Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

Coordinated Motion Generation and Real-time Grasping Force Control for Multifingered Manipulation

Z.X. Li*, Z. Qin, S. Jiang and L. Han†
Dept. of EEE, Hong Kong Univ. of Science & Tech. China
(e-mail: eezxli@ee.ust.hk, fax (852)2358-1485)

February 23, 1998

Abstract

In this paper, we propose a unified Control System Architecture for Multifingered Manipulation (CoSAM^2). CoSAM^2 achieves simultaneously three objectives of multifingered manipulation: (a) Motion trajectory (velocity/force) tracking of a grasped object; (b) Improving the grasp configuration in the course of object manipulation; and (c) Optimizing grasping forces to enforce contact constraint and compensate for external object wrenches. CoSAM^2 is organized in a modular and hierarchic structure so that each module implements a specified function using inputs from its predecessors and a minimum number of sensory data signals. CoSAM^2 is also flexible in accommodating addition of new modules. Here, we give the details for the Coordinated Motion Generation module and the Grasping Force Generation Module.

1 Introduction

Grasping and *fine manipulation of a grasped object* are two main operations performed by a multifingered robotic hand. The other main operation is *dextrous manipulation* in which a grasped object is manipulated from an initial grasp configuration to a more desirable grasp configuration without being dropped. To successfully execute a given task, a sequence or combination of these operations, which will be collectively referred to as *multifingered manipulation*, are to be performed using sensory data feedbacks. Consider, for example, the task of screwing a nut onto a bolt, a typical manufacturing assembly task. First, the parts are localized using say, vision sensors. Then, the nut is grasped and picked up based on accessible grasp points generated using the CAD model of the nut and accessibility constraints. If the grasp is not satisfactory for imparting fine motions on the nut then dextrous manipulation is invoked to manipulate the part to a more desirable grasp configuration. Finally, fine manipulation is performed to fasten the nut onto the bolt.

Over the years, significant strides have been made in realizing features of multifingered manipulation. Several articulated multifingered robotic hands have been developed as research tools to study multifingered manipulation ([21] and [8]); Tactile, force/torque and vision sensors have been developed or utilized to sense contact location and contact forces ([6] and [1]); Several useful contact models have been proposed and experimentally validated ([21] and [5]); Three important classes of kinematic relations underlying a multifingered manipulation system, among which (a) finger kinematics; (b) the grasp map, and (c) the kinematics of contact, have been identified and thoroughly analyzed ([21], [9], [18], and [14]); Dextrous manipulation with rolling contact constraints or finger gaiting has been investigated in ([10], [7], and [16]) along with several useful algorithms for finger motion planning; Coordinated control and compliance control algorithms for multifingered manipulation with either fixed point of contact or rolling contact have been extensively studied ([11], [19], [4], and [17]); Efficient algorithms for generation of optimal grasping forces for multifingered hands have been proposed in ([3] and [2]); Grasp planning and characterization of optimal grasps incorporating even task requirement have been extensively studied in ([13], [15], [12], and [20]).

Despite the enormous amount of research activities in multifingered robotic hands, we are still short from having robotic hands that could perform reliably the types of manipulation tasks we had envisaged them to perform. The dexterity and functionality of the human hand is still unmatched by any robotic systems. Factors that contribute to the inadequacy of present

*This research is supported in part by RGC Grant No. HKUST 685/95E, HKUST 555/94E and HKUST 193/93E.

†C.S. Dept, Teax A&M University, USA

robotic hand systems include: (a) Insufficient sensing abilities, especially the ability to report accurately and reliably contact position and force information, associated with most robotic hands; (b) Difficulties in generating real-time solutions for dextrous manipulation with rolling constraint and finger gaiting; (c) Failure to address the multi-objective nature of multifingered manipulation by existing control algorithms, e.g., a fine manipulation operation requires not only motion trajectory (velocity/force) tracking of a grasped object but also regulation of proper grasp configuration and maintenance of desirable grasping forces; (d) Lack of a unified framework for integrating relevant theory of multifingered manipulation with sensory data inputs to produce a hand control system which generates finger actuator commands based solely on task requirement and environment models; and (e) Decoupling of theoretical studies from experimental investigations. Most experimental works reported so far were mainly on hardware design and sensor development, very little on implementation of manipulation related problems.

In this paper, we propose a unified Control System Architecture for Multifingered Manipulation (CoSAM^2). By incorporating the various kinematic relations of a multifingered robotic hand system with proper sensory data inputs at different stages, CoSAM^2 achieves simultaneously the following objectives of multifingered manipulation:

(a) Motion trajectory (velocity/force) tracking of a grasped object;

(b) Improving the grasp configuration in the course of fine manipulation;

(c) Optimizing grasping forces to enforce contact constraint and compensate for external object wrenches.

CoSAM^2 is organized in a modular and hierarchic structure (see Figure 1) so that each module implements a specified function using inputs from its predecessors and a minimum number of sensory data signals. Furthermore, when changes are made on any of its modules no change or minimal changes will be needed for its neighboring modules. CoSAM^2 is also flexible to accommodate addition of new modules. For example, when efficient algorithms are available to solve the dextrous manipulation problem then a new module, say the *Dextrous Manipulation Planner*, can be simply added as shown by the dotted box in Figure 1 to enhance functionality of CoSAM^2. Several main modules of CoSAM^2 are briefly described as follows:

1. **Object Motion Generator** which computes a desired sequence of velocities and forces of a grasped object based on task specification and sensed object configuration and force information;

2. **Coordinated Motion Generator** which, when invoked for fine manipulation, takes as inputs the desired object velocity and tactile sensor information and generates as outputs desired fingertip velocity for each finger while simultaneously optimizing the grasp quality. When invoked for dextrous manipulation mode, it generates a sequence of rolling and gaiting motion for the fingers so as to manipulate the object to a desirable grasp configuration without dropping the object. In this case, input to the module is simply the desired grasp configuration and tactile sensor information.

3. **Grasping Force Generator** which takes the desired object force as input and generates optimal finger grasping forces for each finger. Tactile information is required to compute the grasp configuration information.

4. **Compliance Motion Controller** which combines desired fingertip velocity computed by Coordinated Motion Generator and desired fingertip force computed by Grasping Force Generator to generate the net incremental motions of the fingertips. Fingertip forces are fed back to the module, and the output from the module will be converted into incremental joint motions of the fingers by the Inverse Kinematics module. Finally, incremental joint motion is executed by the Joint Level Controller module.

Theoretical background of CoSAM^2 along with algorithms suitable for real-time implementation and implementation details of CoSAM^2 on the HKUST Three-fingered Robotic Hand are described in this paper.

2 Coordinated Motion Generation

Consider the k-fingered robotic hand manipulation system shown in Figure 2. We denote by P the palm frame, O the object frame, and F_i the fingertip frame of finger $i, i = 1, \cdots k$. At each point of contact, we let $\alpha_{f_i} \in \mathbb{R}^2$ and $\alpha_{o_i} \in \mathbb{R}^2$ be the coordinates of contact relative to the fingertip and the object, respectively, and L_{f_i} and L_{o_i} be the local frame of the fingertip and the object, respectively. The angle of contact between finger i and the object is denoted ψ_i. The five parameters $\eta_i := (\alpha_{f_i}, \alpha_{o_i}, \psi_i) \in \mathbb{R}^5$ that describe the

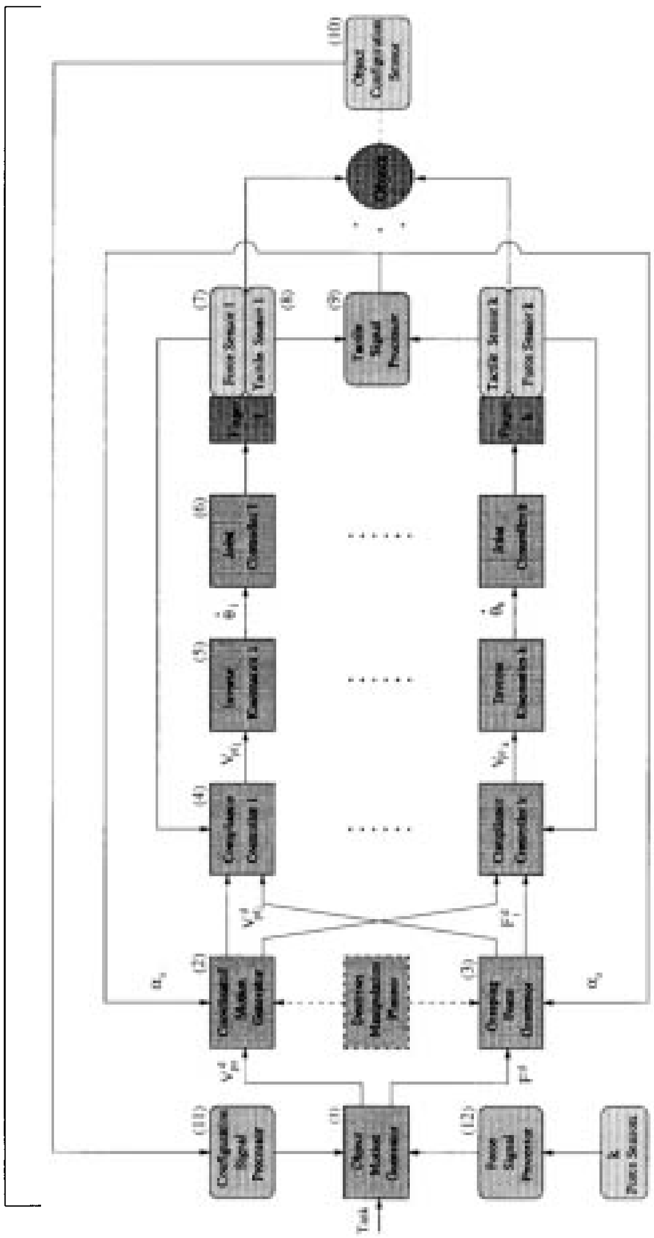

Figure 1: CoSAM[2]: A unified Control System Architecture for Multifingered Manipulation

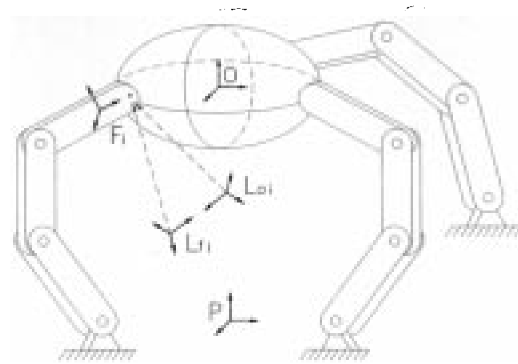

Figure 2: A multifingered robotic hand manipulation system

contact state between finger i and the object is referred to as the *coordinates of contact.* We denote by $\theta_i \in \mathbb{R}^{n_i}$ the joint position vector of finger i, and let $q_i = (\theta_i, \eta_i) \in \mathbb{R}^{n_i+5}$ be the extended joint coordinates of finger i.

We refer to ([22], [18] and [14]) for further notations and the various kinematic relations embedded in the multifingered manipulation system.

We model each finger as a position controlled device. Thus, fingertip velocity V_{pf_i} is treated as pseudo input of the system. The problem we address in this section is the following: *Given a desired object velocity V_{po}, compute the corresponding fingertip velocity V_{pf_i} which, when executed, will ensure the desired object velocity with the effect of contact constraint.*

Expressing the object transformation relative to the palm frame through finger i yields

$$g_{po} = g_{pf_i} \cdot g_{f_i l_{f_i}} \cdot g_{l_{f_i} l_{o_i}} \cdot g_{l_{o_i} o} \tag{1}$$

Differentiating (1) and using the fact that $g_{f_i l_{f_i}}$ and $g_{l_{o_i} o}$ are constant we have

$$V_{po} = Ad_{g^{-1}_{f_i o}} V_{pf_i} + Ad_{g^{-1}_{l_{o_i} o}} V_{l_{f_i} l_{o_i}} \tag{2}$$

Rearranging (2) gives an equation for $\tilde{V}_{pf_i}$, the local expression of V_{pf_i}, in terms of V_{po} and the contact velocity

$$\begin{aligned} \tilde{V}_{pf_i} &= Ad_{g^{-1}_{o l_{o_i}}} V_{po} + V_{l_{f_i} l_{o_i}} \\ V_{pf_i} &= Ad_{g_{f_i l_{o_i}}} \tilde{V}_{pf_i} \end{aligned} \tag{3}$$

A straight forward approach, or the so-called *individual joint control law* ([17]), to compute the fingertip velocity is to first multiply Eq. (3) by B_i^T, the transpose of the wrench basis of the assumed contact model, and then utilize the velocity contact constraint of

$$B_i^T V_{l_{f_i} l_{o_i}} = 0$$

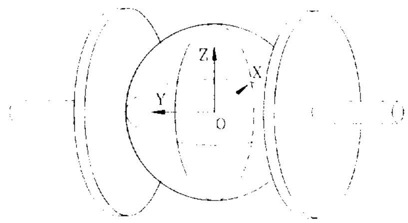

Figure 3: A two-fingered robotic hand manipulating a ball

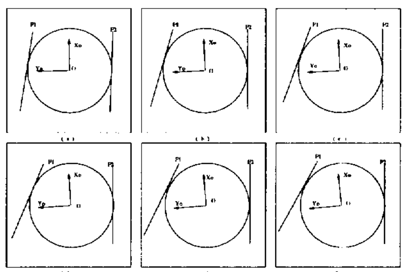

Figure 4: A simulated sequence of intermediate configurations showing undersirable consequence of uncoordinated manipulation.

to obtain the equation

$$B_i^T \tilde{V}_{pf_i} = B_i^T Ad_{g_{ol_{oi}}^{-1}} V_{po} \tag{4}$$

Finally, solving (4) for $\tilde{V}_{pf_i}$ gives a solution of the fingertip velocity V_{pf_i}.

While the *individual joint control law* is simple to implement, it suffers from a serious drawback as suggested by the following example (see [22] for additional examples).

Example 2.1. Consider a two-fingered robotic hand manipulating a ball as shown in Figure 3. The fingertips are both disk-shaped and the initial grasp configuration is antipodal which is considered optimal. Let the desired trajectory of the object be a rotation about the z-axis through a point in space, i.e., a screw motion generated by a twist of the form $\xi = (0, \rho, 0, 0, 0, 1)^T, \rho \neq 0$. Computing the fingertip velocity using (4) and a soft-finger contact model, we found that rolling motion indeed occur at the contact points and the fingers move closer to each other at one end. Eventually, the force-closure condition would fail to hold and the grasped object would risk being dropped. A sequence of intermediate configurations illustrating this undesirable phenomena is shown in Figure 4. □

The preceding example shows that, under the *individual joint control*, the fingers tend to minimize their own motion and have no regard to quality of the resulting grasp. As manipulation proceeds, grasp quality would degrade, leading eventually to failure of the force-closure condition and consequently dropping of the grasped object. To overcome this problem, we impose an additional constraint on fingertip velocities by simultaneously maintaining or optimizing grasp quality. Consider again Eq. (3), rewritten in the form

$$\tilde{V}_{pf_i} = Ad_{g_{ol_{o_i}}^{-1}} V_{po} - Ad_{g_{l_{oi}l_{fi}}} V_{l_{oi}l_{fi}}$$

The contact velocity can be expressed in terms of the rolling velocity which in turn, through the kinematic equations of contact, can be expressed in terms of $\dot{\alpha}_{o_i}$, i.e. we have

$$\boxed{\tilde{V}_{pf_i} = Ad_{g_{ol_{o_i}}^{-1}} V_{po} - T_i(\eta_i)\dot{\alpha}_{o_i}} \tag{5}$$

where

$$T_i(\eta_i) = \mathrm{Ad}_{g_{l_{oi}l_{fi}}} B_i^c R_0 (K_{f_i} + \tilde{K}_{o_i}) R_{\psi_i} M_{o_i}$$

and

$$B_i^c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \quad R_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Let

$$f : \mathbb{R}^{2k} \longrightarrow \mathbb{R} : (\alpha_{o_1}, \cdots \alpha_{o_k}) = \alpha_o \longmapsto f(\alpha_o) \tag{6}$$

be a function defined on the contact coordinates of the object measuring the quality of a grasp. In other words, conditioning of the grasp map $G(\alpha_o)$ is improved if $f(\alpha_o)$ is maximized (or minimized). Then, a simple and sensible solution for $\dot{\alpha}_{o_i}$ in (5) is given by

$$\dot{\alpha}_{o_i} = \lambda_i \nabla_i f(\alpha_o) \tag{7}$$

where $\nabla_i f(\alpha_o)$ is the gradient of $f(\alpha_o)$ with respect to α_{o_i} and $\lambda_i \in (0, 1)$ is a step-size. If minimizing the objective function is desired then negative of the gradient can be used in (7).

Example 2.2. Consider manipulation of a unit ball by a two-fingered robotic hand. We define the grasp quality function to be

$$f : S^2 \times S^2 \longrightarrow \mathbb{R} : (p_1, p_2) \longmapsto ||p_2 - p_1||^2 \tag{8}$$

where $p_i = \phi_i(\alpha_{o_i}), i = 1, 2$, is the point of contact with finger i, and $\phi_i : U \to S^2$ a coordinate map of the ball containing p_i. Note that f attains maximum when the grasp is antipodal and minimum when the two contact points coincide. Thus, it makes sense to improve grasp quality by maximizing $f(\cdot)$. The gradient of f is simply

$$\nabla_1 f = M_{o_1}^{-2}\left(\frac{\partial f}{\partial \alpha_{o_1}}\right)^T = -2M_{o_1}^{-2}\left(\frac{\partial \phi_1}{\partial \alpha_{o_1}}\right)^T (p_2 - p_1)$$

and

$$\nabla_2 f = M_{o_2}^{-2}\left(\frac{\partial f}{\partial \alpha_{o_2}}\right)^T = -2M_{o_2}^{-2}\left(\frac{\partial \phi_2}{\partial \alpha_{o_2}}\right)^T (p_1 - p_2)$$

where $\frac{\partial \phi_i}{\partial \alpha_{o_i}} \in \mathbb{R}^{3\times 2}$ is the Jacobian of ϕ_i.

An interpretation why the *individual joint control law* results in degrading of grasp quality is given as follows: First, the solution of $\tilde{V}_{pf_i}$ obtained from Eq. (4) is viewed as a function of V_{po}. Then, applying the $\tilde{V}_{pf_i}$ to Eq. (5) and rearranging the result gives an expression for $\dot{\alpha}_o$,

$$T(\eta)\dot{\alpha}_o = A(\alpha_o)V_{po} - \tilde{V}_{pf} \tag{9}$$

where

$$T(\eta) = \begin{bmatrix} T_1(\eta_1) & 0 \\ 0 & T_2(\eta_2) \end{bmatrix}, \qquad A(\alpha_o) = \begin{bmatrix} Ad_{g_{ol_{o1}}^{-1}} \\ Ad_{g_{ol_{o2}}^{-1}} \end{bmatrix}$$

Since $T(\eta)$ is full-rank, the unique solution of (9) is given by

$$\dot{\alpha}_o = (T^T(\eta)T(\eta))^{-1}T^T(\eta)\left(A(\alpha_o)V_{po} - \tilde{V}_{pf}\right) \tag{10}$$

Finally, taking the inner product of the right hand side of (10) with the gradient vector of $f(\cdot)$ the result is found to be negative along the desired object trajectory of Example 2.1. Thus, the grasp quality degrades as manipulation proceeds. $\square$

3 Grasping Force Generation

In this section, we consider the problem of generating proper contact or fingertip force $x \in \mathbb{R}^m$ so as to exert a desired object wrench $F^d \in \mathbb{R}^6$ on the grasped object through the effect of the grasp map, i.e., solving the equation

$$Gx = F^d \tag{11}$$

with $x = (x_1, \cdots x_k)^T \in \mathbb{R}^m$ lying in the friction cone $FC = FC_1 \times \cdots \times FC_k$ of the respect contact models. For PCWF the friction cone KC_i is defined by Coulomb's friction law,

$$FC_i = \left\{x_i \in \mathbb{R}^3 | \sqrt{x_{i,1}^2 + x_{i,2}^2} \le \mu_i x_{i,3}, x_{i,3} > 0\right\} \tag{12}$$

where $x_{i,3}$ is the normal force component at the point of contact, $x_{i,1}, x_{i,2}$ the tangential components and μ_i the coefficient of friction.

From the stand point of maintaining grasp constraint, the act of the *Grasp Force Generation* module complements that of the *Coordinated Motion Generation* module. The latter attempts to place the fingers at an optimal grasp configuration by optimizing quality of the grasp, which allows the former to generate proper contact forces within the limit of friction cones so as to enforce the contact constraint. Also note that the grasp map is in general not constant because of possibly rolling contact.

There have been a number of important studies on generation of grasping forces for multifingered manipulation, most of which were based on linear programming formulation with linearized friction constraint ([9], and [19]). Recently, a noval approach using gradient flows on the smooth manifold of symmetric and positive definite matrices has been proposed by Buss, Hashimoto and Moore ([3] and [2]). Their approach was based on an important observation that the friction constraints of (12) were equivalent to positive definiteness of the matrix $P = \text{Blockdiag}(P_1, \cdots P_k)$, where for a PCWF

$$P_i = \begin{bmatrix} \mu_i x_{i,3} & 0 & x_{i,1} \\ 0 & \mu_i x_{i,3} & x_{i,2} \\ x_{i,1} & x_{i,2} & \mu_i x_{i,3} \end{bmatrix} \tag{13}$$

Note that some elements of P are required to satisfy certain constraints. For example, the diagonal elements of P_i in (13) must be the same, i.e., $P_{i,11} = P_{i,22} = P_{i,33}$ for $i = 1, \cdots k$, and some off-diagonal elements of P must be zero, i.e., $P_{i,12} = P_{i,21} = 0$ for $i = 1, \cdots k$. Similarly, the off-diagonal blocks of P must be zero. These constraints are obviously linear on elements of P. Equality of two elements $P_{ij} = P_{kl}$ of $P \in \mathbb{R}^{n\times n}$ can be formulated as $e_i^T P e_j = e_k^T P e_l$, where $e_i \in \mathbb{R}^n$ is the unit vector with an 1 in the i^{th} entry and 0 otherwise. Define the Kronecker product of two matrices A and B by

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1k}B \\ \vdots & \cdots & \vdots \\ a_{n1}B & \cdots & a_{nk}B \end{bmatrix}$$

and the $vec(\cdot)$ operation by

$$vec(A) = (a_{11}, \cdots a_{n1}, a_{12}, \cdots a_{n2}, a_{1m}, \cdots a_{nm})^T$$

Then, the above constraint can be written as $(e_j^T \otimes e_i^T - e_k^T \otimes e_l^T)vec(P) = 0$. Similarly, to constrain the off-diagonal element P_{ij} to zero we write $e_i^T P e_j = 0$

or equivalently $(e_j^T \otimes e_i)vec(P) = 0$. The general form of all such linear constraints can be rewritten as

$$A_1 vec(P) = 0 \qquad (14)$$

where $A_1 \in \mathbb{R}^{\bar{m}_1 \times l}$ is a constant matrix with rank $\bar{m}_1$, $P \in \mathbb{R}^{n\times n}, P > 0$, $l = n^2$, and $\bar{m}_1$ is the number of linear equality constraints.

To exert an object wrench F^d on the grasped object the fingertip force $x \in \mathbb{R}^m$ is required to satisfy Eq. (11) which can also be written as a linear constraint of the form

$$A_2 vec(P) = F^d \qquad (15)$$

where $A_2 \in \mathbb{R}^{6\times l}$ is dependent on the contact coordinates of the object.

Collecting the constraints in (14) and (15) we have the following linear constraint on P

$$A\ vec(P) = q \qquad (16)$$

where with $\bar{m} = \bar{m}_1 + 6$,

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \in \mathbb{R}^{\bar{m}\times l}, \qquad q = \begin{bmatrix} 0 \\ F^d \end{bmatrix} \in \mathbb{R}^{\bar{m}}.$$

Let $P(n)$ denote the set of $n \times n$ symmetric and positive definite matrices. The objective function to be minimized is given by

$$\Phi : P(n) \longrightarrow \mathbb{R} : P \longmapsto \Phi(P) = \mathrm{tr}(W_p P + W_i P^{-1}) \qquad (17)$$

where $W_p, W_i \in \mathbb{R}^{n\times n}$ are weighting factors, and $\mathrm{tr}(\cdot)$ stands for the trace operator. The first term of $\Phi(\cdot)$ provides a linear cost associated with elements of P and the second term tends to infinity as P tends toward singularity or boundaries of the friction cones, see ([3]) for more detailed discussions.

$P(n)$ is a smooth manifold of dimension $n(n+1)/2$, endowed with a Riemannian metric

$$\langle V_p, W_p \rangle = \mathrm{tr}(V_p W_p), \qquad V_p, W_p \in T_p P(n). \quad (18)$$

Computing the gradient flow of (17) using the Riemannian metric (18) yields

$$\dot{P} = -\nabla\Phi(P(t)) = P^{-1} W_i P^{-1} - W_p, \qquad P(0) \in P(n). \quad (19)$$

[3] shows that the gradient flow (19) converges exponentially fast to the unique minimum $P_\infty = W_p^{-1/2}(W_p^{1/2} W_i W_p^{1/2})^{1/2} W_p^{-1/2}$.

Imposing the affine constraint (16) on the gradient flow (19) results in the constrained gradient flow

$$\mathrm{vec}(\dot{P}) = Q\mathrm{vec}(P^{-1} W_i P^{-1} - W_p) \qquad (20)$$

where $Q = (I - A^\# A)$ is the projection operator, $A^\# = A^T(AA^T)^{-1}$ the generalized inverse of A and $P(0) = P_0 \in P(n)$ satisfies the constraint (16). The discrete-time version of (20) is given by

$$\mathrm{vec}(P_{k+1}) = \mathrm{vec}(P_k) + \alpha_k Q\ \mathrm{vec}(P_k^{-1} W_i P_k^{-1} - W_p) \qquad (21)$$

where α_k is a suitably chosen step-size to ensure that $\Phi_{k+1} < \Phi_k$.

A major difficulty in implementing (21) for a multifingered manipulation system with rolling contact is real-time computation of the high dimension matrix Q. For a two-fingered hand with soft-finger contact, $Q \in \mathbb{R}^{64\times 64}$. Using a general algorithm to compute matrix inverses the computation time of Q on a MC68040 (33MHz) is around 3 seconds, which is obviously inadequate for real-time control. To get around with this difficulty we observe that $A_1 \in \mathbb{R}^{\bar{m}_1\times n}$ is a sparse and constant matrix and only $A_2 \in \mathbb{R}^{6\times n}$ is dependent on the contact coordinates of the object. Let

$$AA^T = \begin{bmatrix} A_1A_1^T & A_1A_2^T \\ A_2A_1^T & A_2A_2^T \end{bmatrix} := \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}$$

Computing the inverse of block matrices we have

$$(AA^T)^{-1} = \begin{bmatrix} A_{11}^{-1} + E\Delta^{-1}F & -E\Delta^{-1} \\ -\Delta^{-1}F & \Delta^{-1} \end{bmatrix}$$

where $\Delta = A_{22}A_{12}^T A_{11}^{-1} A_{12} \in \mathbb{R}^{6\times 6}$, $E = A_{11}^{-1}A_{12}$ and $F = A_{12}^T A_{11}^{-1}$. Since A_{11} is constant, A_{11}^{-1} can be computed off-line. What remains for on-line computation is the inverse of the 6×6 matrix Δ. Furthermore, using multiplication of sparse matrices we reduce the computation time of Q to 80ms.

4 Compliance Motion Generation

With the Coordinated Motion module, fingertip velocity $V_{pf_i}^d$ is generated. Realization of $V_{pf_i}^d$ implies that a desired object velocity V_{po}^d will be followed while quality of the grasp is either maintained or improved. On the other hand with the Grasping Force module, fingertip force $F_i^d = Ad^T_{g_{f_i l_{o_i}}^{-1}} B_i x_i$ expressed in the fingertip frame is generated. Realization of F_i^d implies compensation of an object wrench F^d with optimal fingertip forces within the limit of the friction cones. As each finger is modeled as a position controlled device, a compliance matrix $K_{ci} \in \mathbb{R}^{6\times 6}$ can be used to convert F_i^d into equivalent fingertip displacement

$$K_{ci}(F_i^d - F_i^m)$$

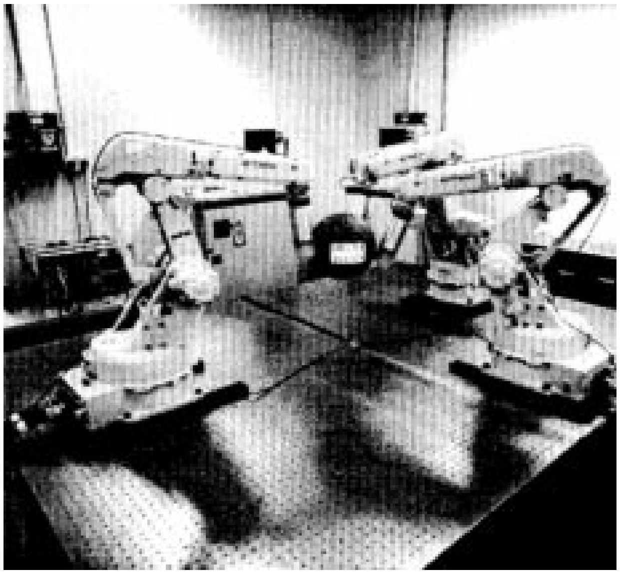

Figure 5: The HKUST multifingered robotic hand

where $F_i^m \in \mathbb{R}^6$ is the actual measured fingertip force. Specification of the compliance matrix is based mainly on experience with the actual hand system but we will assume that it is a symmetric matrix. Also note that if the actual force agrees with the desired force than no action will be necessary.

Combining output from the Coordinated Motion module with that of the Grasping Force module we obtain total displacement of the fingertip

$$\boxed{V_{pf_i} = V_{pf_i}^d + K_{ci}(F_i^d - F_i^m)} \tag{22}$$

which is sent to the Inverse Kinematics module to compute the required joint displacement of the fingers.

5 Implementation and Experiments

In this section, we give results of a simple manipulation experiment conducted based on CoSAM^2 (see [22] for additonal experiments).

The HKUST dextrous robotic hand developed at HKUST for study of multifingered manipulation is shown in Figure 5, more details of the system can be found in [22].

The experiment is to manipulate a ball with two flat fingertips. The initial grasp configuration is $\eta_1 = (\alpha_{o_1}, \psi_1, \alpha_{f_1}) = (8°, 90°, 0, 0, 0)$ and $\eta_2 = (\alpha_{o_2}, \psi_2, \alpha_{f_2}) = (6°, -90°, 0, 0, 0)$, which is not optimal. The desired object trajectory is a translation by *100mm* along the z- axis and the compliance matrix

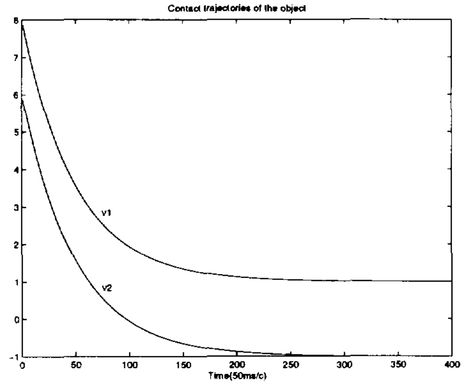

Figure 6: Contact trajectories of the ball

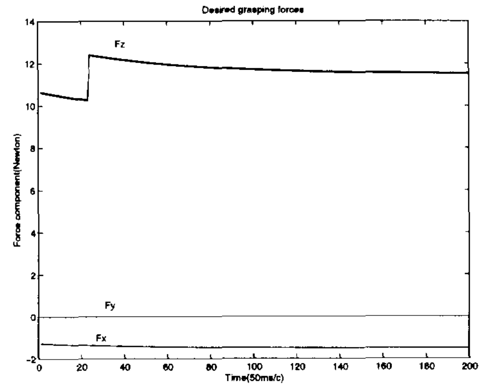

Figure 7: Desired grasping forces

K_{ci} is chosen to be $K_{ci}[2][2] = 0.02$ and all others entries be zero.

The step-size in the Coordinated Motion Generation module is chosen to be $\lambda = 0.1$ and the experimental results are shown in Figure 6, 7, and 8. From this experiment we see that not only the quality of grasp is improved but also optimal grasping forces with new grasp configurations were achieved. Figure 8 gives the measured grasping forces from the contact frame of the fingertip. Obviously, CoSAM^2 achieves all desired objectives. By increasing the step size λ, we can make the grasp configuration converge more quickly to its optimal value but up to a certain point slippage can occur. Thus, a trade-off has to be made between convergence rate and manipulation safety.

6 Conclusion

In this paper, we presented a unified Control System Architecture for Multifingered Manipulation (CoSAM^2). The two main modules of CoSAM^2, the Coordinated Motion Generation module and the

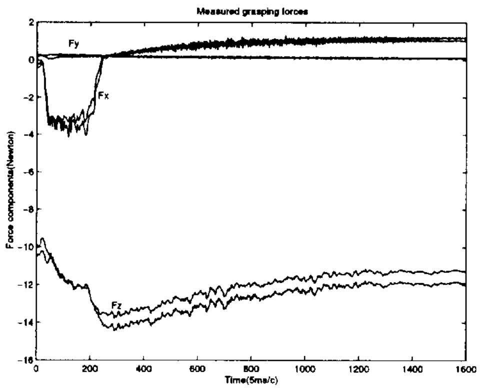

Figure 8: Measured grasping forces

Grasping Force Generation module, were studied in detail.

Several future research problems along the lines of the paper include:

- Generation of a suitable grasp quality function for both two-fingered and three-fingered manipulation with an arbitrarily shaped object, incorporating task and accessibility constraints. Results on potential field based motion planning can be potentially useful here;

- Proper formulation of the Dextrous Motion Planning module along with efficient algorithms under both rolling/sliding contact and finger gaiting. One can make use of Eq. (5) again by treating the second term as a feedforward term rather than as a feedback term. Finger gaiting and nonholonomic motion planning techniques can be used to generate the feedforward term;

- Additional experimental works on multifingered manipulation using sensory data feedbacks;

- An interpreter translating a high-level task to inputs of the various modules of CoSAM^2.

References

[1] P. Allen, A. Miller, P. Oh, and B. Leibowitz. Using tactile and visual sensing with a robotic hand. In *ICRA Proc.*, pages 676–681, 1997.

[2] M. Buss, L. Faybusovich, and J. Moore. Recursive algorithms for real-time grasping force optimization. In *ICRA Proc.*, pages 682–687, 1997.

[3] M. Buss, H. Hashimoto, and J. Moore. Dextrous hand grasping force optimization. *IEEE Trans. on R.&A.*, 12(3):406–418, 1996.

[4] A. Cole, J. Hauser, and S. Sastry. Kinematics and control of a multifingered robot hand with rolling contact. *IEEE Transaction on Automatic Control*, 34(4), 1989.

[5] M. Cutkosky, P. Akella, R. Howe, and I. Kao. Grasping as a contact of sports. Technical report, Department of Mechanical Engineering, Stanford University, 1987.

[6] R. Fearing. Tactile sensing for shape interpretation. In S.T. Venkatarman and T. Iberall, editors, *Dextrous robot hands.* Springer-Verlag, 1989.

[7] J.W. Hong, G. Lafferriere, B. Mishra, and X.L. Tang. Fine manipulation with multifinger hand. In *ICRA Proc.*, pages 1568–1573, 1990.

[8] S. Jacobsen, J. Wood, K. Bigger, and E. Inversen. The utah/mit hand: works in progress. *I.J.R.R.*, 4(3):221–250, 1986.

[9] J. Kerr and B. Roth. Analysis of multifingered hands. *I.J.R.R.*, 4(4):3–17, 1986.

[10] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on R.&A.*, RA2-06:62–72, 1990.

[11] Z.X. Li, P. Hsu, and S. Sastry. On grasping and coordinated manipulation by a multifingered robot hand. *I.J.R.R.*, 8:4, 1989.

[12] Z.X. Li and S. Sastry. Task oriented optimal grasping by multifingered robot hand. *IEEE Trans. on Robotics and Automation*, 4(1):32–44, 1988.

[13] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, (2:541-558), 1987.

[14] D. Montana. The kinematics of contact and grasp. *I.J.R.R.*, 7(3), 1988.

[15] D. Montana. The condition of contact grasp stability. In *ICRA Proc.*, pages 412–417, 1991.

[16] D. Montana. The kinematics of multi-fingered manipulation. *IEEE Trans. on R.&A.*, 11(4):491–503, 1995.

[17] R. Murray. Contr. exp. in planar manip. and grasp. Technical report, U.C. Berkeley, ERL Memo. No. UCB/ERL M89/3, 1989.

[18] R. Murray, Z.X. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[19] Y. Nakamura, K. Nagai, and T. Yoshikawa. Mechanics of coordinative manipulation by multiple robotic mechanisms. In *ICRA Proc.*, pages 991–998, 1987.

[20] E. Rimon and J. Burdick. On force and form closure for multiple finger grasps. In *ICRA Proc.*, pages 1795–1800, 1996.

[21] J. K. Salisbury. Kinematics and force control of articulated robot hands. In M. Mason and J. K. Salisbury, editors, *Robot Hands and the Mechanics of Manipulation.* MIT Press, 1985.

[22] Z. Qin Z.X. Li and S. Jiang. CoSAM^2: A unified control sys. arch. for multifingered manipulation (submitted). *IEEE Trans. on R.&A.*, 1997.