Grasping Objects Using Shadow Dexterous Hand with Tactile Feedback

Sivakumar k [1], Priyanka Ch [2]

P.G. Student, School of Mechanical and Building Sciences, VIT University, Chennai, India[1]

Assistant Professor, School of Mechanical and Building Sciences, VIT University, Chennai, India[2]

ABSTRACT*:* The robotic hand is a wider field in Robotics. Improving the dexterity of the robotic hands play a major role in the development of Humanoid Robots. The aim of this project is to perform a more precise grasping on objects using a Shadow Dexterous Hand. The hand used for this project was designed by an UK based company named Shadow Robot. Joining hands with syntouchllc.they embedded the BioTac (tactile) sensors into the Hand. The BioTac sensors are installed at each tip of the five fingers and they provide nineteen different force outputs depending on the area of finger in contact with the object. The shadow hand supports the Robot Operating System software. ROS packages and libraries are used to simulate and perform control actions on the robotic hand in order to perform precise grasping on objects. Moveit! Package of ROS is used here for planning the joint movements of the robot.

KEYWORDS*:* Shadow hand, ROS, Moveit!, Move Group Interface.

I. INTRODUCTION

In modern robotics, the researchers try to improve the dexterity of the end effector close to that of the human hand for designing anthropomorphic robots. The application for which the robotic hand is to be used plays a major role in designing a hand. They are making a hand with minimalistic approach and with more Dexterity. In the first case the hand is made with minimal amount of engineering involved and it is more simple. These hands are preferred by majority of the customers in the market due to their economic benefits and its simple function to displace objects in a 3D-space. While, the later one is preferred in designing a humanoid robot or in research which involves mimicking a human functional behaviour. The efficiency of a robotic hand is based on the mechanical design to the mathematics involved in grasping algorithms. The hand can perform grasping with the help of the feedback from the vision sensor or the tactile sensor, else it can be done using human demonstration. Lighter objects need more dexterous grasp whereas heavier objects need more robust grasp. Once the vision information or tactile information is received, the actuators need to be controlled to achieve the desired behaviour. Two type of control actions can be performed. They are position control and force control. The joints in the end effector need to be controlled to a desired position for grasping the object. Once it reaches the position, the force applied by the fingers need to be calculated to grasp the object without damaging the object or the hand itself. Following these steps will result in a more dexterous hand manipulation and perform precise grasping even on more delicate objects [1],[2],[3].A survey on the tactile sensing solutions for dexterous in-hand manipulation states that resistive sensing techniques are still the predominant choice for tactile sensing and there exists a lot of resistive techniques in addition to piezoelectric and capacitive sensing techniques. New techniques like Electrical Impedance Tomography (EIT) and the use of embedded passive coils are also recent developments. There are some new materials such as ion-polymer metal composites (IPMCs), Organic Field Emitting Transistors (OFETs) and novel conductive materials which have been introduced for increased sensitivity, functionality and performance. Introduction of nano features such as carbon nano-tubes (CNTs), nano coils and nano wires into the resistive sensors based on conductive elastomer composites proved that conventional conductive particles can improve sensitivity and force range. Hence these conductive elastomer composites can be used in measurements of shear stress during manipulation. The important goal in the integration of a robotic system is to reduce the complexity of wiring to increase the robustness. The main approaches have been found to be encapsulation of sensor elements directly onto flex PCBs, including flexible/stretchable wiring in the sensor structure and direct integration of processing and communication transistors into the sensor array. There is a large understanding of physiology of human skin and more recently the research on neurophysiology of touching and grasping were done but we see that true understanding of the

human dexterous manipulation is still lacking and consequently design of tactile skins for intelligent robotic manipulation has also still not been achieved[4].Researchers concentrate in Improving the robot hand dexterity and performance by modifying hand design and articulation, mechanics of movement, hand dexterity and tactile sensing. It has been stated that bio-oriented dexterous robotic hands could be the challenge of the near future. Once this is done, then number of interesting questions like how new technologies can be integrated to manufacture robotic hands with some desired behaviour concerning to the reliable bio-oriented dexterous robotics can be addressed. It has also been stated that there are quite a large number of attempts to acquire biologically inspired hands and very rear attempts towards employment of brain and cognitive sciences in robotic hand control. However building a robust and accurate controller for a typical design is a complicated task. Use of compliant tendon cables and one way shape memory alloy (SMA) wires in an agonist-antagonist artificial muscle pair configuration for the required flex-ion/extension has a number of features like anthropomorphically accurate size and appearance, kinematically accurate joint motion, compliant and tendon driven muscle-like actuation and biomimetic sensor. A more realistic sensing with electronic processing power has been introduced through which a robotic hand can be able to detect an event of slippage while grasping. This has been further processed to determine tactile and touching events such as making/breaking contact while grasping, slip between the hand and the grasped objects as well as contact location and static pressure. It is more like use of body signals to control finger movements was also discussed [5].Mimicking a human finger using tactile sensor is being implemented and this model was based on soft contact modelling with a full friction description. This sensor was tested on the most common use cases of tactile sensors in robotic grasping. It performed stable grasping without any errors. But problems arose when trying to calibrate the tactile model to correspond exactly to a certain real tactile sensor. Improving the collision detection and computational efficiency are to be concentrated more [6].Using a single control output they have produced co-ordinated movements. The use of families of functions produced highly anthropomorphic hand motions. This method is capable of capturing and reproducing an arbitrary level of motion without affecting the dimensionality of the input space and can be used with various biological signals [7].Adopting the best possible postures to maintain the balance during the reach to grasp movement is an integral part of the grasping. And it also states that ample freedom to choose the posture is important without which the arm will not be able to move in the same manner. Rock climbing was given as an example [8]. The tactile sensing surface, tactile transducing elements for higher force resolution and lower noise level, faster scanning and processing are required in real time. It also demonstrated that tactile sensor array can provide contact force, centre of gravity and orientation for the objects curvature [9].A design support tool named as actuation dexterity is introduced here which is to be considered while designing a device. It is the result of observations made from experiments on five dexterous hands [10]. Finger deflict was calculated here as the difference between the sum of maximum individual finger forces during single finger pressing tasks and the maximal force produced by those fingers during an all finger press task. Finger independence was calculated as the average non task finger forces normalised by the task finger forces and subtracted from 100 percent. Here the critical role of tactile feedback is demonstrated [11].

II. HARDWARE MODEL

A. HAND DESCRIPTION

The shadow hand replicates the functionalities of a human hand. It provides four movements per each finger. The little and thumb fingers are provided with an extra joint for holding the object. The wrist has two joints for pitch and yaw movements. The model of shadow hand used in this research consists of tactile sensors (BioTac) embedded at the tip of each finger and because of this they lose one joint movement per each finger.
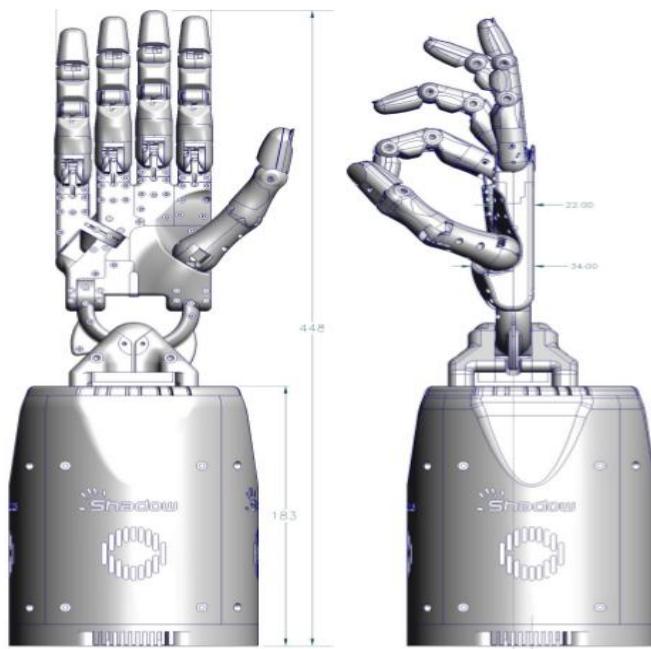
Fig 1 Shadow hand

The above figure represents the Shadow Hand model with the base forearm containing smart motors (one for each joint) and five fingers attached to the palm that is further attached to the wrist. It is to be noted that there is an extra joint provided for the little finger which can be seen at the bottom of the little finger.

B. KINEMATICS OF THE SHADOW HAND

In total the hand produces 24 movements to reproduce as closely as possible the dexterity and kinematics of the human hand. All the five fingers are of the same length but their knuckles are tottered in such a way to give comparable locations for the fingertip similar to the human hand. The forearm of the hand is somewhat wider than the normal human hand because, all the motors actuating the joints are arranged in the forearm of the hand. Smart motors are used here and each smart motor nodes use PWM to drive a Maxon motor. The motor also constitutes a PID controller and uses a pull/pull mechanism onto the tendons in order to control the position of the joints in the hand

Finger	Joints	Movements
First	3	4
Middle	3	4
Right	3	4
Little	4	5
Thumb	5	5

Table 1 Representation of finger joints and movements
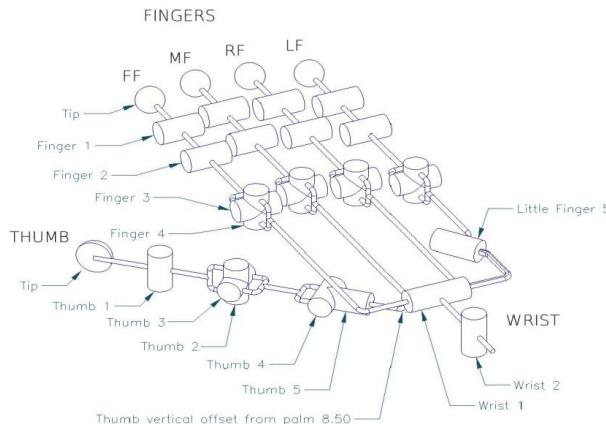
Fig 2 Kinematics of Shadow Hand

The kinematics of the shadow hand is described in the above figure. LF represents little finger, RF for right finger, MF for middle finger, FF for first finger and TH for thumb finger respectively. The joints are represented as fingertip joint, distal joint, middle joint, proximal joint and metacarpal joint.

III. MODELING AND PLANNING

A. ROBOT OPERATING SYSTEM

The robot operating system is the software used for simulating the hand since it contains all the essential packages and libraries needed to perform configuration, calibration, control and simulation of the hand. Nodes, topics, messages and services are some important parameters in Robot operating system. ROS core or master should be running in-order to performing all the above operations. It provides naming and registration services to all the other nodes in the system. When a node (executable) tries to publish or subscribe from or to another node via topics, it passes messages and the ROS master keeps tracking the information. The moveit! Package of ROS is used here to plan the movements of the joints in the simulation. The codes are written in C++ involving the move-group function of moveit!. Move group interface is a primary user interface which provides handy functionality for the users to carryout the operations like setting goal states for the joints, motion planning, adding objects in the working environment and attaching or detaching objects to or from the robot.

B. MODELLING

The initial step in the simulation is to bring the Robotic Hand into the moveit planner. The URDF (Unified Robot Description Format) file contains all the structural information of the robot. The moveit setup assistant is launched through terminal and a new moveit package is created by loading the urdf file of the hand discussed above.
The shadow hand used here contains tactile sensors embedded at the tip of each finger which are represented in dark green colour in the figure. The left pane consists of the planning instructions for the hand. The self-collision is to enable or disable collision pairs. The virtual joint is to make the robot fixed in a 3D- space. In planning groups the joints, links and chains are declared along with the nature of joints (revolute or fixed).The default robotic poses and end effectors for the hand can also be defined. As a final step the files are configured and generated. These files are stored in a used defined catkin workspace. The moveit window with shadow hand appears as shown below.
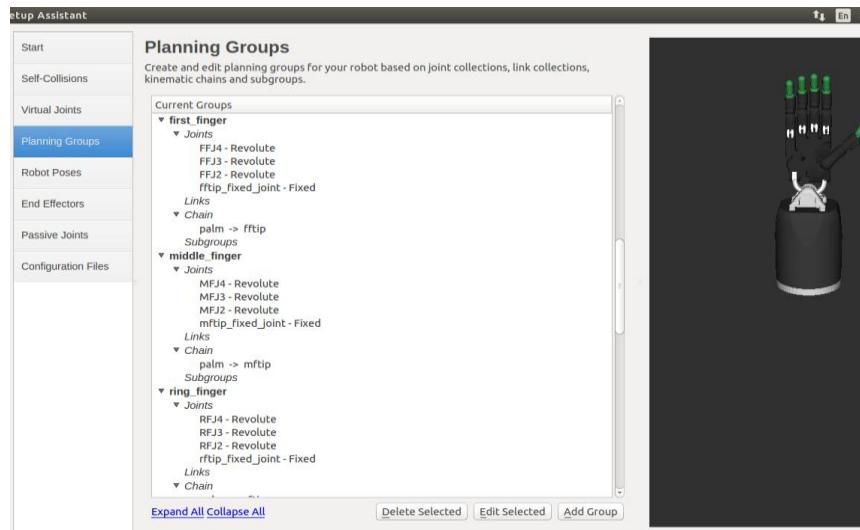
Fig 3 Shadow Hand Model

C. PLANNING

The generated package in the previous step contains a launch file which should be launched into the moveit planner using roslaunch<package_name><launch_file>. Now in the planning window the first finger is selected as the planning group in-order to plan movements for that particular finger. The tip of the first finger appears with a marker to interact with it. In this way the joint movements are planned and Later this is continued for planning movements for all the other fingers. After planning movements for the first finger the window appears as shown.
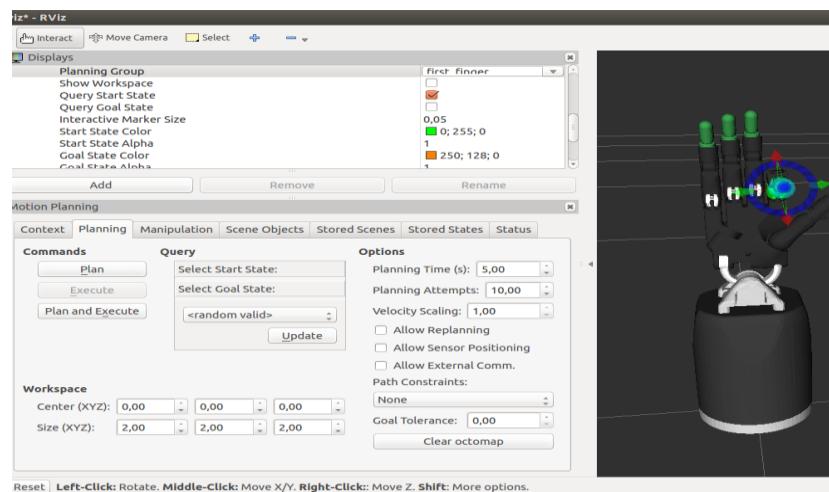
Fig 4 Planned joints using marker

IV. SIMULATION AND CONTROL

This section describes the use of Move group interface functions in C++ in-order to deliver control to the joints of the hand.

A. SIMULATION

Firstly the move group class is set using the name of the group to be planned and controlled. The class is declared as follows

moveit::planning_interface::MoveGroup group("three_fingers");
moveit::planning_interface::PlanningSceneInterfaceplanning_scene _interface;

The above commands involve planning movements for the first three fingers of the hand and the planning scene interface allows the declared class to deal directly with the world. The group.getcurrentstate() function allows us to get information on the current state of the joints in the hand. Once the current states are known, the target postures of each joint is planned and using group.setJointValueTarget(group_variable_values);
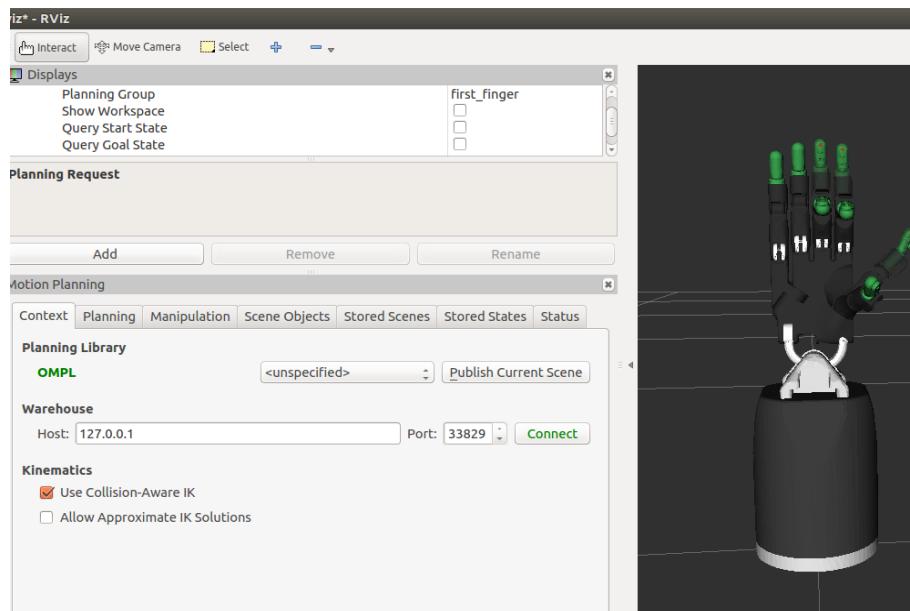
Fig 5 Step by step movements of finger joints

The target values are passed to the hand. The coding is written in such a way that the finger moves in sequential steps so that when it touches the object (identified by the tactile sensors), the loop can be terminated and the joints stop moving and precise grasping can be achieved. The resulting movement of the first three fingers in simulation is as shown below.
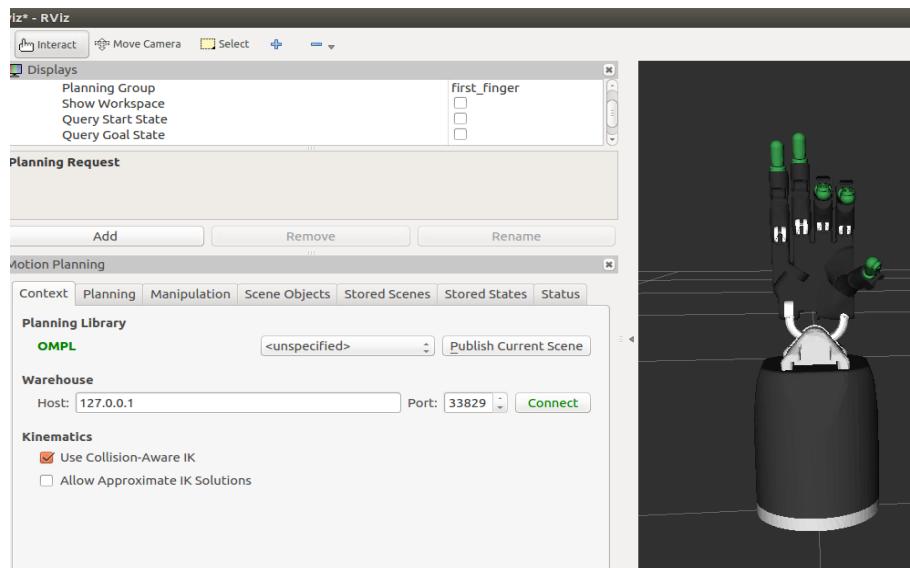
Fig 6 Simulated Shadow Hand

B. CONTROL

The code generated in moveit! is now interfaced with the real robot and communication between moveit! and the real robotic hand is established through the sr_standalone library of the robot operating system. The for loop executes seven times and for each time the joint moves step by step depending on the value provided to each joint in the array. Finally when it touches the object and if the force sensed by the tactile sensor exceeds 200 mV, the loop is terminated. The joint values for the shadow robotic hand is declared in the following order as follows

FFJ4, FFJ3, FFJ2, MFJ4, MFJ3, MFJ2, THJ4, THJ5, THJ3 & THJ2.

```
int joint = 0, times = 10, moves = 0, maxmoves = 3;
moveit::planning_interface::MoveGroup::Plan my_plan;
float movement[]={0.0,0.16,0.16,0.0,0.16,0.16,0.0,0.35,0.0,0.08};

    for( moves = 1; moves <= 7; moves = moves + 1 )
    {
    group_variable_values[1] += movement[1]; // FFJ3
            group_variable_values[2] += movement[2];
group_variable_values[4] += movement[4];
            group_variable_values[5] += movement[5];
    group_variable_values[6] += movement[6];
            group_variable_values[7] += movement[7];
            group_variable_values[8] += movement[8];
    group_variable_values[9] += movement[9];
            group.setJointValueTarget(group_variable_values);
            group.move();
            sleep(2);
    }

    while(is Ok)
    {
```

International Journal of Innovative Research in Science, Engineering and Technology

```
    tactiles = hand.get_tactiles();
    jss = hand.get_joint_states();
    tactiles[0].
    F_thumb = tactiles[4].pdc;
    F_index = tactiles[0].pdc;
    F_middle = tactiles[1].pdc;
    force_zero[0] = tactiles[0].pdc;
    force_zero[1] = tactiles[1].pdc;
force_zero[4] = tactiles[4].pdc;
    }
    if ( F_index - force_zero[0] > 200 ||
        F_thumb - force_zero[4]> 200 ||
        F_middle - force_zero[1] > 200)
{
        break;
    }
```

V. RESULTS AND DISCUSSION

A. RESULTS

As a result of this experiment when the simulation is interfaced with the real robot, the Shadow hand moves in sequence of steps and till it touches the object. The tactile sensors provides the force output and these readings are used to control the position of the joints by terminating the loop inside the move group C++ code and resulting the hand in performing more precise grasping of an object.

Fig 7 Precise Grasping by Shadow Hand using tactile feedback

B. DISCUSSION

Further the use of hand can be extended to grasp delicate or flexible objects like egg, fruits, etc. Use of Cyber gloves with shadow hand can mimic the action performed by a human. When a finger is moved, the sensing wire in the

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

cyber glove undergoes a change in angle .This angle can be passed as arguments to the code and corresponding actions can be transferred to the real hand.

REFERENCES

[1]   EbrahimMattar, "A Survey of Bio-Inspired Robotics Hands Implementation: New Directions in Dexterous Manipulation", Robotics and Autonomous Systems, Volume 61, Issue 5, Pages 517-544, May 2013.

[2]   Schmitz, A.; Maiolino, P.; Maggiali, M.; Natale, L.; Cannata, Giorgio; Metta, G. "Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors", Robotics, IEEE Transactions on,On page(s): 389 - 400 Volume: 27, Issue: 3, June 2011.

[3]   Mittendorfer, P.; Cheng, G. "Humanoid Multimodal Tactile-Sensing Modules", Robotics, IEEE Transactions on, On page(s): 401 - 410 Volume: 27, Issue: 3, June 2011

[4]   Akter, T.; Joseph, J.; Kim, W.S. "Fabrication of Sensitivity Tunable Flexible Force Sensor via Spray Coating of Graphite Ink", Electron Device Letters, IEEE, On page(s): 902 - 904 Volume: 33, Issue: 6, June 2012

[5]   Hanna Yousef, Mehdi Boukallel, KasparAlthoefer (2011) Tactile Sensing for Dexterous in-hand Manipulation in Robotics – A Review, Sensors and Actuators Vol. A 167, pages171-187, 2011.

[6]   H. Kawasaki, T. Komatsu and K. Uchiyama, "DexterousAnthropomorphic Robot Hand with Distributed Tactile Sensor: GifuHand II", IEEE/ASME Transactions on Mechatronics, Vol. 7, no. 3, pp. 296-303, Sep2002.

[7]   P. Michelman, "Precision Object Manipulation with a MultifingeredRobot Hand", IEEE Transactions on Robotics and Automation, Vol. 14,no. 1, pp.105-113, Feb 1998

[8]   A. Bicchi, "Hands for Dexterous Manipulation and Robust Grasping: ADifficult Road toward Simplicity", IEEE Transactions on Robotics andAutomation, Vol. 16, no. 6, pp. 652-662, Dec 2000.

[9]   D. Dornfeld and C. Handy, "Slip Detection Using Acoustic EmissionSignal Analysis", in Proc. IEEE Int. Conf. Robotics and Automation, Vol. 4, pp. 1868-1875, Mar 1987.

[10]   C. Melchiorri, "Slip Detection and Control Using Tactile and ForceSensors", IEEE/ASME Transactions on Mechatronics, Vol. 5, no. 3, pp 235-243, Sep2000.

[11]   R. Maldonado-Lopez, F. Vidal-Verdu, G. Linan and A. Rodriguez-Vazquez, "Tactile Retina for Slip Detection", in Proc. IEEE Int. Conf.Virtual Environments, Human-Computer Interface and MeasurementSystems, La Coruna, Spain, Vol 10, pp 96-101, July 2006.

[12]   D. Gunji, Y. Mizoguchi, S. Teshigawara, A. Ming, A. Namiki, M.Ishikawaand and M. Shimojo, "Grasping Force Control of Multi-fingered Robot Hand Based on Slip Detection Using Tactile Sensor", inProc. IEEE Int. Conf. Robotics and Automation, ICRA 2008, Pasadena,CA, , pp 2605-2610, 19-23 May, 2008.

[13]   Sami Moisio, Beatriz Leon, Pasikorkealaakso, Antonio Morales., "Model of Tactile Sensors using Soft Contacts and its Application", in Robot Grasping Simulation, Robotics and Autonomous Systems Vol. 61, pages1-12, (2013).

[14]   Benjamin A. Kent, John Lavery, Erik D. Engeberg., "Anthropomorphic Control of a Dexterous Artificial Hand via Task Dependant Temporarily Synchronized Synergies", Journal of Bionic Engineering Vol.11, pages 236-248, 2014.

[15]   B.Barkat, S.Zeghloul, J.P.Gazeau., "Optimization of Grasping Forces in Handling Brittle Objects", Robotics and Autonomous Systems, Vol 57, pages 460-468, 2009.

[16]   Jae Kun Shim, Sohit Karol, You-Sin Kim, Na JinSeo, Yoon Hyuk Kim, Yushin Kim, Bum Chul Yoon., "Tactile Feedback Plays a Critical Role in Maximum Finger Force Production", Journal of Biomechanics Vol 45, pages 415-420, 2012.