# Grasping Synchronous with Two Grippers Based in Semi-Humanoid Robot Baxter

luis Arias, , *luis.aac29890@gmail.com,*

***Abstract*—Grasping for manipulator robot has been researched widely around the world. For humanoid robot, grasping, using 2 arms and coordinate motion to pick up objects, is a complex topic and relies on the application.**
**In this article will be studied control for manipulating objects based in vision and force sensors to pick up object using 2 arms. Baxter robot as virtual robot will be used in CoppeliaSim simulator to demonstrate and validate the algorithms mentioned before. The control strategy for motion planning will be validated taking into account the inverse kinematic of the robot, based on CoppeliamSim libraries.**
**The performance of force controller will validate the minimization of the overshooting and either fast response time or stabilization in steady state.**
**Vision intelligent will be demonstrated using Convolutional Neural Network from Deep Learning algorithm. It will be showed as theoretical approach, giving its advantages for future applications.**

***Index Terms*—CNN, Manipulator, Deep Learning, Inverse Kinematic, Optimization.**

## I. Introduction

Grasping is the action to manipulate either objects or others. This is the natural action own of the human being. When human has to pick up something, he uses his arms and finger. Through cognitive actions the brain sends order to the limbs and they move according the requirement. This human action is studied widely in robotic for manipulator, humanoid robot and others. Grasping for human could be a easy task but inner the human there is a complex system that make possible that action. In other hand , for robot is a complicated task, not only for motion planning but also according the kind of object that will be picked up, depend of features like: material, shape, etc.

The coordination with 2 arms is used for applications such as: domestic robot, social robot and collaborative robot.

Grasping for dual arm robot is a big deal and will be studied topic in this article. Most of dual arm robot are semi-human robot and studied to make action like domestic applications.

Throughout the history there has been studies about grasping for robots.

Nikolaus V, (2010) presents an integrated planner for collision-free single and dual arm grasping motions. The proposed Grasp-RRT planner combines the three main tasks needed for grasping an object finding a feasible grasp, solving the inverse kinematics and searching a collision-free trajectory that brings the hand to the grasping pose.
Fangjing S, (2018) proposes novel multi-level Convolutional Neural Networks (CNNs) for finely grasping of unknown objects with multi-fingered dexterous hands, and design a quantitative evaluation method for grasping quality .
Shauri R, (2014) presents the tuning of PID control parameters for position feedback control of motor and the position reference design for grasping task by a 7-degree of freedom (DOF) three-fingered robotic hand. The system of the robotic hand consists of DC-motors, magnetic encoder and a motion controller unit.
Chalermsub S, (2006) proposed method searches for grasping configurations without prior knowledge of object's geometry. The experiment is carried out to validate the proposed method. The result when compared with a random search method shows that the proposed method finds more and better grasping configurations.
Nobuaki N, (1999) in his paper describes the force control of the robot hand which is modeled after the grasping characteristics of the human.
Heng G, (2019) presents a pipeline for grasping unknown objects based on RGB-D images from a stereo camera mounted on the wrist of the robot arm. The proposed grasping pipeline composes of a fully convolutional neural network (FCNN) and a Simplified Grasp Pose Detection (SGPD).
Kensuke H, (2011) proposes a method for planning a grasping posture for a parallel gripper attached at the tip of a robot manipulator.In order to robustly grasp several objects with various shapes,he considers a gripper having flexible sheets attached at the finger surface.
In this article will be studied how is the motion planning for dual arm robot based inverse kinematic, which lecture of force and vision sensor for control design.

There are a quite variety of dual arm robots, for research and education here will be used Baxter robot, which is a semi-human robot.

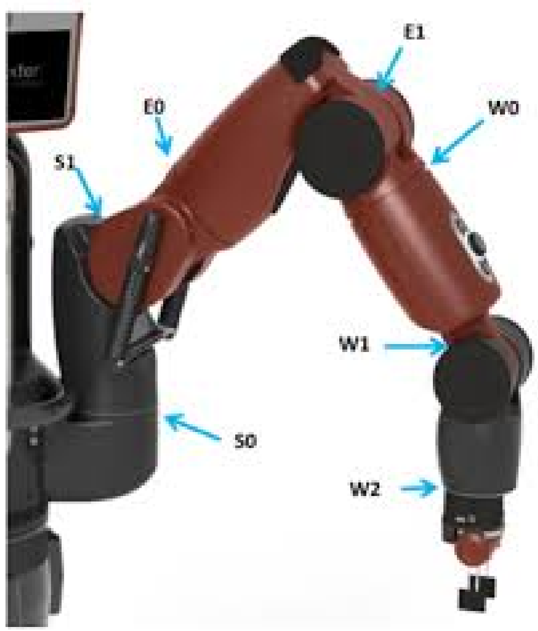
In order to demonstrate and validate the grasping system proposed, this article will be divided into 5 parts:

1) Description of the dual arm robot
2) Design of Force Control
3) Vision Machine
4) Validation
5) Conclusion

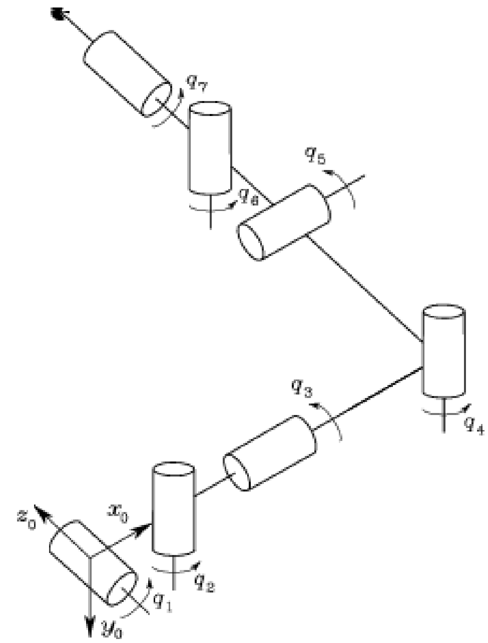
### A. Dual Arm robot Modelling

Throughout history the dual arm was studied for decades, and there are a variety of dual arm robot for applications such as: domestic, education, etc. Baxter is one of the educational robot used in institutes and colleges.
CoppeliaSim, thanks to Rethink company has Baxter model by default for installation.

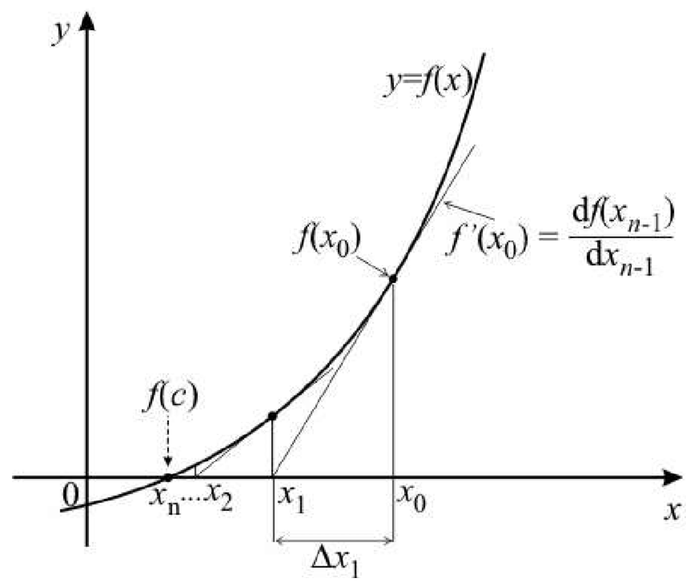
*1) Arm:* The figure above shows the real Baxter robot. This robot has 7 DOF for each arm (revolute joints). In the next figure is showed the configuration of joints.




Modelling the inverse kinematic of 7 DOF arm robot is not as easy as it looks like. It requires advanced mathematical knowledge. For geometrical solutions it could be laborious.



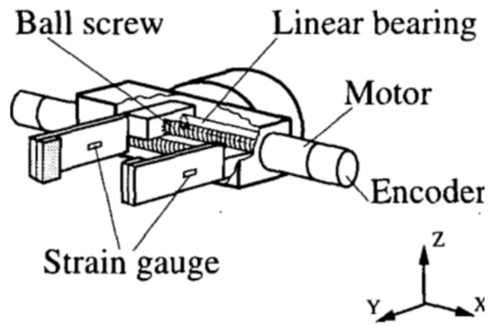

There are another alternatives, the analytical solution using Newton-Raphson approach will be studied.
This is based on iterative approximations of Taylor expansion, to converge the solution that minimize the derivative.

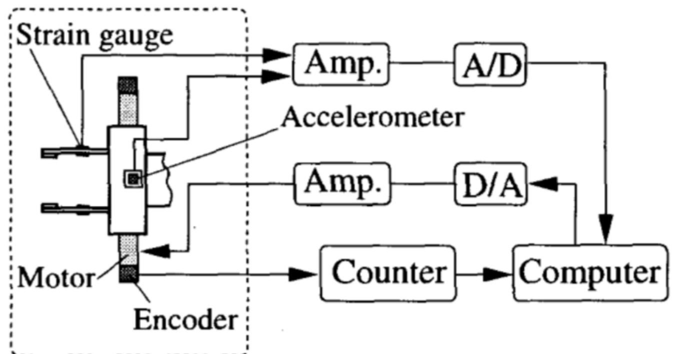


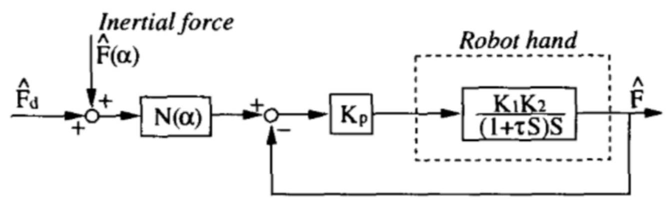

$$f(x_0) = \frac{df(x_{n-1}}{dx_{n-1}} \tag{1}$$

where $f(x_0)$ is the slope for each iteration.

*2) Gripper:* The gripper is the end effector tool based in 2 plates, with micro motor as actuator to open a close the gripper. Also has included force sensors.
In the below figure is showed the real gripper for baxter robot.

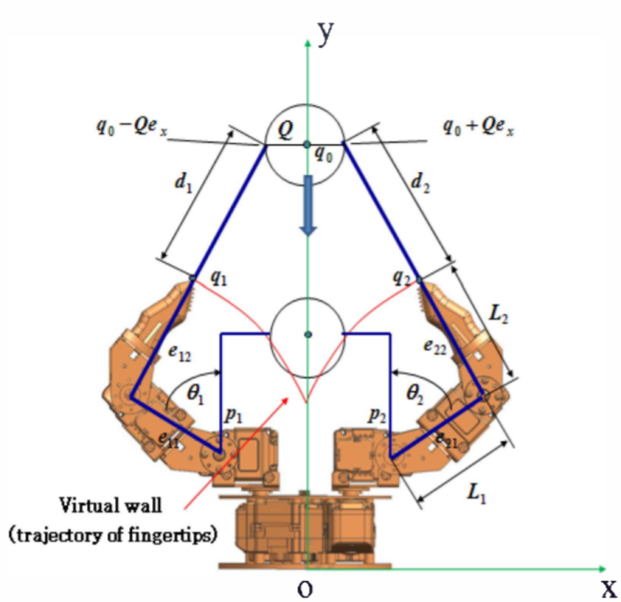



## B. Force Control

Picks up a smooth object is clearly different than when tries to picks up a piece of heavy block. Now, taking in mind the object texture the force sensor has a role important for grasping strategy.
Geometrically, this configuration could be presented like this figure.




The force control was designed, based on dynamic model, taking consideration of the external forces that could be required to compensate the control variable.

To move up the object when the gripper is near to the goal, the force can be modelled as:

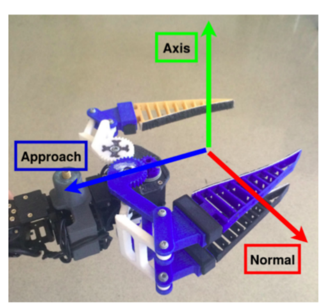
$$m\alpha = f_1 + f_2 - mg \tag{2}$$

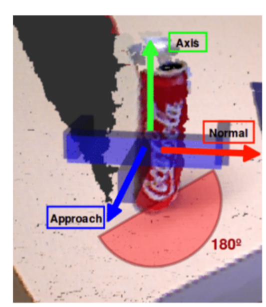
$$|f_1| <= uF_1 \tag{3}$$

$$f_1 = \frac{m(g+\alpha)}{2} \tag{4}$$

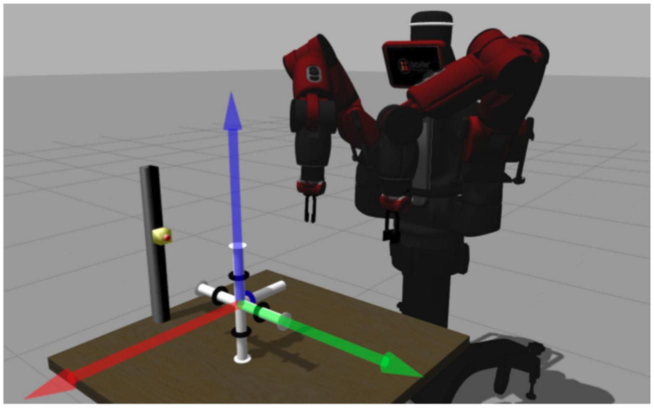
## C. Vision Strategy

When the robot picks up the object. There could be many possibilities for the final configuration of the end effector. Just force control is not enough. Let me explain you with a example: If the robot has a configuration frame to pick up a object, the planner generate the trajectory to reach that configuration and the force control is in charge of control the force of gripper closure to not damage the object and at the same time, save battery energy. However is any agent moves the objects, changes the final configuration, and then the planner has to re-fix the trajectory. To do that, vision computing has a good apparition here.



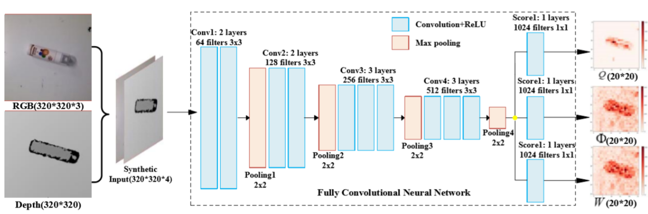

As showed in the above and below figure, the configuration of the end effector frame has to be updated online, using video camera the robot has this capability.



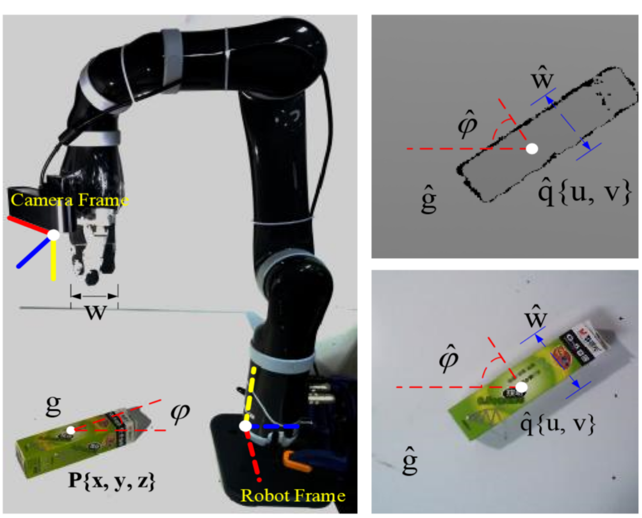
In grasping theory, vision computing and Deep Learning was deeply studied. Convolutional Neuronal Network for computing the frame of the object will be consider.

*1) Convolutional Neuronal Net:* Convolutional Neuronal Net or CNN for short, is a kind of Neural Network for deep learning. In vision processing is applied to learn images.

Nevertheless, in this case to train the Network is not enough RGB image, and will be also used depth image. That requires the training of the neural network for extra features. Taking in mind this approach, the robot will be able to calculate la position and orientation in 3D of the object.




Finally, the planner can be able to process the output of CNN and find the new trajectory to make the grasping action.

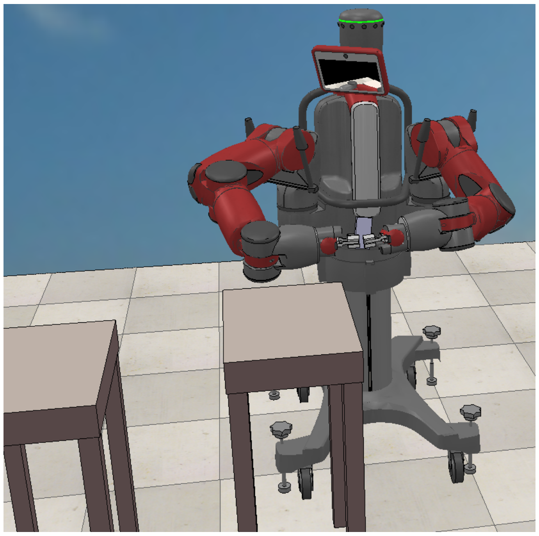



*D. Result*

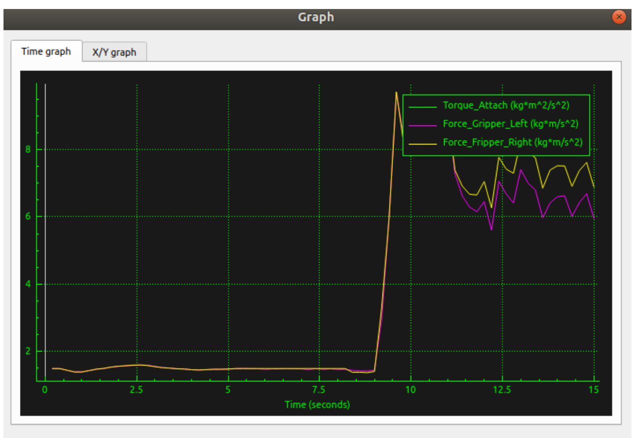
The result for this article was made using CoppealiaSim Simulator. The Inverse kinematic for arm motion was made using LUA code. It provides classes for inverse kinematic. For the simulation Baxter has to grasp the block over the table and coordinate with the right arm to move the block to the beside table.
Baxter has video camera at the end effector level, this is important to get information of the object (position and orientation).
In the following figure baxter robot make the trajectory to picks up the block over the table.

The force control is required when a gripper is touching the final object, to control the gripper closure.

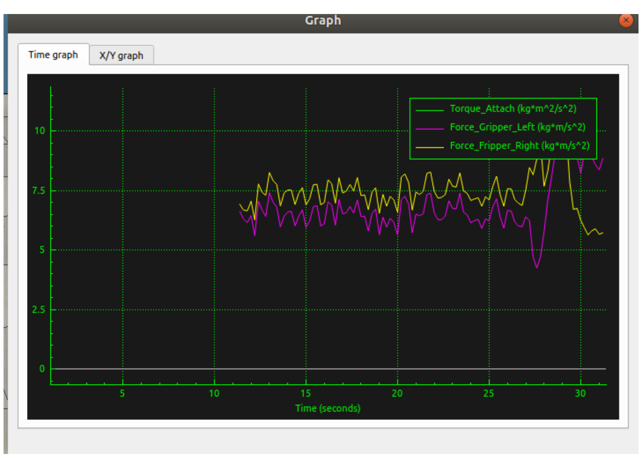
The coordination with two arm was made with No-Threaded script in LUA, and this is the result.



The next figure shows the force signal for the left gripper, when the gripper approaches to reach the object.



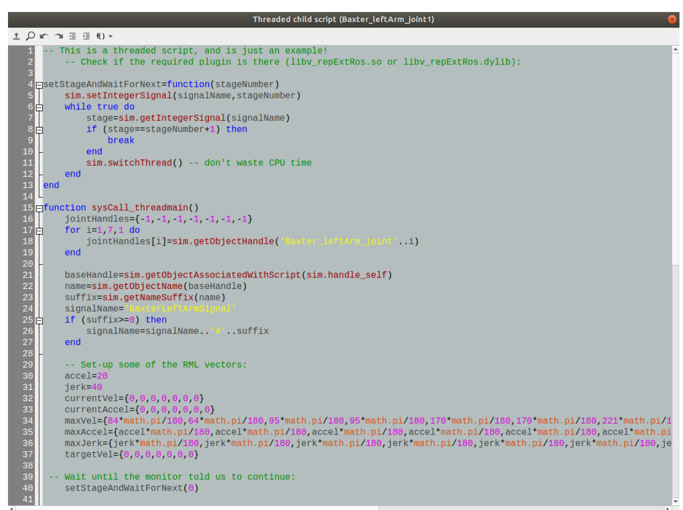

Basically the graph demonstrates how is the response of the force when the gripper picks up the object.




Finally the LUA code was implemented for each arm, that include the gripper tool.

```
waitForLeftArm=function(waitNumber,nextNumber)
    while true do
        stage=sim.getIntegerSignal(leftArmSignalName)
        if (stage==waitNumber) then
            break
        end
        sim.switchThread() -- don't waste CPU time
    end
    sim.setIntegerSignal(leftArmSignalName,nextNumber)
end

waitForRightArm=function(waitNumber,nextNumber)
    while true do
        stage=sim.getIntegerSignal(rightArmSignalName)
        if (stage==waitNumber) then
            break
        end
        sim.switchThread() -- don't waste CPU time
    end
    sim.setIntegerSignal(rightArmSignalName,nextNumber)
end

function sysCall_threadmain()
    monitorJointHandle=sim.getObjectHandle('Baxter_monitorJoint')
    name=sim.getObjectName(monitorJointHandle)
    suffix=sim.getNameSuffix(name)

    leftArmSignalName='BaxterLeftArmSignal'
    if (suffix>=0) then
        leftArmSignalName=leftArmSignalName..'#'..suffix
    end

    rightArmSignalName='BaxterRightArmSignal'
    if (suffix>=0) then
        rightArmSignalName=rightArmSignalName..'#'..suffix
    end

    -- Tell the left arm to start movement:
    waitForLeftArm(0,1)

    -- Move the head towards the left arm:
    sim.setJointTargetPosition(monitorJointHandle,30*math.pi/180)

    -- Wait 2 seconds:
```

Was considered the synchronism between the right and leg arm through LUA code.


Threaded child script (Baxter_leftArm_joint1)

```
-- This is a threaded script, and is just an example!
    -- Check if the required plugin is there (libv_repExtRos.so or libv_repExtRos.dylib):

setStageAndWaitForNext=function(stageNumber)
    sim.setIntegerSignal(signalName,stageNumber)
    while true do
        stage=sim.getIntegerSignal(signalName)
        if (stage==stageNumber+1) then
            break
        end
        sim.switchThread() -- don't waste CPU time
    end
end

function sysCall_threadmain()
    jointHandles={-1,-1,-1,-1,-1,-1,-1}
    for i=1,7,1 do
        jointHandles[i]=sim.getObjectHandle('Baxter_leftArm_joint'..i)
    end

    baseHandle=sim.getObjectAssociatedWithScript(sim.handle_self)
    name=sim.getObjectName(baseHandle)
    suffix=sim.getNameSuffix(name)
    signalName='BaxterLeftArmSignal'
    if (suffix>=0) then
        signalName=signalName..'#'..suffix
    end

    -- Set-up some of the RML vectors:
    accel=20
    jerk=40
    currentVel={0,0,0,0,0,0,0}
    currentAccel={0,0,0,0,0,0,0}
    maxVel={84*math.pi/180,64*math.pi/180,95*math.pi/180,95*math.pi/180,170*math.pi/180,170*math.pi/180,221*math.pi/1
    maxAccel={accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi
    maxJerk={jerk*math.pi/180,jerk*math.pi/180,jerk*math.pi/180,jerk*math.pi/180,jerk*math.pi/180,jerk*math.pi/180,je
    targetVel={0,0,0,0,0,0,0}

 -- Wait until the monitor told us to continue:
    setStageAndWaitForNext(0)
```

## *E. Conlusion*

In this article was developed the Baxter simulation to validate the grasping control based in Force control and CNN for end effector frame update.

The force control was successfully developed to pick up object according to the texture, hardness, etc.

CNN for vision strategy, was consider and will be topic on

future implementation.
Finally Baxter robot was implemented in CoppeliaSim and the trajectory was generated for inverse kinematic include in LUA code.

## REFERENCES


[1] J. Antonio, Two-Handed Grasping with Two-Finger Hands, CONACYT, mexico, 1991.
[2] N. Nabasawa, Force Control of a Robot Hand Emulating Human's Grasping Motion. IEEE , 1999.
[3] H. Guan, An Efficient Robotic Grasping Pipeline Base on Fully Convolutional Neural Network , Chengdu, China, 2019.
[4] P. Lima, Grasp Planning with Incomplete Knowledge About the Object to be Grasped , Lisboa, 2019.
[5] K. Harada., Grasp Planning for Parallel Grippers with Flexibility on its Grasping Surface,International Conference on Robotics and Biomimetics, 2011.
[6] S. Young, A Study on Grasping Control of Hand Fingers 12 Joints , ISth International Conference on Control, Automation and System , 2015.
[7] M. Kokic, Learning Task-Oriented Grasping From Human Activity Dataset ,IEEE ROBOTICS AND AUTOMATION LETTERS , 2020.
[8] N. Mavrakis, Safe Robotic Grasping: Minimum Impact-Force Grasp Selection, International Conference on Intelligent Robots and System, Vancouver, 2017.
[9] S. Revathi, Object Grasping using Convolutional Neural Networks , ViTECoN , 2019.
[10] C. Sangkhavijit, Computing 4-Fingered Force-Closure Grasps from surface Points Using Genetic Algorithm. 2006.
[11] C. Sangkhavijit, PID Position Control of 7-DOF Three-Fingered Robotic Hand for Grasping Task,IEEE International Conference on Control System, Malaysia, 2014.
[12] F. Song, Learning Optimal Grasping Posture of Multi-Fingered Dexterous Hands for Unknown Objects,International Conference on Robotics and Biomimetics, Malaysia, 2018.
[13] N. Vahrenkamp, Integrated Grasp and Motion Planning, IEEE International Conference on Robotics and Automation,Alaska, 2018.
[14] S. Varkey, Learning Robotic Grasp using Visual-Tactile mode, Malaysia, 2018.