

Interactive Grasp Learning Based on Human Demonstration

Staffan Ekvall

Computational Vision and Active Perception
Royal Institute of Technology, Stockholm, Sweden
ekvall@nada.kth.se

Danica Kragic

Centre for Autonomous Systems
Royal Institute of Technology, Stockholm, Sweden
danik@nada.kth.se

Abstract—We describe our effort in development of an artificial cognitive system, able of performing complex manipulation tasks in a teleoperated or collaborative manner. Some of the work is motivated by human control strategies that, in general, involve comparison between sensory feedback and *a-priori* known, internal models. According to recent neuroscientific findings, predictions help to reduce the delays in obtaining the sensory information and to perform more complex tasks.

This paper deals with the issue of robotic manipulation and grasping in particular. Two main contributions of the paper are: i) evaluation, recognition and modeling of human grasps *during* the arm transportation sequence, and ii) learning and representation of grasp strategies for different robotic hands.

I. INTRODUCTION

The development of intelligent interfaces for robot programming in terms of service, medical and industrial robots is one of key research areas nowadays. Service robots and other intelligent devices are used by users that are unexperienced in programming and providing *programming by demonstration* frameworks has been an active area of research for the past few years, [3]. Our goal is to develop an artificial cognitive system acting in everyday environments capable of learning actions commonly induced by humans. Starting point is the development of elementary actions based on kinematics of the robot and some assumption about the environment. Our initial design is motivated by the fact that complex tasks can be modeled as a sequence of elementary ones where each elementary action is represented by a set of tractable constraints originating from the e.g. robot kinematics, task representation, type of sensory input, etc.

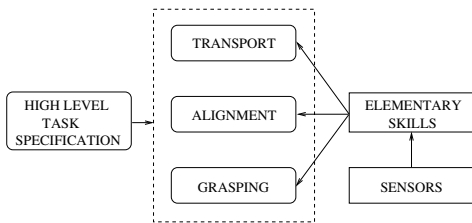


Fig. 1. Designing complex tasks using high level task specification and a set of elementary skills.

The focus here is on structuring actions by understanding, learning and imitating humans in two different settings: i) purely teleoperative one, [9], and ii) in collaboration with

human, [4]. In our previous work, we have presented a Human-Machine Cooperative System (HCMS) for augmented surgical manipulation tasks, [4]. These tasks are commonly repetitive, sequential, and consist of simple steps and can be effectively modeled using a set of basic primitives, where each primitive defines some basic type of motion. These steps can be "open-loop" (simply complying to user's demands) or "closed-loop", in which case external sensing is used to define a nominal reference trajectory. This methodology is also pursued in the current work.

To recognize objects and act upon them, humans typically use hand-eye coordination to identify the object, move the arm to the vicinity of it, preshape and align the hand and finally grasp and manipulate the object, see Figure 1. Even if this basic methodology has been known to us for quite sometime, there are still very few robotic systems that can robustly perform general tasks in realistic environments, [5].

This paper considers the alignment and grasping steps in particular. Two main contributions of the paper are: i) evaluation, recognition and modeling of human grasps *during* the arm transportation sequence, and ii) learning and representation of grasp strategies for different robotic hands where the recognition and mapping of human induced motion directly to a robotic hand are considered. We will show that our ideas can be used both in terms of programming-by-demonstration environments as well as collaborative ones.

Our approach offers a simple learning and control framework for systems with high degrees of freedom. As a comparison, in [11] and [13], the operator controls all degrees of freedom of the end-effector given a complex measuring device (Data Glove, [1]) in a purely teleoperated manner. We are interested in learning the *mapping* between a human hand and different types of robotic hands given a very simple measuring device - Nest of Birds, [2]. The important questions we try to answer are i) given the user's hand posture, what is the best robotic hand posture, ii) how should this mapping be defined so to fully use the dexterity of the robotic hand, and iii) what is the best sensor configuration in order to make this mapping optimal. The optimality of sensor configuration is considered for different hands and different grasp taxonomies, [6].

The paper is organized as follows. We start with a brief discussion of learning by demonstration paradigm in Section II and its application to mapping between of human and robot hands. Section III describes the equipment used in our work.

The developed methodology and framework are presented in Section VI. Experimental evaluation is presented in Section VII and a summary is given in Section VIII.

II. RELATED WORK

Learning-by-Demonstration frameworks, where the robot observes the human performing a task and is afterwards able to perform the task itself, use different sensory modalities for observation - vision, force-torque, haptics. To make the learning problem tractable, a hierarchical representation is commonly used depending on the complexity of the task at hand, [3].

A common learning strategy is to generate a number of training sequences and estimate the representation that is later used for on-line control of robot (robot hand) motion. One of the problems arising in human based learning settings is the one of measuring human performance. A popular approach here is to use sensors such as Data Glove to estimate mappings between high-dimensional configuration space of the glove to a lower dimensional configuration space of a robotic hand.

In [11], a Data Glove is used to learn mappings between a human and a 4-fingered artificial hand using Cartesian representation. Here, graphical representations of the respective human and artificial hand workspaces are used. In general, when considering the mapping between the reach spaces of the human and robot hands, the following problem occurs: While “tight” mappings use the reach space of the robot hand in an optimal way, it may happen that, since the workspace of the human finger only can be determined approximately due to the complex kinematics, some grasps may lead to fingertip positions which lie outside reach space of the artificial hand. [11] deals with this problem by back projecting these unreachable positions to the closest reachable position in the workspace.

Grasp recognition schemes based on the analysis of contact points are very powerful in recognizing different grasps but in order to estimate contact points, detailed geometric models of the object and hand are required. In addition, estimating contact points often requires high computational effort.

In our framework, a different strategy is used. Starting with grasps hierarchies, we map human (“teacher”) grasps to a set, S of “common grasps”. Since, in our framework, object manipulation in domestic settings is considered, we define the set S to consists of grasps common for typical pick-an-place tasks. We base our modeling on the grasps taxonomy shown in Figure 2 and proposed in [6]. The circled grasps are then those used for recognition *during* the arm transportation sequence, representing the set S .

A. Mapping Human Grasps to Robot Grasps

As the kinematics and configuration spaces of a human hand and an artificial robotic hand are generally different, the fingertip positions of the robotic hand cannot correspond exactly to the fingertip positions of the human hand (especially when fingertip grasps are considered). A good mapping between the human and the artificial fingers’ positions is required. In

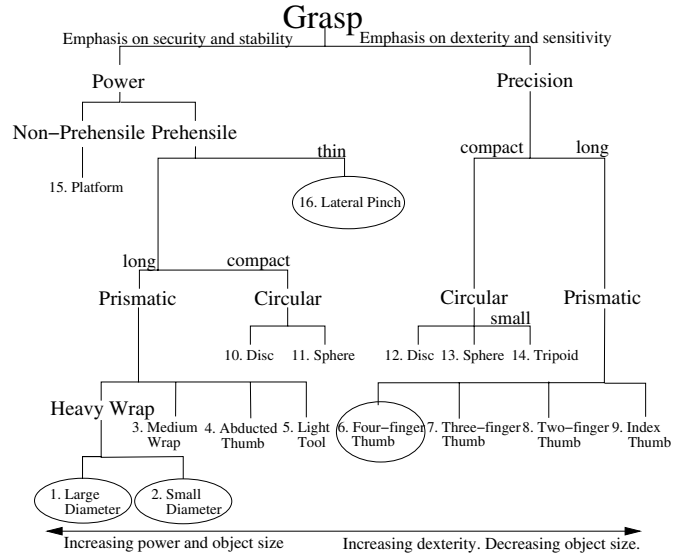


Fig. 2. Cutkosky’s grasp hierarchy.

general, there are two ways of representing mapping between these two spaces, using:

- **Joint space**

Mapping using the joint space representation facilitates the *similarity* between hands’ poses. This is suitable for enveloping or power grasps.

- **Cartesian space**

Mapping using the Cartesian space is more suitable for representing the fingertip positions. This is a natural approach when, for example, precision grasps are considered.

Related to the work presented here, a third group may be added to this list, namely the combination of the above. Here, the positions of the human hand’s fingertips are mapped to some joint values of the robot hand. If the robot and the human have similar hand configurations, the result is likely to be the same as for purely Cartesian mapping. However, if the robot hand’s kinematics is very different from the human one, as for example in the Barrett hand case, Cartesian mapping is not suitable. An example of this is given in Section VII, where we use two fingers to control the closure of the grasp and another finger to control the finger configuration of the robot hand. It has to be noted here that, even if we are using a grasping framework here as an example, the underlying design can be applied to that of learning human arm or leg trajectories in general.

III. THE SYSTEM

In this section, a short description of the system used to evaluate our learning strategy is given.

A. Nest of Birds

Nest of Birds, [2] is a magnetic tracker that consists of an electronics unit, a transmitter and four pose measuring

sensors. The sensors measure transmitter-generated magnetic fields. The electronic unit controls the transmitted signals and directs the sensor measurement. From signals measured by the sensors, Nest of BIRDS calculates the position and orientation of each sensor. Thus, each sensor has six degrees of freedom. This data is returned to the user via a Windows API.

The motivation for using this sensor is the possibility to mount sensors on different parts of human body (arm, hand, leg) and easily exchange their configurations. This allows us to, not only, learn trajectories and poses for human hand-arm-torso-leg motion but also evaluate different sensor configurations and choose the one that captures the representation space the best. Now, in terms of learning-by-demonstration-frameworks for service robots and natural interfaces for human-robot interaction, we believe that our approach has much stronger potential compared to dedicated sensors such as the Data Glove.

B. GraspIt!

In this project, we have used a grasping simulator, called GraspIt! [7], to analyze and visualize the poses of a variety of different robot hands which are not available to us in practice. The obvious advantage of using the simulator is the possibility to automatically repeat the experiments with slightly changed conditions and system parameters in a reinforcement learning manner.

GraspIt! can import a wide variety of different hands and robots, model environments with objects and all of these can be manipulated within a virtual 3D workspace. A custom collision detection and contact determination system prevents bodies from passing through each other and can find and mark contact locations. A dynamics engine can compute contact and friction forces over time.

IV. FEATURE EXTRACTION

The Nest of Birds individual sensors have been mounted on a glove, as shown in Figure 3. The center sensor, mounted on the back side of the glove, serves as a reference sensor. It measures the position and orientation of the hand. The remaining sensors are mounted on the thumb, index finger and little finger, respectively, and provide position measurements.



Fig. 3. The glove used for human input.

Each of the sensors estimate a 3D-position \mathbf{p} , calculated according to Equation 1, giving a total of nine values. The

position of the sensor is represented by x , y and z , and the reference sensor is represented by x_r , y_r and z_r . The rotation matrix M is calculated using the Euler angles ϕ , θ and γ , given by the reference sensor. As seen in Equation 1, the features derived from the sensors are both translational- and rotational-invariant, since the positions are multiplied by the transpose (inverse) of the rotation matrix.

$$\begin{aligned} p_x &= (x - x_r) \cdot M_{11} + (y - y_r) \cdot M_{21} + (z - z_r) \cdot M_{31} \\ p_y &= (x - x_r) \cdot M_{12} + (y - y_r) \cdot M_{22} + (z - z_r) \cdot M_{32} \\ p_z &= (x - x_r) \cdot M_{13} + (y - y_r) \cdot M_{23} + (z - z_r) \cdot M_{33} \end{aligned} \quad (1)$$

V. ACTION RECOGNITION

One important issue in a system such as ours is the recognition of the human intent in order to, for example, provide appropriate assistance *before the contact with the object has occurred*. We have modeled four different grasp types for the purpose of recognizing the typical grasp for four different objects: a toy car, a pen, a glass and a cup.



Fig. 4. The objects for which we are recognizing grasps.

Figure 4 illustrates these objects and, using the chosen grasp taxonomy, grasps are shown in Figure 2:

- Car is grasped with the *Large Diameter Grasp*.
- Glass is grasped with the *Small Diameter Grasp*.
- Cup is grasped with the *Lateral Pinch Grasp*.
- Pen is grasped with the *Four-finger Thumb Grasp*.

We investigate the possibility to recognize these grasps given a very simple measuring system. It is important to note here that most of the grasps shown in Figure 2 will be identical if only four sensors are used.

Similar to our previous work, we have investigated Hidden Markov Models (HMMs) for recognition, [4]. Each sensor contributes with one input dimension, giving a total of three dimensions for the HMM. Each of these can take on 27 different values, representing a change in position in 3D space (relative to the reference sensor). An element in the 3-dimensional input vector \mathbf{o} is determined according to

$$o_i = \begin{cases} 0 & \text{if } \|\Delta \mathbf{p}_i\| < 1 \text{ cm} \\ \arg \min_{j=1..26} (\|\frac{\Delta \mathbf{p}_i}{\|\Delta \mathbf{p}_i\|} - \mathbf{v}_j\|) & \text{otherwise} \end{cases} \quad (2)$$

$\mathbf{p}_{1,2,3}$ are calculated according to Equation 1. $\Delta\mathbf{p}_{1,2,3}$ are calculated as the difference between $\mathbf{p}_{1,2,3}$ and $\mathbf{p}_{1,2,3}^{old}$ representing the last measurement for which $\mathbf{o} \neq \mathbf{0}$. $\mathbf{v}_{1..26}$ are 26 discrete vectors representing the six basic directions in 3D, and their combinations, e.g. “up-right-forward”. This way, the direction is determined as the one of the 26 discrete vectors.

We assign a left-to-right structure to our HMM (SLR) and each model operates in parallel. By inspection of the transition matrices for different number of states, it was experimentally evaluated that five states was the optimal number. For training, 20 hand sequences are used for each grasp. The Baum-Welch algorithm is used to train the HMM. Only a few iterations are required to obtain stable values for $\lambda_j(A_j, B_j, \pi_j)$.

For each model λ_j , the total likelihood at time t is represented by $P(\mathbf{o}_{-N} \dots \mathbf{o}_0 | \lambda_j)$. Given λ_j , calculating the likelihood for the observations $\mathbf{o}_{-N} \dots \mathbf{o}_0$ using the Forward Procedure in general is a simple task. For an online implementation, when the observations are available continuously, it is slightly more difficult, since we do not know which observations belong to the current model.

We have solved the problem so that, for each new observation, the optimal sequence length is first calculated by

$$seqL = \arg \max_{i=minSeqL}^{maxSeqL} \left(\max_{j=1}^4 \underbrace{(P(\mathbf{o}_{-i}, \dots, \mathbf{o}_0 | \lambda_j))}_{\text{Forward Procedure}} \cdot \underbrace{(N_{obs})^{-N_d \cdot (maxSeqL-i)}}_{\text{Penalty term}} \right) \quad (3)$$

where $minSeqL$ and $maxSeqL$ are the minimum and maximum sequence length for a grasp. N_{obs} and N_d are the number of possible observations and dimensions in the HMM-model. In our case, we had $minSeqL = 3$, $maxSeqL = 20$, $N_{obs} = 27$ and $N_d = 3$.

Since short sequences have higher likelihood than long ones, with Equation 3 short and long sequences can easily be compared. Here, a penalty term that increases as the sequence length decreases was added. The penalty term represents the likelihood of observing a sequence of random \mathbf{o} , in addition to already available observations. It is assumed here that the finger movement likelihoods are uniformly distributed.

Once the sequence length has been established, the likelihood P_j for each model j is estimated as

$$P_j = P(\mathbf{o}_{-seqL}, \dots, \mathbf{o}_{-1}, \mathbf{o}_0 | \lambda_j) \quad (4)$$

The likelihood for the correct grasp is greatly decreased if the user performs an unexpected movement. If the grasp is predicted at that time, the prediction is likely to be wrong. But if the previous likelihoods are weighted according to

$$W(P(t)) = \sum_{i=-20}^0 \left(\frac{P(t_i)}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{(t_0-t_i)^2}{2\sigma^2}} \right) \quad (5)$$

the prediction is more likely to be correct. Equation 5 describes a filtering with a Gaussian kernel, where $(t_0 - t_i)$ is the time

passed since observation \mathbf{o}_i , and $P(t_i)$ is the likelihood calculated at time t_i , when \mathbf{o}_i was observed. We have experimentally found that $\sigma = 1$ is a good compromise.

VI. POSTURE MAPPING

We have decided to use artificial neural network (ANN) modeling to represent the mapping between the user and the robot hand configuration spaces. The basic steps involve the sampling of a number of typical human poses, matching those directly to robot hand poses and using those for training. The resulting ANN then represents the entire mapping surface between these two spaces.

The neural network used is a two-layer feedforward network as illustrated in Figure 5. Nine input variables calculated from Equation 1 are fed into the network and the outputs are the joint values in degrees. A sigmoid function is used in both the hidden and output layer. The output layer size is equal to the number of degrees of freedom of the robot hand.

For the Robonaut-hand case, see Figure 9, nine hidden neurons are used. For the Barrett-hand, see Figure 7, four hidden neurons is sufficient. To match the sigmoid function, the input and target values are scaled to the range [0,1]. The ANN is trained using the back-propagation algorithm.

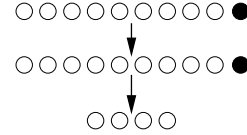


Fig. 5. A two-layer network is used for mapping. The black circles represents the bias units (always=1).

A. Training

First, the user wearing the glove demonstrates a pose to the system which is captured when a key is pressed. Then, Graspit! is used so that the user drags the links and joints of the hand to achieve the corresponding pose. These two poses constitute then a training sample for the ANN. When enough training samples have been gathered (usually only a few is sufficient), the ANN is trained. This step takes only a few seconds on a regular Pentium 450MHz PC.

VII. EXPERIMENTAL EVALUATION

A. Action Recognition

For testing, all four objects were grasped ten times. The likelihoods $W(P_j)$ for each object j were calculated according to Equation 4 and 5 as the grasp was performed. Figure 6 illustrates the grasp likelihoods, as a car is being grasped. The likelihood for the car is initially lower than for the glass, but as the hand is formed to grasp the car, the likelihood increases which proves the validity of Equation 4. As a grasp is being executed, $seqL$ typically increases from 3 to 10, indicating that the sequence segmentation also works. In the graph, the likelihood for car is higher already at $t = 0.35$ s, which is about when the grasp is half-done; the hand has been formed

TABLE I
GRASP RECOGNITION

HMM-Classification	Object Grasped			
	Car	Pen	Cup	Glass
Car	7	0	0	1
Pen	3	10	0	0
Cup	0	0	10	0
Glass	0	0	0	9

TABLE II
GRASP INTENTION RECOGNITION

HMM-Classification	Object Grasped			
	Car	Pen	Cup	Glass
Car	5	2	0	1
Pen	4	7	7	0
Cup	0	1	3	0
Glass	1	0	0	9

and is ready to perform the grasp. This supports our idea of recognizing grasp intentions, in addition to just recognizing grasps.

Table I shows the recognition results. 90 % of the grasps were successfully recognized. The car was the most difficult object to recognize, as the grasp movements for the car and the pen were similar. Table II shows the most probable object when the grasp is half-done; i.e the likelihoods are calculated at $t = t_{done}/2$ where t_{done} is the time when the grasp is complete. The initial recognition rate in this case is 60 %.

Despite the low recognition rate for the grasp prediction, the information may be fused with other sources(e.g. visual cues) as a feed forward information both in terms of teleoperative and collaborative settings. Preparing the grasp on-line, before the final alignment with respect to the object is done, is a common property of human grasping and has not yet been explored to a large extent in robotic systems.

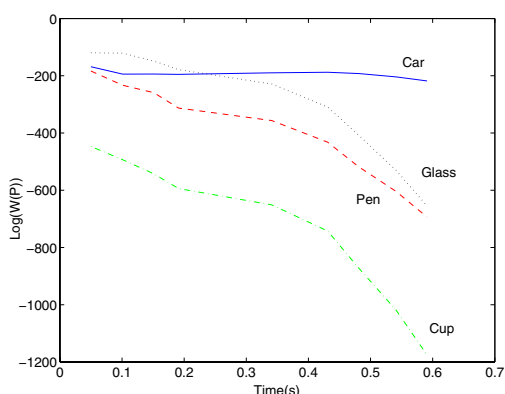


Fig. 6. Online recognition of hand actions. The graph shows the filtered likelihoods of each HMM as the car is being grasped.

B. Grasp Mapping

Once the grasp is recognized, it can be mapped to a robot hand. We have tested the system using two kinematically



Fig. 7. An example mapping of a human grasp using the Barrett Hand.



Fig. 8. An example mapping of a human grasp using the Barrett Hand.

different robot hands, the Barrett hand and the Robonaut hand.

The Barrett Hand has four degrees of freedom: one of the fingers is static while the movement of two remaining fingers is defined by a spread angle. The NASA Robonaut Hand [12] has a total of fourteen degrees of freedom. It consists of a two degree of freedom wrist, and a five finger, twelve degree of freedom hand, which is broken down into two sections: a dextrous work set which is used for manipulation, and a grasping set which allows the hand to maintain a stable grasp while manipulating or actuating an object. The initial results are very promising and the control is intuitive and simple, even in the case of the 14-DOF Robonaut hand.

Figures 7 and 8 illustrate some typical mappings from a human hand to the Barrett hand in which each sensor-mounted finger controls a finger of the Barrett hand. Figures 9-10 illustrate the mapping from human hand to the Robonaut hand. Note here that we do not have enough sensors to cover all fingers, so therefore the little finger has to control three fingers of the robot hand simultaneously. Of course, it is possible to demonstrate another behavior, for example, letting the index finger and the little finger control two fingers each. Even if we are constrained here by the number of sensors, the underlying methodology can easily be extended if more sensors are available.

We have experienced that the kinematic differences between the Barrett and the human hand made it very difficult to control the fourth degree of freedom of the Barrett hand, i.e. the spread angle motion. Therefore, this joint was fixed during the first training sequence and only three degrees of freedom were controlled (the angle closure of each finger). However, to demonstrate that our system is capable of controlling all four DOF, we also present an alternative training scheme, see Figure 11-12. This setup allows grasping with all robot fingers at once, by using the thumb and the index finger. The fourth DOF is controlled by the little finger. This type of control is not very intuitive but may be necessary for certain tasks.



Fig. 9. An example mapping of a human pointing grasp using the Robonaut Hand.

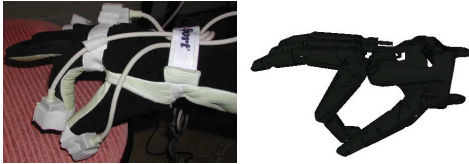


Fig. 10. An example mapping of a human pinch grasp using the Robonaut Hand.

VIII. CONCLUSIONS

We have presented our initial design of a system capable of performing grasping tasks in a learning-by-demonstration framework. The design methodology is motivated by human control strategies that, in general, involve comparison between sensory feedback and *a-priori* known, internal models.

Two main contributions of the paper are evaluation, recognition and modeling of human grasps *during* the arm transportation sequence and learning and representation of grasp strategies for different robotic hands. Our approach offers the possibility of evaluating hand designs depending on the human grasps typical for certain environments. Currently, a similar



Fig. 11. An example mapping of a human grasp using the Barrett Hand.



Fig. 12. An example mapping of a human grasp using the Barrett Hand.

evaluation idea is used in GraspIt! but the evaluation is based on grasp stability rather than on the grasp “importance” given the task knowledge.

One of the problems we have encountered is that certain poses of the human hand “lock” the manipulator hand. The fingers of the manipulator hand may collide even though the fingers of the human hand do not collide, depending on how the mapping is learned. An example of the can be seen in Figure 8. This represents a problem since if one of the fingers when the hand is closed locks another finger, the fingers have to be “unfolded” in the correct order, i.e. the finger on top should be opened first. But if the user unfolds his/her grasp in another order, it results in a situation in which the manipulator hand is closed while the human hand is open. We are currently solving this problem by incorporating a simple path planner, i.e. given a certain pose, it plans how to reach the new pose.

For grasp intention recognition, Cutkosky’s hierarchy is not useful, since many grasps have the same initial movement. Future work will be concerned with developing a less complicated hierarchy, in which for example the *Large Diameter* and *Small Diameter* will fall in the same category. Finally, the tests have been performed only in the simulator. Our current work considers testing these ideas on a real PUMA-arm and a Barrett-hand attached to it.

REFERENCES

- [1] The 5DT Data Glove 5, [<http://www.5dt.com/products/pdataglove5.html>]
- [2] Nest of Birds, [<http://www.ascension-tech.com/products/nestofbirds.php>]
- [3] H. Friedrich, R. Dillmann and O. Rogalla, Interactive Robot Programming Based on Human Demonstration and Advice, Christensen et al (eds.): Sensor Based Intelligent Robots, LNAI1724, pp.96-119, 1999.
- [4] D. Kragic and G.D. Hager, Task Modeling and Specification for Modular Sensory Based Human-Machine Cooperative Systems In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. IROS’03, Las Vegas, USA
- [5] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic and H.I. Christensen, Systems Integration for Real-World Manipulation Tasks, In IEEE International Conference on Robotics and Automation, ICRA 2002, Washington, USA
- [6] M.R. Cutkosky, On grasp choice, grasp models and the desing of hands for manufacturing tasks, IEEE Transactions on Robotics and Automation, 5(3):269-279, 1989.
- [7] A. T. Miller and P. K. Allen. GraspIt!, A versatile simulator for grasping analysis, Proc. of the ASME Dynamic Systems and Control Division, volume 2, pages 1251 1258, Orlando, FL, 2000.
- [8] S. B. Kan and K. Ikeuchi, Robot task programming by human demonstration: mapping human grasps to manipulator grasps, Intelligent Robots and Systems ’94. ’Advanced Robotic Systems and the Real World’, IROS ’94. Proceedings of the IEEE/RSJ/GI International Conference on, Volume: 1, 12-16 Sept. 1994. Pages: 94-104 vol.1
- [9] T.B. Sheridan, Telerobotics, Automation, and Human Supervisory Control, Cambridge: MIT Press, 1992.
- [10] I. N. Durlach and S. N. Mavor, Virtual Reality: Scientific and Technological Challenges, 1994, pp. 304-361.
- [11] M.Fischer, P.van der Smagt and G.Hirzinger, Learning techniques in a dataglove based telemanipulation system for the DLR hand, Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, Volume: 2, 16-20 May 1998, Page(s): 1603-1608 vol.2
- [12] C.S. Lovchik and M.A. Diftler, The Robonaut hand: a dexterous robot hand for space, Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, Volume: 2, 10-15 May 1999 Page(s): 907-912 vol.2
- [13] C.P. Tung and A.C. Kak, Automatic Learning of Assembly Tasks Using a DataGlove System, Intelligent Robots and Systems 95. ’Human Robot Interaction and Cooperative Robots’, Proceedings. 1995 IEEE/RSJ International Conference on, Volume: 1, 5-9 Aug. 1995, Pages: 1-8 vol.1