DEXTROUS TELEMANIPULATION WITH FORCE FEEDBACK IN VIRTUAL REALITY

Latifa TURKI
Philippe COIFFET

Laboratoire de Robotique de Paris (LRP)
CNRS-UVSQ-UPMC
10/12 Avenue de l'Europe- 78140- Vélizy- France
e-mail : turki@robot.uvsq.fr

ABSTRACT

The traditional approach to control a remote operated mechanism consists of providing the operator of joysticks to position the slave mechanism and TV displays to visualize the remote worksite. This approach does not necessary lead to "good" teleoperation performance. An alternative approach is to use not only the sensors mounted on the remote mechanism but also try to make the most of human natural skills. In the context of dextrous telemanipulation, we have to control a robotic hand by an exosqueleton (or DataGlove) worn by the operator. To achieve such an operation, we are investigating in the LRP the problems associated with the transformation of the human hand motions to teleoperate a robotic slave hand as well as the replication on the master operator hand of the contact forces generated during manipulative procedures performed by the robotic hand. The telemanipulation system consists of the "Master", a dextrous hand with force feedback which measures 14 finger joints developed at the LRP [1], the "Slave", an articulated hand with different structures and a graphical interface developed to validate our control procedures.

1 Introduction

The control of multifingered robot hands is playing increasingly an important role in assembly operations and other precision manipulation tasks that require a high level of dexterity. In most instances of teleoperation reported to date, the slave will be a serial link manipulator equipped with a parallel jaw-gripper as its end-effector. The human operator will be provided with a switch for opening and closing the gripper in order to grasp and ungrasp the desired object during task execution. Such, grippers however, have some limitations in view of advanced telemanipulation tasks. These limitations can be overcome by equipping the slave arm with an end-effector capable of emulating human hand like motions and sensing abilities [2]. Such capabilities exist in multifingered robotic hands. For literature on dextrous manipulation using robotic hands, the majority has dealt with kinematic design of such hands, automatic generation of stable grasping configurations [13] as well as the control problem of the coordination of movements [8]. All this literature shows us that autonomous control of multifingered hands is not an easy task.

By comparison to automated control, telemanipulation is an inexpensive method in terms of computation expense [10]. By use of Virtual Reality tools, error recovery and task planning functions are assumed by a human hand master, significantly reducing the algorithmic complexity. The operator using the master device can act in a virtual world containing the slave mechanism work cell and is no longer a passive observer. This includes a feature where the operator can control the simulated (virtual) slave hand via the master hand. In Virtual Reality applications, hand masters are used to control graphics displays and as interfaces for computer controllers [7]. Commercial hand masters include the VPL DataGlove (VPL Research Inc., Redwood City, CA), the Exos Dextrous Hand Master (Exos Inc., Lexington, MA) and the Cyber Glove (Virtual Technologies, Stanford, CA) [4]. For more advanced applications, force reflecting master gloves have been developed, including a pneumatic-powered system [3].

Make a robotic hand execute the same task than the master hand is not an easy task. Human fingers motion is restricted to that allowed by the master's kinematics. Also, for some systems there may be significant geometrical differences between robot and master hands and so, kinematic mapping is not direct. Telemanipulate a dextrous robot hand then requires an algorithm that performs transformation of human hand poses to the robot hand. This transformation however, presents both conceptual and analytical problems.

In this paper we propose an algorithmic method which consists on duplicating master poses into the slave side in target approach phase, and ensuring static stability once the object is grasped. As the force feedback is very important for the efficiency of telemanipulation, we'll talk about the transformation of forces between the two hands and propose force control schemes for 2 kinds of grasp.

2 The Telemanipulation System

The telemanipulation system [12] consists of the master, a dextrous hand with force feedback developed at the LRP [5], the slave, an articulated hand and the control procedures developed on an HP graphic workstation (Figure 1).

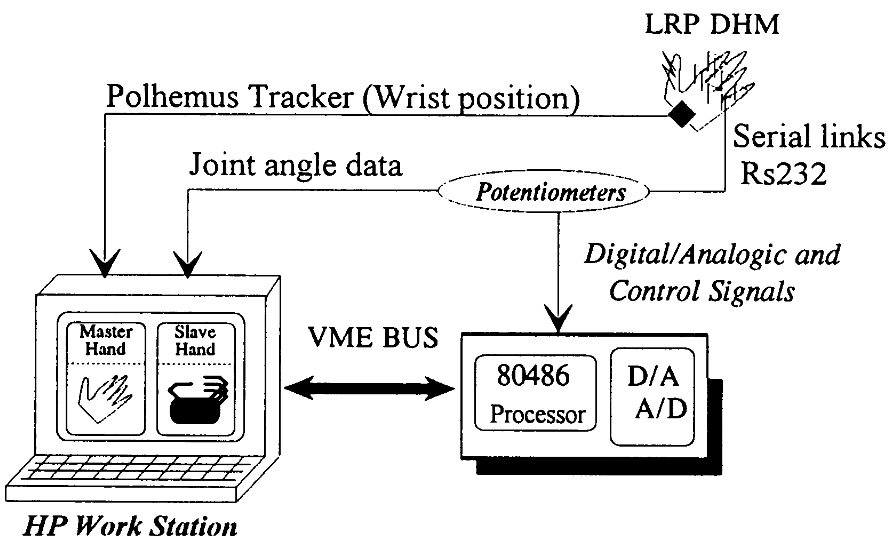

Figure 1 : The telemanipulation system

2.1 LRP Dextrous Hand Master

The LRP DHM mechanism (Figure 2) consists of several tendons connected to the top of each finger segment. It measures 14 finger joints (two for the thumb and three for each other finger) and applies normal and variable forces on each phalanx [1]. The exosqueleton is comprised of 3 pairs of parallel link mechanisms spanning the length of each finger and attached to an immobile base on the back of the hand where the Polhemus Tracker lies (sensor which gives 3D position in space). Each joint is actuated through a tendon-sheath transmission by a DC torque motor placed remotely from the hand. The angles of the finger joints are geometrically deduced from the cable motion which is tracked by potentiometers mounted on the motor shafts. Miniature force sensors are placed on the back of the palm to monitor the cable strain allowing by that force control.

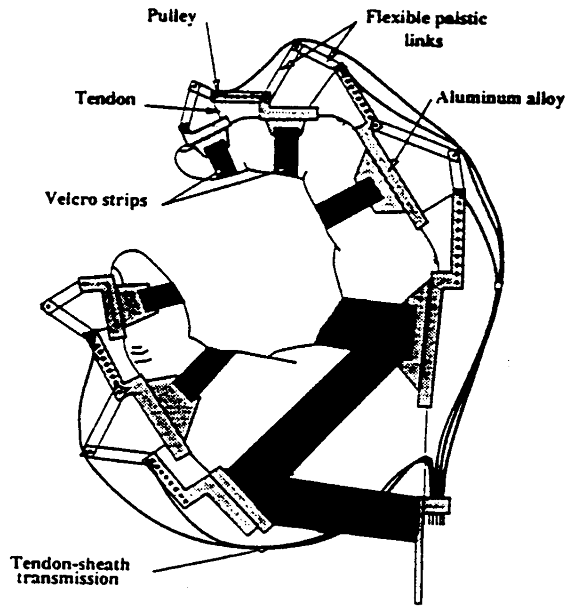

Figure 2 : The LRP Dextrous Hand Master

Human hand kinematics studies [11, 6] show at least 5 DOF's for each finger. A simplified model is dictated by the 14 DOFs of the LRPDHM. The simplified model first approximates each joint as a rotation about a fixed axis. For each finger i, we define a frame *Rfmi (xfmi, yfmi, zfmi)* attached to the fingertip. Frame *Rmi (xmi, ymi, zmi)* coincides with the first joint of the same finger. The origin of frame *Rmi* (base of each linkage) is displaced from the frame *Rm* (back of the hand which lies with the Polhemus Tracker) by a distance ${}^{m}\underline{d}_{mi}$ expressed in frame *Rmi* (Figure 3).

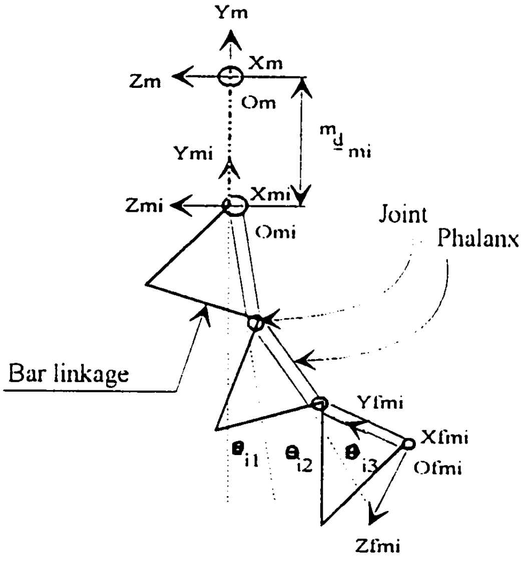

Figure 3: Side View of the human finger model
All joints are assumed to rotate about fixed axes

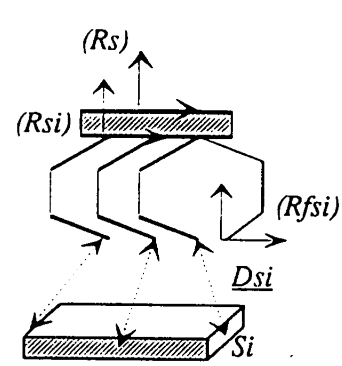

Figure 4: Slave mechanical hand

The forward kinematics of the master hand are given in section 3.2.2.

2.2 Robotic multifingered hand and common reference frame

For a telemanipulation task, the master/slave transformation procedure depends on the slave mechanism. We distinguish 3 cases:

- Both master and slave hands have exactly the same structure (same number of fingers and DOF per finger). In this case, the hand transformation procedure consists only on mapping joint-angle data from master to slave. A sensor calibration for both human and robot hands can improve the efficiency of the algorithm.
- Both master and slave hands have the same number of fingers but with a different number of DOF per finger. In this case a simple fingertip mapping is applied (see section 3.2.1).
- The slave hand is kinematically completely different from the master hand. To execute the same task in both sides, the low level controller presented in section 3.2.2 is applied.

In all cases, mapping fingertip positions require a common reference frame on the human and robot hands. Previous fingertip mapping research placed the common origins at the base of the thumbs [10,9]. We use common origins at frame *Rm* for the master hand and at the wrist for the slave hand (frame *Rs*). These locations were chosen because the slave wrist tracks the master "wrist" (location of the Polhemus Tracker). On Figure 4, *Si* correspond to fingertip/object points of contact. These points are calculated off-line so as to ensure the grasp equilibrium. *Dsi* represents the distance between each fingertip *i* and the correspondent point contact *Si* to reach.

Before going to the next section and introduce grasp stability, lets define the task to be executed by the slave hand. The task is subdivided into 3 phases :
1. Reach phase: From an initial state, the slave hand has to reach safely the object to be grasped. The reach phase corresponds then to target approach phase in which each finger *i* tries to reach point *Si* on the object. Reach phase is the first goal priority of this paper.
2. Grasp phase: After we have reached the desired points contact, we need at this stage to ensure static stability of the grasp operation.
3. Manipulation phase: It consists of manipulating the object until the desired task is executed (e.g. peg in a hole task). In this case, dynamic stability must be achieved.

3 Hand transformation

After an analysis of some specific requirements to achieve a grasp operation by a multifingered hand, we present an approach for synthesising the control required in the slave hand to achieve a telemanipulation task.

3.1 Synthesis of grasp operation

In a general case, we resume the grasp operation as follows (Figure 5):

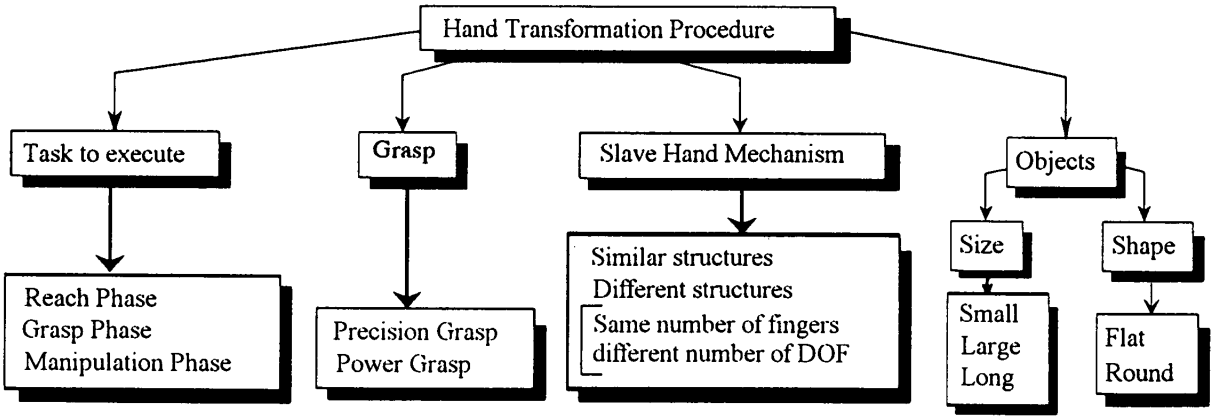

Figure 5 : Grasp planning operation

Perception is not considered in our paper.

Usually, when an object is fixed with respect to the robotic hand manipulating it, the grasp is presently defined as *stable*, otherwise the grasp is *unstable*. In general, grasping may be divided into two areas : static and dynamic. The static grasp is one in which the object does not move relative to the hand throughout the operation. On the other hand, if the object orientation or position relative to the hand is changed during the manipulation, the grasp is dynamic.

In any grasp operation by a multifingered hand, we have to define an approach to:

- determine all possible static grasping regions on the object to be grasped based on the geometry of the object and the assembly task to be performed,
- define a finite number of grasps on these regions specific to a given gripper,
- determine the optimal grasp from the remaining static grasping regions. We use standard forces that fingers apply to any object (no slippage).

A duplication of master movements in the slave hand alone is not sufficient to achieve the grasp operation. We have to be sure that the object, when it is grasped, must not slip from the slave hand. Reason why we need to calculate the point contacts Si that ensure static equilibrium. Concerning our system, all the objects represented on HP graphic work-station are already known before the task is started. Objects are characterised by their size: small-large-long and their shape: flat-round. Geometric description of the components of the object is given as a CAD/CAM database. Contacts between the object and the fingers are assumed to be point contacts without friction. For each object, we define n points of contact corresponding to n fingers of the slave hand.

Finally we consider 2 kinds of grasp :

- Precision grasp : for an n finger mechanism, we have n points of contact.
- Power grasp : The object is completely in the hand, we have more than n points of contact.

Only the first situation is considered in the section that follows.

3.2 Hand transformation procedures

Lets Imagine a robotic hand operating in an *hostile* environement. The operator sends motion to the robotic hand and reorient it until the object is reached. When the robotic hand grasps the object, as the slave environement is invisible, we need to approach a hand posture that permits a stable grasp. The kinematics of the master/slave mechanisms may be different but should conduct to accomplish the same task. As mentioned above, to transform hand movements between the virtual hand (image of the master hand on the graphics display) and a slave mechanical hand we need not only a duplication of the master poses into the slave side in target approach phase but the achievement of static grasp equilibrium before the object is manipulated. While manipulating the object, we need to ensure a dynamic stable grasp in the slave side. We consider 2 cases :

3.2.1 Simple Fingertip Mapping

This is the case in which both master and slave hands have the same number of fingers but with a different number of DOF per finger. The control scheme by fingertip mapping is represented on Figure 6.

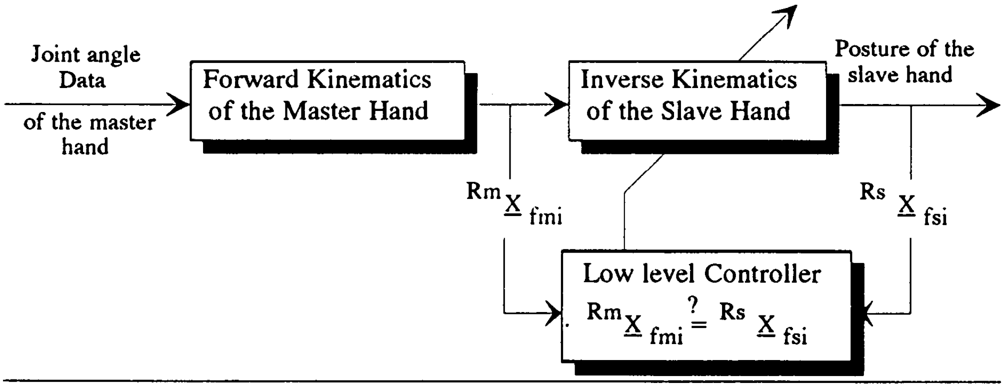

Figure 6 : Fingertip Mapping for 2 similar mechanisms

If we are in presence of 2 master/slave mechanisms kinematically different, the fingertip mapping is not direct. This case is treated in the next section.

3.2.2 Low level control

From the geometry of the object and the number of slave fingers, we use at the moment human skills to find the optimal points of contact noted *Si* that ensure static equilibrium. This is done off-line as an initial step. As the slave wrist ideally tracks the master wrist, an infrared (or ultrasonic) sensor mounted on the slave wrist gives us at each step the distance D_{cont} between the point encountered on the object by the radiation rays and the origin *Os* of frame *Rs* (Figure 7). To achieve movement transformation, we need to (1) calculate the forward kinematics of the master hand, (2) map the fingertip positions to the slave hand via a common reference frame, (3) generate slave hand inverse kinematic solutions, (4) cancel the distance between each fingertip and the corresponding point of contact on the object.

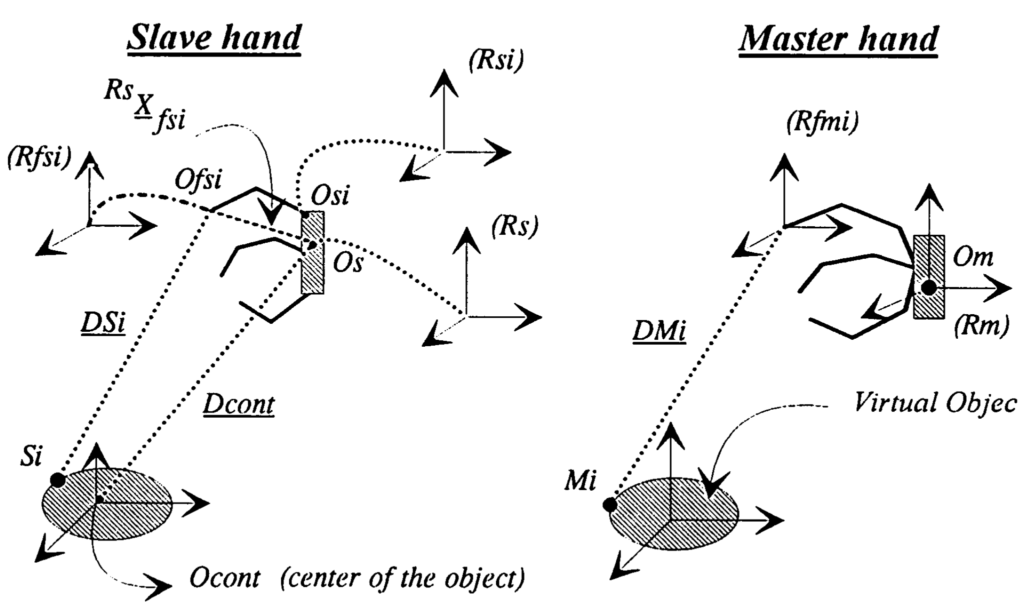

Figure 7 : Low level Controller

Forward and Inverse kinematics

The forward kinematics are calculated from the LRP DHM joint angle data using the simplified human hand model. First the forward kinematics of each finger are calculated with respect to frame *Rmi* (eq.1). The position of each human finger is transformed to frame *Rm* by use of equation 2 :

$$
{}^{Rmi}\underline{X}_{fmi} = \begin{pmatrix} 0 \\ -\sum_{j=1}^{j=m} lij.\cos(\sum_{k=1}^{k=j} \Theta ik) \\ -\sum_{j=1}^{j=m} lij.\sin(\sum_{k=1}^{k=j} \Theta ik) \end{pmatrix} \quad \text{(eq. 1)}
$$

$$
{}^{Rm}\underline{X}_{fmi} = A_{Rmi}^{Rm}.{}^{Rmi}\underline{X}_{fmi} \qquad \text{(eq. 2)}
$$

where for each finger i (i=0,1,2,3,4) and joint j, ${}^{Rmi}\underline{X}_{fmi}$ is the human fingertip position vector with respect to Rmi, A_{Rmi}^{Rm} is the 3-by-3 orientation matrix of frame Rm/Rmi, l_{ij} is the length of phalanx j and Θ_{ij} the joint angle.

The inverse kinematics of the slave hand are calculated with respect to coordinate frame *Rsi* by use of the inverse jacobian matrix J of each finger:

$$
\underline{\Delta\Theta i} = J^{-1}.\left[A_{Rfsi}^{Rfs}\right]^{-1}.{}^{Rs}\underline{\Delta X}_{fsi}
$$

where $\underline{\Delta\Theta i}$ is the $mx1$ joint angle vector of finger i and ${}^{Rsi}\underline{X}_{fsi}$ is the robot fingertip position vector with respect to frame *Rsi*.

Algorithm Structure

From Figure 1.6, we can write :

$$
\underline{D}_{si} = {}^{Rs}\underline{X}_{fsi} + \underline{D}_{cont} + \underline{O}_{cont}S_i \quad \text{(eq. 3)}
$$

$$
{}^{Rs}\underline{X}_{fsi} = A_{Rsi}^{Rs}.{}^{Rsi}\underline{X}_{fsi} \qquad \text{(eq. 4)}
$$

At each step, D_{cont} is given practically by the ultrasonic/infrared sensor mounted on the slave wrist. Lets recall that the distance $O_{cont}Si$ is known as points *Si* have been already determined off-line. ${}^{Rs}\underline{X}_{fsi}$ represents the desired trajectory of the slave fingertips given by the master hand during movement.

At each step, when the positional transformation is done, we have to check if we are still approaching the points of contact *Si*. This is done by adapting the slave fingertips according to the error done on distances <u>*Dsi*</u> and <u>*DMi*</u> (Figure 8). At each step, the computed distance <u>*Dsi*</u> is compared with the desired distance <u>*DMi*</u> (<u>*DMi*</u> is graphically, then numerically known). The errors between both of them, resulting from the difference kinematics between the two hands, is cancelled by a classical feedback controller. The flowchart of simulation is shown on Figure 8.

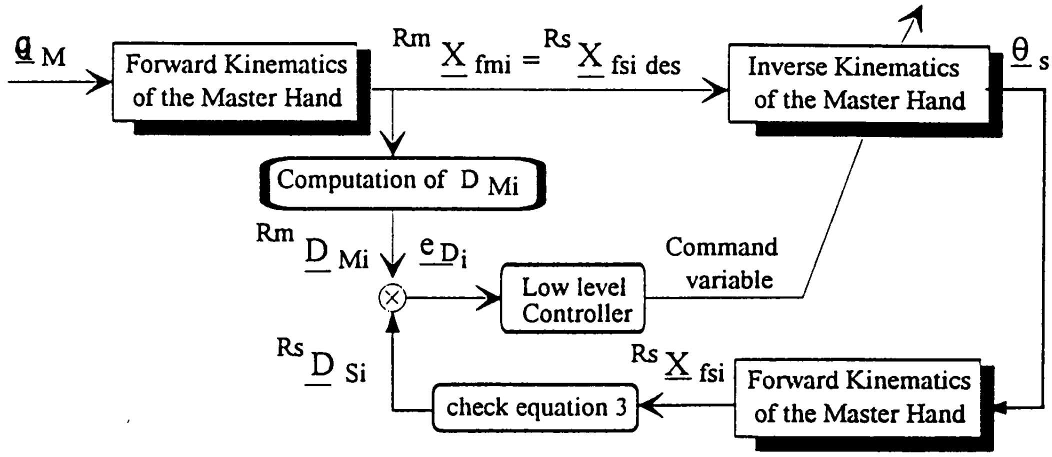

Figure 8 : Control Procedure

4 Forces transformation

The replication on the human hand of the forces acting on the robot hand during manipulation procedure can help to improve the effect of telepresence sensation. When we add the force feedback, the operator knows the effect of his actions (open and close fingers). Feeling the force on the hand, he works instinctively. The object for the operator is virtual, but the situation is real. The forces produced by the slave hand, whatever is the grasp, would ideally correspond to the forces that the master hand would exert during the direct interaction with an object.

The fact is that, when grasping actions or complex manipulation procedures are performed during the contact with the object, forces are generated on different sides of the hand fingers. As the LRP DHM measures forces on each phalanx, to simplify we consider mainly 2 kinds of grasp mentioned before :

- Precision Grasp : We consider that the interaction forces act on the fingertips (Fig 9.1)
- Power Grasp : Interaction forces act this time on each phalanx (Fig 9.2)

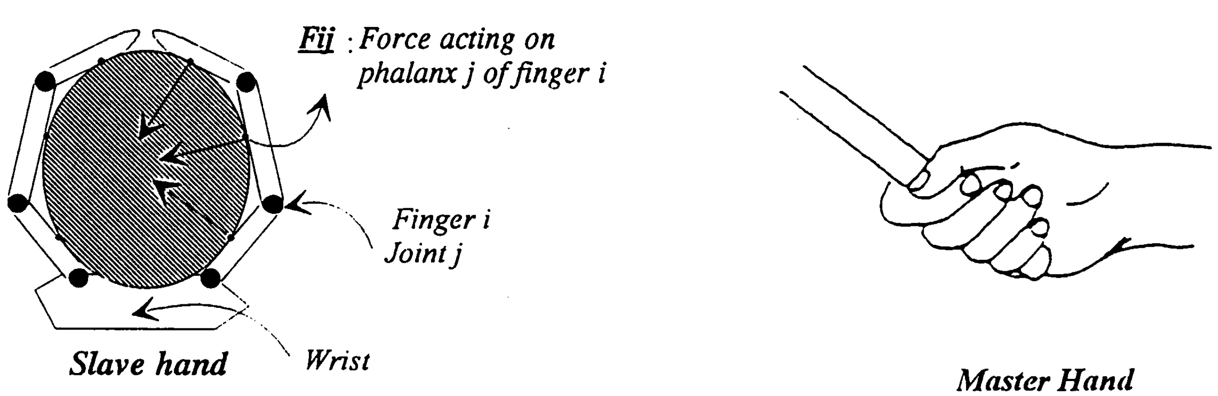

Figure 9.1: Illustration of a Power Grasp

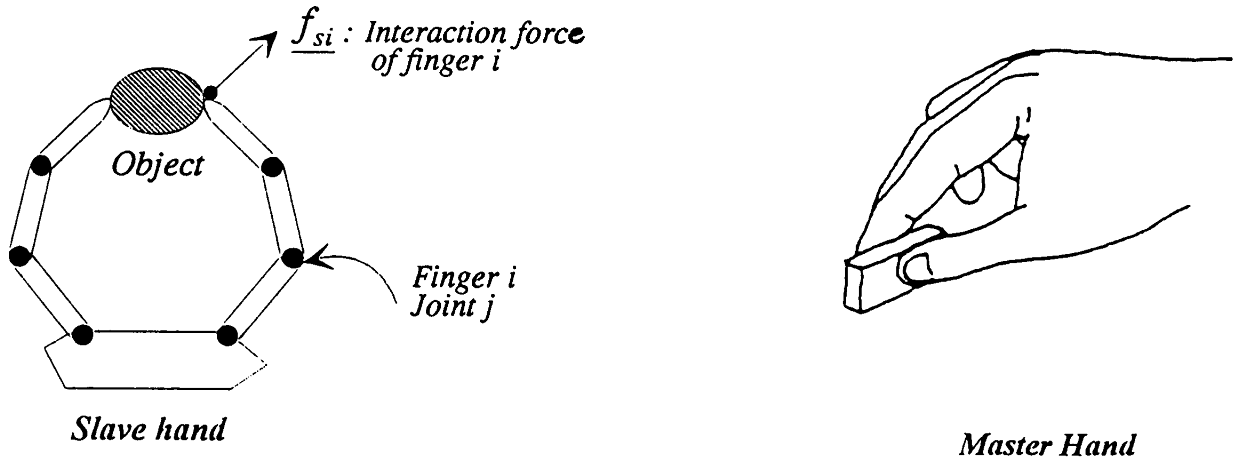

Figure 9.2: Illustration of a Precision grasp

In the sections that follow, we present the general problem of reproducing forces on the master hand for each kind of grasp by referring to the schemes of Figure 10.1 and 10.2. We suppose first that all joints of the slave hand are motorised and torque controlled (the joint torque is an image of the current in the motor). Concerning the master hand, the forces are measured on each phalanx as mentioned in section 2.2.The fingertip force vector is bi-dimensional (no moment transmitted at point contact).

4.1 Power Grasp

When contact occurs between the remote robotic hand and the external object the generated contact force vector on each finger called f_{S_i} is reflected in terms of joint torque vector called $\tau_{S_{ij}}$ (finger i, joint j) of the same finger. In general we assume that n contact points and force systems occur. The controller of the slave hand commands the actuation system by considering the error between the effective variable $\tau_{S_{ij}}$ and the desired variable $\tau_{M_{ij}}$ (joint torque of finger i and joint j) of the master hand. The latter is obtained by use of the equation $\tau_{M_{ij}} = F_{ij}.d_j$, where d_j is deduced from the cable motion of phalanx j (see section 2.2) and F_{ij} recorded at the interaction between the master hand and the virtual object (Figure 10.1). Notice that, on Figure 9.1, the force F_{ij} measured on each phalanx of the master hand is opposed to the one exerted of the slave hand.

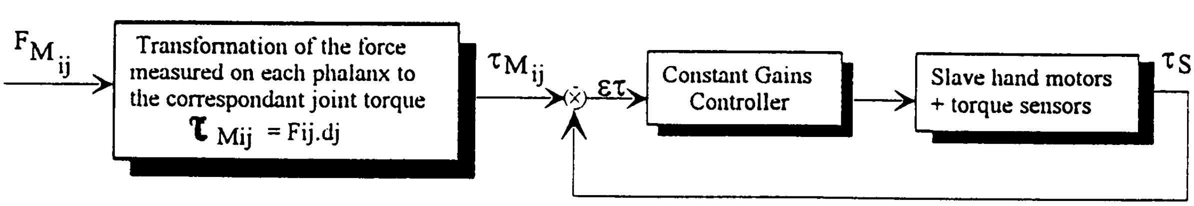

Figure 10.1

4.2 Precision Grasp

In this case we have to compute each fingertip force f_{M_i} resulting from the interaction virtual object/master hand. This one is obtained by use of the equation $f_{M_i} = J^{-T}.F_{ij}.d_j$, where J is the jacobian matrix of finger i. The slave force feedback controller will command this time the force feedback actuation system according to the error between the values f_{S_i} exerted by the slave hand in interaction with the real object and the forces f_{M_i} (Figure 10.2).

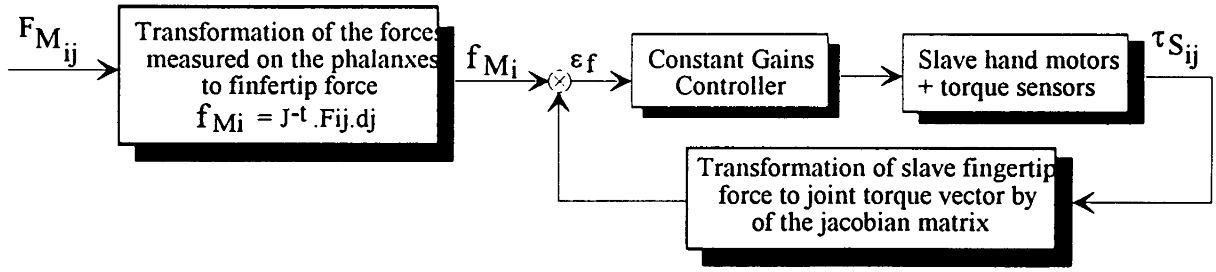

Figure 10.2

5. Conclusion and Future works

In this paper we proposed an algorithmic method for the telemanipulation of a multifingered robotic hand kinematically different from the master hand (LRPDHM). Even when fingertip position mapping is possible, errors from the human hand model, master glove data may reduce the mapping accuracy. In this paper we considered only the target approach and reach phase. The grasp and manipulation phase, for which we have to ensure static as well as dynamic stability will also be considered.

The fact that the LRP DHM also measures forces, we proposed in this paper force control schemes to control slave hand/object interaction forces from the master hand. These schemes will be implemented in the future to control Salisbury's robotic hand of the LIFIA Laboratory as a slave hand from the LRP DHM as a master hand.

REFERENCES

[1] Bouzit, M. & Coiffet, P. [1993] "Design of the LRP Dextrous Hand Master with force feedback". Virtual Reality Systems, FALL'93, Sept. 93, N.Y, U.S.A.

[2] Burdea, G. & Speeter, T. [1990] "Portable Dextrous Force Feedback Master for Robot Telemanipulation". Proceedings of NASA Conference on space telerobotics, Pasadena, CA, Jan/Feb, Vol. II, pp153-161.

[3] Burdea, G. et al. [1992]. "A portable Dextrous Master with Force Feedback. Presence: Teleoperators and Virtual Environments, Vol. I, pp. 18-28.

[4] Burdea, G. & Coiffet, P.[94]. "La réalité Virtuelle". Edition HERMES.

[5] Coiffet, P. et al. [1992]. "Réalité Virtuelle et Robotique". Revue d'automatique et de productique appliquées, 5(2) : 81-89.

[6] Conney III, W. P. et al. [1981]. "The kinesiology of the thumb trapeziometacarpal joint". J. Bone and Joint Surgery, Vol. 63-A, pp. 1371-1381, 1981.

[7] Foley, J.D.[1987]. "Interfaces for advanced computing".Scientific American, pp. 126-135.

[8] Hilhorst, R. A. & Tanie, K. [1994]. "Dextrous manipulation of objects with unknown parameters by robot hands". Proceedings of the 1994 IEEE International Conference on Robotics and Automation.

[9] Hogan, J & Tan, X. [1989]. "Calibrating a VPL DataGlove or teleoperating the UTAH/MIT hand". Proceedings of the IEEE International Conference on Robotics and Automation. Scottsdale, Arizona, pp. 1752-1757, May 14-19. 1989.

[10] Speeter, T. H. [1992]. "Transforming Human Hand motion for Telemanipulation". Presence Journal, Volume 1, Number 1, Winter 1992.

[11] Thompson, D. E. & Giurintano, D. J. [1989]. "A Kinematic Model of the Flexor Tendons of the Hand". J. Biomechanics, Vol. 22, pp. 327-334.

[12] Turki, L. & Coiffet, P. [1995]. "A survey of Dextrous Telemanipulation with force feedback using the LRP Virtual Reality System". Proceedings of the 1995 IMACS/IEEE Symposium on SAS, Berlin 1995.

[13] Zhang, X. & Nakamura, Y. [1994]. "Robustness of power grasp". Proceedings of the 1994 IEEE International Conference on Robotics and Automation.