2012 12th IEEE-RAS International Conference on Humanoid Robots
Nov.29-Dec.1, 2012. Business Innovation Center Osaka, Japan

Robust Grasping for an Under-actuated Anthropomorphic Hand under Object Position Uncertainty

Yanyu Su, Yan Wu, Kyuhwa Lee, Zhijiang Du and Yiannis Demiris

***Abstract*—This paper presents a grasp execution strategy for grasping an object with one trial when there is uncertainty in the object position. This strategy is based on three grasping components: 1) robust grasp trajectory planning which can cope with reasonable amount of initial object position error, 2) sensor-based grasp adaptation, and 3) compliant characteristics of the under actuated mechanism. This strategy is implemented and tested on the iCub humanoid robot. Two experiments and a demo of the iCub robot playing the Towers of Hanoi game are carried out to verify our system. The results demonstrate that the iCub using this approach can successfully grasp objects under certain position error with its under-actuated anthropomorphic hand.**

I. Introduction

Robot grasping of daily objects is a fundamental function of many robotic applications in a daily environment. In an unstructured human environment, uncertainty of object pose is a major challenge for robot grasping [1]. Many methodologies have been proposed to tackle this problem.

The most direct approach is to estimate the object pose. [2] proposes to use machine vision and/or range-finders to reduce the level of uncertainty. Such approaches are usually combined with tactile sensors to cover the residual uncertainty from the estimation [3], but are heavily dependent on the hardware limitations. Tactile sensors can also be used independently to localise objects by touching [4], [5]. This method is reliable, but requires some trials to obtain a touch signature of the object and is computationally expensive to map [2]. Moreover, repeated trials are not an option in many applications.

Within an object pose error region, trajectory planning is used to generate a trajectory leading to successful grasp [6]. Constrained to a permissible initial object pose error, this approach uses a search algorithm to find the optimal grasping trajectory from a set which have a permissible initial object pose error region not smaller than requested. Although this method does not depend on external sensors, it requires prior knowledge of object shapes, gross object pose, accurate model of grasping process and expensive computations for an under-actuated hand.

Yanyu Su is jointly with the State Key Laboratory of Robot and System, Harbin Institute of Technology, China and the Electrical and Electronic & Engineering Department, Imperial College London, United Kingdom. yanyu.su@imperial.ac.uk

Yan Wu, Kyuhwa Lee and Yiannis Demiris are with Department of Electrical & Electronic Engineering, Imperial College London, United Kingdom {yan.wu08, k.lee09, y.demiris}@imperial.ac.uk

Zhijiang Du is with the State Key Laboratory of Robot and System, Harbin Institute of Technology, China. zjdu01@hit.edu.cn

It is possible to exploit the compliant characteristics of an under-actuated hand to naturally grasp object within a pose error region [7]. This method is robust to uncertainty of objects, however, it is also difficult to build the actuation model of its grasping process. Many open problems are still remaining on how to efficiently use such advantage.

Since close-loop dexterity control using visual feedback is computationally expensive and inaccurate because of the correspondence problem and occlusion, vision or range-finders used for initial object detection and pose estimation are more appropriate than being involved in control. The open option for reliable control strategy to grasp objects for an under-actuated robot hand is to combine trajectory-based planning method with tactile sensing to reduce computational cost and touch repetitions. To our knowledge such approach has not been done on under-actuated anthropomorphic hand before possibly due to the complexity of integration.

A recent advancement in grasping with tactile sensors sheds light on the simplification for such integration on an under-actuated hand [8]. Inspired from a human neuroscience study, this work presents a grasp controller which allows a pair of parallel gripper, similar to a two-finger precision grasp, to gently pick-up and set-down daily objects after a grasp point is selected. This idea of applying tactile feedback to enhance grasp stability can be extended to an under-actuated hand for a three-finger power grasp with two new features: 1) enhance sensitivity of tactile feedback with joint position of the finger, 2) robust to pose uncertainty.

In this paper, we investigate the approach of integrating the optimal grasping strategy given the permissible initial object position error (PIOPE), tactile-sensor-based grasping [8] and the compliant characteristics of an under-actuated anthropomorphic robot hand, assuming the prior knowledge of estimated object shape and pose, and the hand has approached the object and is ready for grasping.

The rest of the paper is organised as follows. The robotic platform is introduced in Section II followed by the system design in Section III. We present the model and controller details in Section IV and V before discussing three experiments to verify the strategy in Section VI. Finally, we conclude our work in Section VII with future work.

II. Robotic Platform

The robotic platform for our investigations is the iCub humanoid robot, developed by the RobotCub Consortium. It has two 7 degree-of-freedom (DoF) arms, each of which is attached with a 9-DoF anthropomorphic hand ("iCub-hand", shown in Fig. 1A). The iCub-hand has 20 joints, some of

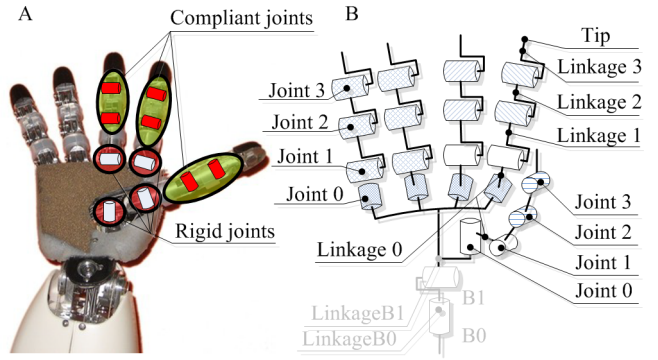

Fig. 1: A labelled picture of the right hand of the iCub (A) and the corresponding schematics of the hand (B), where the coupled joints are marked with the same pattens.

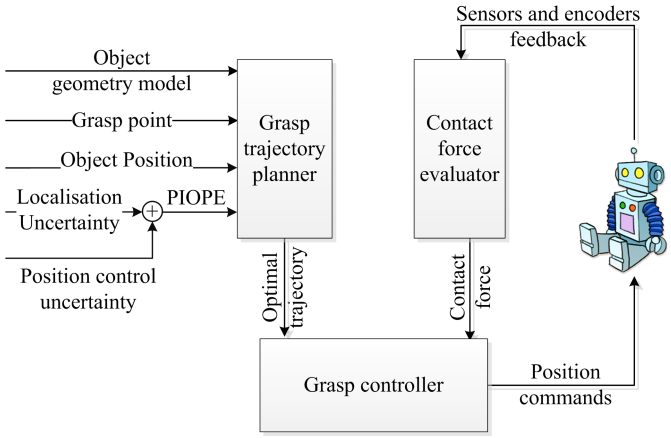

Fig. 2: The schematic overview of our system design. The system consists of 3 modules and requires 5 external input signals. The *Grasp trajectory planner* generates an optimal trajectory based on the given signal. The trajectory is then sent to the *Grasp controller* which runs a state machine to execute the grasp. The *Contact force evaluator* provides feedback to the Grasp controller based on the contact forces received from the sensors.

which are coupled and under-actuated including the distal joints of all fingers. This design, show in Fig. 1B, enables the phalanges of these fingers to possess compliant characteristics. Each finger-tip is equipped with 12 capacitive pressure sensors covered by the black silicon foam. The tactile sensors installed at each finger-tip produce 8-bit readings sampled at $50Hz$.

III. System Design and Overview

In this work, we propose a grasping system that takes uncertainty of object position into account. The proposed system, shown in Fig. 2, consists of three modules. It depends on five input signals from external sources which are assumed to be given in this work.

The format of the inputs is as follows: 1) the object geometry model (*OBJ*) is represented by a point cloud; 2) the grasp point ($\boldsymbol{P}_{gd}$) is represented by a set of position where the robot fingers will touch the object, and is represented by a $N_f \times 3$ matrix, where N_f is the number of fingers; 3) the object pose is a 6-D vector in the robot frame of reference; 4) the localisation uncertainty ($\boldsymbol{e}_l$) and the position control uncertainty ($\boldsymbol{e}_p$) are the errors of the corresponding methods, and are represented by 3-D column vectors, thus the required PIOPE ($\boldsymbol{e}_r$) must be no less than the propagation of them.

The inputs are provided to the *Grasp trajectory planner*, discribed in section V.A, which generates an optimal trajectory for all finger joints by minimizing the distance between the contact point estimated by a simulator and the input grasp point constrained by the required PIOPE. The simulator also evaluates if the grasp can be successfully achieved before applying genetic algorithm to generate the desired optimal trajectory.

The *Grasp controller*, described in section V.C, receives the trajectory and executes the grasping process by a state machine with 3 discrete states: 1) *Touch phase*: fingers follow the optimised trajectory towards the object and individually stops upon contact; 2) *Grasp phase*: fingers apply appropriate forces to hold the object in place; 3) *Open phase*: forces are removed before fingers are fully opened.

This grasping process is monitored by the *Contact force evaluator*, descirbed in section V.B, to provide the Grasp controller with updates of finger contact forces. These forces are derived from sensor readings of the robot's tactile sensors, hall-effect sensors.

IV. Kinematic and Kinetostatic Analysis of Grasp

Having fewer actuators than joints, the underactuated hand needs external constrains to determine the motion of fingers. Up to recent researches, the external constrains are specific to applications, for example, [9] uses elastic averaging to map actuator velocity to grasped object velocity for manipulating grasped objects and, [10] uses the kinematostatic and the quasi-static models for the parallel underactuated hand. In this section, kinematics and kinetostatics of the iCub-Hand fingers are analysed for the purpose of simulating the proposed grasp process.

The mechanical configuration of the thumb, index and middle fingers of the iCub-Hand are the same except that the middle finger does not have an abducting/adducting joint. Hence, the same model is adopted for these fingers as shown in Fig. 3 apart from the abducting/adducting joint of the middle finger being fixed. Two assumptions are made in this model: 1) geometry of the finger is modelled as simple lines, 2) joint friction and contact torque are negligible.

A. Terminology

We denote θ_j^f as the position of the joint j on finger f, where $f = 1,2$ *and* 3, corresponding to the thumb, index and middle finger respectively and, joint $j = 0,1,2$ *and* 3, corresponding to the descriptions in Fig. 1B. The position vector of the finger joints can therefore be expressed as:

$$\boldsymbol{\theta} = [\theta_0^1\ \theta_1^1\ \theta_2^1\ \theta_3^1 \quad \theta_0^2 \ldots \theta_3^2 \quad \theta_0^3 \ldots \theta_3^3\ ] \tag{1}$$

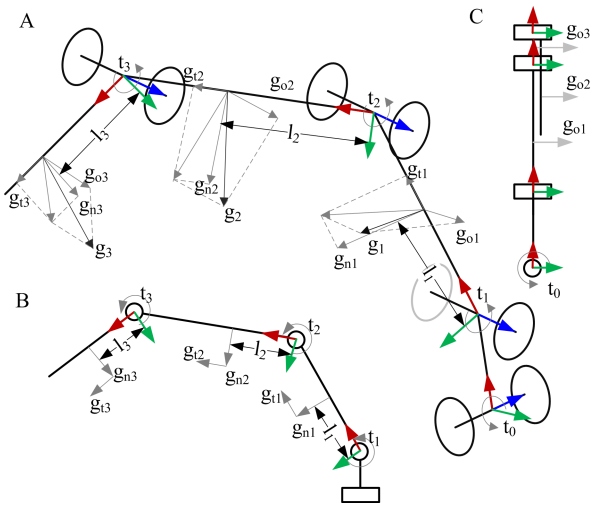

Fig. 3: Kinetostatic model of iCub-hand finger (A). As abducting/adducting motion is independent to flexing/extending motion, they can be separately analysed with models of flexing/extending motion (B) and abduction/adduction motion (C)

and the position vector of the actuators are:

$$\boldsymbol{\theta}_m = [\theta^1_{m0}\ \theta^1_{m1}\ \theta^1_{m2,3} \quad \theta^2_{m0}\ \theta^2_{m1}\ \theta^2_{m2,3} \quad \theta^3_{m1}\ \theta^3_{m2,3}\ ] \tag{2}$$

where m denotes the actuator, which can drive multiple fingers or joints.

We use standard D-H parameters to analyse the kinematics for each finger of the iCub-hand. The forward kinematics is expressed as:

$$\boldsymbol{T}^f_i = T^f_i(\boldsymbol{\theta}^f) \tag{3}$$

where $\boldsymbol{T}^f_i \in R^{4\times4}$ is the rot-translation matrix for phalange i (4 denoting the tip of finger) and $\boldsymbol{\theta}^f = [\theta^f_1 \theta^f_2 ... \theta^f_4]$. For convenience, $\boldsymbol{T}^f_i$ is divided into a rotational matrix $\boldsymbol{R}^f_i \in R^{3\times3}$ and a position vector $\boldsymbol{p}^f_i \in R^3$.

B. Flexing/extending Kinetostatics

Flexing/extending involves joints 1, 2 and 3 as shown in Fig. 3B. Using virtual work of this system [11], the static model of the finger is expressed as:

$$(\boldsymbol{t}^f_e)^T \boldsymbol{\omega}^f_e = (\boldsymbol{g}^f_e)^T \boldsymbol{J}^f_e \boldsymbol{S}^f_e \boldsymbol{\omega}^f_e \tag{4}$$

where $\boldsymbol{t}^f_e$ is the total exerted torque, $\boldsymbol{\omega}^f_e$ is the corresponding joint velocity, $\boldsymbol{g}^f_e$ is a contact force vector. Matrix $\boldsymbol{J}^f_e$ depends on the contact location, relative orientation of the phalanges. Matrix $\boldsymbol{S}^f_e$ describes transmission mechanism. When θ^f_1, θ^f_2 and θ^f_3 are within working space (0° to 90°), each term is listed as follows:

$$\boldsymbol{t}^f_e = [t^f_{m1} \quad t^f_{m2,3} - k^f_2 \Delta\theta^f_2 \quad -k^f_3 \Delta\theta^f_3]^T \tag{5}$$

where t^f_{m1} and $t^f_{m2,3}$ is actuation torque, k^f_i is the stiffness of spring attached to joint i, $\Delta\theta^f_i$ is the coordinate of joint i relative to its rest configuration.

$$\boldsymbol{\omega}^f_e = [\dot{\theta}^f_{m1} \quad \dot{\theta}^f_{m2,3} \quad \dot{\theta}^f_3]^T \tag{6}$$

where $\dot{\theta}^f_{m1}$, $\dot{\theta}^f_{m2,3}$ and $\dot{\theta}^f_3$ are first-order derivative of θ^f_{m1}, $\theta^f_{m2,3}$ and θ^f_3.

$$\boldsymbol{g}^f_e = [g^f_{n1} \quad g^f_{n2} \quad g^f_{n3}]^T \tag{7}$$

where g^f_{ni} are resolved contact force along z-axis of corresponding joint frame as shown in fig. 3A.

$$\boldsymbol{J}^f_e = \begin{pmatrix} l^f_1 & 0 & 0 \\ l^f_2 + a^f_1 c^f_2 + \mu^f_2 a^f_1 s^f_2 & l^f_2 & 0 \\ l^f_3 + a^f_1 c^f_{23} + a^f_2 c^f_3 + \mu^f_3 a^f_1 s^f_{23} + \mu^f_3 a^f_2 s^f_3 & l^f_3 + a^f_2 c^f_3 + \mu^f_3 a^f_2 s^f_3 & l^f_3 \end{pmatrix} \tag{8}$$

where l^f_i is the contact point in corresponding joint frame, μ^f_i is defined as $\mu^f_i = g^f_{ti}/g^f_{ni}$, a^f_i is the length of each phalange. s^f_3, c^f_{23} and etc denote the trigonometric functions e.g. $s^f_3 = \sin(\theta^f_3)$, $c^f_{23} = \cos(\theta^f_2 + \theta^f_3)$.

$$\boldsymbol{S}^f_e = \begin{pmatrix} \frac{r^f_{m1}}{r^f_1} & 0 & 0 \\ 0 & \frac{r^f_{m2,3}}{r^f_2} & -\frac{r^f_3}{r^f_2} \\ 0 & 0 & 1 \end{pmatrix} \tag{9}$$

where r^f_i are radii of the joint pulley, r^f_{m1} and $r^f_{m2,3}$ are radii of the actuator pulley.

C. Abducting/adducting Kinetostatics

Only joint 0 is involved in abduction/adduction as shown in fig.3C. Assuming it works in its working space (0° to 15°), the same analysis applies:

$$t^f_{m0}\dot{\theta}^f_{m0} = (\boldsymbol{g}^f_a)^T \boldsymbol{J}^f_a S^f_a \dot{\theta}^f_{m0} \qquad \begin{aligned} \boldsymbol{g}^f_a &= [\tau^f_1 g^f_{n1} \quad \tau^f_2 g^f_{n2} \quad \tau^f_3 g^f_{n3}]^T, \\ \boldsymbol{J}^f_a &= [a^f_0 + l^f_1 s^f_1 \quad a^f_0 + l^f_1 s^f_1 + l^f_2 s^f_2 \\ &\quad\ a^f_0 + l^f_1 s^f_1 + l^f_2 s^f_2 + l^f_3 s^f_3]^T, \\ S^f_a &= \tfrac{r^f_{m0}}{r^f_0} \end{aligned} \tag{10}$$

with the τ^f_i defined as $\tau^f_i = g^f_{oi}/g^f_{ni}$.

D. Kinematics for touch phase

During the touch phase, each finger moves from its initial position towards the object and stops upon contact. Hence, the kinetostatic can be simplified by setting $\boldsymbol{g}^f_e$ and $\boldsymbol{g}^f_a$ to $\mathbf{0}$ in (4) and (10) then the actuator and finger joints follow:

$$\begin{aligned} &\theta^1_0 = \frac{r^1_{m0}}{r^1_0}\theta^1_{m0}, \quad \theta^2_0 = \frac{r^2_{m0}}{r^2_0}\theta^2_{m0}, \\ &\theta^3_0 = 0, \quad \theta^f_1 = \frac{r^f_{m1}}{r^f_1}\theta^f_{m1}, \\ &\theta^f_2 = \frac{k^f_3}{k^f_2 + k^f_3}\frac{r^f_{m2,3}}{r^f_2}\theta^f_{m2,3}, \\ &\theta^f_3 = \frac{k^f_2}{k^f_2 + k^f_3}\frac{r^f_{m2,3}}{r^f_3}\theta^f_{m2,3} \end{aligned} \tag{11}$$

The kinematics during touch phase is then obtained by combining (3) and (11).

V. System Implementation

A. Grasp trajectory planner

Deriving the optimal finger trajectory that minimises the distance between the contact point and the desired grasp point under the required constraints is essentially an optimisation problem. This problem can be solved in two steps: 1) building a simulator for the proposed grasp execution process; 2) using genetic algorithm to find the desired trajectory.

1) Assumptions and conditions: As discussed in the previous sections, simulation of grasp execution requires expensive computations, but with the help of the sensor-based grasp execution technique, it is possible to make some assumptions to simplify the stable grasp condition, the models of the under-actuated hand and the soft silicon foam at the fingertips.

As tactile sensors are only installed beneath the silicon foam, to obtain the correct sensor readings for contact force computation, only fingertips are used to grasp the object. The following assumptions and conditions are also made to facilitate the simulation:

- Grasp process follows quasi-static conditions.
- Finger phalanges are simple rods for easy computation.
- Robot arm and objects are stationary during experiments. Deformation of silicon foam on tactile sensor is negligible.
- Successful grasp condition is defined as all fingertip tactile sensors are in contact with object and the angle between the normal vectors of tactile sensor and object surface is below a reasonable threshold.
- Only traces that satisfy the stable grasp condition with only fingertip touching the object during the intermediate process are considered valid.

2) Grasp simulator: A Matlab-based simulator is implemented to simulate the touch phase and determine if a given trajectory can lead to a successful grasp based on the kinetostatics for the touch phase, successful grasp conditions and assumptions. In this simulator, objects are represented by Delaunay triangulations. Contact detection is calculated by finding the shortest distance between the object surface and each finger. This simulator is expressed by following function:

$$\begin{aligned} \boldsymbol{p}_g &= P_g(OBJ,\, \boldsymbol{c},\, \boldsymbol{\Theta}^1,\, \boldsymbol{\Theta}^2,\, \boldsymbol{\Theta}^3) \\ b_g &= B_g(OBJ,\, \boldsymbol{c},\, \boldsymbol{\Theta}^1,\, \boldsymbol{\Theta}^2,\, \boldsymbol{\Theta}^3) \end{aligned} \tag{12}$$

where $\boldsymbol{p}_g$ is the grasp point in the object frame, $b_g = 1$ if stable grasp archived otherwise $b_g = 0$, OBJ is the model of the object, $\boldsymbol{c}$ is the object position, $\boldsymbol{\Theta}^f$ is the matrices for trajectories of each actuators. Three simulation instances are shown in Fig.4.

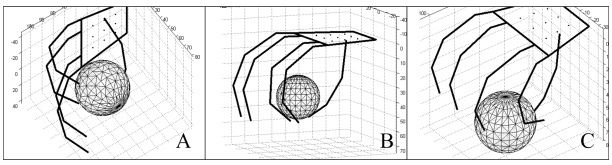

Fig. 4: Examples simulation of successful grasp of a sphere (A) and (B), and unsuccessful grasp (C).

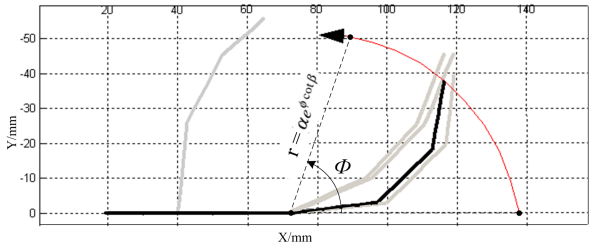

Fig. 5: A side view example of a typical fingertip trajectory ($\alpha = 68$, $\beta = 1.39$) for index finger. The red curve is the fingertip trajectory, where r is the distance of fingertip to metacarpophalangeal joint, ϕ is the angle to the palm plane.

3) Grasp trajectory template: We adopt the fingertip trajectory of human grasping [12] as a template for grasp trajectory generation. This template can be transformed into the finger frame of reference of the robot at phalange 1 by:

$$\boldsymbol{p}^f_{d4} = \boldsymbol{R}^f_1 rotz(\phi)[\alpha e^{\phi \cot\beta} \quad 0 \quad 0]^T \tag{13}$$

where ϕ is the angle given in Fig.5, α and β are grasp-specific parameters. (13) is transformed into joint space by inverse kinematics:

$$\boldsymbol{\Theta}^f = fun_t(\theta^f_0,\, \alpha,\, \beta,\, \boldsymbol{\phi}) \tag{14}$$

where $\boldsymbol{\phi}$ is the vector of ϕ from initial to end positions, θ^f_0 is position of joint 0, which is usually fixed during grasping.

4) Grasp trajectory optimisation: The optimisation variables are the three parameters in (14): $\boldsymbol{\theta}_0 = \{\theta^f_0 \,|f = 1,2,3\}$, $\boldsymbol{\alpha} = \{\alpha^f \,|f = 1,2,3\}$ and $\boldsymbol{\beta} = \{\beta^f \,|f = 1,2,3\}$, while $\boldsymbol{\phi}$ is set as needed in experiment.

PIOPE is represented by a set of object centre points:

$$\boldsymbol{P}_{piope} = \{\boldsymbol{p}^k_o | k = 1,2,..N_p\} \tag{15}$$

where $\boldsymbol{p}^k_o$ is the position of objects, N_p is number of positions in the PIOPE.

By substituting these optimisation variables into (14) and (12), the simulation can be expressed as:

$$\begin{aligned} \boldsymbol{P}^k_g, &= P_o(OBJ,\, \boldsymbol{p}^k_o,\, \boldsymbol{\theta}_0,\, \boldsymbol{\alpha},\, \boldsymbol{\beta}) \\ b^k_g &= B_o(OBJ,\, \boldsymbol{p}^k_o,\, \boldsymbol{\theta}_0,\, \boldsymbol{\alpha},\, \boldsymbol{\beta}) \end{aligned} \tag{16}$$

If the object is located in a point within the desired PIOPE, successful grasp can be attained. Thus, the optimisation constraint is to minimise the distance between the contact points $\boldsymbol{P}^k_g$ and the desired grasp point $\boldsymbol{P}_{gd}$:

$$\arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum(||\boldsymbol{P}^k_g - \boldsymbol{P}_{gd}||), \quad subject\ to: N - \sum b^k_g = 0 \tag{17}$$

We solve this optimisation problem using genetic algorithm with the following parameters for best performance: population is set to 8, boundary of θ_0^1 to $[0°, 90°]$, θ_0^2 to $[0, 15°]$, α to $[60, 72]$ (mm) and β to $[90°, 105°]$, other options are set to default in the standard MATLAB toolbox.

B. Contact force evaluator

To detect contact and evaluate the contact forces, the contact force evaluator is built by integrating tactile sensors, hall-effect sensors, as inspired from [8], [13].

1) Contact detection and force evaluation using tactile sensors: The fingertip tactile sensor value g_{sa}^f is given as:

$$g_{sa}^f = max(g_{cl}^f) \tag{18}$$

where g_{cl}^f represents the value from each capacitive sensor on the finger tip, and the finger force disturbance g_{fa}^f is given as:

$$g_{fa}^f = HF(Z)g_{sa}^f(Z) \tag{19}$$

where $HF(Z)$ is a discrete-time first-order Butterworth high-pass filter with a cut-off frequency of $4Hz$, Z denotes signal in z-domain.

The g_{sa}^f is used as the evaluation of contact force. Tactile sensor based method is robust to disruptions. However, it requires the contact stiffness to be higher than stiffness of the finger joint spring. Soft or light objects cannot therefore be detected by this method.

2) Contact detection using hall sensors: The position difference of the coupled complaint joints and the actuators can be expressed as:

$$g_{sp}^f = f_{sp}^f(\theta_3^f, \theta_2^f) \tag{20}$$

g_{sp}^f is used to evaluate contact in touch phase.

This value can be used to detect application of external forces on the finger and is available from a standard iCub module. As compared to the tactile sensor based method, this method can detect object with much low surface stiffness but is less robust to disruptions.

3) Contact detection: Thresholds are empirically implemented to detect contact for each finger:

$$(g_{fa}^f > h_{fa}^f) \vee (g_{sp}^f > h_{sp}^f) \tag{21}$$

where g_{fa}^f and g_{sp}^f are given in [8]; h_{fa}^f and h_{sp}^f are the respective thresholds.

C. Grasp controller

Similar to the approach in [8], we divide the grasping process into three phases: touch, grasp and open.

1) Touch phase: In the touch phase, each individual finger follows the trajectories given by the grasp trajectory planner. The same velocity and acceleration are applied to all fingers to synchronise the finger movements. Each finger stops if contact is detected on the finger.

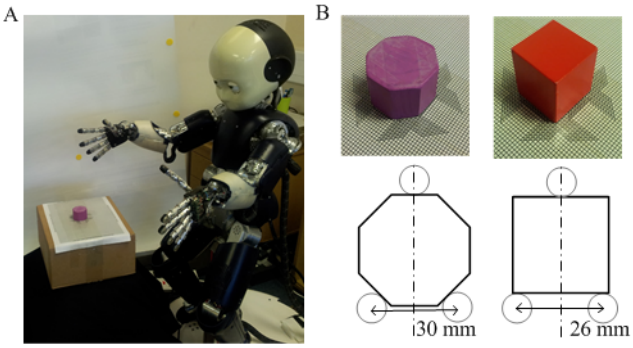

Fig. 6: Experimental setup (A), the selected objects (B) and the desired grasp points (C)

Item	Required PIOPE(mm)			Size(mm)	
	X	Y	Z	Width	Height
Cube	±6.0	±4.0	±5.0	30.0	30.0
3D Octagon	±8.0	±6.0	±5.0	40.0	30.0

TABLE I: Object models and required PIOPEs in experiments

2) Grasp phase: The goal of the grasp phase is to apply appropriate forces to hold the object. According to [8], the force applied on g_d is given as:

$$g_d^f = \max_t(g_{sa}^f)\frac{k_h}{|\omega_t|} \tag{22}$$

where $\max_t(g_{sa}^f)$ is the maximum force observed by the tactile sensor during the contact settling time in touch phase, k_h is the parameter representing grasp hardness.

Grasp phase starts after contact is detected on all fingers in touch phase. Only joint 2 and 3 are active in this process. The applied force is simply controlled by a PID controller using result from contact force evaluator as feedback, desired contact force g_d^f as reference and sending joint position as output.

3) Open phase: The open phase is called when the grasping process is interrupted by an open signal. This module commands the fingers to remove forces from the object, return to initial state for grasping.

VI. Experiments

Our grasping system is implemented and tested on the iCub humanoid robot. Thumb, index and middle fingers of the iCub-Hand are used in all experiments. Two experiments and a demonstration of the iCub playing the Towers of Hanio game are conducted to evaluate the performance of the approach.

A. Experiment Setup

As shown in Fig.6A, the object is placed on a piece of coordinate paper (2mm resolution) in the workspace of the iCub. The position of the coordinate paper is carefully calibrated throughout all experiments. The position of the iCub-Hand is fixed to minimise errors induced by arm movements.

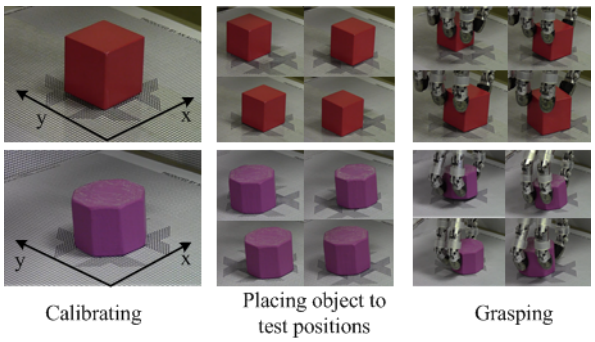

Fig. 7: The process of Experiment A from left to right: calibrating the coordinate paper, place object at a test position displaced from the reference centre, execute and evaluate the grasp.

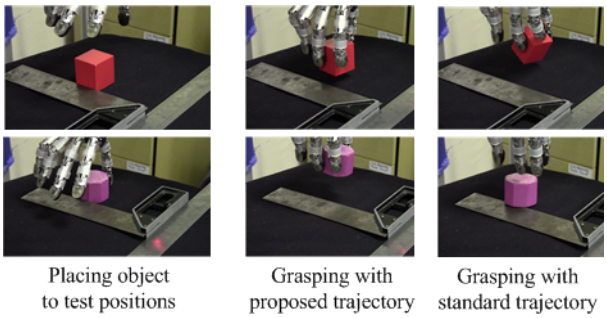

Fig. 8: The process of Experiment B from left to right: place object at a test position displaced from the refernece centre, execute a grasping method and repeat the process for the other method

Two objects of typical shapes are selected for the experiments as shown in Fig. 6B, a cube and a 3D octagon with corresponding required PIOPEs, geometric models and desired grasp points tabulated in TABLE I and Fig. 6C

We define successful grasp as follows: 1) object is lifted for 10 sec; 2) maximum object displacement is less than 10% of the size of the object; 3) all three fingers are in contact with the object.

B. Experiment A

To validate the grasp strategy, the test objects are placed in a set of positions within the boundary of the required PIOPE for grasping. This process is shown in Fig. 7. Four trials are tested for each of the distance range:

- $\pm$0% of required PIOPE on XY
- $\pm$50% of required PIOPE on X, Y and XY
- $\pm$100% of required PIOPE on X, Y
- $\pm$110% of required PIOPE on X, Y

For each distance, we record the count of successful trials, the approximate object pose, the joint positions and the distance between the desired grasp points and the actual contact points.

C. Experiment B

The standard iCub grasping module requires the position of the object, the finger velocity, contact force threshold and time-out threshold as its input. It uses either spring module or tactile sensors as contact force feedback. To investigate the improvement of our system over the standard iCub grasping module, a comparison experiment is carried out to grasp the objects at the same position.

Experimental trials are taken at different positions along the Cartesian axes of the coordinate paper to obtain the actual PIOPE for each method. A ruler with higher accuracy (0.5mm)is used. The experiment process is shown in Fig. 8

Position	Cube		3D Octagon	
	Success Rate	Dis.(mm)	Success Rate	Dis.(mm)
$\pm$0% XY	4/4(100%)	0.8	4/4(100%)	0.9
$\pm$50% X	4/4(100%)	4.2	4/4(100%)	5.4
$\pm$50% Y	4/4(100%)	3.5	4/4(100%)	4.8
$\pm$50% XY	4/4(100%)	4.8	4/4(100%)	6.1
$\pm$100% X	4/4(100%)	7.7	4/4(100%)	9.8
$\pm$100% Y	4/4(100%)	6.5	4/4(100%)	7.6
$\pm$110% X	4/4(100%)	8.9	4/4(100%)	11.5
$\pm$110% Y	4/4(100%)	8.1	0/4(0%)	N/A

TABLE II: Results for Experiment A. Dis denotes the mean distance between the desired and the actual grasp points of the three figners.

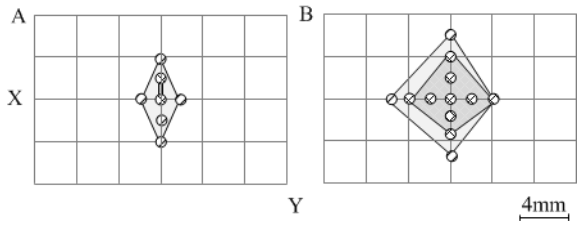

Fig. 9: The PIOPE diagrams of the Cube (A) and the 3D Octagon (B) for the two methods in comparison. Light grey and dark grey areas denote the PIOPEs of our method and the standard iCub module respectively.

D. Results and Discussions

1) Experiment A: The results for Experiment A are tabulated in TABLE II. All displacements within the PIOPEs of the objects result in successful grasp and reasonable discrepancy in grasp points. The system is able to grasp the Cube when it placed 10% outside its PIOPE. The proposed system fails in the trials at $\pm$110 Y position of the 3D octagon, this is due to the initial error is too big that index and middle finger cannot contact the object at the same time.

The discrepancies in distances between desired grasp points and evaluated actual contact points mainly come from three sources: 1) the displacement of the object causes a part of the same value as the displacement; 2) the trajectory plan method gains overall optimised result, so the distances for most initial error positions are not zero; 3) unwanted displacement of the object during touch and grasp phase also causes distances. The last two sources are related to trajectory planning, which can be evaluated by subtracting the displacement of the object from the the distances, the biggest distance of cube is 2.5 mm (8% of the object size) within the PIOPE, and 3.8 mm (9.5% of the object size) of 3D octagon.

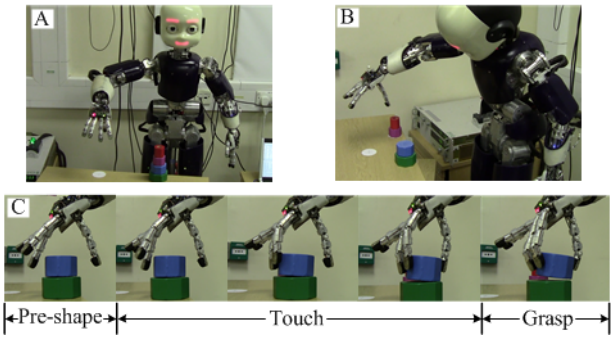

Fig. 10: The 4-level Hanoi Towers game setup (A), iCub playing the Hanoi Towers game (B) and the three phases of grasping the the blue 3D Octagon (C). The towers are 3D Octagons in different colors and sizes (the smallest one on the top) putting on three white points of a wooden surface.

2) Experiment B: The experimental results are plotted onto two PIOPE diagrams for comparison as shown in Fig. 9. For both the cube and the 3D octagon, the PIOPEs of our method are visually larger than the standard iCub module. This is especially significant for the cube digram. The PIOPE of the iCub module is a narrow strip. This result suggests that our system can increase the robustness of grasping under object pose uncertainty.

As compared to a simple grasp trajectory planning method, for the optimisation of which the typical time consumption is several hours [6], the average time consumption for the experiments is 1 minute (on Intel i7 3.5 GHz CPU platform). The average grasping time for our system is 15 sec while traditional tactile sensor based grasping method, such as [5], takes about 45 sec.

E. Towers of Hanoi Demo

Two features of playing a 4-level Hanoi Towers game make it a suitable complex task for testing our grasping system: 1) blocks of different sizes are moved from place to place, resulting in 120 times of moving and grasping actions at different locations. This can test the robustness of the system over accumulated arm position errors; 2) smaller blocks are on top of larger ones which require a reasonable PIOPE region.

The Hanoi Towers demo, shown in Fig. 10, is carried out successfully with the proposed method. A video of this demonstration can be found at the website of the Personal Robotics laboratory.

VII. Conclusion

In this paper, we presented an integrated grasp execution system for an anthropomorphic hand to grasp objects in one trial. This system is based on three grasping components: 1) optimal grasping trajectory planning with initially-given position error region of the object, 2) sensor based grasp adaptation, and 3) compliant characteristics of under actuated mechanism.

To implement this strategy several work is done in this paper: 1) the kinetostatics of grasping process is analysed; 2) the sensor based grasp execution strategy is applied to the iCub-hand; and 3) a simulator is built for the grasping process of this system, based on which a trajectory planner is applied to gain desired trajectory leading to a grasp robust to position uncertainty of the object. Three experiments are carried out to verify our system both statistically and empirically. The computational cost and results demonstrate that this approach can successfully grasp objects with a tendon-driven under actuated anthropomorphic hand.

As many daily objects are similar and sharing the same local geometry features near grasp points, our future work is to investigate how to apply the proposed method in grasping those unknown similar objects.

VIII. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 Challenge 2 Cognitive Systems, Interaction, Robotics under grant agreement No [270490]- [EFAA].

References

[1] C. Kemp, A. Edsinger, E. Torres-Jara, and D. Robots, "Challenges for robot manipulation in human environments [Grand Challenges of Robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, pp. 20–29, Mar. 2007.

[2] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Robust grasping under object pose uncertainty," *Autonomous Robots*, vol. 31, pp. 253–268, July 2011.

[3] A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2586–2591, IEEE, Oct. 2010.

[4] L. Natale, "A sensitive approach to grasping," *Proceedings of the sixth international workshop*, 2006.

[5] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1811–1816, Oct. 2009.

[6] H. Dobashi, A. Noda, Y. Yokokohji, H. Nagano, T. Nagatani, and H. Okuda, "Derivation of optimal robust grasping strategy under initial object pose errors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2096–2102, IEEE, Oct. 2010.

[7] A. M. Dollar, R. D. Howe, and S. Member, "Simple, Robust Autonomous Grasping in Unstructured Environments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, no. April, pp. 4693–4700, IEEE, Apr. 2007.

[8] J. M. J. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, pp. 1067–1079, Dec. 2011.

[9] L. U. Odhner and A. M. Dollar, "Dexterous manipulation with underactuated elastic hands," *2011 IEEE International Conference on Robotics and Automation*, pp. 5254–5260, May 2011.

[10] C. Quennouelle and C. Gosselin, "Quasi-static modelling of compliant mechanisms: application to a 2-DOF underactuated finger," *Mechanical Sciences*, vol. 2, pp. 73–81, Feb. 2011.

[11] L. Birglen, C. M. Gosselin, and T. Laliberté, *Underactuated Robotic Hands*. Springer, 2008.

[12] D. G. Kamper, E. G. Cruz, and M. P. Siegel, "Stereotypical fingertip trajectories during grasp.," *Journal of neurophysiology*, vol. 90, pp. 3702–10, Dec. 2003.

[13] A. Schmitz, U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "Design, realization and sensorization of the dexterous iCub hand," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 186–191, IEEE, Dec. 2010.