, 2011.
[15] T. Feix, *et al.*, "A comprehensive grasp taxonomy," in *RSS: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, June 2009. [Online]. Available: http://grasp.xief.net
[16] U. Prieur, *et al.*, "Modeling and planning high-level in-hand manipulation actions from human knowledge and active learning from demonstration," in *IEEE/RSJ IROS*, 2012.
[17] K.-C. Nguyen and V. Perdereau, "Fingertip force control for grasping and in-hand manipulation," HANDLE Training Workshop, Benicassim, Spain, February 2012.
[18] F. Veiga and A. Bernardino, "Towards bayesian grasp optimization with wrench space analysis," in *IROS: Workshop Beyond Robot Grasping*, 2012.
[19] I. Kapandji, *The Physiology of the Joints : Volume One Upper Limb*, 1982.
[20] "Architectural design of the human intrinsic hand muscles," vol. 17.
[21] A. Cutts, *et al.*, "Ratios of cross-sectional areas of muscles and their tendons in a healthy human forearm." *J. of Anatomy*, vol. 176, pp. 133–137, June 1991.
[22] S. Murai, *et al.*, "Combinatorial roles of extrinsic and intrinsic muscles in extension strength of the distal interphalangeal joint," *J. of Orthopaedic Res.*, vol. 30, no. 6, pp. 893–896, June 2012.
[23] K. R. S. Holzbaur, *et al.*, "Upper limb muscle volumes in adult subjects," *J. of Biomechanics*, vol. 40, no. 4, pp. 742–749, 2007.
[24] S. Koh, *et al.*, "Intrinsic muscle contribution to the metacarpophalangeal joint flexion moment of the middle, ring, and small fingers," *J. of Hand Surgery*, vol. 31, no. 7, pp. 1111–1117, Sept. 2006.
[25] T. A. Schreuders and H. J. Stam, "Strength measurements of the lumbrical muscles," *J. of Hand Therapy*, vol. 9, no. 4, pp. 303–305, Dec. 1996.