Task-oriented grasping human synergies based using modular robotic hands

Master Student: Filippo Sanfilippo
Supervisors: Prof. Dr. Jianwei Zhang, Prof. Domenico Prattichizzo, Ph.D. Houxiang Zhang, Ph.D. Student Gionata Salvietti

University of Hamburg, Group TAMS

Jenuary 2011

Summary I

Summary II

Summary III

Introduction: task-oriented grasps

- Grasping is still a crucial topic in robotics research. Modular robotics could be one of the most promising solutions.

What we present

- Modular robotic hand models capable to grasp an object
- A method to obtain the optimal mapping of postural human synergies for non-human hands.
- An algorithm capable to determine which is the minimum and the optimal hand configuration in order to grasp an object and to accomplish a planned post-grasp operation. It also provides the stable grasp and the optimal force necessary to grasp the object.
- A modified version of our algorithm that is also capable to determine the achievable postural synergies.

Motivation and inspiration

Advantages

- Versatility. A modular construction allows an easy adaption on different requirements and situations.
- Robustness. Modular robots offer high versatility robustness and robustness against failure, as well as the possibility of self reconfiguration.
- Low cost. The production, maintenance and repair costs of the modular robots are reduced.

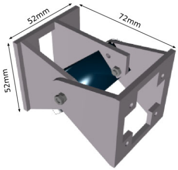
Disadvantages

- To avoid coupling of the joints and to allow a modular construction, direct driven joints are necessary.

A simple planar modular gripper

Initially, we decided to make a simple gripper model with the following features:

- two fingers;
- three phalanges for each finger;
- the model is based on Y1 module[1]



Y1 module characteristics

- simple and cheap;
- One degree of freedom, actuated by a servo;
- Material: PVC;
- Weight: 50gr;
- Rotation Range: 180 degrees

[1] built in 2004 by an international group consisting of TAMS and BUAA

Motivations

- We decided to use these modules for their simplicity and flexibility. Furthermore they are available in the lab where I am doing my thesis (TAMS Group, University of Hamburg).
- It should be noted that these modules have never been used for building a robotic hand. Salvietti used these modules for introducing a task priority based approach in order to manage grasping and locomotion functions starting from the hyper redundant manipulator theory[2]. However, in this work, the grasping tasks were performed on the ground by hyper redundant robots.

[2] G. Salvietti, H.X. Zhang, J. Gonzalez-Gomez, D. Prattichizzo, J.W. Zhang, A Task Priority Framework for Modular Robot with Grasping Capability

Using a simulator for fast developing

- We decided to use a simulator to develop our gripper model. We had to decide which simulator to use. Juan Gonzalez-Gomez developed a plug-in for the famous simulator OpenRave. This plug-in allow the user to simulate the locomotion of modular robots. In our opinion, however, OpenRave does not offer many tools for assessing the grasp quality.
- On the other hand one of the most used, powerfull and appreciated grasp simulator is GraspIt!. GraspIt! is an interactive simulation, planning, analysis, and visualization system for robotic grasping. The downside is that there is not any modular plug-in for GraspIt!.
- So considering all this, we decided to use GraspIt! and to build our modular model for this simulator.

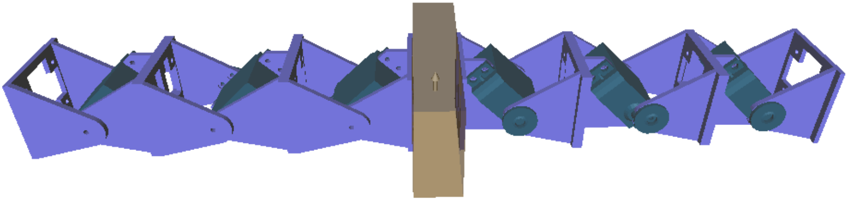
GraspIt! simulator

We decided to use GraspIt! because it allows a easy, fast and precise grasping evaluation. It also offers:

- stable Grasp Computation
- analytical Inverse Kinematics Solvers
- support to Matlab and to other script engines
- possibility to develop your own plugin
- support to ROS

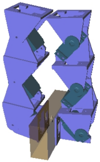
GraspIt! model

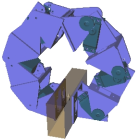
- In GraspIt!, a Robot is made up of multiple links, connected into kinematic chains.
- In general, in GraspIt!, a robot configuration file contains the following data:
  - the palm, this is simply a pointer to the body file that contains the palm.
  - degrees of freedom
  - kinematic chains
- We defined all these fields.



Synergies

- Besides, we also set the hand synergies for our own model.
- In GraspIt!, the synergies are called eigengrasps. In particular, we defined two different synergies:
  - proximal joints flexion
  - distal joints flexion
- While, at this step, we found our choices to produce good and fast results, then we will propose our algorithm for the optimal choice of eigengrasps for modular robotic hand and, more generally, for non-human hands.
- The advantage of using eigengrasps is that they form a low-dimensionality basis for grasp postures, and can be linearly combined to closely approximate most common grasping positions.

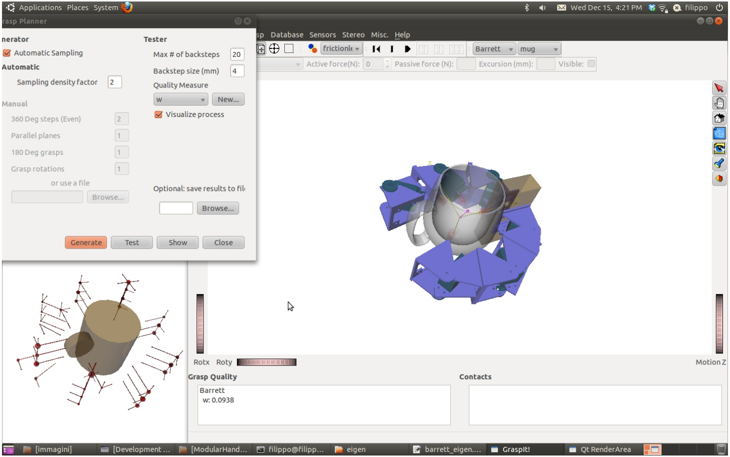




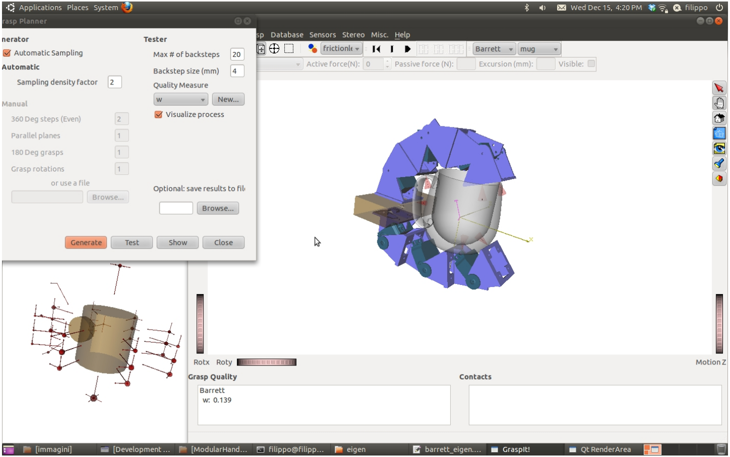
Grasp of a cup

- Despite the size of the modules that is too large to implement a gripper in order to grasp human daily life objects, we tried to grip a cup with the developed model.
- Using the GraspIt! Primitive-based Planner we obtained some interesting results.
- The planned found different grasps that are form closed. However, among the different grasps returned by the GraspIt! planner there are some that are not useful for the post-grasp tasks.
- Functional grasp: the gripper has to accomplish the grasp as well as the planned post-grasp operation. For example, if we want to grasp a cup and the planned post-grasp task is the operation of pouring-in/out, we cannot grasp the cup by placing the finger over it, but we should grasp the cup from its side.

Functional grasp of a cup

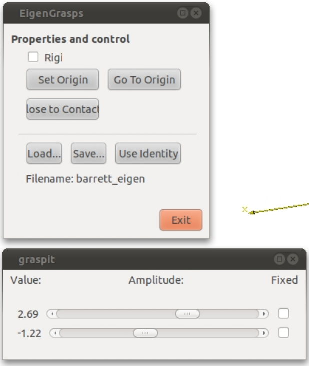


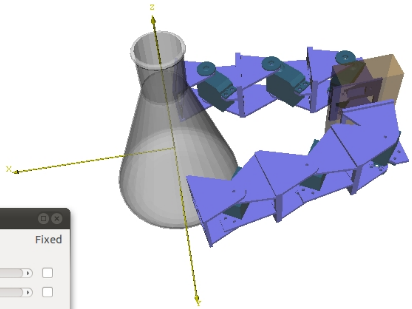
Not functional grasp of a cup



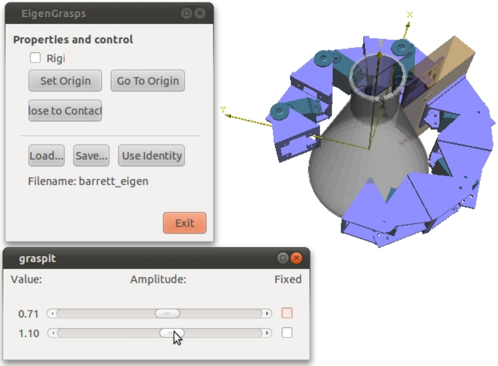
Grasping a flask using the gripper synergies

We also performed some grasps of a flask using the modular gripper that we have modelled. For doing this we used the two hand synergies.





Grasping a flask using the gripper synergies



A possible approach

- Assuming that position and orientation of the manipulator's wrist and palm have been accurately determined so that it can hits the target object, the fingers have to slowly close around the object until they cannot close further.
- Each link has to stop only when it reaches its joint angle limit or when one of the successive segments in its chain collides with the object.
- Anyway, using this approach, the contact points are only touched by the fingers. No force or only little force due to offset error is applied through these contact points. Hence, it is necessary to determine the optimal actuator efforts and the corresponding contact forces.

Imitating the human method of grasping

- Holding and manipulating delicate objects such as a glass of water, introduces the concept of safe grasping. When a robot manipulates a delicate object, it must grasp it in a stable way to prevent slipping by applying the minimum necessary grasping force to avoid object breaking or deformation.
- Safe grasping with robotic manipulators could be done by force closure method which needs a sophisticated control capable of providing stable grasping of the object with minimum possible normal force.
- Adigh and Ahmadi proposed a new controller inspired by human method of grasping[3]. The proposed controller, which only needs measurement of the contact forces, is then employed in conjunction with jacobian transpose method for grasping by multi-link fingers.
- We want to use this method for the real implementation of our modular gripper.

[3] Adigh and Ahmadi, Safe grasping with multi-link fingers based on force sending

A human-like control law

- The controller proposed by Sadigh and Ahmadi feedbacks the normal and tangential force and based on it decide whether to close the gripper jaws or not. It is necessary to use a control law of the following form for $x_i' \in \mathbb{R}^1$ which shows the closing speed of i-th gripper jaw:

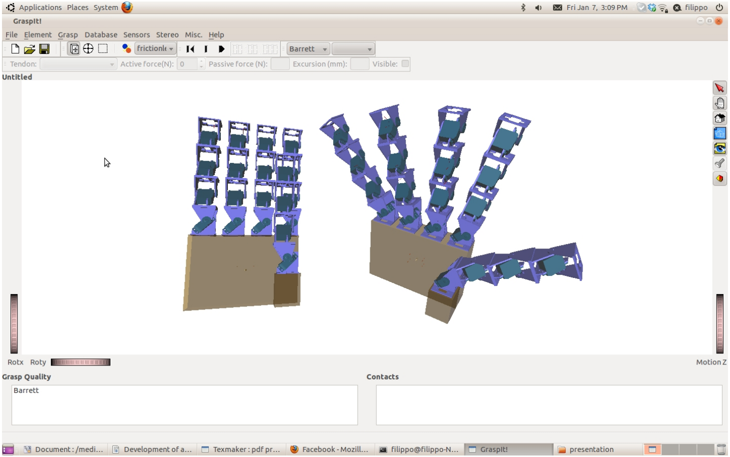
$$x_i' = \frac{V - k_N\left(N_i - \dfrac{k_\mu |f_i|}{\mu_i}\right)}{2}$$

- in which k_N, k_μ and V are some constants to be chosen to meet desired performance. On the other hands μ_i, N_i, and f_i are, respectively, friction coefficient and the measured values of normal and friction forces of the i-th finger. It also worth mentioning that x_i' is considered positive when fingers are moving towards each other. The first term in this controller gives a constant intention of closing the gripper with a speed of V ; while the second term gives an intention of opening the gripper with a speed proportional to $N_i - \dfrac{[K_\mu |f_i|]}{\mu_i}$ which is an indication of excessive applied normal force.

A five fingers modular hand

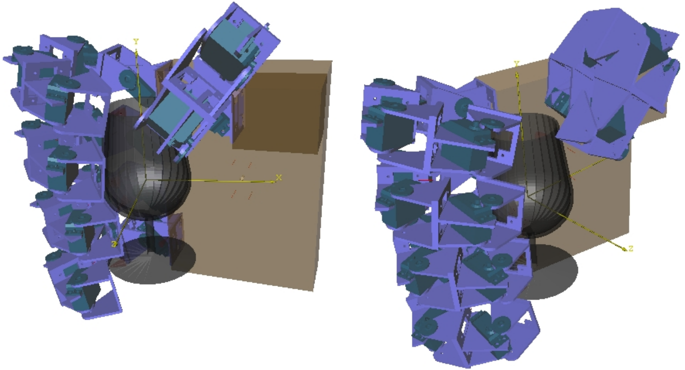
- We also realized a model of a hand with five identical fingers. Each finger has 4 DOFs.
- The first degree of freedom allow each finger the movement of adduction and abduction (Ab/Ad). The remaining three DOFs allow each finger the movement of flexion and extension (F/E). These pairs of movements are in the same plane for the four middle fingers, but for the thumb the planes of movements for Ab/Ad and F/E are slightly shifted forward compared to the other fingers.
- Anyway the model is very simple and the hand palm is approximated by a simple parallelepiped. At the top of this parallelepiped four fingers are placed. The fifth finger is located on another smaller parallelepiped attached to the palm so that it can imitate the human hand thumb.
- The model proportions are inspired by the human hand, even if the module size is greater than the human phalange.

A five fingers modular hand



Some simple grasping tasks

Using the new modular hand model we repeated the experiment of grasping a glass.



Dimensionality reduction

- Despite the simplicity of a modular hand model, with the increase in the number of its fingers and modules, it becomes rival to the human hand in complexity. In this case, one of the hardest problems is the creation of control algorithms.
- If we wish to reproduce human-like grasping it would seems natural to draw inspiration not only from the hardware of the human hand, but also from the software; that is, the way the hand is controlled by the brain.
- Recent studies in neuroscience research have shown that the grasping human hand movements can be described by trajectories in a configuration space of much smaller dimension than the kinematic count would suggest. Such configuration space is known as postural synergies.

The first two synergy components

- These hypothesis are supported by the work of Santello[4]. He asked different subjects to shape the right hand as if to grasp and use a large number of familiar objects. Static hand posture was measured by recording the angular position of 15 joint angles of the fingers and of the thumb. Although subjects adopted distinct hand shapes for the various objects, Santello noticed that the joint angles of the digits did not vary independently.
- Principal components analysis showed that the first two components could account for more than 80% of the variance, implying a substantial reduction from the 15 degrees of freedom that were recorded.

[4] M. Santello, M. Flanders, and J. F. Soechting, Postural hand synergies for tool use

Synergies and modular hands

- The Principal Components of the human hand synergies are also called eigengrasps. They are fundamental because they form a low-dimensionality basis for grasp postures, and can be linearly combined to closely approximate most common grasping positions.
- Based on these observations, we want to employ the two dominant human eigengrasps in order to imitate the human hand behaviour using our modular model.

Advantages

- Easy design. The concept of hand synergies allow us to simplifying the design and analysis of robotic hands.
- Flexible control algorithms. We can operate identically across all the possible modular hand models. The key to our approach is that the eigengrasps encapsulate the kinematic characteristics of each hand design.

Synergies and modular hands

- The concept of synergies is closely related to the human hand, but if we analyze the problem as a simplification of the grasp search space we can extend this concept also to the modular robotic hand.
- However, modular hand can be very different from the human one.

How to map the human synergies on a non-human hand?

- Since the configuration of our modular hand can change depending on the object to grasp and on the post-grasp task to be performed, we need to find a way to imitate the human hand behaviour.

Human hand synergies imitation

- Imitation is a powerful learning tool, but in this case the demonstrator that is the human hand and the imitator that is the modular robotic hand may not share the same embodiment (degree of freedom, body morphology, constraints, affordances, and so on).
- Alissandrakis, Nehaniv, and Dautenhahn proposed a method[5] to formalize body mapping using a unified (linear) approach via correspondence matrices, which allow one to capture partial, mirror symmetric, one-to-one, one-to-many, many-to-one, and many-to-many associations between various DOFs across dissimilar embodiments.

[5] Alissandrakis, Nehaniv, and Dautenhahn, Correspondence Mapping Induced State and Action Metrics for Robotic Imitation

Human hand synergies imitation: our contribution

Our idea

- We want to use a general imitation algorithm in order to map the humans postural synergies on the modular robotic hand.
- It should be noted that, using the concept of imitation, we are able to map the human synergies on all the possible modular hand configuration. In this way we can extend the eigengrasps advantages to the modular hand model. This is our scientific contribution.
- More generally, using this approach, we are also able to map the human synergies to a non-human hand, regardless the latter is modular or not or if it has a different embodiment.

Imitation as a correspondence problem

- Imitation is equivalent to solve a correspondence problem. We need to create an appropriate mapping between demonstrator and imitator.
- The demonstrator is the human hand, the imitator is our modular robotic hand.
- In particular, we should map the human postural synergies defined by the first two principal components on the modular robotic model in order to achieve corresponding state and effects.

Our assumption

For our goals, we can consider the postures as states to imitate and their changes in value as actions to be performed.

Problem definition

- Both human and modular robotic hands can be described as simplified kinematic models, comprising of a rooted acyclic connected graph of links.
- Each link has a base and a tip end, and is described by the quintuple ($\imath$, l_i, p_i, θ_i, φ_i) as follows:

- $\imath$ index number of the segment;
- l_i segment length;
- p_i index number of the parent segment;
- θ_i and φ_i azimuth and polar angles, respectively, for the spherical coordinates (l_i, θ_i, φ_i) that indicate how the segment is positioned in 3-D space (relative to the end of its parent segment).

Problem definition: states and actions

- The state of such a kinematic model can be defined as the vector containing the values of the degrees of freedom. Depending on whether the relative or the absolute angle values are used, for an embodiment with n segments, two different state vectors can be considered:

- $S_{relative} = [\theta_1 \varphi_1 \theta_2 \varphi_2 \ldots \theta_n \varphi_n]$
- $S_{absolute} = [\Theta_1 \Phi_1 \Theta_2 \Phi_2 \ldots \Theta_n \Phi_n]$

- An action can be defined as the difference between two consecutive state vectors S and S^I:

- $A = S^I - S$

State metrics

- In order to evaluate the similarity of behavior, with respect to states and actions, between an agent β (modular hand) imitating another agent α (human hand), we have to use appropriate metrics.
- firstly, let us assume that both agent embodiments have the same number of DOFs n. A first global state metric can be defined as:

- $\mu_{state}(S^{\alpha}, S^{\beta}) = \sum_{j=1}^{n} |S_j^{\alpha} - S_j^{\beta}|$

Where S_j^{α} and S_j^{β} are the values of the state vectors for the two agents.

Action metrics

- A first global action metric can be defined as:

- $\mu_{action}(A^{\alpha}, A^{\beta}) = \sum_{j=1}^{n} |A_j^{\alpha} - A_j^{\beta}|$

Where A_j^{α} and A_j^{β} are the values of the action vectors for the two agents.

Metrics minimization

An agent performing actions, in our case synergic actions, so as to minimize one (or a weighted combination) of these two metrics would successfully imitate a demonstrator in respect to states and/or actions.

Correspondence matrix

- So, let consider now two agents, demonstrator α (human hand) and imitator β (modular hand) with n and m DOFs, respectively. In this case an nxm correspondence matrix can be defined as:

$$C = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{pmatrix}$$

- where the $w_{i,j}$ values are real-valued weights, determining how the jth DOF of the imitator β depends on the ith DOF of the demonstrator α.

Metric definitions with correspondence matrix

- Since we want to develop modular robotic hands, in most cases they will have less DOFs than the human hand. So, in these cases, the agents do not have the same number of DOFs, and we have to map many DOFs to a single DOF.
- For doing this kind of mapping using a correspondence matrix, other metric definitions have to be induced.
- First, the state vector S^α and the action vector A^α of the demonstrator can be multiplied with the correspondence matrix:

$$S = S^\alpha \times C$$
$$A = A^\alpha \times C$$

Induced state and action metrics

- Remembering the state metric definition we can obtain:

$$\mu^{c}_{state}(S, S^{\beta}) = \sum_{j=1}^{n} |S_j - S_j^{\beta}| \epsilon_j$$

where S^{β} is the imitator's attempted matching state, and the corrective term

$$\epsilon_j = \begin{cases} 0, & \text{if} \quad \sum_{i=1}^{n} w_{i,j}^2 = 0 \\ 1, & \text{otherwise} \end{cases}$$

takes the value zero if the jth column of the correspondence matrix contains only zeros (effectively omitting the imitator's jth DOF).

- The imitator can match the state S by assuming state S^{β} so as to minimize the metric μ_{state}.
- In the same way we can also derive the new action metric:

$$\mu^{c}_{action}(A, A^{\beta}) = \sum_{j=1}^{n} |A_j - A_j^{\beta}| \epsilon_j$$

where where A^{β} is the imitator's attempted matching action.

Induced state and action metrics minimization

- The idea is that the demonstrator performs a series of actions, and the imitator tries to minimize the correspondence induced relative state metric. We want to imitate the first two components of the human postural synergies. So, in this instance, the human hand has to perform the actions related to the two first components of synergies, and the given non human hand should follow similar movements.
- Using the components of S for which $\epsilon_j \neq 0$ as the current sub goals for each DOF j, the imitator performs actions that attempt to reduce the contribution of error in each such component.

Object-oriented and Task-oriented grasping

Object-oriented

- We would like to take advantage from the flexibility offered by the modularity. First of all, we would like to adapt our modular hand to the object to be grasped.
- For this reason, it will be useful to develop an algorithm capable to determinate the minimum manipulator configuration for a specific object (how many fingers and how many modules we need).

Task-oriented

- We should also taking into account the planned post-grasp operation that has to be accomplished with the object.
- We would like to obtain, as much as possible, functional grasps. So we have to find the minimum hand configuration that allow the gripper to accomplish the grasp as well as the planned post-grasp operation.

Form- or Force-closure grasps are not enough

- In order to find the minimum configuration, we have to assess the grasp quality. This measure should be correlated to the post-grasp tasks.

And how can we evaluate a such task-oriented grasp quality?

- Commonly used criteria for grasp evaluation take only forces into account. Most of the currently proposed methods don't use any knowledge about the object functionality.

Reusability in form of actions

- A possible way to take into account the object functionality consists to describe its reusability in form of actions being performed after that the object has been grasped.
- Each grasp has a proper semantics according to the operation being performed with the grasped object. This semantics cannot be covered by a grasp evaluation criterion solely related to forces. Baier and Zhang proposed a criterion[6] to addresses this problem.
- They use four categories in order to describe the operations being performed with the grasped object. By describing these post-grasp-operations the semantics of the grasp can be defined.

[6] Baier and Zhang, Reusability-based Semantics for Grasp Evaluation in Context of Service Robotics

A set of actions

Furthermore this set of actions is described in a way which is understandable to humans, so that it is easy to use. The set P of post-operations is defined by:

- Pour-in
- Pour-out
- Handover
- Movement

Differences with respect to the classic grasping problem formulation

- Our problem differs from the classical grasping problem formulation.
- The first reason, as we said, is that the commonly used criteria for grasp evaluation are not task-oriented.
- Normally, the robotic hand is known. The framework that we want to develop does not know the hand configuration, but rather it has to determine it.

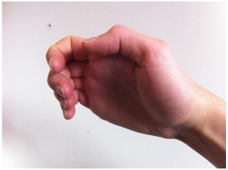
Grasp Preshapes

- In order to simplify the research of the minimum hand configuration we could take advantage from some grasp preshapes. The latters will be very useful for the grasp planning phase of our optimisation algorithm.
- Automatic grasp planning for robotic hands is a difficult problem because of the huge number of possible hand configurations. However, humans simplify the problem by choosing a different prehensile posture, called preshape, appropriate for the object and task to be performed.
- All the possible grasp preshapes depends on the complexity of the hand.
- Prats, Sanz and del Pobil have dentified a set of four task-oriented hand preshapes[7].

[7] Prats, Sanz and del Pobil, Task-Oriented Grasping using Hand Preshapes and Task Frames

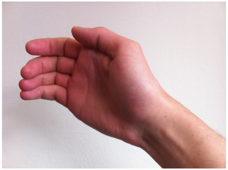
Task-oriented hand preshapes

Hook power



Precision

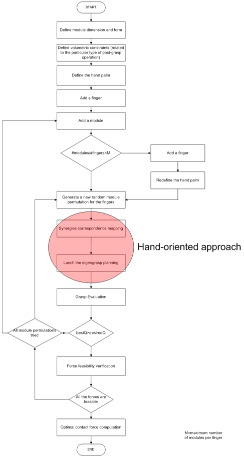
Hook precision



Cylindrical

A possible object-oriented algorithm

We want to determine which the minimum configuration is for a modular hand in order to grasp a given object for accomplishing a certain task. We want also to determine the optimal actuator efforts and corresponding contact forces.



Define module dimension and form

- We assume that the modules have all the same size.
- We can model the distal finger link of the hand as a cylinder, thus the ideal form for the modules would be a cylinder.
- We could also assume that the actuator limits are the same for all the modules.

Define volumetric constraints

- Depending on the particular type of post-grasp operation we have to define the appropriate volumetric constraints that allow to perform the specific task in a proper way. We also have to take into account the gain or loss in weight of the object due to its use after the grasp.
- As mentioned earlier, referring to the post-grasp operations set described by Baier and Zhang we can define the volumetric constraints.
- All the object semantic information could be represented as shape primitives, which are treated by the grasp planning as obstacles or must-touch regions of the object to influence the resulting grasps.

Define the hand palm

- Since we want to take advantage of the modular concept we have to apply this philosophy to the hand palm as well as we have done with the fingers.
- The hand palm will have a number of metacarpals equal to the number of fingers.
- The metacarpals will be made with the same modules of the fingers.

One finger hand or multi-fingers hand

One finger hand

- The palm will be made with only a metacarpal.
- In this case, all the grasp preshapes should be obtained using the only available finger.

Multi-fingers hand

- The palm will have two or more metacarpals.
- With two or more fingers, it's possible to realize a opposable thumb. One of the fingers can be seen as a thumb.
- All the hand preshapes are easily feasible. As the number of fingers increases, the hand preshapes will be more easily achievable.

Add fingers and modules

- The algorithm starts with a one finger manipulator. The only finger consists of just one module. Then the grasp planning is launched. It will find the best grasp for the current hand configuration.
- If the grasp quality is not satisfying, another module will be added. Hence, the system determines and assesses again the best grasp achievable with the current hand configuration.
- If the number of modules for finger exceeds the limit M (previously established and linked to problems of physical nature), another finger will be added to the hand.

Redefine the hand palm

- Each time that the system adds another finger to the hand, the palm has to be redefined.
- In practice, another metacarpal has to be added.

Generate a random module permutation for the fingers

When the hand has more than one finger, the system generate a random module permutation. Here an example:

finger	Number of modules per finger at the fist permutation	Number of modules per finger at the second permutation	Number of modules per finger at the third permutation	Number of modules per finger at the fourth permutation
thumb	1	2	3	4
index	4	3	2	1

Generate a random module permutation for the fingers

- One of all the possible permutations will be randomly drawn. Hence, the system determines and assesses the best grasp achievable with the current hand configuration.
- If the grasp quality is not satisfying, another permutation will be evaluated.

Grasp planning: forward or backward solutions

- Grasp planning tries to find the configuration of the robotic hand in which the hand grasps the object with contact points between the fingers and the object surface.
- The grasp planning problem can be solved in either forward or backward direction. The forward solution involves the finger forward kinematic to close the fingers and uses the collision detection technique to detect the finger joint positions at collision.
- The backward method is object centered.

Grasp planning: our choose

- A main drawback of the backward method is the need of collision checking, however, the inverse kinematic algorithm needs an unambiguous position of the contact point.
- For these reasons, we will use the forward solution for implementing our algorithm. Anyway, it should be noted that, thanks to the modularity of the algorithm, the planning phase is independent of the other steps.

Two possible approaches

- At this step, about the forward solution, we have two possible approaches:
  - object based grasp planning, that it is based on the concept of primitive decomposition for the grasped object;
  - hand based grasp planning, that is based on the concept of eigengrasps.
- initially, we will use an object oriented approach, using a primitive-based planner. Then, we will modify our algorithm in order to make it hand oriented using the hand synergies.

Grasp planning using a simulator

- In all the cases, the main goal of the grasp planning is the possibility to evaluate many hand postures quickly, and from a functional point of view (i.e. through grasp quality measures).

It should be noted that, thanks to the modularity of the algorithm, the planning phase is independent from the other steps.

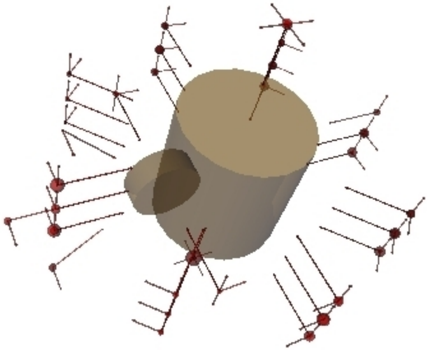
- To speed up the implementation phase of our algorithm we could use GraspIt! simulator. Its grasp planners are grouped in three families:

- the Primitive-based Planner;
- the Eigengrasp Planner;
- the Database Planner.

- For our algorithm, the first two families are useful. While the third kind of planner is not useful for us because it presupposes the existence of a database of pre-computed grasps. The latter doesn't exist.

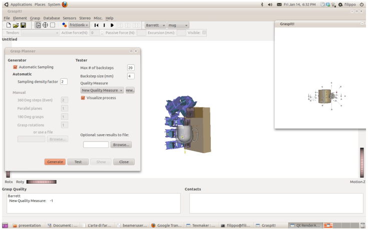
Object based grasp planning

This planner uses some hand preshapes before grasping to shrink the hand configuration space.



Grasp Testing

After the grasp starting positions have been generated, each grasp must be performed and evaluated. Since the grasp evaluation is by far the most time consuming operation, the system checks for infeasible hand configurations at each step of the grasp execution to avoid unnecessary evaluations.



Grasp Evaluation

After the object is grasped, the contact points between the hand and the object are collected to evaluate the grasp quality.

- We should consider firstly the traditional stability-oriented quality measures. So the grasp planner has to compute the two widely used quality metrics Q1 and Q2 that characterize the Grasp Wrench Space (GWS). This metod allow the system to determine if the grasp has force closure.
- Nevertheless, since our aim is to find stable grasps suitable with a given post-grasp operation, we should also take into account a task-oriented grasp quality measure (Q3).

A task-oriented grasp quality measure

Aleotti and Caselli have presented an effective method for task-oriented grasp quality evaluation based on a novel grasp quality measure[8].

- The proposed approach is based on the concept of programming by demonstration and interactive teaching, wherein an expert user provides in a teaching phase a set of exemplar grasps appropriate for the task. Following this phase, a representation of task-related grasps is built (Functional Wrench Space).
- When a grasp is generated by the planning system, its suitability for the task is assessed by checking it with respect to the FWS. Namely, the quality measure of a grasp is defined as the largest factor e 0 by which the GWS of the grasp to be evaluated can be scaled to fit in the demonstrated FWS.

[8] Aleotti and Caselli, Interactive teaching of task-oriented robot grasps

A global grasp quality measure

- At the end of this step, the grasp evaluation process should return the global grasp quality measure Q (that is the weighted sum of the three measures Q1, Q2 and Q3).

$$Q = \alpha Q_1 + \beta Q_2 + \gamma Q_3$$

- When the best grasp, among all the possible grips corresponding to the current hand configuration, exceeds or is equal to the desired grasp quality, the next step will appen.

Force feasibility verification

- At this step we have the locations of the contact points on the object and the hand, the corresponding friction models, the kinematic structure of the hand, the actuator limits. Knowing external load on the object and hand, we have to determine if the load can be balanced.
- If the force feasibility verification is positive the next procedure step will be executed, otherwise another module will be added to the hand and the algorithm will repeat its iteration.

Optimal contact force computation

- The found minimal hand configuration is accompanied by the correspondent grasp. If the fingers close exactly to these joint positions, the contact points are only touched by the fingers. No force or only little force due to offset error is applied through these contact points. Hence, it is necessary to determine the optimal actuator efforts and corresponding contact forces.
- Many studies have investigated this problem, to compute optimal forces with given contact point information.
- We could use the method[9] proposed by L. Han, J. Trinkle and Z. Li because it provides the optimal forces and part of it is already integrated in GraspIt!.

[9] L. Han, J. Trinkle and Z. Li, Grasp analysis as linear matrix inequality problems

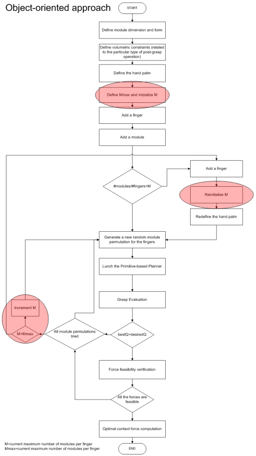
Termination

The algorithm ends when the best grasp, among all the possible grips corresponding to the current hand configuration, exceeds or is equal to the desired grasp quality and the force feasibility verification is positive. At the end of the process,always, the system returns:

- the minimum hand configuration that allow to grasp the object and to accomplish the post-grasp operation;
- the stable grasp;
- the optimal force necessary to grasp the object.

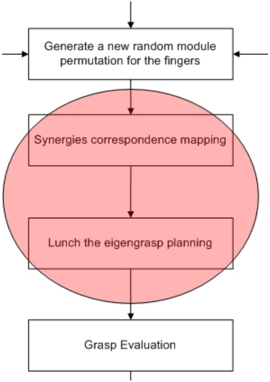
M as a non-fixed parameter

- M represents the maximum number of modules for finger. It is an important parameter because it determines the maximum number of the degrees of freedom of each finger.
- Increasing the value of this parameter we can realise non-human-like fingers and we may obtain hyper-redundant manipulator.
- We propose a possible variant of our algorithm that is more general than the first one. In our basic algorithm, the parameter M is fixed. A possible variation is to initialize $M = 1$. Then, at every step of the algorithm, M is increased until it reaches a certain M_{max}. When this limit is reached, another finger is added to the modular hand.



A possible hand-oriented algorithm

- The algorithm that we firstly developed doesn't take into account the human hand synergies. In order to overcome this lack we present a modified version of our basic algorithm that considers also the possibility to grasp objects using the human eigengrasps.
- Thanks to the modularity and flexibility of our basic algorithm, we can preserve its structure.
- However we have to add and modify some steps.



Eigengrasps planning

- Intuitively, the energy function used for classical planning problem formulations has to be related to the quality of the grasp.

Classical energy function

$E = f(p, w)$

- If d is the number of intrinsic hand DOF's then $p \in \mathbb{R}^d$ represents the hand posture and $w \in \mathbb{R}^6$ contains the position and orientation of the wrist. So, in this case, the parameter space has $d + 6$ dimensions.
- In order to reduce the parameter space dimensions, we can perform the optimization in eigengrasp space, as opposed to DOF space. The energy function takes the form:

New energy function

$E = f(a, w)$

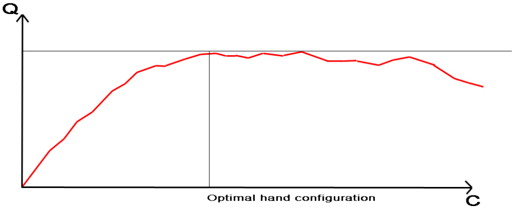
where $a \in \mathbb{R}^2$ is the vector of eigengrasp amplitudes. This effectively reduces the parameter space to 8 dimensions (2 eigengrasp amplitudes plus 6 extrinsic DOF's) from as high as 26 dimensions in the case of the human hand.

A possible algorithm for finding the optimal configuration

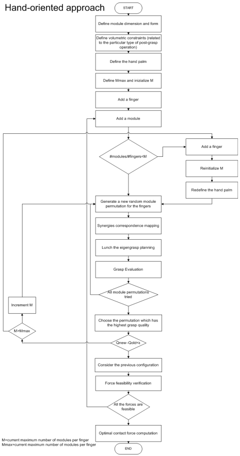
- Finding the minimum hand configuration is interesting, but even more useful would be to find the optimal hand configuration in order to grasp a given object and to accomplish a certain post-grasp task.
- To do this we have to modify once again our algorithm. It has to terminate when it is useless to add more fingers or modules because the quality does not increase or, even worse, it decreases.
- Initially, increasing the number of fingers and modules used, we can expect a grasp quality improvement. However, from a certain point we can expect that the grasp quality doesn't increase any more because the hand complexity doesn't allow a functional-grasp.

Termination

What we can aspect



Where $Q = Q_1 + Q_2 + Q_3$ and C represents the modules number.



The Columbia Grasp Database (CGD)

Nowadays, the most famous grasp database is the Columbia Grasp Database (CGD).

Disadvantages

- The downside of Columbia Grasp Database is that it is not task-oriented. Its approach to grasp planning is purely geometric. The lack of domain-specific knowledge means that some of our grasps are semantically incorrect, such as a mug grasped by placing the fingers inside, although they are still form closed.
- Furthermore, the measures of the grasp quality is too closely related to the particular type of the hand. It is not possible to know which is the best configuration of the hand for grasping an object in order to accomplish a particular task.

A new task oriented grasp database

For these reasons, it coulb very useful to build a new grasp database. Rather than considering object-hand combinations, it would be much more useful to consider object-task combinations. In the new grasp database, for each object-task entry, the output will be:

- the minimum hand configuration required to grasp the object (how many fingers, how many modules);
- the stable grasp;
- the traditional stability-oriented quality measures Q1 and Q2;
- the task-oriented grasp quality measure Q3 (that is defined as the largest factor e greater than 0 by which the GWS of the grasp to be evaluated can be scaled to fit in the demonstrated FWS);
- the global grasp quality criterion Q (that is the weighted sum of the three measures Q1, Q2 and Q3);
- the optimal force necessary to grasp the object;
- the achievable postural synergies.

Conclusions

What we have done:

- Modular robotic hand models capable to grasp objects using the human synergies
- A method to obtain the optimal mapping of human synergies for non-human hands.
- An algorithm capable to determine which is the minimum and the optimal hand configuration in order to obtain functional grasp. It also provides the stable grasp and the optimal force necessary to grasp the object.
- A modified version of our algorithm that is also capable to determine the achievable postural synergies.

What we have to do:

- Implement our algorithms.
- Realise a human sized modular robotic hand.

Thank you for your attention

I would greatly appreciate your precious questions and advises!