A Generalisation of:
A Simple Set of Grasps for a Dextrous Hand[1]

Damian Lyons

COINS Technical report 85-37

November, 1985

Laboratory for Perceptual Robotics
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

A version of this paper was presented to the IEEE *Conference on Robotics and Automation*, 25-28th of March, 1985, St.Louis, Missouri.

Abstract.

This paper describes a high-level control mechanism for a dextrous hand. It suggests a general framework for grasping and manipulation, and develops a particular set of grasps in this framework. An important feature of this framework is that it implictly emodies task requiements as parameters to the grasping process. It overviews briefly a distributed environment for robot control currently under development, and discusses the implementation of the grasping framework in this environment. Finally the paper speculates about the role the grasp can play in obstacle avoidance and vice-versa.

---

[1] Preparation of this paper was supported in part by grants ECS-8108818 and DMC-8511959 from NSF

1 Introduction

One of the most challenging problems in Artificial Intelligence (AI) currently is the attempt to build a robot with a measure of versatile behavior comparable to a human. The hand of man is, arguably, his most versatile appendage. The control of a ***dextrous robot hand*** addresses therefore those areas in which current robot technology is deficient: effective use of multiple degrees of freedom, integration of visual and touch information, and a versatile task specification. Several such dextrous hands have appeared in the robotics literature[4][11] [13][16][18].

In this paper we describe a general framework for grasping and manipulation. The purpose of this framework is firstly to provide a global structure into which to fit current work on grasping, and secondly to define the mechanism by which task constraints control grasping. This research is part of an on-going project at the Laboratory for Perceptual Robotics (LPR) at the University of Massachusetts to provide a high-level control architecture for a dextrous hand.

We are developing a special distributed computational model for the control of complex robot systems such as dextrous hands [14]. A distributed robot control system offers the advantage of efficient use of parallel computing resources[17]. Additionally, there are characteristics of the robot domain which are inherently parallel [1][6]; we are constructing our model to implicitly account for these. It has become evident in robotics that versatile behavior is not possible without adequate sensory information, and an appropriate high-level control architecture to integrate perception and action. Robotics research based on these concepts has been referred to as ***perceptual robotics*** [8] [10]; this is a cornerstone of our distributed model.

2 Previous Work in this Area

In the current literature there exists no overall framework for grasping and manipulation. As a matter of fact, almost all the work has been bottom-up; the construction of dextrous hand mechanisms, or determining what features a particular hand mechanism can offer.

Hanafusa and Asada [1977] were among the first to consider the stability of a multifingered grasp. For one-DOF elastic fingers, they defined stability as the ability of the grasp configuration to counter small object displacements with correcting forces. This work has been extended by Baker et al. [1985] using a modified hand model. They investigate stability for two classes of grip configuration: Their *triangular grip*, where a maximal inscribed circle on a 2-d polyhedral object touches the boundary at three points, with less than 180 degrees between successive points, and the points then correspond to finger positions; and their *parallel grip*, where the maximal inscribed circle touches the boundary at two points.

One of the few papers to address the task requirements on grasping is Cutkosky [1984,1985], who develops an explicit list of *testing criteria* to investigate a particular grasp configuration (characterized by the arrangement of fingers on the object, and the stiffness and kinematic design of the fingers) in terms of: Stiffness in response to externally imposed loads; resistance of a grasp to slipping in the face of applied torques and forces; and the ability of the grasp configuration to counter small displacements with correcting forces. His fundamental motivation is that the robot may choose between a number of grip configurations to determine which are the most suited in view of expected requirements. He makes the point that in the presence of static friction there are many grips which will satisfy static equilibrium (net forces and torques on the object are zero), and it is possible to find the one best suited for a given task. This approach comes nearest to ours.

Salisbury [1982] designed a three-fingered, articulated robot hand, and the necessary theory to enable the hand to be used for object manipulation. He analysed object-finger contact conditions into a number of categories and used these to define the *mobility* and *connectivity* of a grasped object and hand. His immediate goal was not to analyze grasp stability but to determine what contact conditions can best guarantee complete object restraint when finger joints are locked and complete object control when the finger joints are active. He used this information to select a good design of dextrous hand. For this hand he developed the *grip transform* matrix; which can be used to transfer small motions and forces to a grasped object.

From our point of view, previous work is lacking in a number of areas. Firstly, it provides no clear or uniform structure in which to see the problems of grasping and object manipulation. This has come about because of the strong bottom-up tendancy in the field. A uniform framework would structure the work which has already been done and indicate where future work should be done to complete the whole picture. Secondly, few if any hints are given as to how to relate the requirements of a particular task to the way in which the hand is used. In this paper we attempt to rectify both of these problems by formulating such a general framework for grasping and manipulation, and by providing explictly for task requirement input in the framework.

In the next section we present our general framework, and define a set of grasps to work with within this framework, which we'll call a *simple set of grasps.* We then indicate how current work can be fit into our framework in a *mutually enriching* fashion. In a simple assembly example we explore our approach from the task requirement point of view. Finally we consider implementing this framework in a distributed programming environment.

3 A General Model of Grasping and Manipulation

The ways in which a complex robot system such as a dextrous hand can interface to the environment are many. We partition the set of all such interactions into a number of *domains of interaction*; each with specific characteristics. When the hand is required to grasp an object, a particular domain is selected to work within by considering object features and the operation to be carried out. Within each such domain, hand control is accomplished by a specific *approximation* to the power of the complete hand. Correct choice of domain will mean that the aspects of the hand highlighted by the approximation will be the ones important for this object and operation, whereas the aspects ignored by approximation will be the ones not relevant. Using this partitioning concept we define an abstraction of the hand called a *Grasp* which will embody the *approximate model for a domain of interaction.*

We identify *specific characteristics* with each domain of interaction. The task requirements and target object characteristics can then be used to choose an appropriate domain of interaction. A similar concept is suggested by Cutkosky [1985] for selecting gripping positions which are best suited for a task. However, the metrics proposed are different, and Cutkosky is selecting the gripping configuration *only.* A grasp G is a triplet $(\mathbf{P}_g, \mathbf{A}_g, \mathbf{M}_g)$ consisting of:

- *Preshape Component,* $\mathbf{P}_g$:
  The preshape component simultaneously configures the hand to the grasp preshape configuration and carries it to the vicinity of the target object. The preshape configuration consists of a parameterizable hand 'shape', and a preshape co-ordinate system. The hand 'shape' provides a receptacle for the object; the finger placement is such as to facilitate secure gripping in the acquisition stage, and later in object manipulation. We shall refer to this 'receptacle' as the *preshape volume.*

- *Acquisition Component,* $\mathbf{A}_g$:
  The hand approaches the object along a defined ***approach axis*** of the preshape coordinate system. When the object center of mass[2] coincides sufficiently closely with the origin of the preshape coordinate frame[3], which we shall refer to as the *focus* of the preshape, the preshape phase terminates and the acquisition begins. We leave the discussion of ***static grip equilibrium***, ***grasp stability***, and ***acquisition techniques*** to later sections. There are three cases for the gripping phase:

  1. Simple Grip. The object can easily enter the approach volume up to the *focus*. The grip can proceed by moving the fingers in until they touch the object.
  2. Two Phase Grip. The object can enter the approach area, but cannot proceed far enough into the approach volume to trigger the grip. The solution is to pick the object up with the fingertips first and then maneuver it into its final position.
  3. Constrained Grip. Due to the surrounding environment the object cannot enter the approach area of the grasp. For example the preshaped hand may be too big to come near the object in a cluttered workspace. At this stage it is necessary to consider deforming the preshape for the purpose of obstacle avoidance.

  The constrained grip can be approached by reducing the size of the preshape volume; however this may compromise grasp function. The constrained case can also be seen as a special case of the two-phase grip. For example a fingertip grip may be applied to acquire the object, which is then manipulated into some other grip position. This manipulation is outside the scope of this paper, and from now on only the simple grip is considered.

- *Manipulation Component,* $\mathbf{M}_g$:
  The manipulation component provides grasp-specific information on how the object can be manipulated. Not all grasps will provide the same amount of dextrous (finger) manipulation ability, and wrist and arm movement may be necessary to achieve a particular object configuration. Knowledge about the ***limits*** on dextrous manipulation, as well as appropriate information for object to finger transforms and object to wrist/arm transforms are necessary.

---

[2] Our later object definition will broaden the concept of center of mass to, in general, allow grasping on any part of an object

[3] This is a hand/fingertip relative point

We have now described our basic framework for grasping and manipulation. However, the framework is still to global and abstract to be useful as it stands. For this reason, we develop a set of grasps within the framework.

3.1 Domain Characteristics

The first step in realizing this approach is to determine useful characteristics with which to partition hand usage into specific domains of interaction. This paper is written in the context of the *robotic assembly* domain. Most practical examples of this work today are done by robots equipped with two-fingered, parallel-jawed grippers. However examination of these examples is not enough since a dextrous hand offers potential *not available* with a two-fingered gripper. There does, however, exist an extensive literature on human grasping[15][19]. We argue that the following are a useful set of characteristics:

Grasp Function:

By grasp function is meant the part played by the grasp in completing some assembly operation. Throughout this paper it is assumed that *object usage is known in advance of object grasping.* One important functional characteristic is the amount of precision movement required in the operation. In a dextrous hand, precision movement can only be carried out using the fingers. However, finger movement can only be done for small objects. Another important functional characteristic is the security of affixment between hand and object. Some tasks, such as wielding a hammer, may require firm affixment. Others, such as placing a small object, may not require such firm affixment.

Object Characteristics.

Two important object characteristics are *size* and *shape*. Objects which are large with respect to the hand cannot be lifted by a fingertip grasp. We roughly classify size into *large* and *small* with respect to the hand. Objects which have flat faces can often be grasped better in certain hand configurations than those with curved surfaces, and vice-versa. We divide objects into *round* or *flat* depending on whether their flat edges or curved edges are the dominant characteristics. We also define objects as long or short with respect to their aspect ratio. In a later section we discuss how to embody *continuous* rather than discrete characteristics. We place a constraint on the analysis at this point — the object description is assumed to refer to the complete object *or* to the grasp-site on the complete object, whichever is appropriate. Grasp-site is an *input parameter* to our model, and one which allows for direction of grasping activity by a high-level process[21].

Principle of Parsimony:

The *description* and *computation overhead* associated with certain grasps will be larger than those associated with others. The grasp chosen for a particular combination of object and operation should be no more complex than required. This provides a way to choose between grasps when necessary.

Object Location:

There is no doubt that the location of the target object also plays a part in determining appropriate grasps. A constraint on a preshape configuration is that it be able to move close enough to the object to acquire it. There has been some work in this area with regard to two-fingered grippers [12] [22]. The approach adopted by this paper is to *modify* a chosen preshape according to the environment. Thus the local object environment does not play a direct part in the initial partitioning into grasps.

These characteristics can be seen as describing an abstract space in which all possible grasps and objects in our framework are points. One basic problem in the grasping literature surface here, however. It is difficult to consider a grasp *separately* from the hand mechanism on which it is implemented. So far, our analysis has been independent of hand mechanism and we would like to keep it that way. To maintain this independence we use the *virtual finger* mechanism, first discussed by Arbib et al.[1].

Virtual Fingers

It was noted in [1] that the same "grasp" may use different physical fingers, depending on object characteristics. They noted that the mug-grasp can be described using three components: a downward force from above the handle; an upward force from within the handle; and a third force to stabilize the handle from below. However, different size mugs would require these three components to be supplied by *different* fingers; thus the grasp must be defined independently of physical fingers, and only mapped onto physical fingers when the details of the object are known. They called these logical units *virtual fingers*. The concept of the virtual finger provides independence from both physical hand structure and from particular object models. In its abstract sense, a virtual finger is simply an element of task description, and could be represented as a force vector or system of force vectors[9].

We here define a virtual finger as a particular *logical* mechanism, having an associated *mapping set*. For simplicity, we assume the logical mechanism is a *3-DOF* finger as in figure one. For each particular hand mechanism a mapping from the joints on the virtual

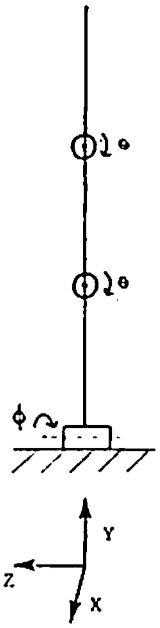

Figure 1: The virtual finger as a 3-DOF logical mechanism

finger to those on the physical finger must be provided. The *i*th virtual finger is denoted by VF_i; and its *mapping set* by $VFset_i$, which contains the indices of fingers considered part of VF_i. We shall allow the use of $VFset$ to specify *similar* actions[14]; that is similar response to sensory input; as well as *identical* actions such as when all fingers execute identical motor operations.

3.2 An Acceptable Set of Grasps

We now construct an *acceptable* set of grasps to work with within our general framework; defining acceptable by the following arguments:

Salisbury's [1982] goal was a hand design which completely restrained the grasped object, as well as being able to impart arbitrary forces and small motions to it. We argue that in an assembly task, a *range* of manipulation and affixment abilities are called for. For example, stacking objects on top of each other will not require the same manipulation abilities as threading a nut onto a bolt. In addition, precision of manipulation is not a goal at all for some types of grasping; instead affixment of hand to object is the control

Contact Types	With Friction	Without Friction
Point Contact	5	3
Line Contact	4	1
Planar Contact	3	0

Figure 2: Contact Conditions

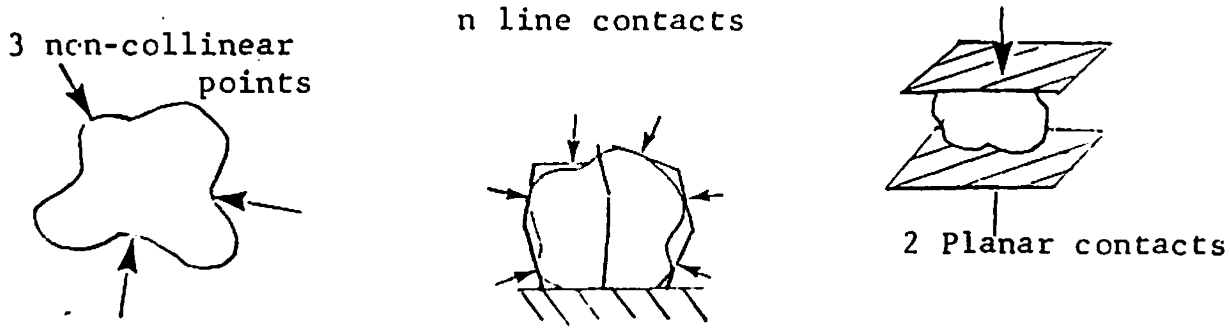

Figure 3: Three useful contact configurations

parameter. For example, in hammering a nail the hammer must be tightly clasped; in delicate insertion it may be preferable for the hand to *slip* from the object rather than to damage the object. We define an acceptable set of grasps in our framework as a set which allows a range of control on manipulation and affixment abilities. To develop an acceptable set of grasps, we use Salisbury's analysis of constraints and freedoms with the note that a grasped object will reciprocate the category of contact offered by the hand. This will be true if we assume that the contact sites on the object are *planar* with respect to the fingers. The figure shows the *remaining* freedoms due to contacts of point with plane, line with plane, and plane with plane, respectively. We shall consider the fingertips to exhibit point contact on the object, finger phalanges to exhibit line contact, and the palm, or a number of fingers, to exhibit planar contact. For example consider a fingertip contacting an object; this is a case of point contact, and there are 5 (3 with friction) dof remaining to the object because of this fingertip contact.

The three contact configurations in figure 3 are the basis for a set of acceptable grasps. In order to outline why, we need to define rough measures of how well an object can be manipulated within a grasp, and how rigidly an object can be held within a grasp. Connectivity between the grasped object and the palm, with the hand joints active, is used as the measure of how well the object can be manipulated in this grip. The connectivity between two bodies in a kinematic system is defined as ***the number of independent parameters necessary to completely specify their relative positions.*** If the connectivity between object and palm with the joints active is 6, then the hand can exert any manipulation on the object. Connectivity less than 6 means that the manipulative abilities of the hand are constrained.

Three non-collinear point contacts can be chosen to completely restrain a grasped object[18]. If more contacts are then added, the object is ***overconstrained.*** This redundancy may provide the grip configuration with the ability to withstand disturbances at some of its contact points (due to slippage, etc.), in a superior fashion to a minimally constrained object. In order to rigidly grasp an object, the grip configuration must be very stable in the presence of disturbing forces. One approach to this is to examine the surface of the object for contact points which provide the most stable grip[7],[2]. The solution depends on the detail with which the object boundary is known. A way to facilitate stable grasp without too much computational overhead and with minimum object knowledge is to overconstrain the object with contact points. Unfortunately in dextrous hands, contact points are a scarce resource, the more joints which are used as contact points, the less joints are available for fine manipulation of the object. The maximum ***excess constraints*** (from the DOF contact table) on the object at the contact points is used as a measure of the rigidity of the grip.

Case 1 of figure 3, (3(1)), minimally constrains the object, while 3(2) and 3(3) increasingly overconstrain the object with contact points. Consider, however, implementing these cases on a dextrous hand: 3(1) can be implemented using fingertips as the point forces; 3(2) by using the finger phalanges as the two planar surfaces; 3(3) by wrapping each finger around the object in turn. The connectivity between object and palm with joints active is a maximum in configuration 3(1); and should preferably be six. Since 3(2) blocks some of the finger joints, the connectivity must be less than that in 3(1). However, 3(3) blocks ***all*** finger joints, the object cannot be manipulated at all by the fingers, and the connectivity is zero. Thus the contact configurations in figure 3, when implemented on a dextrous hand provide a range of abilities in manipulation and affixment: 3(1) allowing the most manipulation but least affixment; 3(3) allowing no manipulation, but very firm affixment; 3(2) allowing some manipulation and some affixment.

Object size and shape plays a part in the applicability of the grip configurations. Fingertip contact, such as that in figure 3(1), can only be achieved if the object is small with respect to the hand. Planar contacts such as that in figure 3(2) are best applied to objects which have planar (preferably parallel) faces, and objects which have a long axis (in which case the long axis should lie along and between the planes). Figure 3(3) has wide object applicability, and is more suited for objects with curved faces than is 3(2).

3.3 A Simple Set of Grasps

Based on the three contact configurations of figure three we construct the following three grasps (and see figure 4):

An Encompass Grasp: has the characteristics that the fingers are used to envelope the object in a compliant fashion. The goal of the grasp is to attach the object rigidly to the hand. manipulation can only occur through movements of the wrist – the fingers operate just to securely grip the object.

A Lateral Grasp: has the characteristics that the object is gripped not with the fingertips, but with the inner planar surface of the digits. Unlike the encompass grasp the fingers do not envelope the object, they merely hold it in a vice-like grip, similar to that of a two-fingered gripper. Using the phalanges rather than the fingertips overconstrains the object at the contact points, however not enveloping the fingers around the object still leaves *some* joints free to effect some precision manipulation. It provides a firm way to grip flat surfaces, the object size dictating how many fingers are necessary to oppose the thumb, and still allows some control of the orientation of the long axis of the grasped object (by revolving around the axis created by the thumb and fingers) and thus is useful for precise manipulation of long objects.

A Precision Grasp: has the characteristics that the fingertips are positioned on the object in such a way that the object is restrained, yet arbitrary fine motions can be imparted to it by the hand configuration. Manipulation is primarily a finger related function, the wrist being used only for gross motions.

3.4 Filling in the Framework

Now that the framework has been constructed, we shall consider how well it fulfills its objective of allowing us to classify existing work in this field, and determine where work need to be done. It is immediately noticable that the majority of work in the grasping field centers (understandably) around the precision grasp. Salisbury's manipulation work is essentially the manipulation component of the precision grasp. Baker et al.'s work on

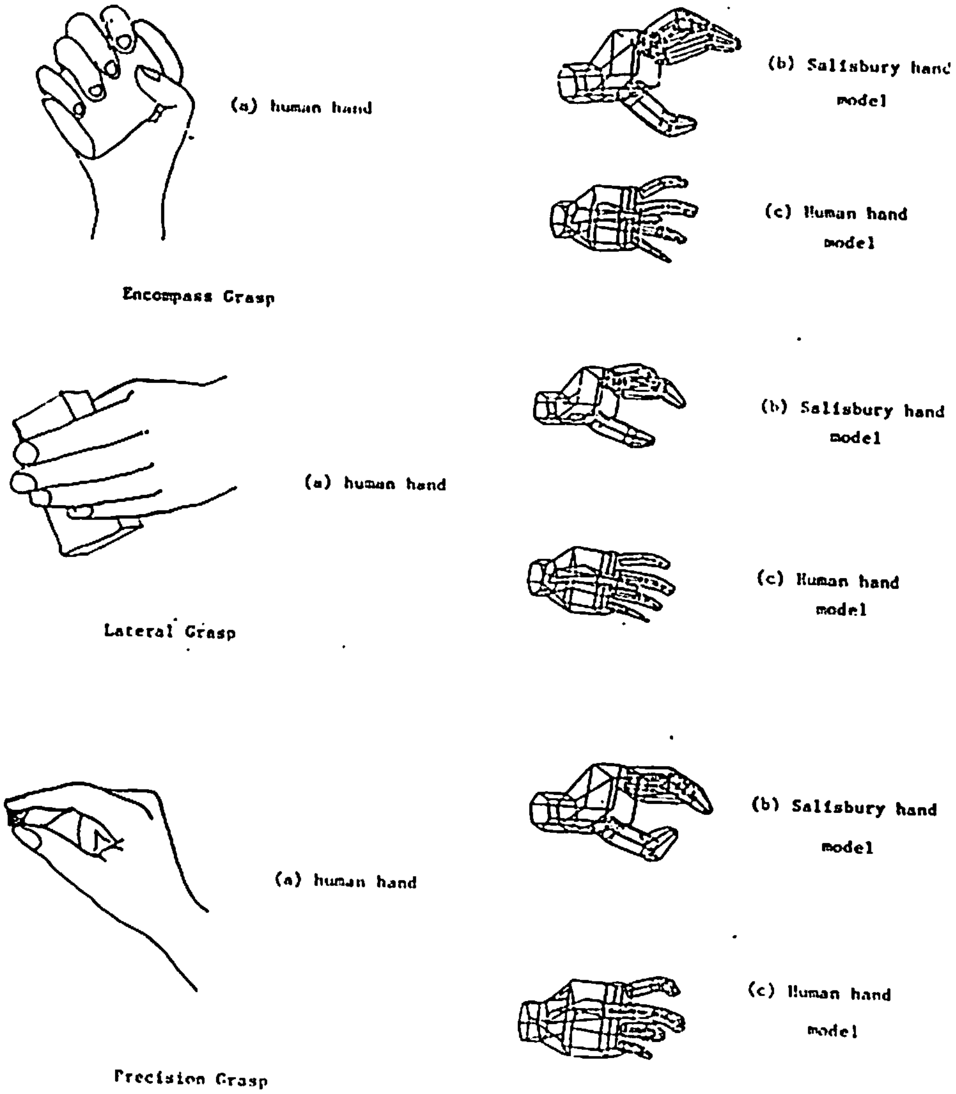

Figure 4: Stereotyped grasp configurations

stability of grasp is part of the acquisition and preshape components of this grasp.

We can observe however that little work is present on adapting the preshape to particular object sizes and shapes, and on ensuring the grasp stability while the fingers close. Additionally it can be observed that there is currently little linkage between the parameters necessary for manipulating an object, and the task-level specification for the manipulation.

Fearing's[5] work on manipulation can be seen to relate to the acquisition components of all grasps, as well as to the more limited forms of manipulation available for example in the lateral grasp. Next to the precision grasp, the encompass grasp seems the next most generally useful grasp. However there is little if any work in the literature on *any* component of this grasp. An analytical approach to grasp stability may be computationally intensive, and also requires knowledge of the object shape and size. The encompass approach is more brute-force, but does not have the guarantees of the analytical approach. A mixture of these approaches could be profitable.

3.5 Task Requirements

In this section a method for selecting grasps from task context is presented. We discuss a number of operation subsequences from a simple assembly task (figure 5.1) and consider what grasp or grasp sequence would be appropriate. These sequences are then generalized to produce an *index table* linking object and task context to appropriate grasps. Finally a more flexible method of grasp selection is considered including a possible role for learning and adaptation in grasp selection. Consider the assembly shown in figure 5.1. The following are the necessary construction steps:

Lift the base-plate assembly from the pallet and place it (with lax positional tolerance) in the workspace. Lift a motor from the motor pallet, and place it into the hole in the base-plate. Take the top-plate from its pallet and fit it on top of the motor. Take four spacers, one at a time, from the spacer feeder and place them into the holes in the corner of the top-plate through to the base-plate. Finally take four bolts, one at a time, from the bolt feeder and insert them through the holes in the top-plate, through the spacers and into the base-plate. Hand-tighten the bolts. Remove the assembly and place it on a pallet.

First it is necessary to acquire and place the base-plate. From figure 5.1 this is a *long, flat* object. The assembly instructions indicate no precision is necessary in placement – this means an Encompass (ENC) or Lateral (LAT) grasp can be chosen, avoiding the unnecessary manipulation overhead of the Precision (PRE) grasp. Of the two grasps, the

Step#	Grasp	To-Object	To-Do
1.	LAT	Base-Plate	Coarse Position in Workspace
2.	ENC	Motor	Coarse Position on Base-Plate
	PRE	Side-of-Motor	Fine Alignment
3.	LAT	Top-Plate	Coarse Positioning
	PRE	Edge-of-Top	Fine Alignment
4-7.	PRE	Side-of-Spacer	Insertion
8-11	PRE	Side-of-Bolt	Insertion
	PRE	Head-of-Bolt	Screw In

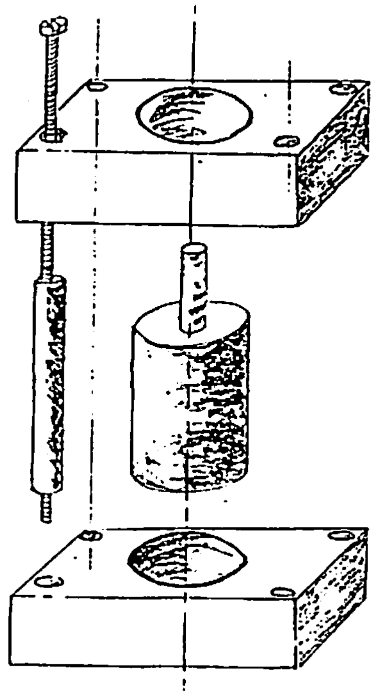

Figure 5: Example assembly task, and grasp sequence

LAT is suited to the acquisition of fat objects, and is chosen.

Next the motor needs to be acquired and placed in the base-plate. This operation will require precision manipulation to place and insert the motor. If the motor was *small*, a PRE could be applied. However if the motor is too large (or too heavy) for fingertip manipulation, a LAT or ENC will need to be applied; however *neither* of these provides the necessary precision degrees of freedom to place and insert the motor. This is solved by using one of ENC or LAT, whichever is more appropriate to the object shape, to coarsely place the motor, and follow this by the application of a PRE to a sub-site on the object to precision maneuver it into position; thus dispensing with the need for the grasp to counter gravity while manipulating the object. Since the object is *round*, the ENC is chosen to coarsely place the motor, and followed by a PRE applied to the side of the motor.

The top-plate is almost identical to the base-plate, except that it must be precision placed. Since it is a *flat* object, the LAT is chosen to coarsely place (the single precision DOF of the LAT being an added bonus) it, and followed by the application of a PRE to the side of the plate to maneuver it precisely.

The spacers are large round rods which need need to be inserted through the top-plate into the base-pate. The insertion task in this case needs precision in five axis; x, y and z to place the tip, and ϕ, ψ around z and y to insert the spacer. θ around x is not necessary since the spacer is symmetrical around x (its long-axis). A PRE is chosen; since the object is *long* all fingers may well be called into play and a *parallel* grip configuration implemented. It is much easier to hold and insert the rod this way then by grasping it the plane perpendicular to its long axis. Note however that the fingers must be removed one-by-one as the rod is inserted. Inserting the bolts is similar to inserting the spacers, however once the insertion operation has been completed it is necessary to re-grasp the head of the bolt and screw it in – a two degree of freedom task; z, and ϕ around z. A PRE is appropriate for this task.

An index table structure can be built to generalize this process. When precision is required in dealing with an object, PRE can be chosen immediately if the object is small enough. Otherwise ENC or LAT have to be chosen to coarsely place the object, and followed by a PRE to some appropriate sub-site on the object, to complete the precision maneuvering. The choices of LAT or ENC would depend on the object shape; ENC dealing with round objects and LAT with flat. If firm affixment is required then PRE can be ruled out, and again LAT or ENC used depending on object shape; with ENC being favored over LAT, since it generates better grasp security.

If both precision and firm affixment are necessary then its necessary to balance the grasps off against each other. For small objects the PRE is appropriate. For larger objects LAT and ENC are chosen on the basis of shape. It may be necessary to follow these up by application of PRE to an object sub-site, for the final maneuver. If neither precision nor affixment are required, the simplest grasp which suits the objects shape is chosen. For the majority of cases this is the ENC grasp. Table 5.2 illustrates these selection rules.

We can formulate the logic of grasp selection from task requirements as a set of logic equations. In following equations it is assumed that $small = \overline{large}$, $short = \overline{long}$, $flat = \overline{round}$:

$$\begin{aligned}
PRE &= (small\&short\&P\&F) \mid (small\&P\&\bar{F}) \\
LAT &= (long\&flat\&\bar{P}\&\bar{F}) \mid (large\&flat\&P\&\bar{F}) \mid \\
&\quad (large\&flat\&P\&F) \\
ENC &= ((long\bar{\&}flat)\&\bar{P}\&\bar{F}) \mid \\
&\quad (large\&round\&P\&\bar{F}) \mid (F\&\bar{P}) \mid (large\&round\&P\&F)
\end{aligned}$$

If three *grasp-experts* are constructed, each embodying one of these equations, some inherent grasp-selection parallelism can be exposed. Using these equations the grasp-experts would be mutually exclusive, and never need to 'confer' to reach a decision. If continuous quantities are substituted for the discrete functional, shape and size characteristics; for example each defined as a suitability rating between 0 and 1; then the logic equations can be replaced by a set of probability equations to yield the appropriateness of each grasp to the operation. In this case the grasp-experts would need to confer to resolve their decisions.

In an attempt to simplify hand-object interaction we have selected three domains of interaction and characterized them using the grasp construct. Howeverthe result of analysing task-context has generated not an object and task-context to grasp mapping function, but instead an object and task-context to grasp *sequence* mapping function. The proper output of the three grasp-experts should be a *sequence* of grasps to complete the operation, not a single grasp. Through feedback of some evaluation functions it might be possible to construct and adapt such sequences to optimize task performance. A possible rule for altering grasp sequences could be based on combinations of precision and non-precision moves, such as was necessary to precisely position large objects.

4 Implementation

A Distributed Robot Programming Environment has been constructed which accepts programs written in the terminology of the distributed computing model and simulates the execution of a true distributed system. Special primitives have been constructed in the interpreter for motor output and for dynamic sensory input [8]. At the moment the output from the interpreter is used to drive a kinematic hand model (the *hand simulation*) which in turn feeds back position, contact and pseudo-visual information.

This hand simulation [1] was designed around a general hand data structure, which could be parameterized to the details of a particular hand model. The research in this paper was done using two hand models; one of a generic human hand, one of a robot hand similar to the Salisbury hand [18], and a number of object models. Two different models were used to emphasise the fact that the results are *independent* of any particular hand model. At the moment the simulation considers only rudimentary dynamics; this must be expanded before the results here can be applied to a real robot hand.

4.1 Outline of the distributed Programming Model

Each joint in the hand is represented by a computing agent. This `limb` schema accepts input for each of the degrees of freedom at that joint through its input *ports*. An object or obstacle is also represented as a computing agent in the model. An object is represented as a potential sink and an obstacle as a potential source; that is, an object will attract limbs and an obstacle will repel limbs.

The `object` or `obstacle` schema offers advice to the `limb` schema through connections to its input ports. In this way a dynamic network of computing agents is set up around the `limb` schema. All activate computing agents (called *instantiations* of a schema, SI) execute concurrently, and communicate only through their port connections. Computing agents can create other computing agents (can *instantiate* other schema) and specify how they are connected into the existing network.

4.2 Grasp Implementation

The initial input to the grasping system is defined to be: an order to grasp, an indication of the target object, and a specification of the requested functionality. Given this, the grasp selection equations can be used to choose a suitable stereotyped hand shape. The first phase of the grasp, the *preshape* can now begin.

The stereotyped grasp is parameterized to suit the *particular* target object. For this the *Virtual Finger* concept is useful. The hand shape is described in terms of the positions of one to three virtual fingers; and then the mapping from virtual fingers to real fingers is done considering object size. Thus one virtual finger may map to two or three real fingers for a large object. In addition, of course, the joint angles of the virtual fingers are also parameterized by the object size (see the implementation section).

In order to consider the termination of the preshape phase and the next phase, the *gripping* phase, we need to introduce some terminology. First of all we define approach and orientation vectors for the open hand as seen in Figure 6(top). Next we consider how this changes when a grasp is formed.

For each grasp type the preshape defines a volume between the fingers called the *approach volume*. The entrance to this volume is via a surface formed by joining the fingertips of the preshaped hand by straight lines; this area is called the *approach area*. We define the approach axis for the grasp to be along the normal to this area through its centroid (Figure 6(bottom)). This compares well with the standard definition of the approach vector for a two-fingered gripper. The hand approaches the object along its approach axis, the aim being to capture (some of) the object in the approach volume. When the object reaches a point in the approach volume called the *focus* (this position is a function of the particular grasp), the preshape phase terminates, the reach movement stops and the grip phase begins.

The final phase of the grasp is the *manipulation phase*. The grasp was chosen on the basis of the mobility it could provide for the grasped object. We take a table-driven approach to this final phase of the movement. For each grasp there are specified degrees of freedom (see Figure 7). When a manipulation command is given, the table is accessed to determine how to transfer this grasp degree of freedom to the components of the hand.

The Preshape Phase: There are two parts to the way the hand preshapes for any grasp; the fingers and thumb separate out to accommodate the size of the object, and they curve inwards to form the approach volume. Each grasp preshape is defined by the actions of a number of virtual fingers. The encompass grasp is a *one* virtual finger task, the lateral grasp a *two* virtual finger task and the precision grasp a *three* virtual finger task.

The encompass grasp can be described in terms of the actions performed by one virtual

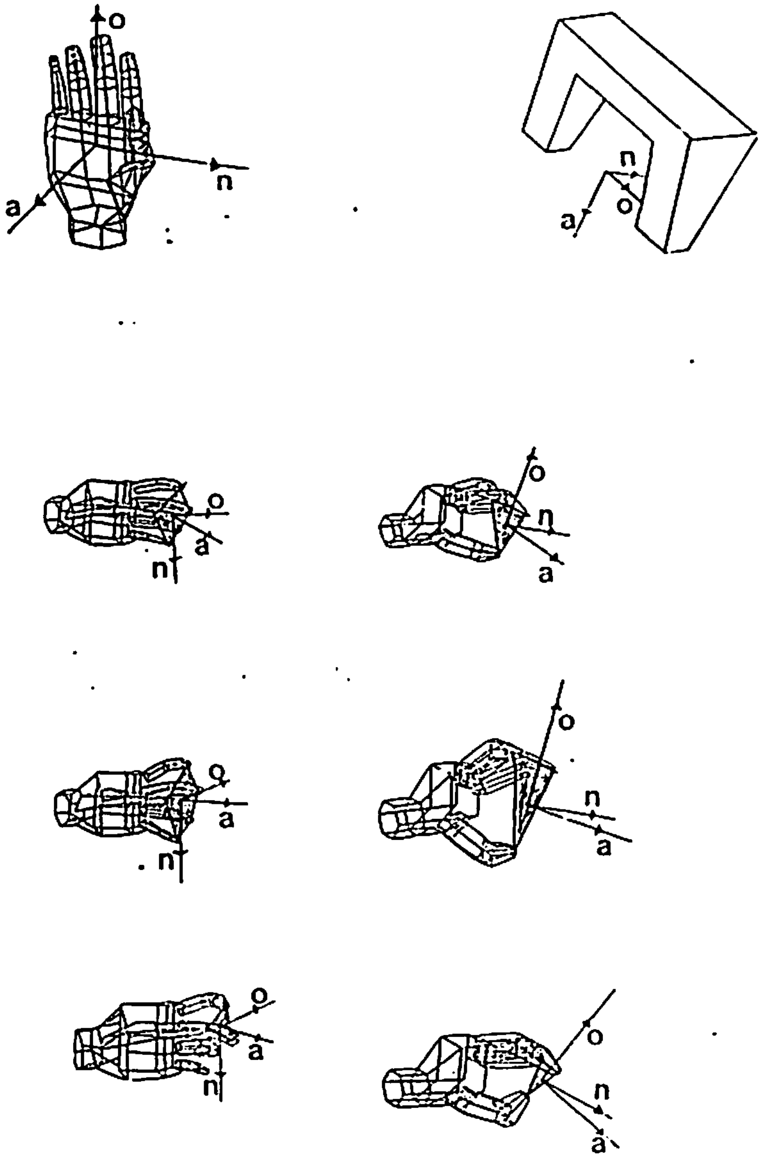

Figure 6: Preshape Orientation Vectors

Grasp	Manipulation Command	Action
Encompass	Rotate(Axis,Amount)	Rotate wrist
	Translate(Axis,Amount)	Translate wrist
Lateral	Rotate(Axis,Amount)	If axis=Z then rotate object between fingers and thumb, else rotate wrist
	Translate(Axis,Amount)	Translate wrist
Precision	Rotate(Axis,Amount)	If rotations within range rotate object using fingertips else rotate wrist
	Translate(Axis,Amount)	If translations within range translate object using fingertips else translate wrist

Figure 7: Manipulation Table

finger. First an obstacle schema is instantiated at the location of the base of the virtual finger and secondly a object schema is instantiated at some point relative to the finger. This virtual finger can map to any amount of real fingers (depending on the object size); for each real finger it maps to, the same obstacle and object schema are instantiated. The obstacle SI force the fingers to repel each other and separate, the object SI force the fingers to curve inwards towards the object position. The preshape can be parameterized to a particular real-world object by correct choice of the virtual finger to real finger mapping, and the fields of the object and obstacle SI.

The lateral grasp preshape can be described in terms of two virtual fingers. The obstacle SI chosen for the first virtual finger defines the separation of the fingers which form one 'jaw' of the grasp. The obstacle SI chosen for the other virtual finger defines the size of the opening between the 'jaws' of the grasp.

The precision grasp can be described in terms of three virtual fingers. The obstacle SI for each finger define the shape of the approach area and are related to expected object size and shape. Parameterizing each virtual finger separately makes this grasp more versatile, than say, the encompass grasp. A single object SI is defined for all three virtual fingers to provide the finger curvature.

The Grip Phase: The only case implemented for the grip phase is that described previously as the simple grip. For this case the target object is within the approach volume. To complete the grip for any of the three grasps all that is necessary is that the virtual fingers close in to contact the object. For the encompass grasp the object is pressed against the palm. For the lateral grasp the inner surface of the thumb and fingers squeeze the object. For the precision grasp only the fingertips contact the object.

This action is modelled by setting an object SI at the position of the real-world object. In the encompass grasp all limb SI are drawn by it, for the lateral grasp only those limbs forming the inner surface of the thumb and fingers, and for the precision grasp only the fingertips (and 'thumbtip').

5 Discussion

In this paper we have describe a framework which allows us to view grasping and manipulation as a single structure. The framework is based on constructing a number of domains of interaction for a dextrous hand, which we call Grasps, and then using task requirements to select an appropriate domain of interaction for the hand to work in. We have constructed a set of three grasps, which we call the simple set of grasps. This is an acceptable set of grasps, in that it provides a range of manipulation and affixment abilities. Given this framework we can now catagorize other work in the grasping and manipulation literature and can also detect the absence of work in some areas. Additionally we can now formalize the effect of task requirements on hand usage.

The ulterior motive for the development of this framework for grasping and manipulation was the need to determine characteristics of the complex robot domain for use in formulating our distributed model, as well as an example domain for programs in the model. The framework does not represent the total manipulation abilities of any dextrous hand, and does not represent completely the manipulative power of the human hand. A deeper look at human versatility is needed to inspire extensions here[9].

Additionally the framework does not offer extra detail at the mechanism level for grasping and manipulation; the dynamics issues in grasping have not been considered at all. However we are integrating our approach with such issues [20]. Again, the main objective of our framework is to provide a global, overall structure in which to place more detailled work, and as a task-level interface for grasping and manipulation.

This paper does not deal with some of the most interesting cases of the grip phase. These cases are related to constraining obstacles around the target object, and argue for consideration of the *surrounding environment* in the choice of grasp. However there may also be reasons for considering these complex cases as indicating a possible strength of this whole approach.

A dextrous hand can, to some extent, exhibit obstacle avoidance behavior. This is evident in human reaching where the preshape may deform to miss an obstacle, or to fit into a confined space. The effort in this may only be worthwhile if the arm trajectory is notably simplified. However, since the question of obstacle avoidance must be addressed in order to expand the grasping power of the hand anyway, the provision of local obstacle avoidance capability in the hand can be justified. This area is currently being researched.

6 Acknowledgements

I would like to thank M.A. Arbib and T. Iberall for comments on an earlier draft of this paper, all the members of the LPR at the University of Massachusetts for their assistance and comments on this research.

REFERENCES

[1] Arbib, M.A., Iberall, T., and Lyons, D. "Coordinated Control Programs for Movements of the Hand" Technical Report #83-25, Department of Computer and Information Sciences, University of Massachusetts at Amherst 1983.

[2] Baker, B.S., Fortune, S.J., and Grosse, E.H. "Stable Prehension with a Multi-Fingered Hand: Extended Abstract," *Proceedings 1985 International Conference on Robotics and Automation*, March 25-28, 1985 St.Louis.

[3] Cutkosky, M., "Mechanical Properties for the Grasp of a Robot Hand". *CMU-RI-TR-84-24*, Robotics Institute, Carnegie-Mellon University, Penn. Sept. 1984.

[4] Crossley, F. and Umholtz, F. "Design for a Three Fingered Hand" *Mechanism and Machine Theory* Vol.12 1977.

[5] Fearing, R.S., "Simplified Grasping and Manipulation with Dextrous Robot Hands" *MIT AI Memo: 809* Nov., 1984.

[6] Geschke, C., "A System for Programming and Controlling Sensor-Based Robot Manipulators," *IEEE Trans. on PAMI*, Vol. PAMI-5, No. 1, Jan. 1983, pp. 1–7.

[7] Hanafusa, H., and Asada, H., "Stable Prehension by a robot hand with Elastic Fingers" *Proceedings of the 7th ISIR* Tokyo, Japan, October 1977, pp 361–368.

[8] Iberall, T. and Lyons, D. "Perceptual Robotics: Towards a Language for the Integration of Sensation and Perception in a Dextrous Robot Hand" *IEEE Workshop on Languages for Automation*, November 1-3 New Orleans, Louisiana 1984.

[9] Iberall, T., Bingham, G. and Arbib, M.A. "Opposition Space as a Structuring Concept for the Analysis of Skilled Hand Movements" *Generation and Modulation of Action Patterns Symposium* October 22-26th Bielefeld, W.Germany 1984.

[10] Iberall, T. and Lyons, D. "Towards Perceptual Robotics" *IEEE Conference on Systems, Man and Cybernetics* October 9-12 Halifax, Nova Scotia 1984.

[11] Jacobson, S.C., Wood, J.E., Knutti, D.F., and Biggers, K.B., "The Utah/MIT Dextrous Hand: Work in Progress" *SDF/MIT 1st International Symposium of Robotics Research*, Bretton Woods, NH 1983.

[12] Laugier, C., and Pertin, J., "Automatic Grasping: A case Study in Accessibility Analysis" in: Advanced Software in Robotics, (A. Denthine and M.Geradin, Eds.) Elsevier Science Publishers, B.V. (North-Holland) 1984, pp 201–214

[13] Lian, D., Peterson, S. and Donath, M. "A Three-Fingered, Articulated Robot Hand" *Proceedings 13th ISIR* April 17-21 Chicago Illinois 1983.

[14] Lyons, D. and Arbib, M.A., "A Task-Level Model of Distributed Computation for the Sensory-Based Control of Complex Manipulators" *IFAC Symposium on Robot Control*, Barcelona, Spain, Nov. 6-8th, 1985.

[15] Napier, J. "The Prehensile Movements of the Human Hand" *Journal of Bone and Joint Surgery* Vol.38b No.4 pp902-913, 1956.

[16] Okada, T., "Computer Control of Multijointed Finger System for Precise Object-Handling" *IEEE Trans.SMC* Vol.SMC-12 No.3 May/June 1982.

[17] Ramamratham, K., Pocock, G., Lyons, D., and Arbib, M.A., "Towards Distributed Robot Control Systems" *IFAC Symposium on Robot Control*, Barcelona, Spain, Nov. 6-8th, 1985.

[18] Salisbury, J.K. "Kinematic and Force Analysis of Articulated Hands" Ph.D. Dissertation, Stanford University and Report STAN-CS-82-921, Department of Computer Science, Stanford University 1982.

[19] Tubiana, R. (Ed) *The Hand* Phila:W.B.Saunders and Co, 1981.

[20] Venkataraman, S., and Arbib, M.A., "A Distributed Control Model for Finger Control on the Salisbury Hand". *IFAC Symposium on Robot Control*, Barcelona, Spain, Nov. 6-8th, 1985.

[21] Vijaykumar, R. and Arbib, M.A., "Dynamic Plannig for Sensor-Based Robots" *IFAC Symposium on Robot Control*, Barcelona, Spain, Nov. 6-8th, 1985.

[22] Wolter,J, Volz, R., and Woo, A.C., "Automatic Generation of Gripping Positions" *IEEE Trans. SMC* Vol. SMC-15, No.2, March/April 1985, pp 204–213.